Degree Final Project

Degree in Industrial Technology Engineering

Analysis and design of a lateral force estimator for a Formula Student vehicle

Author: Adrián Blázquez Romañá
Director: Carlos Ocampo Martínez
Summon: June, 2020

Escola Tècnica Superior d'Enginyeria Industrial de Barcelona

Abstract

Vehicle dynamic controls are a very important aspect in the automotive industry, especially in electric or hybrid vehicles, as they allow the car to obtain a better driving performance and power-train management systems. In racing cars, the control systems related to power distribution such as the Torque Vectoring are especially important and they open an amount of possibilities related to the tire forces distribution in order to increase the vehicle dynamics behavior and obtain the maximum efficiency possible. They include the control and use of tire forces, which is a difficult task specially because of their nonlinearities, as they are the only points of interaction with the road and they limit the dynamic behavior of the car.

In this work, the lateral vehicle dynamics are analyzed and modeled so a tire cornering force estimator can be designed and integrated to the current Torque Vectoring control system of a Formula Student vehicle to improve its performance. Four different observers (a Luenberger, Extended Kalman Filter, Unscented Kalman Filter and a Moving Horizon Estimator) that will be analyzed and compared are designed with MATLAB and CasADi and simulated with the IPG CarMaker software in order to validate them. It is a full simulation work and it will require of a future implementation in the actual car.

The lateral dynamics of the car are modeled using a seven-variable state-space representation, which is discretized and validated during the project. The different parameters of the observers are tuned so they estimate the tire forces properly and the estimators are finally compared according to some Key Performance Indicators (KPIs) so one of the four estimation options is chosen as the best observer to integrate in the car.

The final conclusion of the work is that the Moving Horizon Estimator proposed is the best option to estimate the tire cornering forces properly as it meets all the necessary requirements to be integrated in the car and it can be used to provide the inputs of the Torque Vectoring system.

Acknowledgments

First of all, I would like to thank all the ETSEIB Motorsport team members for all the moments shared during this year and everything I learned in my first real introduction to the motorsport world. Without them, all the work during this project would not make any sense as there would not be any car in where the estimator designed could be implemented. Along the same lines, I would like to thank especially the vehicle dynamics control department of the team, Eduard, Oriol and Marc, not only for accepting me in the section, but also for all the knowledge that I have acquired thanks to them and all the help they have given me during the work.

On the other hand, I would also like to thank Carlos Ocampo, not only for accepting to guide my project, but also for all the help he has given to me during the realisation of the work. For all the advises and suggestions based on his knowledge and personal experience, that have allowed me to do this work the way it has been until this final result.

Finally, thanks to all my family and friends for helping and supporting me during all the four years of university studies and during the realisation of the work. I really appreciate what they have done.

Contents

Abstract i

Acknowledgments ii

List of Figures vi

List of Tables vii

List of Abbreviations viii

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 State of the art . . . 2
- 1.3 Objectives . . . 2
- 1.4 Outline of the project . . . 3

2 Vehicle modeling 4
- 2.1 Tire dynamics . . . 4
  - 2.1.1 Longitudinal force . . . 5
  - 2.1.2 Lateral force . . . 6
  - 2.1.3 Vertical force . . . 8
- 2.2 Vehicle dynamics . . . 9
- 2.3 Vehicle dynamic model . . . 10
  - 2.3.1 Tire modeling . . . 12
  - 2.3.2 Vehicle model . . . 13
  - 2.3.3 Disturbance and noise modeling . . . 14
- 2.4 Model validation . . . 15
- 2.5 Summary . . . 20

3 Lateral force estimation 21
- 3.1 Data acquisition . . . 21
- 3.2 Estimator design . . . 21
  - 3.2.1 Luenberger Estimator . . . 24
  - 3.2.2 Extended Kalman Filter . . . 25
  - 3.2.3 Unscented Kalman Filter . . . 28
  - 3.2.4 Moving Horizon Estimator . . . 30
- 3.3 Summary . . . 32

4 Simulation and Comparison 33
- 4.1 Test tracks . . . 33

4.2 Key Performance Indicators . . . 33
4.3 Tustin and Euler analysis . . . 35
4.4 MHE horizon analysis . . . 36
4.5 Skidpad Analysis . . . 41
4.6 Autocross Analysis . . . 46
4.7 Estimators comparison . . . 51
4.8 Summary . . . 52

5 Economical analysis **53**

6 Environmental impact **55**

7 Conclusions **56**
7.1 Main Contribution . . . 56
7.2 Conclusions on the project . . . 56
7.3 Further work and proposals . . . 57

8 References **58**

List of Figures

1.1 Torque vectoring scheme 1

2.1 Tire forces and moments [Kuster et al., 2015] 5
2.2 Longitudinal force against slip ratio. Taken from [Kang et al., 2005] 5
2.3 Longitudinal tire dynamics. 6
2.4 Tire slip angle [OldBastardRacing, 2018] 7
2.5 Lateral force against slip angle. Taken from [OptimumG, 2012] 7
2.6 Car subjected to longitudinal and lateral acceleration [Seward, 2014] 8
2.7 Vehicle motion 9
2.8 Pro-Ackermann (left) Anti-Ackermann (right) configurations 10
2.9 4WD model representation scheme 11
2.10 Tire slip angles modeled and simulated during an Autocross run. 15
2.11 Longitudinal velocity modeled and simulated during an Autocross run. 16
2.12 Lateral velocity modeled and simulated during an Autocross run. 17
2.13 Yaw Rate modeled and simulated during an Autocross run. 17
2.14 Tire lateral forces modeled and simulated during an Autocross run. 18
2.15 Three axis sensor rotation. 18
2.16 Pitch angle during a Skidpad event. 19
2.17 Roll angle during a Skidpad event. 19

3.1 Tire lateral force estimator scheme. 22
3.2 Kalman Filter scheme. 25
3.3 Unscented transform scheme, obtained from [Wan and Van Der Merwe, 2000] 28

4.1 Skidpad track layout 34
4.2 Elapsed time for the Tustin (left) and Euler (right) method. 35
4.3 Eigenvalues comparison for Tustin (left) and Euler (right) method. 35
4.4 Longitudinal velocity for different horizon lengths. 36
4.5 Lateral velocity for different estimation horizon lengths. 36
4.6 Yaw rate for different estimation horizon lengths. 37
4.7 Front left (up) and right (down) lateral forces for different estimation horizon lengths. . . 37
4.8 Rear right (up) and left (down) lateral forces for different estimation horizon lengths. . . 38
4.9 Expanded lateral forces estimation. 39
4.10 CPU time for the different estimation horizon lengths. 39
4.11 Longitudinal velocity comparison during an Skidpad run. 41
4.12 Lateral velocity comparison during an Skidpad run. 41
4.13 Yaw rate comparison during an Skidpad run. 42
4.14 Tire lateral forces comparison during an Skidpad run. 42
4.15 State variable relative error (kinematic variables) during an Skidpad run. 44
4.16 State variable relative error (lateral forces) during an Skidpad run. 44

4.17 Computational time for the NLO, EKF, UKF and MHE observers. 45
4.18 Longitudinal velocity comparison during an Autocross run. 46
4.19 Lateral velocity comparison during an Autocross run. 47
4.20 Yaw rate comparison during an Autocross run. 47
4.21 Tire lateral forces comparison during an Autocross run. 48
4.22 State variable relative error (kinematic variables) during an Autocross run. 49
4.23 State variable relative error (lateral forces) during an Autocross run. 50

List of Tables

2.1 Tire slip angles relative errors. 16
2.2 State variables errors 18

4.1 State variable mean errors (in %) according to the IPG response. 40
4.2 State variable mean errors (in %) according to the sensor input. 40
4.3 State variable errors (%) for an Skidpad event simulation. 43
4.4 State variables mean errors (in %) according to the sensor input. 45
4.5 Convergence time. 46
4.6 State variable errors (in %) for an Autocross event simulation. 49
4.7 State variables mean errors (in %) according to the sensor input. 50
4.8 Convergence time. 51
4.9 Estimators comparison. 51

5.1 Economical analysis of the different tasks of the project. 53
5.2 Economical analysis of the material and software required for the project. . 54

List of Abbreviations

Symbol	Units	Meaning
a_x	m/s^2	Longitudinal acceleration
a_y	m/s^2	Lateral acceleration
a_z	m/s^2	Vertical acceleration
v_x	m/s	Longitudinal velocity
v_y	m/s	Lateral velocity
$\dot{\psi}$	rad/s	Yaw Rate
$\ddot{\psi}$	rad/s^2	Yaw Acceleration
β	rad	Vehicle slip angle
α	rad	Tire slip angle
σ_x	-	Slip Ratio
ω	rad/s	Wheel angular velocity
v_{wx}	m/s	Wheel longitudinal velocity
I_w	kgm^2	Wheel inertia moment
δ	rad	Steering angle
F_x	N	Longitudinal tire force
F_y	N	Lateral tire force
F_z	N	Vertical tire force
σ	m	Relaxation length
C_α	N/rad	Cornering stiffness
K	N/m	Lateral stiffness
τ	s	First order time constant
$\bar{F}_y$	N	Static tire lateral force
h_c	m	Height to center of mass
l_r	m	Rear distance to center of mass
l_f	m	Front distance to center of mass
t_r	m	Rear track
t_f	m	Front track
l, L	m	Wheel base
t_s	s	Sample period

Symbol	Units	Meaning
m	kg	Vehicle mass
I_z	kgm^2	Vehicle vertical inertia moment
g	$\mathrm{m/s}^2$	Gravity acceleration
R	m	Wheel radius
Γ	Nm	Wheel torque
ρ_{air}	$\mathrm{kg/m}^3$	Air density
C_D	-	Drag coefficient
A_x	m^2	Frontal area
W	N	Vehicle weight
ΔW_x	N	Longitudinal load transfer
ΔW_y	N	Lateral load transfer

Chapter 1

Introduction

1.1 Motivation

This work is included within the Formula Student project. Formula Student is an engineers competition in which some university students from all over the world design and construct a racing car in order to compete between them. The main objective of the competition is to allow the students to be part and carry out a real engineering project, providing them with personal and professional experience related to the engineering world and, specially, the automotive industry.

The ETSEIB Motorsport team has participated in this competition for the past 13 years. The evolution of the technology brings a lot of new possibilities each year to improve the past vehicles and design and construct an improved racing car. Along the same lines, the car has evolved from a combustion to an electric powered car. In the last year project, the car went from one electric motor in the rear axle to a four-wheel drive (4WD) car, with an electric engine per wheel. This new condition allows the team to incorporate what is known as a Torque Vectoring control algorithm [De Novellis et al., 2014], which inclusion implies the existence of a lot of possibilities to improve the dynamic behavior of the vehicle and to obtain the best performance and energy management possible in the competitions.

The design of a complex and competitive car for the Formula Student competition is the main motivation for this work, and it can be done by improving the current Torque Vectoring control system [Trunas Bruguera, 2019] of the ETSEIB Motorsport's car. This type of control algorithm is specially useful when the car is cornering and it requires of some inputs such as the lateral tire forces which cannot be measured (see Figure 1.1).

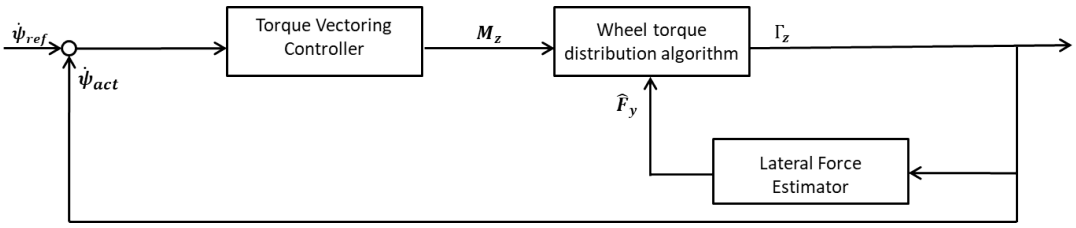

Figure 1.1: Torque vectoring scheme

In Figure 1.1, Ψ_{ref} is the reference yaw rate, Ψ_{act} is the actual yaw rate, M_z is the moment in the Z axis, Γ is the wheel torque and $\hat{F}_y$ is the estimated lateral force. The different

controllers and algorithms the Torque Vectoring includes are already being studied and designed in [Morera Torres, 2020], but it requires of the lateral forces inputs to work. These forces are currently being obtained by calculating the total lateral force per axle with a simplified model and weighting them according to the vertical load per tire. The results obtained are not appropriate and could lead to high errors in the torque distribution. The possibility of including these cornering forces properly in the algorithm via an estimation process is the base of this project and the main motivation of it. There are many observer options that can actually be added in the car to obtain the best performance possible and make the control system optimal. Four different estimators will be analyzed and designed during this work so the proper one can be used as an input for the current Torque Vectoring control of the car. Considering the importance of the values the forces get in the output of the algorithm, it will be considered as a preferable estimator those that assure the convergence, not only in order to get an optimal response, but also for security.

1.2 State of the art

The automotive industry have been changing for the last past years as new improves have been reached. One of these important aspects are all the improvements related to the electrical energy and the different uses it has either the power-train or the electronics inside the car. All these vehicles, which can be completely electric or hybrid cars, require of some control algorithms, not just to manage and distribute the power correctly and optimally, but also to improve its performance and limitations. Although the majority of these control systems get the inputs from different sensors incorporated in vehicle, there are a lot of variables that cannot be measured directly and should be estimated. For the racing cars (the case this project is studying), the main variables that need to be estimated are those which cannot be obtained with any kind of sensor or those whose sensors are expendable (this is especially in terms of weight saving). These variables are mainly divided between kinematic and dynamic ones.

Regarding the kinematics, the main variables estimated are the yaw rate and the vehicle slip angle [Ghoneim and Chin, 2001], which are typically used as a way of finding the reference yaw rate in Torque Vectoring algorithms. The majority of these observers are designed by Kalman Filter based estimators. As racing cars work in their dynamic limits most of the time, nonlinear models should be considered instead of the basic linear model, and that is why the Extended Kalman Filter is typically used when estimating variables [Nam et al., 2012]. Some other variables usually estimated in racing cars are the velocities [Imsland et al., 2006], specially the lateral speed that not all of the sensors can measure.

Regarding the dynamic variables, the longitudinal, lateral and vertical tire forces [Ray, 1995] are the most commonly estimated as the vehicle dynamics in a car are really complex. Forces are difficult to measure and they usually are estimated via Kalman Filters, such as the Kalman–Bucy Filter [Ray, 1997], the Extended Kalman Filter or the Unscented Kalman Filter [Hamann et al., 2014]. The road friction coefficient is also typically estimated in racing cars as a complement of the tire forces.

1.3 Objectives

The main objective of this project is to design a tire lateral force estimator that can be integrated as an input for the actual Torque Vectoring system of the ETSEIB Motorsport car so it improves its performance and dynamic behavior. To achieve this goal, the follow-

ing specific objectives must be met:

- Explore the state of the art of the vehicle state estimation for racing cars, especially that related to nonlinear system and dynamic modeling.
- Study the vehicle dynamics of a car, focusing on the tire dynamics.
- Develop a proper nonlinear model of the tire and the vehicle.
- Design different observers that estimate suitably the state variables.
- Define the most suitable estimator, according to the simulations done, such that it can be considered for implementation into the control system of the car.
- Give a basis in estimation concepts for the future new members of the team.

1.4 Outline of the project

This work is divided and organised with the following structure:

Chapter 2

This chapter introduces the tire and vehicle dynamics in order to get a proper model that can be used for the tire lateral force estimator. Once the car is modeled, it will be validated with the use of the IPG CarMaker software.

Chapter 3

This chapter defines the sensors available and the nonlinear state-space is completely defined with its corresponding state vector, input vector and measurement vector. The discrete-time version of the model is proposed. Finally, four different observers are introduced and designed so they can be compared and analyzed.

Chapter 4

In this chapter, the four estimators described in Chapter 3 are compared in two different Formula Student event track layouts. They are compared considering three Key Performance Indicators and one of them is chosen as the best option to integrate it in the car.

Chapter 5

This chapter introduces the economical analysis of all the project, divided between personal cost and material cost.

Chapter 6

In this chapter, the environmental impact of all the work is studied and analyzed.

Chapter 7

The final chapter presents the conclusions of the work and the results obtained during its realisation. The future work based on what has been presented in this work is also exposed in Chapter 7.

Chapter 2

Vehicle modeling

As it has been commented in Chapter 1, the main objective of this project is to estimate the lateral forces of the tire. In order to design a suitable estimator, it is important to know how a Formula Student car behaves and getting a proper mathematical model from it.

In this chapter, a basic introduction to the vehicle dynamics of a car and the important parameters that should be considered are discussed.

2.1 Tire dynamics

The interaction of the tire with the road produces the forces that allow the vehicle to move as they are the main external forces the car gets. These forces depend on a lot of parameters such as the road friction coefficient and the tire properties like the material, profile or stiffness. As a consequence of the large number of variables involved, tire dynamics are really complex to understand and also to model.

Splitting up the space in a three-axis system (X, Y, Z), the forces and torques acting in the tire can be modelled and studied easier. All the acting forces and moments considered here are represented in Figure 2.1.

In Figure 1, all the forces, moments and angles of each axis are referred to as

- F_x: Longitudinal or tractive force.
- F_y: Lateral or cornering force.
- F_z: Vertical force.
- M_x: Overturning moment.
- M_y: Rolling resistance moment.
- M_z: Self aligning moment.
- α: Slip angle.
- γ: Camber angle.

In this project, just the main forces will be considered in order to model the tire and the vehicle, this means that the self-aligning moment, the rolling friction force and the rolling moment will be neglected and just longitudinal, lateral and vertical forces will be considered.

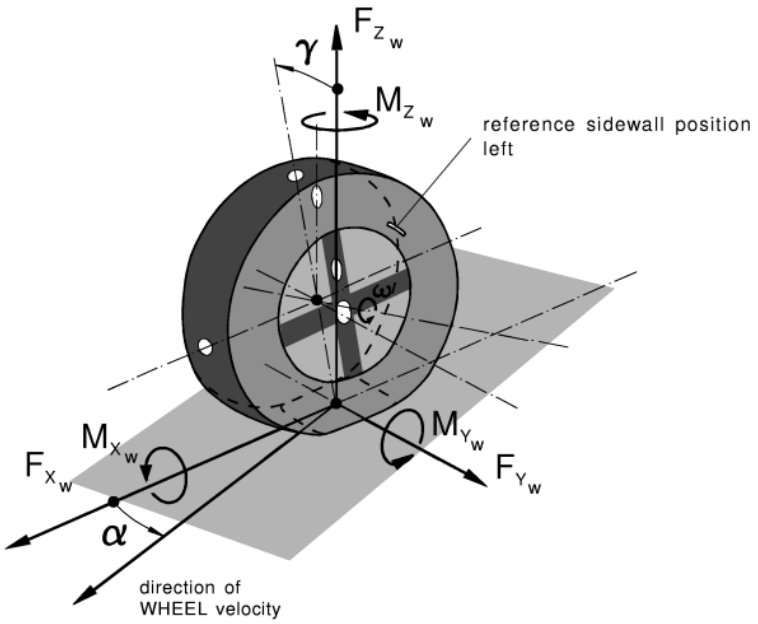

Figure 2.1: Tire forces and moments [Kuster et al., 2015]

2.1.1 Longitudinal force

As it is shown in Figure 2.2, a tire needs a certain longitudinal slip, called slip ratio, in order to make this force between the road and the tire. This longitudinal force is the one that allows the car to accelerate or brake and also depends on the vertical force, so it is usually represented as a normalized force $\frac{F_x}{F_z}$.

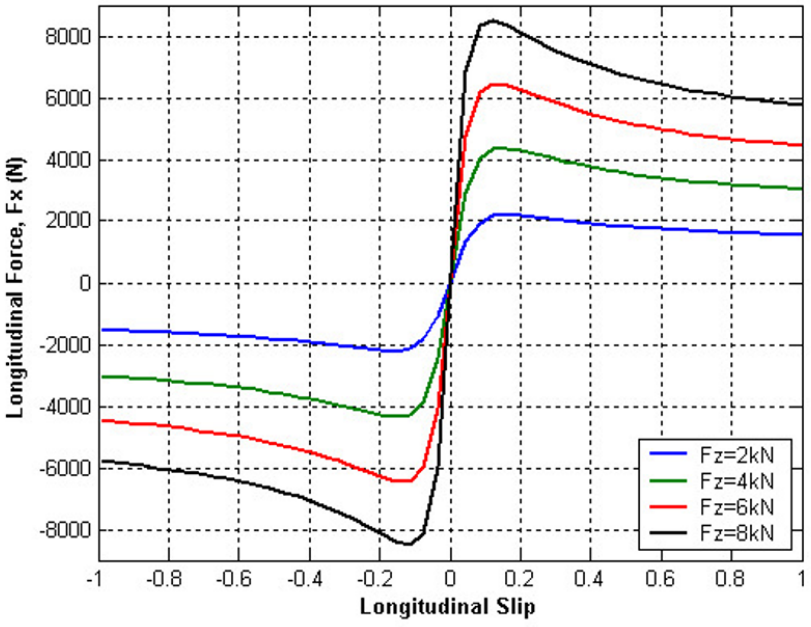

Figure 2.2: Longitudinal force against slip ratio. Taken from [Kang et al., 2005]

The longitudinal slip ratio is defined as

$\sigma_x = \frac{R\omega - v_x}{R\omega}$, while accelerating, and

$\sigma_x = \frac{v_x - R\omega}{v_x}$, while braking,

where v_x is the longitudinal velocity [m/s], R is the wheel radius [m] and ω is the wheel angular speed [rad/s]. Notice how the slip ratio is a dimensionless variable.

Figure 2.3 shows the longitudinal tire dynamics neglecting the rolling force. From this scheme, the wheel dynamics equation is deduced as

$$\Gamma - RF_x = \alpha I_w, \tag{2.1}$$

where Γ is referred to the wheel torque [Nm], R and I_w are the wheel radius [m] and inertia [kgm^2], respectively, α is the wheel angular acceleration [rad/s^2] and F_x is the longitudinal force [N].

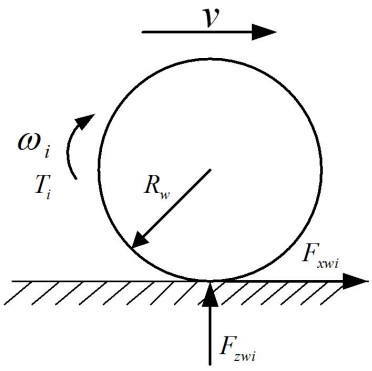

Figure 2.3: Longitudinal tire dynamics.

According to (2.1), it can be seen that the traction force can be computed and controlled by knowing the angular acceleration and wheel torque. The inverter, inverter controller and the resolver sensor allow to determine this parameters, so this force can be treated as a known state of the vehicle.

2.1.2 Lateral force

Lateral tire forces present a similar behavior as the longitudinal. These forces, that appear when the vehicle is cornering, depend on the normal force, but they also depend on the slip angle (instead of the slip ratio). This slip is defined as the angle between the direction of the tire and the velocity's direction (see Figure 2.4).

It can be seen in Figures 2.2 and 2.5, that both longitudinal and lateral forces present a linear behavior for small slip ratio and slip angle, respectively. This approximation can be done for a normal car, but not for a Formula Student vehicle, as high slip angles are reached. The cornering force keeps increasing until it reaches a maximum lateral force value and starts to slowly decrease. There are different curves represented in Figure 2.5

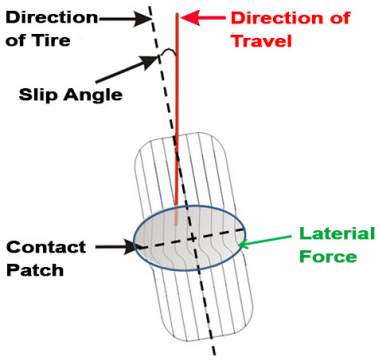

Figure 2.4: Tire slip angle [OldBastardRacing, 2018]

as the force the tire is able to do depend on the normal load applied, with higher values reached when higher is the vertical force. This means the orange graphic corresponds to a vertical load than the blue one.

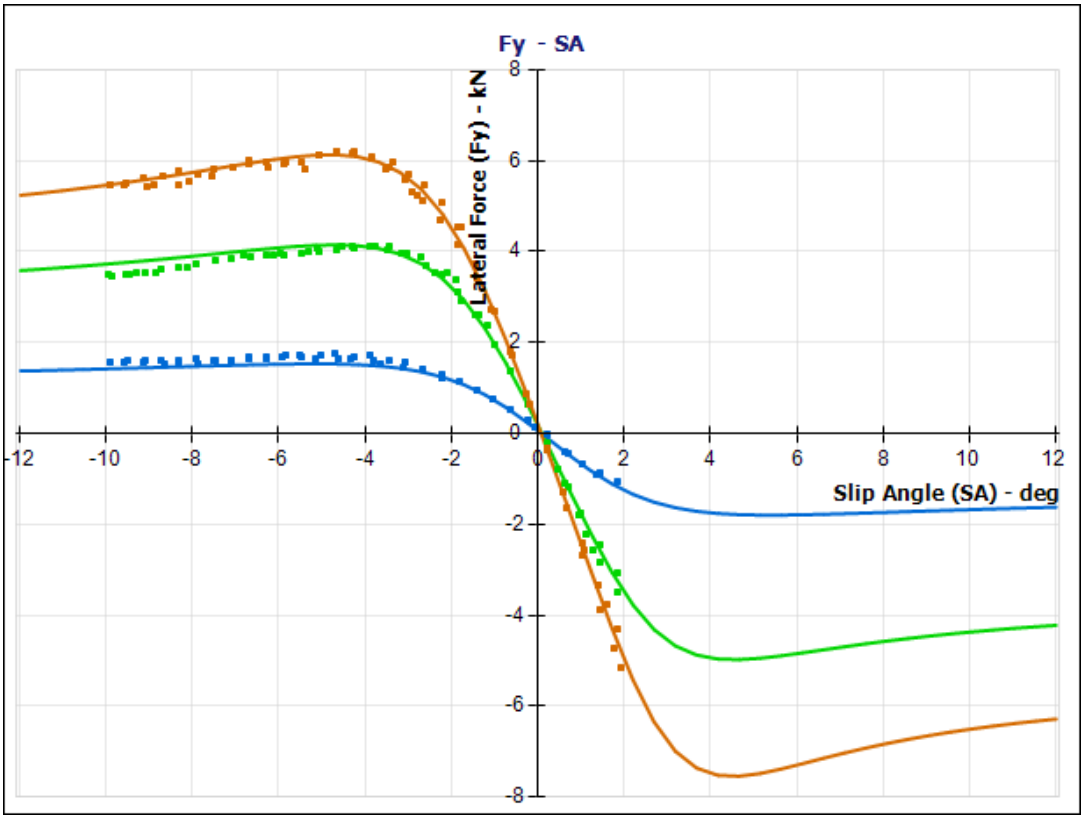

Figure 2.5: Lateral force against slip angle. Taken from [OptimumG, 2012]

There are some mathematical equations for this cornering force that try to model its behavior as a function of mainly the slip angle and the vertical load. The most used one is the Pacejka, also known as the magic formula [Pacejka, 2005]. The expression

$$F_y = D_y \sin[C_y \arctan(B_y\alpha_y - E_y(B_y\alpha_y - \arctan(B_y\alpha_y)))] + S_y \tag{2.2}$$

outputs the shape of the cornering force for a certain vertical load. Some parameters (Pacejka tire coefficients) are used to modify this curve, some of them with physics meanings (such as the cornering stiffness of the tire) and others are just scaling factors. In (2.2),

F_y is the lateral force [N] and α_y is the tire slip angle [rad]. The remaining parameters in (2.2) are

- D_y: Peak value,
- C_y: Shape factor,
- B_y: Stiffness factor,
- E_y: Curvature factor, and
- S_y: Vertical shift.

This Pacejka formula (2.2) describes the lateral force as a function of some physic parameters such as tire pressure, tire properties, vertical load, inclination angle, temperature, among others. In order to simplify this model, some variables will be neglected so the final equation depends just on the vertical force and the wheel slip angle.

2.1.3 Vertical force

As it has been commented previously, the vertical force applied to the tire has an important role and affects directly the longitudinal and lateral forces. This force, which appears as a consequence of the gravity and also because of the aerodynamic lift force, is the one that specifies the amount of longitudinal and lateral forces the tire can produce. While accelerating/braking or cornering, the related acceleration causes a load transfer (longitudinal or lateral, respectively). In Formula Student vehicles, this effect is quite important as high accelerations are reached while driving the car.

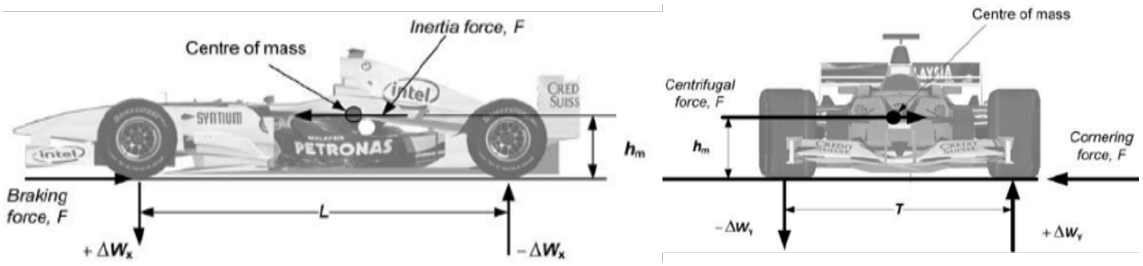

Figure 2.6: Car subjected to longitudinal and lateral acceleration [Seward, 2014]

Figure 2.6 shows a car subjected to longitudinal and lateral acceleration, in which ΔW_x and ΔW_y represent the load transfers [N] (longitudinal and lateral, respectively), h_m is the height between the tire contact point with the ground and the vehicle's center of gravity [m], m is the mass [kg], L is referred to the wheelbase [m] and T to the track [m]. By doing a force and momentum analysis of the car, the load transfer formulas can be obtained as follows:

$$\Delta W_x = \pm \frac{F h_m}{L} = \pm \frac{m a_x h_m}{L}, \tag{2.3}$$

$$\Delta W_y = \pm \frac{F h_m}{T} = \pm \frac{m a_y h_m}{T}. \tag{2.4}$$

In this project, the Formula Student vehicle studied (ETSEIB MotorSport) has four position sensors (one on each wheel). With those sensors and by knowing the suspensions dynamics, vertical loads are able to be computed.

2.2 Vehicle dynamics

In order to design a lateral force estimator, it is important to model the tire forces, but also to model the vehicle dynamics, so one can know and understand how the car is acting. In this section, just the fundamentals dynamics will be explained as they will be the most important to consider while modeling the car. To get this study done, the three axis system, yaw, roll and pitch, will be used. According to this system, the vehicle has three rotational degrees of freedom (one per axis) and three linear velocities.

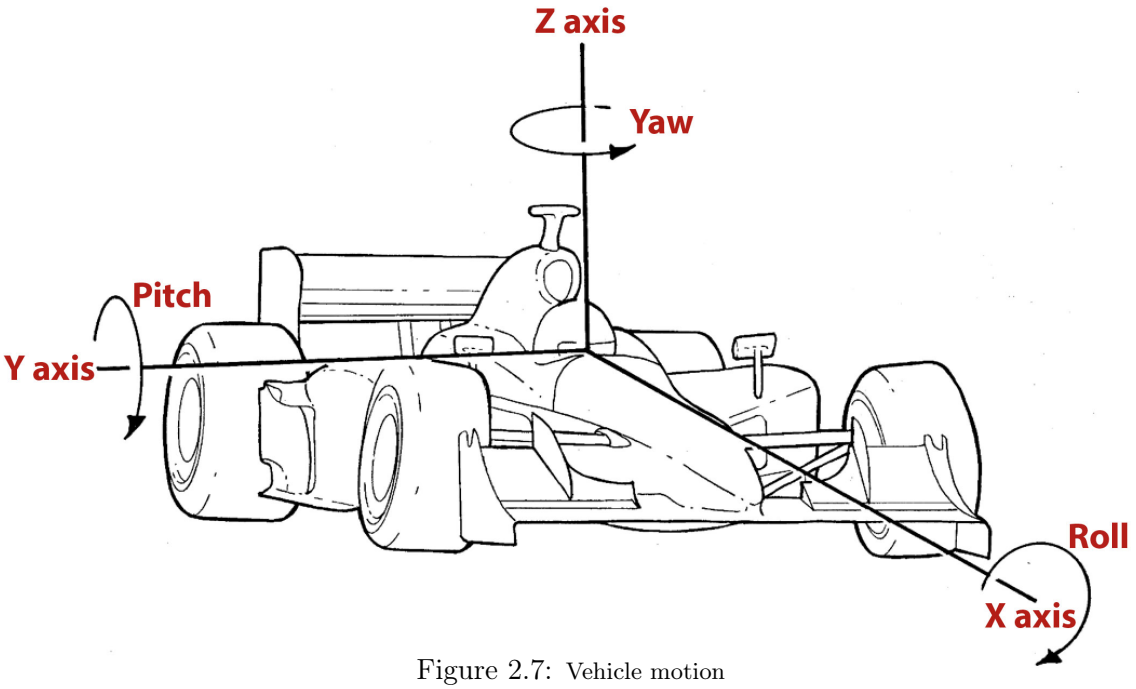

Figure 2.7: Vehicle motion

Pitch and roll rates appear while the car is accelerating/braking or cornering, respectively, as a consequence of load transfer (see Section 2.1.3). When car accelerates/brakes, there is a longitudinal load transfer towards the rear/front wheels, and, as a consequence, the car experiments pitch. The same happens with roll when the car is cornering. The rank of values that pitch or roll may reach depends on a lot of variables like suspension stiffness or the use of sway bars. To see more information about this dynamic structures, see [Milliken et al., 1995].

As it has been commented before, these dynamics behind a car are really complex so some simplifications must be done. In the case this work is studying, this roll and pitch angles will be neglected as low angles are reached (see Section 2.4). According to these simplifications, just yaw rate will be considered.

Apart from the tire forces, aerodynamic forces must be considered as it is a Formula Student vehicle and drag and downforce will appear while the car reaches considerable velocities. Downforce can increase braking and cornering performance highly as vertical load in the wheels also increases. On the other hand, drag force acts in the opposite direction of motion, which means that is a motion resistance force due to the relative velocity between the car and the wind. Both forces depend on the velocity and the area projected in that direction; only drag force in the longitudinal direction will be considered as either velocities in other directions or the area projected reach low values and can be neglected.

Finally, it is important to mention the steering model. As this project studies a racing car, the vehicle does not have the same steering ratio for each wheel. This statement means that for a certain steering wheel angle input, each front wheel turns a different angle (in this case, just front direction is considered). This set up is known as Ackermann configuration [Troupe, 2011]. There are two different types of Ackermann steering geometries: Pro and Anti Ackermann. The difference between them is that in the Pro-Ackermann configuration, the inner wheel turns more than the outer one, the opposite happens for the Anti-Ackermann (see Figure 2.8).

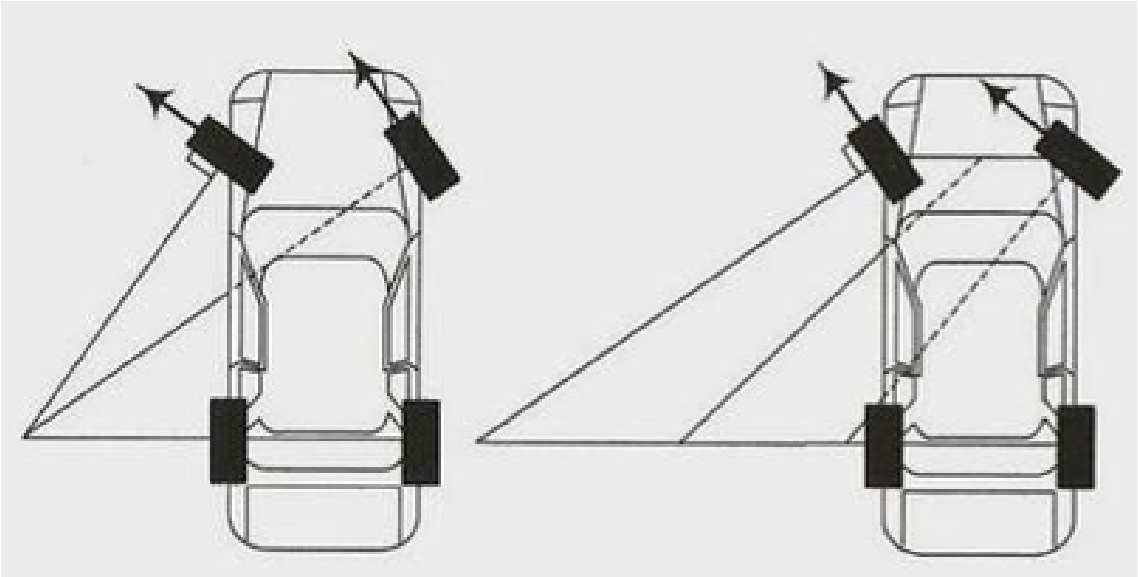

Figure 2.8: Pro-Ackermann (left) Anti-Ackermann (right) configurations

2.3 Vehicle dynamic model

Now that the basic dynamics have been presented, a model should be characterized in order to estimate the lateral force. There are some typical models, like the bicycle model [Abe, 2015], which can be either linear or nonlinear, that a lot of authors considered in this kind of studies. In this project, a four wheel traction car is considered, as it has a motor in each wheel. It is also important to notice that the vehicle is going to work in the nonlinear part of the force curves as, as mentioned before, high slip angles are reached while cornering. Finally, as the objective of this project is to estimate the lateral force of each tire, the model to be used should be a dynamic model instead of a kinematic model. Considering this facts, the linear model will not be the best option to model the vehicle. As a summary, a nonlinear four wheel drive (4WD) model will be used.

In Figure 2.9, a scheme of the 4WD model with the important parameters that will be taking into account is presented.

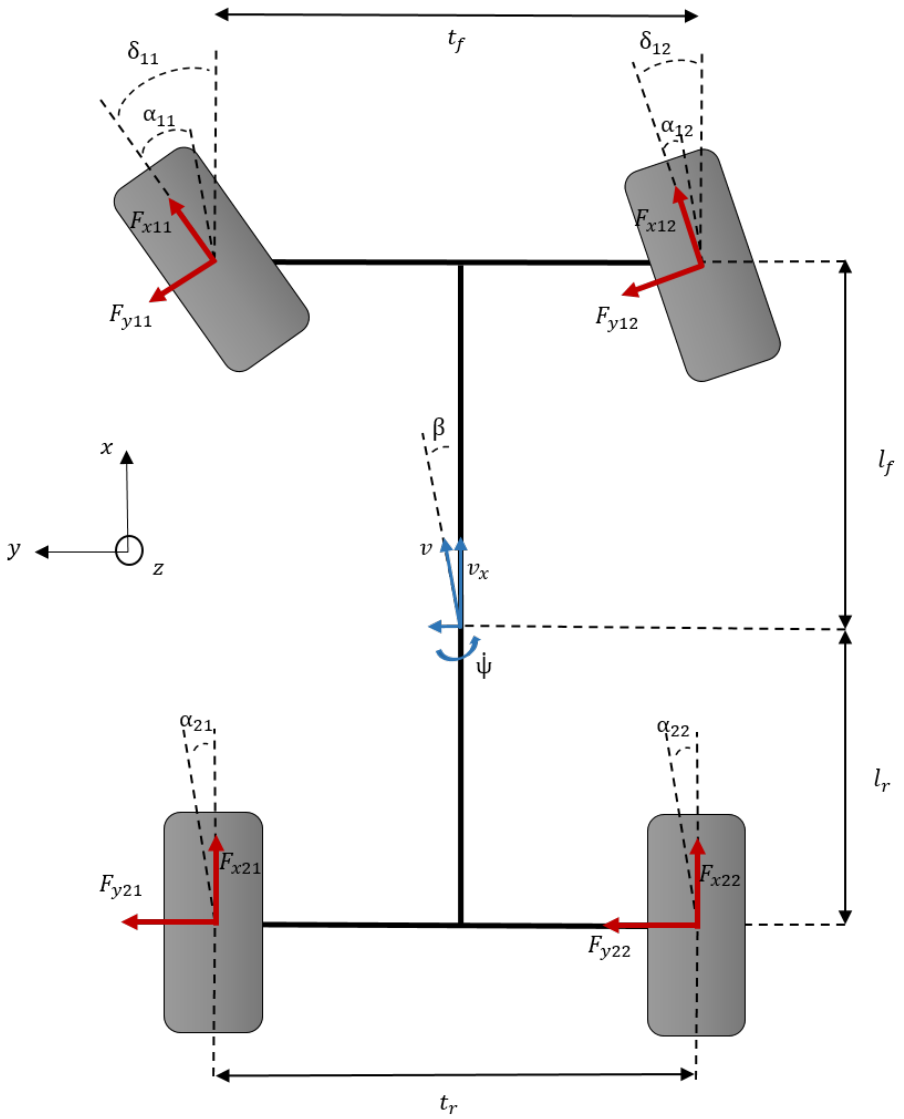

Figure 2.9: 4WD model representation scheme

In Figure 2.9, one can see the important parameters considered to model the car. Each tire has its corresponding longitudinal and lateral force (F_{xij}, F_{yij}) [N], as well as the slip angle α_{ij} [rad]. It is also important to notice the Ackermann geometry is incorporated as the steering angle is different in each wheel (δ_{1j}) [rad]. In the formulation used, the subscript i is for the front or rear axle (1 or 2, respectively), and the subscript j is for the left or right wheel (1 or 2, respectively). The center of gravity velocities v_y and v_x are represented [m/s], as well as the geometrical measures like the front and rear track, t_f and t_r [m], and the distance between the front/rear axle and the center of gravity, l_f and l_r [m]. Finally, the β angle is used to define the vehicle slip angle [rad].

According to this model representation, to find the differential equations that describe the motion and behavior of the car, the force and moment analysis should be done. For the kinematic analysis, it is important to consider that the base (vehicle) is rotating in the Z axis, as it has a yaw rotation ($\dot{\psi}$) [rad/s]. It is also important to add the drag force to this scheme.

2.3.1 Tire modeling

Now that the tire dynamics have been explained, it is time to model the tire. In order to obtain a differential equation of the lateral force, a first order transient model will be used [Rajamani, 2011], i.e.,

$$\tau \dot{F}_y + F_y = \bar{F}_y, \tag{2.4}$$

where τ is the time constant [s] and $\bar{F}_y$ is the quasi-static force [N], calculated by the magic formula defined in Section 2.1.2. In this model, τ can be defined as $\tau = \frac{C_\alpha}{K V_{wx}}$, where C_α is the tire cornering stiffness [N/rad] (ratio of cornering force over slip angle), K is the tire lateral stiffness [N/m] (ratio of lateral force over lateral displacement) and V_{wx} is the wheel longitudinal velocity [m/s]. The quotient of $\frac{C_\alpha}{K}$ is also known as relaxation length σ [m], whose values are usually between 0.12 and 0.45 m, with higher values corresponding to higher velocities and heavier loads. Some studies consider this length to have a similar value as the wheel radius. Equation (2.4) can be rewritten as

$$\frac{\sigma}{V_{wx}} \dot{F}_y + F_y = \bar{F}_y. \tag{2.5}$$

Considering the chassis as a rigid-body, the velocities at any point of the car can be obtained by knowing the center of gravity velocities. By doing this transformation, the longitudinal velocities in each wheels are

- Front left wheel:

$$v_{wx11} = \left(v_x - \frac{t_f \dot{\psi}}{2} \right) \cos(\delta_{11}) + \left(v_y + l_f \dot{\psi} \right) \sin(\delta_{11}). \tag{2.6}$$

- Front right wheel:

$$v_{wx12} = \left(v_x + \frac{t_f \dot{\psi}}{2} \right) \cos(\delta_{12}) + \left(v_y + l_f \dot{\psi} \right) \sin(\delta_{12}). \tag{2.7}$$

- Rear left wheel:

$$v_{wx21} = \left(v_x - \frac{t_f \dot{\psi}}{2} \right). \tag{2.8}$$

- Rear right wheel:

$$v_{wx22} = \left(v_x + \frac{t_f \dot{\psi}}{2} \right). \tag{2.9}$$

The nomenclature for all the variables and parameters is the same used in Figure 9 and v_{wxij} is used to refer to the longitudinal wheel velocity.

The final aspect to model is the quasi-static force. As commented previously, the model that is going to be used is a simplified version of the magic formula. In this version, the cornering force is only a function of the vertical load and the tire slip angle, neglecting other parameters such as the pressure. With this simplified version, just the slip and vertical forces should be computed. Although the vertical loads can be obtained by a suspension position sensor, an equation based on acceleration will be proposed for each force as follows:

$$F_{z11} = \frac{m(l_r a_z - h_c a_x)}{2l} - \frac{m h_c a_y (l_r a_z - h_c a_x)}{l t_f a_z}, \tag{2.10}$$

$$F_{z12} = \frac{m(l_r a_z - h_c a_x)}{2l} + \frac{m h_c a_y (l_r a_z - h_c a_x)}{l t_f a_z}, \tag{2.11}$$

$$F_{z21} = \frac{m(l_f a_z + h_c a_x)}{2l} - \frac{m h_c a_y (l_f a_z + h_c a_x)}{l t_r a_z}, \tag{2.12}$$

$$F_{z22} = \frac{m(l_f a_z + h_c a_x)}{2l} + \frac{m h_c a_y (l_f a_z + h_c a_x)}{l t_r a_z}. \tag{2.13}$$

According to its definition, the tire-slip angle can be calculated as

$$\alpha_{11} = \delta_{11} - \arctan\left(\frac{v_y + l_f \dot{\psi}}{v_x - t_f \dot{\psi}/2}\right), \tag{2.14}$$

$$\alpha_{12} = \delta_{12} - \arctan\left(\frac{v_y + l_f \dot{\psi}}{v_x + t_f \dot{\psi}/2}\right), \tag{2.15}$$

$$\alpha_{21} = -\arctan\left(\frac{v_y - l_r \dot{\psi}}{v_x - t_r \dot{\psi}/2}\right), \tag{2.16}$$

$$\alpha_{22} = -\arctan\left(\frac{v_y - l_r \dot{\psi}}{v_x + t_r \dot{\psi}/2}\right). \tag{2.17}$$

Now, both lateral and longitudinal forces can be calculated.

2.3.2 Vehicle model

Once the forces are modeled, by applying the second Newton's Law and the derivative of each variable, the differential equations of the vehicle's motion are

$$\begin{aligned}\dot{v}_x =& \dot{\psi} v_y + \frac{1}{m}[F_{x11}\cos(\delta_{11}) + F_{x12}\cos(\delta_{12}) + F_{x21} + F_{x22} - F_{y11}\sin(\delta_{11}) \\ & - F_{y12}\sin(\delta_{12}) - 0.5\rho_{air} C_D A_x v_x^2],\end{aligned}$$

$$\begin{aligned}\dot{v}_y =& -\dot{\psi} v_x + \frac{1}{m}[F_{y11}\cos(\delta_{11}) + F_{y12}\cos(\delta_{12}) + F_{y21} + F_{y22} + F_{x11}\sin(\delta_{11}) \\ & + F_{x12}\sin(\delta_{12})],\end{aligned}$$

$$\begin{aligned}\ddot{\psi} =& \frac{1}{I_z}\{l_f[F_{y11}\cos(\delta_{11}) + F_{y12}\cos(\delta_{12}) + F_{x11}\sin(\delta_{11}) + F_{x12}\sin(\delta_{12})] - l_r[F_{y21} \\ & + F_{y22}] + \frac{t_f}{2}[F_{y11}\sin(\delta_{11}) - F_{x11}\cos(\delta_{11}) - F_{y12}\sin(\delta_{12}) + F_{x12}\sin(\delta_{12})] \\ & + \frac{t_r}{2}[F_{x22} - F_{x21}]\}.\end{aligned}$$

Here, I_z is referred to the inertia on the Z axis [kgm^2], ρ_{air} to the air density [kg/m^3], C_D to the vehicle drag coefficient and A_x to the frontal area of the car [m^2]. Combining the four lateral force differential equations from (2.5) with this three motion equations, a

seven-states system can be designed. A state-space representation of this nonlinear model is proposed as

$$\begin{aligned} x^+ &= f(x, u, d, \omega), \\ y &= h(x, v), \end{aligned}$$

where x^+ is the evolution of the state variable, $x \in \mathbb{R}^{n_x}$ is the state vector, $u \in \mathbb{R}^{n_u}$ is the input vector, $y \in \mathbb{R}^{n_y}$ is the output vector, $d \in \mathbb{R}^{n_d}$ are the disturbances and $\omega \in \mathbb{R}^{n_\omega}$ and $v \in \mathbb{R}^{n_v}$ are the model and measure noises, respectively. Moreover, $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_d} \times \mathbb{R}^{n_\omega} \longmapsto \mathbb{R}^{n_x}$ is the state mapping and $h : \mathbb{R}^{n_x} \times \mathbb{R}^{n_v} \longmapsto \mathbb{R}^{n_y}$ is the output mapping.

In this case, as mentioned before, a seven-states system is proposed, so the state vector would be

$$x = [v_x, v_y, \dot{\psi}, F_{y11}, F_{y12}, F_{y21}, F_{y22}]^T.$$

According to the variables needed and the ones obtainable, the input vector is given by

$$u = [\delta_{11}, \delta_{12}, F_{x11}, F_{x12}, F_{x21}, F_{x22}, F_{z11}, F_{z12}, F_{z21}, F_{z22}]^T.$$

The vector field f is obtained by adding the differential equations found in Sections 2.4.1 and 2.4.2. Its corresponding discretization and the definition of the vector field h will be developed in Section 3.3.

2.3.3 Disturbance and noise modeling

The final variables to model are the disturbance and noises (model and measure noise). These noises do not depend on any other variable such as could be velocity or yaw rate and, because of this, they will be treated as white noises with their corresponding variation. The possibility of having a disturbance input like a bump will also be treated as noise. This means that d and ω can all be considered a single white noise with a certain variation that will have to be found. According to the definition of white noise, the functions that define the state-space representation can be rewritten as

$$x^+ = f(x, u) + \omega, \tag{2.18a}$$

$$y = h(x) + v. \tag{2.18b}$$

These noises, with no correlation between them, will be modeled following a normal probability distribution with mean equal to zero and a variance σ_x^2 distributed in a diagonal matrix Q (for the model noise) and R (for the measurement noise). Defining them as a diagonal matrix means that the noise variation of one variable is not correlated with any other variables.

In the case of the array R, the sensors used (see Section 3.1) have a standard deviation known, so the variance of these measures can be obtained by squaring this deviation. The matrix R, which is a 5×5 array according to the sensors available, is then

$$R = \begin{pmatrix} \sigma^2_{v_x} & 0 & 0 & 0 & 0 \\ 0 & \sigma^2_{v_y} & 0 & 0 & 0 \\ 0 & 0 & \sigma^2_{\dot{\psi}} & 0 & 0 \\ 0 & 0 & 0 & \sigma^2_{a_x} & 0 \\ 0 & 0 & 0 & 0 & \sigma^2_{a_y} \end{pmatrix} = \begin{pmatrix} 0.1^2 & 0 & 0 & 0 & 0 \\ 0 & 0.1^2 & 0 & 0 & 0 \\ 0 & 0 & 0.5^2 & 0 & 0 \\ 0 & 0 & 0 & 1^2 & 0 \\ 0 & 0 & 0 & 0 & 1^2 \end{pmatrix}.$$

On the other hand, the Q matrix cannot be determined this easily as the variance of the dynamic model is not known. To obtained it, a set of simulations will be done (Chapter 4) and the different diagonal elements will be tuned to define this matrix according to the simulation results. As there are seven state variables, the size of Q must be 7×7, i.e.,

$$Q = \begin{pmatrix} \sigma^2_{f_1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma^2_{f_2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma^2_{f_3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma^2_{f_4} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma^2_{f_5} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma^2_{f_6} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma^2_{f_7} \end{pmatrix}.$$

2.4 Model validation

In this section, the model proposed in (2.18) is tested with the simulation software IPG Automotive [GmbH, 2020] so it can be validated and used in the lateral force estimator. The four equations proposed to calculate the different slip angles, (2.14) to (2.17), as well as the seven state process model in (2.18) will be simulated and compared with the resultant variables computed by the IPG. The model proposed is a continuous representation and it should be in discrete-time form in order to develop the estimator. According to Chapter 3, where the discretization process is explained, the model will be validated using a sample time of 20 ms and the Euler method.

Figure 2.10 shows the different slip angles (obtained with (2.14) to (2.17)) against the IPG software results of an Autocross event (see Section 4.1).

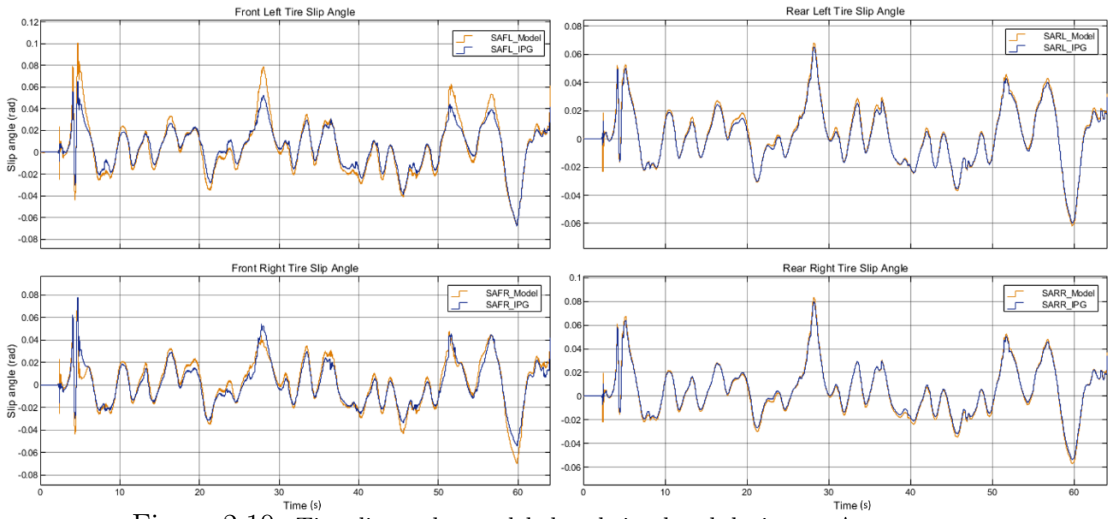

Figure 2.10: Tire slip angles modeled and simulated during an Autocross run.

The modeled slip angles and the simulated ones are similar with low relative errors. It can be noticed how the front slip angles (both left and right) have a higher error due to the addition of the steering angle of the wheel. These steering angles are currently being modeled with a MATLAB Look-Up-Table with the current Ackermann configuration transformation. The rear wheels, as they do not have any steering geometry, present a lower mean error than the front ones tires. To get a better perception of the results, the mean errors and standard deviation are analysed in Table 2.1.

Table 2.1: Tire slip angles relative errors.

Slip angle	Max error (%)	Mean error (%)	Std error (%)
Front left	8.1379	1.2259	4.7785
Front right	6.9253	1.0647	3.8936
Rear left	3.1186	0.4675	2.4656
Rear right	2.7457	0.4275	2.0494

The errors obtained are definitely small enough to consider that the slip angle per tire is suitably modeled. Finally, all different states and their respective IPG results are studied.

Figure 2.11 shows how the modeled longitudinal velocity fits with the real velocity. The error between them has a mean of the 0.9979% and a standard deviation of the 0.0334%. The relative error has a low dispersion and mean with a maximum value of the 1.0232%. The error increases when the vehicle starts getting speed and then it is stabilized.

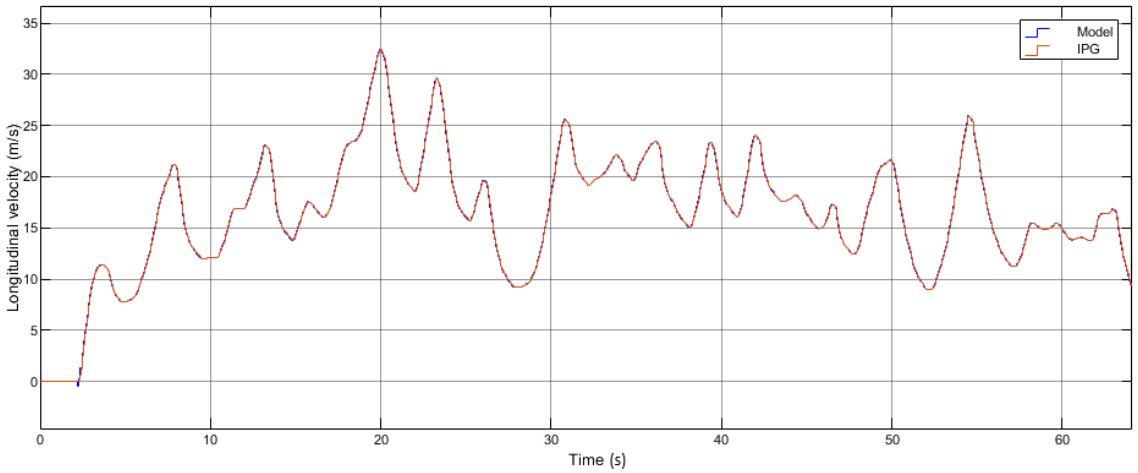

Figure 2.11: Longitudinal velocity modeled and simulated during an Autocross run.

In Figure 2.12, the lateral velocity modeled and simulated are displayed. The maximum error obtained is 1.3562%, with a mean of the 0.2638% and a standard deviation of the 0.6878%. The mean error is lower than he longitudinal speed one, but it has a higher standard deviation as it can be noticed in Figure 2.12.

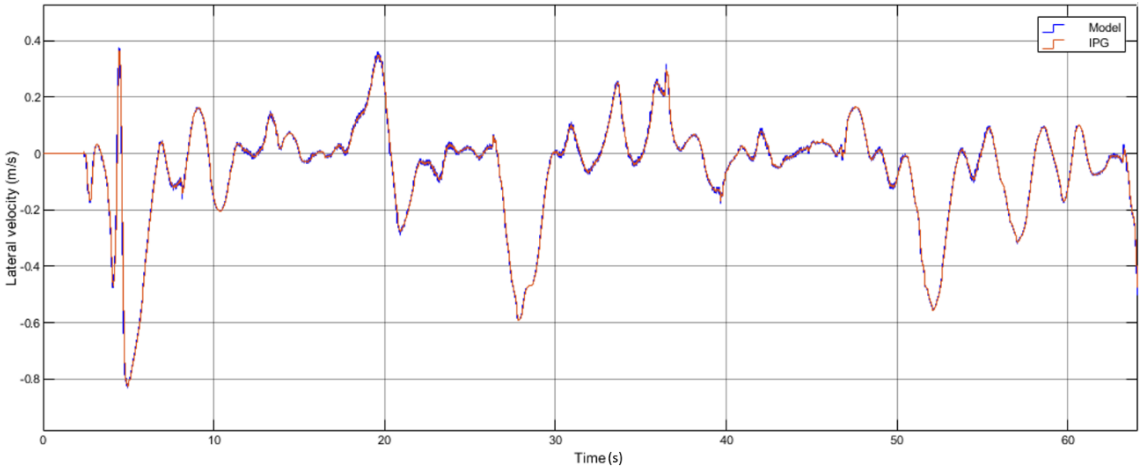

Figure 2.12: Lateral velocity modeled and simulated during an Autocross run.

Figure 2.13 represents the yaw rate obtained with the simulation software IPG and the modeled one for an Autocross run (see Section 4.1). The errors are similar to the longitudinal and lateral speed, with a mean of 0.2468% and a standard deviation of 0.8876%, which are low enough to consider that all three variables are suitably modeled. The yaw rate has a maximum relative error of 2.7512%, which is reached in the beginning of the Autocross run. All these error values are summarized in Table 2.2.

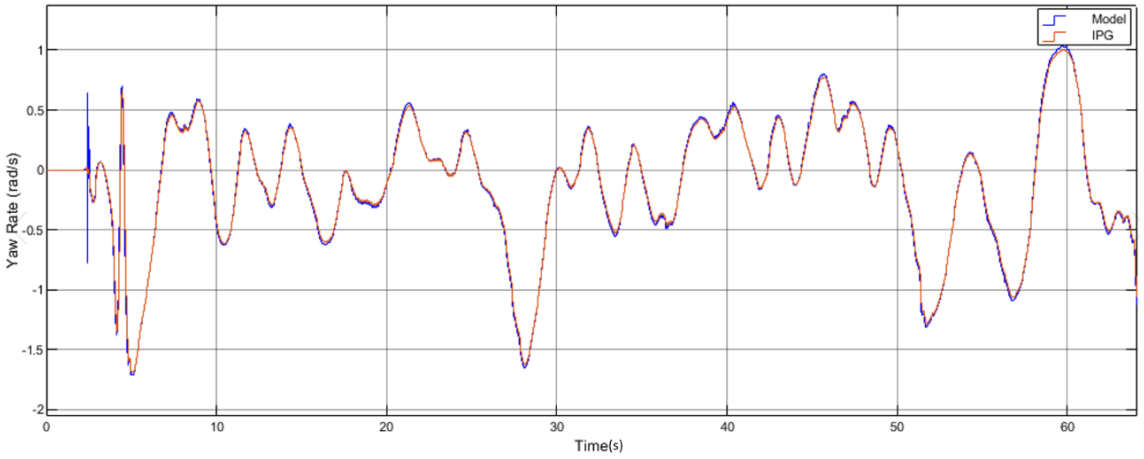

Figure 2.13: Yaw Rate modeled and simulated during an Autocross run.

In Figure 2.14, the four lateral forces with their corresponding IPG value are represented and Table 2.2 summarizes all the errors. Lateral force errors have a low mean and standard deviation, which means it is a proper model. The maximum values of these errors are for the lateral forces as it is a dynamic model with some simplifications done. They are reached when the cornering forces have their maximum values as the nonlinearities of the curve (Figure 2.5) are higher. As it happened with the slip angles, the front wheels have a higher mean error caused due to the slip angle error commented previously. Despite this issue, these mean errors are low and it can be considered that cornering forces are properly modeled.

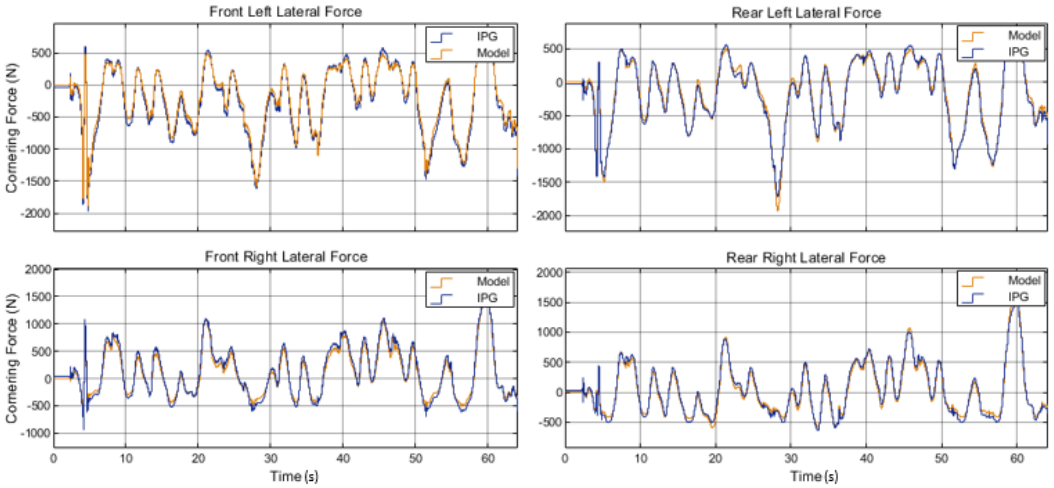

Figure 2.14: Tire lateral forces modeled and simulated during an Autocross run.

Table 2.2: State variables errors

Variable	Max error (%)	Mean error (%)	Std error (%)
Longitudinal velocity	1.0232	0.9979	0.0334
Lateral velocity	1.3562	0.2638	0.6878
Yaw rate	2.7512	0.2468	0.8876
Front left F_y	4.5692	1.1076	3.3181
Front right F_y	3.2521	0.8078	1.3116
Rear left F_y	2.4168	0.5171	1.7802
Rear right F_y	2.2333	0.5173	1.6555

It is also important to analyze the different values the neglected kinematic variables can reach, such as the roll and pitch angles. In the following pages, the analysis of the roll and pitch angles done to consider the possibility of neglecting them is exposed. It has been done by analyzing the IPG simulations software data. A Skidpad event (see Section 4.1) has been studied.

When cornering or accelerating, the sensor used to measure either velocity or acceleration gets an additional component as its base is rotating. For example, if there is a high roll angle in a moment, the lateral acceleration measured has also a gravity component due to this angle. This axis rotation is represented in Figure 2.15.

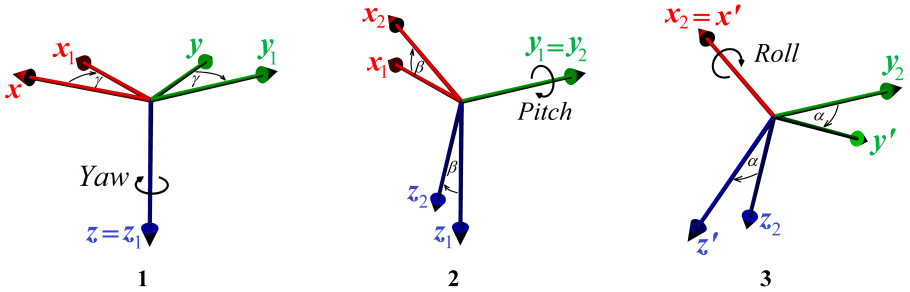

Figure 2.15: Three axis sensor rotation.

This additional component should be deducted in order to get the real measure in that axis. It can be noticed how this axis rotation has a small effect, that can be neglected, if these angles have low values. In order to assume it to be zero, pitch and roll angles should reach small values during all the maneuver. Figure 2.16 shows the pitch angle during the Skidpad.

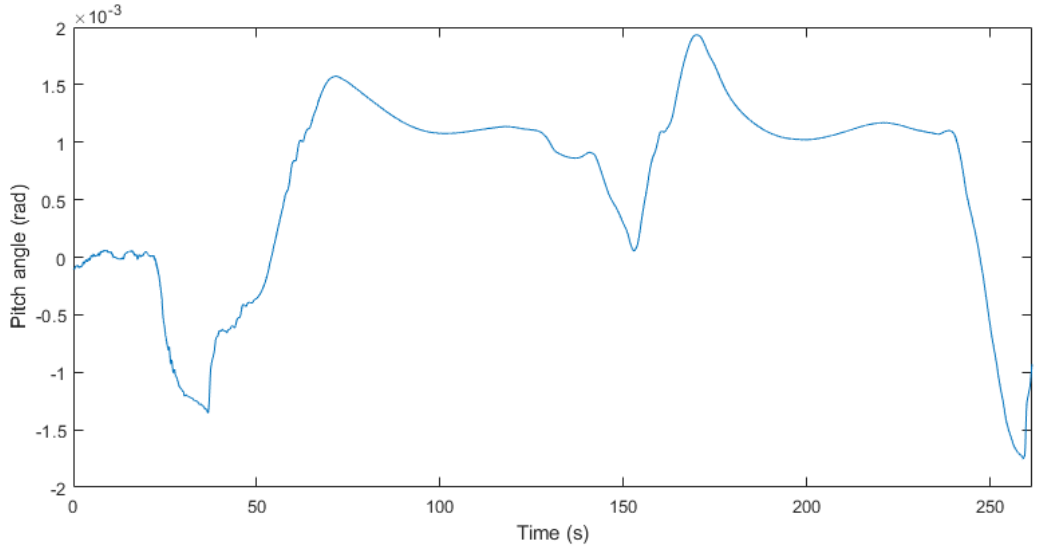

Figure 2.16: Pitch angle during a Skidpad event.

The pitch angle affects directly the longitudinal component of the velocity or acceleration. It can be noticed how the maximum value for this angle is 1.1932×10^{-3}rad and an average angle of 9.4946×10^{-4}rad. According to these small errors, the component added because of the pitch can be properly neglected. In Figure 17, the roll angle during a Skidpad event is represented.

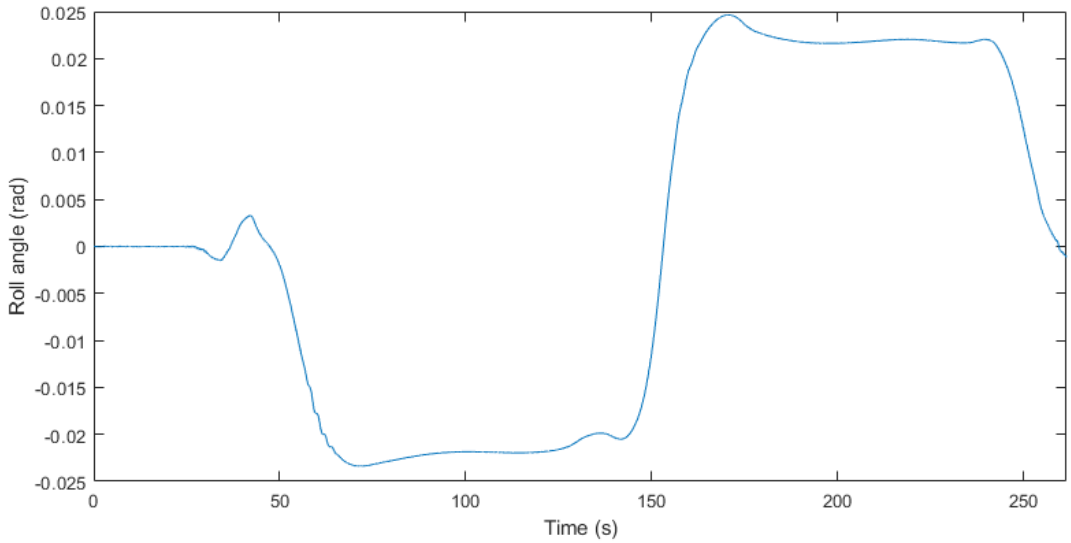

Figure 2.17: Roll angle during a Skidpad event.

The roll angle, which affects directly the lateral component measured by the sensor, reaches a maximum value of 0.0246rad and an average angle of 0.0159rad. Although it is slightly higher than the pitch angle, it is still low enough to consider it null. In conclusion, both,

roll and pitch angles, can be neglected.

To sum up, it is shown the simplified system is suitably modeled and the different variables neglected fulfill the different characteristics necessary to consider them null.

2.5 Summary

In this chapter, a brief introduction to the vehicle dynamics was exposed. The different tire forces were presented and modeled as well as the vehicle plant. To do so, some variables such as the roll or pitch angles were studied and neglected to simplify a real complex model. In order to use the state-space representation proposed, measurement and process noise were simplified as white noise, which follow a normal probabilistic distribution with null mean and a different standard deviation for each variable. Finally, a discrete version of the model is tested and validated so it can be used to design the tire lateral force estimator. In Chapter 3, the different measurement inputs available will be presented and the observers will be completely defined. Four different estimators will be studied and designed so they can be compared later.

Chapter 3

Lateral force estimation

3.1 Data acquisition

After modeling the car and before starting to design the estimator, it is important to know which variables can be measured directly by the use of sensors and which of them should be calculated using these variables. Some information about this topic has been briefly explained in previous sections. In this chapter, a deeper information will be given so the estimator inputs can be definitely defined.

First of all, an inertial navigation system (INS) is used in order to measure all the kinematic variables and the longitudinal and lateral velocity as well as the three components of the acceleration can be obtained with the sensor. Although the sensor available for the team (SBG ELLIPSE2-N-G4A3-B2) only starts sending information after a total velocity of 15km/h has been reached, after it happens it will start working perfectly even though the velocity is lower than this border limit, so there will not be any problem while getting all the input information. The INS also gives the yaw, roll and pitch rate. The values of the roll and pitch angles have been previously studied and the assumption of neglecting them has been decided (as commented in Section 2.4). The yaw rate, velocities and accelerations will be the measure inputs used in this project to estimate the lateral tire forces per wheel. In case of failure of the sensor, the team also has available an inertial measurement unit (IMU) that can provide the team with the accelerations (longitudinal, vertical and lateral) and with the yaw rate.

On the other hand, some potentiometers have been installed in each suspension geometry. With this sensor, the position and compression of each spring is always known and by knowing the spring's stiffness, the vertical load per wheel can be determined. They also can be calculated using accelerations as shown in (2.10) to (2.13).

Finally, the inverter and the resolver provide the motor torque and rotational speed, which can be used to calculate the wheel torque by using the transmission ratio (in this car, equal to 11,96, dimensionless). These parameters can also be used as input measures to compute the longitudinal tire force. All the data and measures are transmitted by CAN through the Electronic Control Unit (ECU).

3.2 Estimator design

The state and input vectors have been already established in Section 2.3.2, and just the data measured vector remains to be defined. According to the sensors available, these

variables can be the velocities and accelerations and the measurements vector is defined as

$$y = [v_x, v_y, \dot{\psi}, a_x, a_y]^T.$$

Figure 3.1 shows an open-loop scheme of the tire lateral force estimator proposed in this project. The *Pacejka* block is referred to the magic formula calculation defined in (2.2).

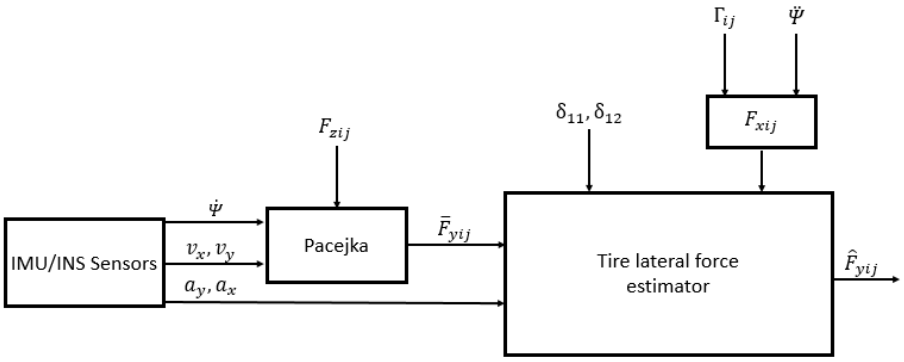

Figure 3.1: Tire lateral force estimator scheme.

The nonlinear model is given by

$$\dot{x} = \begin{pmatrix} \dot{v_x} \\ \dot{v_y} \\ \ddot{\psi} \\ \dot{F}_{y11} \\ \dot{F}_{y12} \\ \dot{F}_{y21} \\ \dot{F}_{y22} \end{pmatrix} = f(x,u) + \omega,$$

$$= \begin{pmatrix} \dot{\psi} v_y + \frac{1}{m}[F_{x11}\cos(\delta_{11}) + F_{x12}\cos(\delta_{12}) + F_{x21} + F_{x22} - F_{y11}\sin(\delta_{11}) \\ -F_{y12}\sin(\delta_{12}) - 0.5\rho_{air} C_D A_x v_x^2] \\ -\dot{\psi} v_x + \frac{1}{m}[F_{y11}\cos(\delta_{11}) + F_{y12}\cos(\delta_{12}) + F_{y21} + F_{y22} + F_{x11}\sin(\delta_{11}) \\ +F_{x12}\sin(\delta_{12})] \\ \frac{1}{I_z}[l_f[F_{y11}\cos(\delta_{11}) + F_{y12}\cos(\delta_{12}) + F_{x11}\sin(\delta_{11}) + F_{x12}\sin(\delta_{12})] \\ -l_r[F_{y21} + F_{y22}] + \\ +\frac{t_f}{2}[F_{y11}\sin(\delta_{11}) - F_{x11}\cos(\delta_{11}) - F_{y12}\sin(\delta_{12}) + F_{x12}\sin(\delta_{12})] \\ +\frac{t_r}{2}[F_{x22} - F_{x21}]] \\ \frac{v_{wx11}}{\delta_{11}}(\bar{F}_{y11} - F_{y11}) \\ \frac{v_{wx12}}{\delta_{12}}(\bar{F}_{y12} - F_{y12}) \\ \frac{v_{wx21}}{\delta_{21}}(\bar{F}_{y21} - F_{y21}) \\ \frac{v_{wx22}}{\delta_{22}}(\bar{F}_{y22} - F_{y22}) \end{pmatrix},$$

and the measurements as follows:

$$y = \begin{pmatrix} v_x \\ v_y \\ \dot{\psi} \\ a_x \\ a_y \end{pmatrix} = h(x) + v,$$

$$= \begin{pmatrix} v_x \\ v_y \\ \dot{\psi} \\ \frac{1}{m}[F_{x11}\cos(\delta_{11}) + F_{x12}\cos(\delta_{12}) + F_{x21} + F_{x22} - F_{y11}\sin(\delta_{11}) \\ -F_{y12}\sin(\delta_{12}) - 0.5\rho_{air}C_D A_x v_x^2] \\ \frac{1}{m}[F_{y11}\cos(\delta_{11}) + F_{y12}\cos(\delta_{12}) + F_{y21} + F_{y22} + F_{x11}\sin(\delta_{11}) \\ +F_{x12}\sin(\delta_{12})] \end{pmatrix}.$$

This state space representation of the nonlinear system is in the continuous-time domain. Real systems cannot be computed continuously and they should be discretized. To get this transformation done, the Euler first-order method [Ascher and Petzold, 1998] is used in order to obtain a proper discretization. Being a first-order method means that it approaches a curve into a straight line, neglecting high-order terms. A derivative can be expressed as

$$\dot{x} = \frac{\Delta x}{\Delta t} = \frac{x_k - x_{k-1}}{t_s},$$

where x_k is the state variable at a time k and t_s is the corresponding sampling time. Note that a first initial condition (x_0) is needed when applying this method. The last aspect to define to have the system properly discretized is the definition of the sampling time. In a Formula Student vehicle there are a lot of components that work at different frequencies and it is important to establish correctly this sampling period. This period should be the highest one, so every component sends the messages and data at the same frequency. In the case studied in this project, the information travels at a 50 Hz frequency, which corresponds to a 20ms period. This is the main ECU frequency. With all these variables defined, the system can be now expressed as

$$x_k = f^*(x_{k-1}, u_k) + \omega_{k-1},$$
$$y_k = h(x_k) + v_k,$$
$$x_k = f^*(x_{k-1}, u_k) + \omega_{k-1} = \begin{pmatrix} v_{x,k-1} + t_s\dot{v}_x \\ v_{y,k-1} + t_s\dot{v}_y \\ \dot{\psi}_{k-1} + t_s\ddot{\psi} \\ F_{y11,k-1} + t_s\dot{F}_{y11} \\ F_{y12,k-1} + t_s\dot{F}_{y12} \\ F_{y21,k-1} + t_s\dot{F}_{y21} \\ F_{y22,k-1} + t_s\dot{F}_{y22} \end{pmatrix}.$$

Although the Euler method can be considered as the simplest one to compute and does not introduce a high computational cost, the Tustin rule [Iserles, 2009] will also be tried. This method requires a higher computational cost but it also preserves stability after the transformation to the discrete domain. The difference between both methods will also be shown in Chapter 4, with all the graphics and tables of the results of each method. In a nonlinear state-space representation, the Tustin rule discretizes the system as follows:

$$x_k = x_{k-1} + \frac{t_s}{2}[f(x_{k-1}) + f(x_k)]. \tag{3.1}$$

It can be noticed how the unknown state is in both sides of the equation. To solve it, the Newton-Raphson method will be used considering that it is a seven state vector. The equation can be redefined as

$$F(x_k) = x_{k-1} + \frac{t_s}{2}[f(x_{k-1}) + f(x_k)] - x_k = 0, \tag{3.2}$$

and the following algorithm of the Newton's method is executed after. A first $x_n = x_0$ guess is chosen and the algorithm is computed until it converges, i.e.,

$$x_n = x_{n-1} - J_{n-1}^{-1} F(x_{n-1}), \tag{3.3}$$

where J is the Jacobian matrix of F function. This method will only be used in the Luenberger estimator proposed in Section 3.2.1 as the results are similar with both discretization rules.

3.2.1 Luenberger Estimator

The first option suggested is a nonlinear Luenberger observer [Luenberger, 1971]. This estimator mechanics are quiet simple but also robust. Starting from the nonlinear state space, it uses a feedback matrix L to properly estimate the unknown state.

$$\dot{\hat{x}} = f(\hat{x}, u) + L(\hat{x}, u)(y - \hat{y}), \tag{3.4a}$$
$$\hat{y} = h(\hat{x}, u), \tag{3.4b}$$

where $\hat{x}$ is the state estimate. With some algebra and considering the local linearisation, it can be deduced that the estimation error ($e_{\hat{x}} = x - \hat{x}$) is

$$\dot{e}_{\hat{x}} = [A(\hat{x}, u) - L(\hat{x}, u)C(\hat{x}, u)]e_{\hat{x}}, \tag{3.5}$$

where $A = \frac{\partial f}{\partial x}(x)$ and $C = \frac{\partial h_i}{\partial x_j}(x)$ are the Jacobian matrices of f and h vector fields, respectively. This error must tend to zero when time increases to obtain a proper state estimation. The feedback matrix L acts like a gain which weights the measurement innovation y_k at each sampling time k. In this particular case, a constant feedback matrix is chosen so the weight given to the innovation remains always constant. That chosen matrix is

$$L = \begin{pmatrix} 0.5 & 0.01 & 0.01 & 0.01 & 0.01 \\ 0.01 & 0.9 & 0.01 & 0.01 & 0.01 \\ 0.01 & 0.01 & 0.5 & 0.01 & 0.01 \\ 0.01 & 0.01 & 0.01 & 0.01 & 15 \\ 0.01 & 0.01 & 0.01 & 0.2 & 15 \\ 0.01 & 0.2 & 0.01 & 0.01 & 15 \\ 0.01 & 0.5 & 0.5 & 0.01 & 15 \end{pmatrix}.$$

3.2.2 Extended Kalman Filter

The second observer proposed for the lateral force estimation is the one known as Kalman Filter [Bishop et al., 2001]. As mentioned before, this model is completely nonlinear and the linear Kalman Filter will not be even considered. This estimator, the Extended Kalman Filter (EKF) [Wilkin et al., 2005] [Doumiati et al., 2010], is a variation that solves the model linearizing it with the Jacobian matrices of f and h vector fields in the actual state.

The Kalman Filter works in two steps, a prediction and an update step. In the first one, the estimator predicts the future state according to the previous state and the actual inputs of the system with the model equations (f vector field) defined. After this first calculation, the measurements of the actual state are compared with the prediction and the state is updated according to the variance matrices Q and R defined for the noises. This statement means that if the measurement noise matrix (R) is much lower than the model matrix (Q), the value of the measure will have more weight than the predicted state, as the sensor gets a better value of the real state. This estimator also works with a covariance matrix P of the main state. Figure 3.2 shows an scheme of the process followed by the Kalman Filter, where it can be noticed that an initial state and covariance matrix must be defined to start estimating the next states.

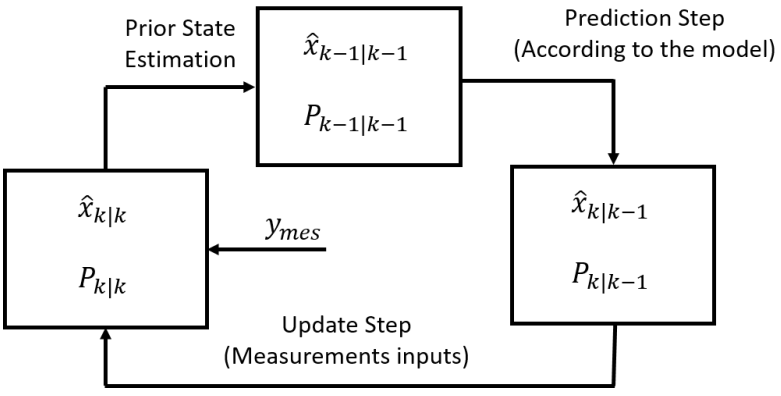

Figure 3.2: Kalman Filter scheme.

Starting from a previous state, in the prediction step a predicted covariance matrix and state are computed as follows:

$$\hat{x}_k = f^*(\hat{x}_{k-1}, u_k), \tag{3.6}$$

$$P_k = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T, \tag{3.7}$$

where A_k and W_k are the Jacobian matrices, defined as

$$A_{i,j} = \frac{\partial f_i^*}{\partial x_j}(\hat{x}, u),$$
$$W_{i,j} = \frac{\partial f_i^*}{\partial \omega_j}(\hat{x}, u).$$

According to this case study, how the variables are defined and because the noise is modeled considering it as white and non correlated with the variables, (3.7) can be rewritten and the prediction step is

$$\hat{x}_k = f^*(\hat{x}_{k-1}, u_k),$$
$$P_k = A_k P_{k-1} A_k^T + Q_{k-1}.$$

Here, $\hat{x}_k$ and P_k are corrected in the update step using the measured data obtained. The final estimated state can be calculated as

$$K_k = P_k C_k^T (C_k P_k C_k^T + V_k R_k V_k^T)^{-1},$$
$$\hat{x}_k = \hat{x}_k + K_k(y_k - h(\hat{x}_k)),$$
$$P_k = (I - K_k C_k) P_k,$$

where C_k and V_k are the Jacobian matrices

$$C_{i,j} = \frac{\partial h_i}{\partial x_j}(\hat{x}),$$

$$V_{i,j} = \frac{\partial h_i}{\partial v_j}(\hat{x}).$$

Here, K_k is the Kalman gain, which gives the corresponding weight to the measure or model, y_k is the measure done in a time k and I is the identity matrix of suitable dimensions. The same that has been done with the Jacobian matrix W_k can be done with the Jacobian matrix V_k, and the update step is rewritten as follows:

$$K_k = P_k C_k^T (C_k P_k C_k^T + R_k)^{-1},$$
$$\hat{x}_k = \hat{x}_k + K_k(y_k - h(\hat{x})),$$
$$P_k = (I - K_k C_k) P_k.$$

To start the algorithm, the initial values should be defined. In the beginning of the maneuver, the car will be stopped, with no velocity or acceleration. It is simple to notice how all the variables of the state vector should be zero before starting driving. However, the null values of the initial state may result into an unstable predicted state. Moreover, to compare between estimators it is important to consider the convergence time, so although the real initial state is zero, in these estimators a different initial state will be considered. It is proposed the state $x_0 = [1, 1, 0.2, 5, 5, 5, 5]^T$.

Apart from the state vector, the initial covariance matrix P_k should be determined. This matrix represents how disperse can be the predicted value from the real one. The diagonal vector shows the variance of the seven state variables and the other elements represent the covariance of those index variables. This statement means that the element i, j shows the covariance between element in the row i and element in the j column. To start the simulations, the following matrix has been chosen:

$$P_0 = \begin{pmatrix} 0.1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10 \end{pmatrix}.$$

After running some tests, the entries of the Q matrix are chosen to be

$$Q = \begin{pmatrix} 0.1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10 \end{pmatrix}.$$

This estimator might not work properly if the model is highly nonlinear. It also may have a high computational cost because of the complex Jacobian matrices that should be calculated at every iteration. These Jacobian matrices are:

$$A = \begin{pmatrix} A_{11} & t_s\dot{\psi} & t_s v_y & -\frac{t_s}{m}\sin(\delta_{11}) & -\frac{t_s}{m}\sin(\delta_{12}) & 0 & 0 \\ -t_s\dot{\psi} & 1 & -t_s v_x & \frac{t_s}{m}\cos(\delta_{11}) & \frac{t_s}{m}\cos(\delta_{12}) & \frac{t_s}{m} & \frac{t_s}{m} \\ 0 & 0 & 1 & A_{34} & A_{35} & -l_r\frac{t_s}{I_z} & -l_r\frac{t_s}{I_z} \\ A_{41} & A_{42} & A_{43} & 1-\frac{t_s v_{x11}}{\sigma_{11}} & 0 & 0 & 0 \\ A_{51} & A_{52} & A_{53} & 0 & 1-\frac{t_s v_{x12}}{\sigma_{12}} & 0 & 0 \\ A_{61} & 0 & A_{63} & 0 & 0 & 1-\frac{t_s v_{x21}}{\sigma_{21}} & 0 \\ A_{71} & 0 & A_{73} & 0 & 0 & 0 & 1-\frac{t_s v_{x22}}{\sigma_{22}} \end{pmatrix},$$

with

$$A_{11} = 1 - \frac{t_s}{m}\rho C_D A_x v_x,$$
$$A_{42} = \frac{\bar{F}_{y11} - F_{y11}}{\sigma_{11}}\sin(\delta_{11})t_s,$$
$$A_{34} = \frac{t_s}{I_z}[l_f\cos(\delta_{11}) + \frac{t_f}{2}\sin(\delta_{11})],$$
$$A_{35} = \frac{t_s}{I_z}[l_f\cos(\delta_{12}) - \frac{t_f}{2}\sin(\delta_{12})],$$
$$A_{43} = \frac{\bar{F}_{y11} - F_{y11}}{\sigma_{11}}t_s[l_f\sin(\delta_{11}) - \frac{t_f}{2}\cos(\delta_{11})],$$
$$A_{53} = \frac{\bar{F}_{y12} - F_{y12}}{\sigma_{12}}t_s[l_f\sin(\delta_{12}) + \frac{t_f}{2}\cos(\delta_{12})],$$
$$A_{73} = \frac{\bar{F}_{y22} - F_{y22}}{\sigma_{22}}\frac{t_r}{2},$$
$$A_{41} = \frac{\bar{F}_{y11} - F_{y11}}{\sigma_{11}}\cos(\delta_{11})t_s,$$
$$A_{52} = \frac{\bar{F}_{y12} - F_{y12}}{\sigma_{12}}\sin(\delta_{12})t_s,$$
$$A_{51} = \frac{\bar{F}_{y12} - F_{y12}}{\sigma_{12}}\cos(\delta_{12})t_s,$$
$$A_{61} = \frac{\bar{F}_{y21} - F_{y21}}{\sigma_{21}}t_s,$$
$$A_{63} = -\frac{\bar{F}_{y21} - F_{y21}}{\sigma_{21}}\frac{t_r}{2},$$
$$A_{71} = \frac{\bar{F}_{y22} - F_{y22}}{\sigma_{22}}t_s,$$

and

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{\rho C_D A_x v_x}{m} & 0 & 0 & -\frac{\sin(\delta_{11})}{m} & -\frac{\sin(\delta_{12})}{m} & 0 & 0 \\ 0 & 0 & 0 & \frac{\cos(\delta_{11})}{m} & \frac{\cos(\delta_{12})}{m} & \frac{1}{m} & \frac{1}{m} \end{pmatrix}.$$

3.2.3 Unscented Kalman Filter

Another variation of the Kalman Filter for nonlinear systems is the Unscented Kalman Filter (UKF) [Wan et al., 2001]. Instead of local linearizing like the EKF does, this estimator is based on the unscented transformation, which mainly transforms a set of points (Sigma Points) through the nonlinear model to determine the mean and variance of the states. As it is a Kalman Filter, this estimator can also be described in two phases, the prediction step and the update step. In Figure 3.3, it is shown how this transformation works and the main difference with the local linearizing approach. It also should be mentioned that the Sigma Points that are used during the transformation must be $2n + 1$ points, being n the dimension of the state studied.

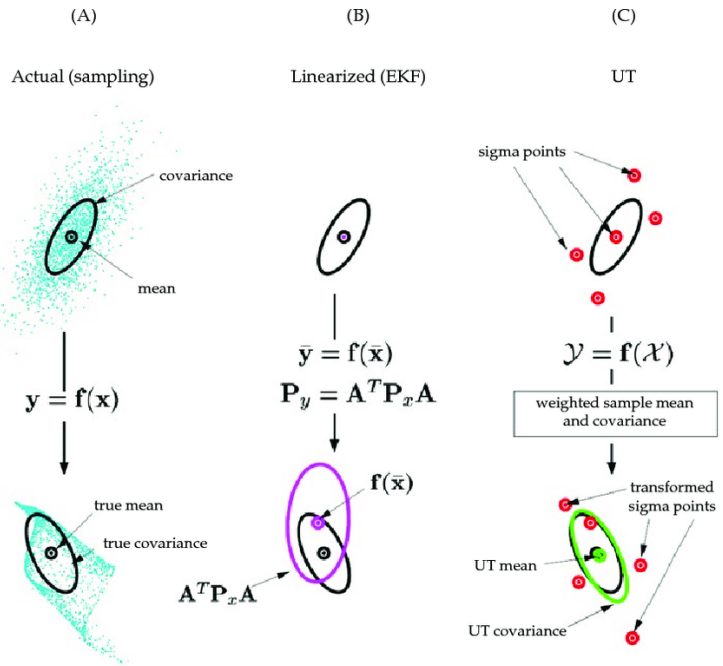

Figure 3.3: Unscented transform scheme, obtained from [Wan and Van Der Merwe, 2000]

This transformation works better for highly nonlinear systems and the results accuracy is much proper. Although it does not use Jacobian matrices, the unscented transform also has a high computational cost that should be studied and compared with the other estimators. Apart from the Q and R matrices, the UKF has three other tuning parameters, which are α, β and κ. For white and non-correlated noise, the following procedure is proposed to obtain the estimated states [Wan et al., 2001]:

First, the calculation of the $2n + 1$ Sigma Points is performed as

$$\chi = [\mathbf{x_k} \;\; (\mathbf{x_k} + \sqrt{c\mathbf{P_k}})_i \;\; (\mathbf{x_k} - \sqrt{c\mathbf{P_k}})_j]^T, \qquad i,j = 1,...,n,$$

where $c = \alpha^2(n+\kappa)$ and the subscripts i and j are the i-th and j-th rows of the matrix P_k. Besides, χ is a matrix of size $n \times (2n+1)$. Considering this set of Sigma Points, the predicted state and covariance matrix are

$$\chi^x = f(\chi, u),$$
$$\hat{x}_k = \sum_{i=0}^{2n} W_{mi}\chi_i^x,$$
$$P_k = Q + \sum_{i=0}^{2n} W_{ci}[\chi_i^x - \mathbf{x_k}][\chi_i^x - \mathbf{x_k}]^T,$$

with $W_{m0} = 1 - \frac{n}{c}$, $W_{c0} = W_{m0} + 1 - \alpha^2 + \beta$ and $W_{mi} = W_{ci} = \frac{1}{2c}$ for $i = 1,...,n$. Then, a new set of Sigma Points $\mathcal{Y}$ are calculated and the corresponding calculus of the update step are done, i.e.,

$$\mathcal{Y} = h(\chi),$$
$$\mathbf{y_k} = \sum_{i=0}^{2n} W_{mi}\mathcal{Y}_\mathbf{i},$$
$$P_y = \sum_{i=0}^{2n} W_{ci}[\mathcal{Y}_\mathbf{i} - \mathbf{y_k}][\mathcal{Y}_\mathbf{i} - \mathbf{y_k}]^T,$$
$$P_{xy} = \sum_{i=0}^{2n} W_{ci}[\chi_\mathbf{i} - \mathbf{x_k}][\mathcal{Y}_\mathbf{i} - \mathbf{y_k}]^T,$$
$$K = P_{xy}P_y^{-1},$$
$$\hat{x}_k = \hat{x}_k + K(\mathbf{y_{mes}} - \mathbf{y_k}),$$
$$P_k = P_k - KP_yK^T.$$

Here, $\mathbf{y_k}$ is the output estimation, $\mathbf{y_{mes}}$ is the measurement input vector, and P_y and P_{xy} are the covariance matrices related to the Sigma Points. For this estimator, the same initial values for the state and the covariance matrix are used. The same with the noise arrays Q and R. After some simulation tests done, the parameters have been set to $\alpha = 10^{-3}$, $\beta = 2$ and $\kappa = 0$ ($\beta = 2$ is typically used for Gaussian noises, which is the case studied). The UKF algorithm should compute the square root of a matrix to obtain the sigma points and it may cause problems if one of its values is negative in any iteration. To solve this problem, the Cholesky decomposition [Watkins, 2004] is typically used. This factorization might cause problems when the algorithm cannot obtain a positive semidefinite covariance matrix, and the estimation can go unstable. In [Ma et al., 2010], a singular value decomposition (SVD) of this matrix is proposed. This method guarantees the positive semidefinite condition and the estimation is always stable. To implement the SVD, the Sigma Points calculated should be replaced with

$$[U, D, V] = F_{SVD}(P_k),$$
$$C = U(\sqrt{(n+\kappa)\alpha^2 D}U^T),$$
$$\chi = [\mathbf{x_k} \;\; (\mathbf{x_k} - \sqrt{\mathbf{C}})_i \;\; (\mathbf{x_k} + \sqrt{\mathbf{C}})_j]^T, \qquad i,j = 1,...,n,$$

where $F_{SVD}(*)$ is the singular value decomposition function of the matrix $(*)$ and can be calculated with the Matlab command `> >[U,D,V] = svd(*)`. The results are shown in Chapter 4.

3.2.4 Moving Horizon Estimator

The last estimator studied is the Moving Horizon Estimator [Rao et al., 2003]. The MHE may include the statement of both linear and nonlinear optimization problems with constraints. It considers the constrained optimization problem in a finite horizon to get the new state, and as new measurements become available, the old measurements are discarded from the horizon window to get the new state estimation. The MHE computes the predicted state with the last $t-N$ data available (where N is the estimation horizon length). The main difference between the other estimators proposed and the MHE is the addition of constraints to the problem and it allows to set the physic limits and constraints that the different state variables may have. Notice how the discrete-time representation of (2.18a) and (2.18b) should also be added as constraints since they are the equations of the model. They will be referred as

$$\begin{aligned} x_{k+1} &= f(x_k, u_k) + \omega_k, \\ y_k &= h(x_k) + v_k. \end{aligned}$$

Then, the optimization problem to be solved is

$$\begin{aligned} \min_{x_0, \left\{\omega_{k=0}^{k=t-1}\right\}} \quad & \Phi_t \left(x_0, \left\{\omega_{k=0}^{k=t-1}\right\}\right), \\ \text{s.t.} \quad & x_{k+1} = f(x_k, u_k) + \omega_k, \\ & y_k = h(x_k) + v_k, \\ & x_{k+1} \geq L, \quad x_{k+1} \leq M, \\ & \omega_k \geq N, \quad \omega_k \leq B, \\ & x_k \in \mathbb{R}^7, \quad \omega_k \in \mathbb{R}^7, \quad v_k \in \mathbb{R}^5, \end{aligned} \tag{3.8}$$

where L, M, N and B represent the upper or lower limits of each variable, Φ_t is the cost function and $\left\{w_{k=0}^{k=t-1}\right\}$ is the state disturbance sequence from $k=0$ to $k=t-1$. Considering a limited horizon of time means that two different situations will appear while solving (3.8) as the current time of estimation $k=t$ can be lower or higher than the horizon length. The cost function can be reformulated considering the two different cases.

· For $t \leq N$:

$$\begin{aligned} \min_{\hat{x}_0, \left\{\omega_{k=0}^{k=t-1}\right\}} \quad & \Phi_t \left(\hat{x}_0, \left\{\omega_{k=0}^{k=t-1}\right\}\right) = \\ \min_{\hat{x}_0, \left\{w_{k=0}^{k=t-1}\right\}} \quad & \sum_{k=0}^{t-1} ||\omega_k||^2_{Q^{-1}} + \sum_{k=0}^{t-1} ||v_k||^2_{R^{-1}} + \Psi_0. \end{aligned} \tag{3.9a}$$

· For $t > N$:

$$\begin{aligned} \min_{\hat{x}_{t-N}, \left\{\omega_{k=t-N}^{k=t-1}\right\}} \quad & \Phi_t \left(\hat{x}_{t-N}, \left\{\omega_{k=t-N}^{k=t-1}\right\}\right) = \\ \min_{\hat{x}_{t-N}, \left\{w_{k=t-N}^{k=t-1}\right\}} \quad & \sum_{k=t-N}^{t-1} ||\omega_k||^2_{Q^{-1}} + \sum_{k=t-N}^{t-1} ||v_k||^2_{R^{-1}} + \Psi_{t-N}, \end{aligned} \tag{3.9b}$$

where the notation $||z||_A^2 = z^T A z$. The first term of this expression is referred to the process noise and the second corresponds to the matching or not of the measurement data. The third term, Ψ, is called the arrival cost, which summarizes the estimated information prior to the new horizon, from $k = 0$ to $k = t - N - 1$. Finding an expression to compute this arrival cost is hard, and many authors propose to use an approximation of the unconstrained problem. Taking advantage of the estimators designed before, an EKF arrival cost is used as proposed in [Rao et al., 2003]. With this approach, the arrival cost can be calculated as

$$\Psi_{t-N} = (x_{t-n} - \hat{x}_{t-N}) P_{t-N} (x_{t-n} - \hat{x}_{t-N}), \tag{3.10}$$

where P_{t-N} can be obtained with the EKF algorithm in Section 3.2.2. Additionally, the cost function is modified with a forgetting factor λ, which gives exponentially less weight to older errors. This factor is added in by multiplying the cost function with λ^{k-N}, being $0 \leq \lambda \leq 1$. A $\lambda = 0.9$ has been chosen for this particular case.

The last aspect to define before studying the different tuning parameters of the MHE are the constraints that will be used in the problem (3.8). As solving the optimization problem guarantees these constraints will be fulfilled, each variable will have an upper and lower limit so it moves in a physically possible range of values. The constraints that will be used are

$$\begin{aligned} x_{k+1} &= f(x_k, u_k) + \omega_k, \\ y_k &= h(x_k) + v_k, \\ 0 \text{ m/s} \leq v_x &\leq \omega_{max} R \text{ m/s}, \\ -10 \text{ m/s} \leq v_y &\leq 10 \text{ m/s}, \\ -5 \text{ rad/s} \leq \dot{\psi} &\leq 5 \text{ rad/s}, \\ -2500 \text{ N} \leq F_{yij} &\leq 2500 \text{ N}. \end{aligned}$$

Here, ω_{max} is the maximum rotational speed the wheel can get [rad/s] and R is the wheel radius [m]. Note how the longitudinal velocity must be a positive value as the case studied is a Formula Student vehicle and it cannot be driven in reverse. There will not be any process noise upper B or lower N bound. The L and M matrices from (3.8) can be now defined as

$$L = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2500 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2500 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2500 & 0 \\ 0 & 0 & 0 & -0 & 0 & 0 & -2500 \end{pmatrix},$$

$$M = \begin{pmatrix} \omega_{max}R & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2500 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2500 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2500 & 0 \\ 0 & 0 & 0 & -0 & 0 & 0 & 2500 \end{pmatrix},$$

The optimization problem of the MHE (3.8) will be implemented in MATLAB and solved using the CasADi tool [Andersson et al., 2019], which is an open-source tool for nonlinear optimization and algorithmic differentiation. For this particular case, the IPOPT solver is used.

3.3 Summary

In this chapter, the lateral force tire estimator scheme has been completely defined, establishing the measurement inputs and the discretization method used to get a discrete-time state space realization. Then, four different observers have been presented (Luenberger, Extended Kalman Filter, Unscented Kalman Filter and a Moving Horizon Estimator) with their corresponding noise or gain matrix. For the MHE, the problem is exposed defining the cost function, arrival cost and all the constraints. The estimation horizon length will be studied and determined in Chapter 4, as well as the comparison between all the observers. The same initial value for each variable has been chosen so the convergence time can be studied for all the estimators equally. One of the estimators will be chosen as the best option to incorporate it in the car.

Chapter 4

Simulation and Comparison

In this chapter, the four observers designed are compared between them by simulating some different conditions and circuits according to some Key Performance Indicators. Before comparing the results, the Tustin rule and the Euler method are studied and a estimation horizon length for the MHE is chosen. As the vehicle studied and modeled is a Formula Student vehicle, the different tests used to simulate the observers will be some of the Formula Student competition events.

4.1 Test tracks

In a Formula Student competition, there are three different type of tracks corresponding to different events: acceleration, Skidpad and Autocross or endurance. The first one, as its name describes, it is a simple 75m straight and the car has to finish it in the less time possible. In a straight line, the tire does not do any lateral force and it should not be used to test the observers. The second event mentioned is the Skidpad, whose track consists of two pairs of concentric circles in a figure of eight pattern (see Figure 4.1). The centers of these circles are 18.25m apart. The inner circles have a 15.25m diameter and the outer circles' diameter is 21.25m. The driver must do four laps in circles, two in clockwise and two in anticlockwise. Notice how this is also the most stationary test as the steering wheel input should remain almost constant while cornering and the different state variables should also behave almost constant. Finally, the Autocross or endurance are events with a track layout that consists on some straights, turns and slaloms. The main difference between them is the length of the track. The Autocross layout is the smallest one, being it less than 1.5km, so the Autocross event will also be studied and simulated instead of the endurance. To sum up, the Skidpad and Autocross tracks layout have been chosen to simulate the observers.

4.2 Key Performance Indicators

Once the test conditions have been defined and before starting the simulations to compare all of the observers, it is important to define the key performance indicators (KPI). These parameters will determine which of the estimators is the one that best adapts to the needs of a Formula Student vehicle, and, therefore, can be incorporated into the ETSEIB Motorsport car. Three different indicators have been chosen as KPI.

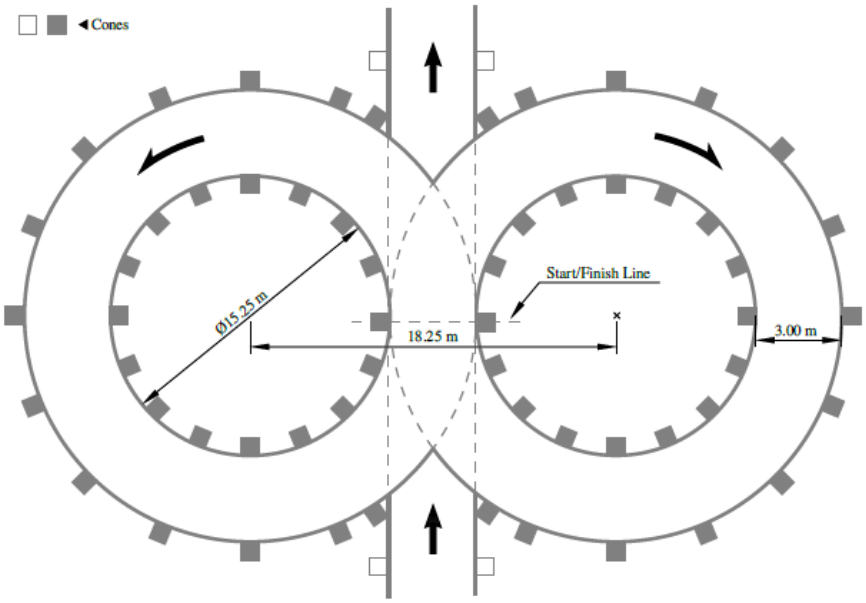

Figure 4.1: Skidpad track layout

- Computational cost: The time it takes to the Central Processing Unit (CPU) to run the four estimator's code is an important parameter to consider. If the elapsed time is higher than 20ms, it will not be possible to incorporate the observer to the car. A 64.1s Autocross run will be studied with MATLAB. The simulation will be conducted with a system with an AMD Ryzen 5 2600 6-Core (3.4GHz) CPU and, although some solver options have been modified in the MHE observer to decrease this CPU time, the code could still be improved.
- Convergence time: It is also important the observer converges as fast as possible and it can be used as an appropriate input for the torque vectoring system. It will be considered that it has converged when it reaches the real state value plus its mean error and the same initial state will be considered for all the estimators.
- Normalized error: The last indicator will be the normalized error. The estimator should provide a value close to the real one with a small error. Although the IPG software can provide the results of all the seven state variables, the forces cannot be measured with any sensor and also the error between the estimated variables and their corresponding measured values will be considered. The mean errors between the simulation software and the estimated values should be less than the 1.5% for the kinematic variables and less than the 5% for the dynamic ones to consider the state is properly estimated. The measurable variables have been modeled according to the sensor the team has available and the error between them and the real values should be smaller than the 0.5%. This KPI will be calculated as follows:

$$\epsilon_z = 100 \left| \frac{x_k - \hat{x_k}}{\max(x_k)} \right|, \tag{4.1}$$

where ϵ_z is the normalized error in percentage. A first analysis and study of the MHE will be done to choose the best estimation horizon length (Section 4.4) and, once the estimator is completely defined, it will be compared with the other three observers simulating an Skidpad event (Section 4.5) and an Autocross run (Section 4.6). Before starting with the comparison, a general analysis of the discretization method is carried out in Section 4.3.

4.3 Tustin and Euler analysis

As commented in Section 3.2, the nonlinear Luenberger will be used to test the difference between the Euler first-order discretization method and the Tustin rule. The error obtained with both methods is the same for all the seven variables. The main difference can be found in the computational cost. Figure 4.2 shows the elapsed time in milliseconds.

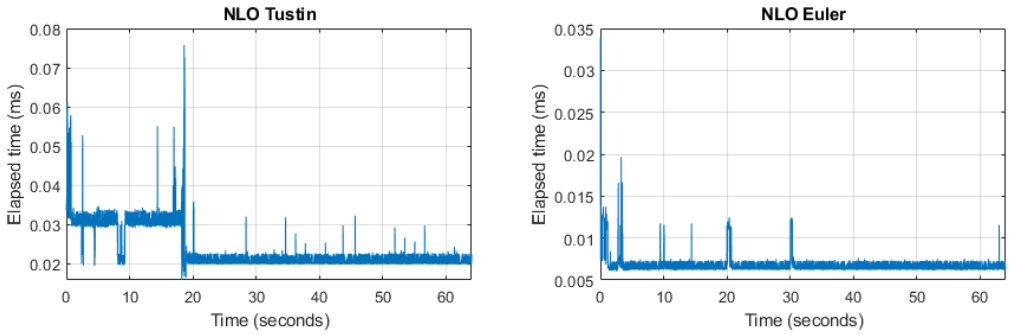

Figure 4.2: Elapsed time for the Tustin (left) and Euler (right) method.

Being the time in milliseconds implies that both methods can be implemented in the car as it is always below the 20ms sampling period of the car's main ECU. The time cost is lower for the Euler first-order discretization method and it does not present high peaks like the Tustin method does. These peaks are caused because the Tustin rule needs to solve (3.2) for each iteration and it may take longer to find the solution in some cases. Finally, the error eigenvalues are figured in the real and imaginary axis. Starting from (3.5), the error eigenvalues are obtained from the matrix $A - LC$.

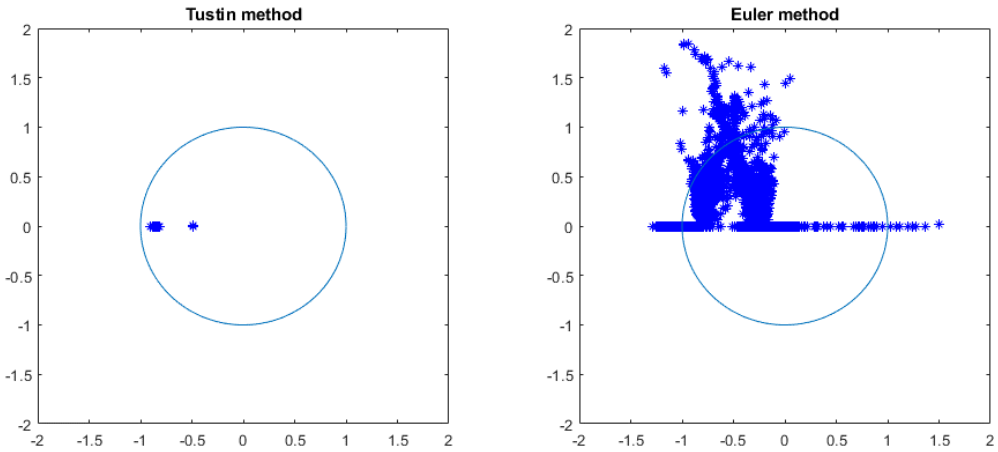

Figure 4.3: Eigenvalues comparison for Tustin (left) and Euler (right) method.

In Figure 4.3, it can be noticed how the Tustin rule ensures that all of the eigenvalues are inside the unit circle as they all have a modulus lower than unity. In the other hand, the Euler discretization method cannot assure that, but the system still converges because the input measurements redirect the system in every iteration and makes it stable again. It can be noticed how the eigenvalues are also more concentrated with the Tustin method than with the Euler rule, in which they are more scattered.

4.4 MHE horizon analysis

The last aspect to tune of the MHE is the estimation horizon length (N). A 64.1s Autocross run will be simulated for different lengths ($N \in \{3, 5, 7, 10, 15, 25\}$) and they will be compared according to the KPIs defined in Section 4.2. Figures 4.4 to 4.8 show the comparison between the MHE results (for each different estimation horizon length) and the IPG software values.

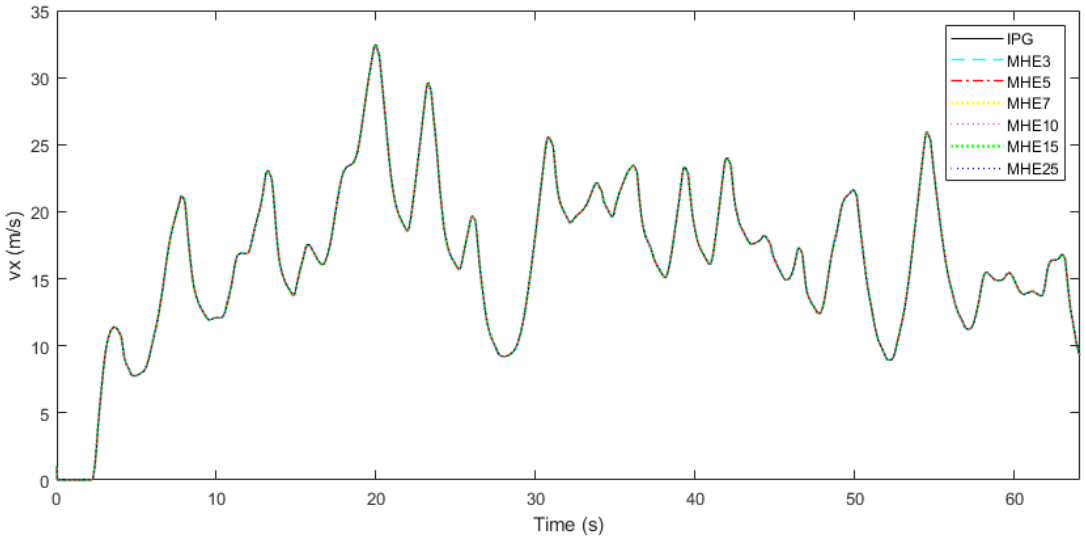

Figure 4.4: Longitudinal velocity for different horizon lengths.

In Figure 4.4, it can be seen how the MHE estimates properly the longitudinal velocity during all the simulation test. There is not a substantial difference for any horizon. All of them have a mean error around the 0.02% of the IPG value.

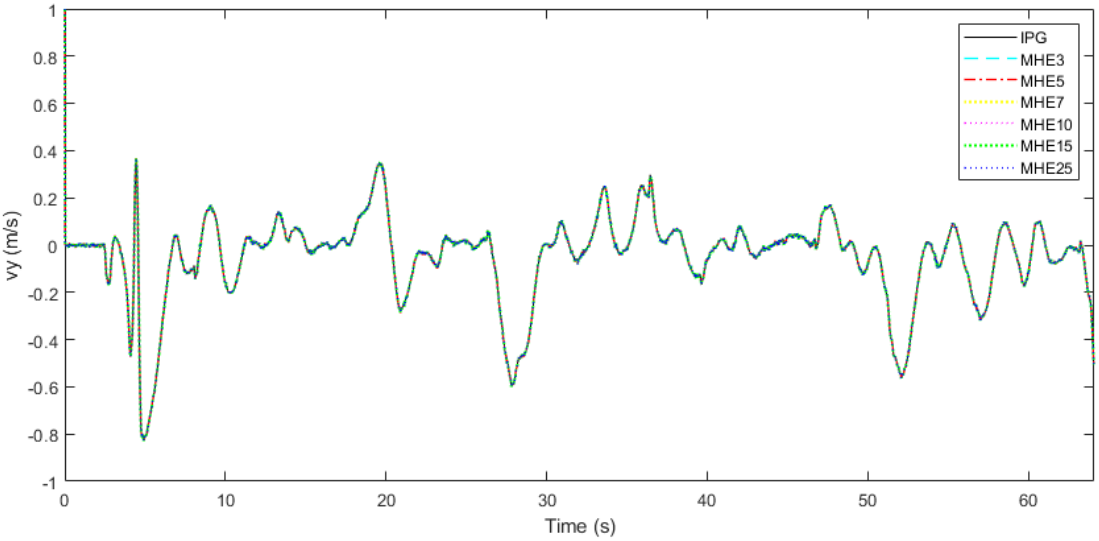

Figure 4.5: Lateral velocity for different estimation horizon lengths.

Figure 4.5 represents the lateral speed for the estimation horizon lengths mentioned previously. As it happens with the longitudinal velocity, the different observers proposed follow properly the IPG simulation results with a mean error of approximately 0.52% for all the cases.

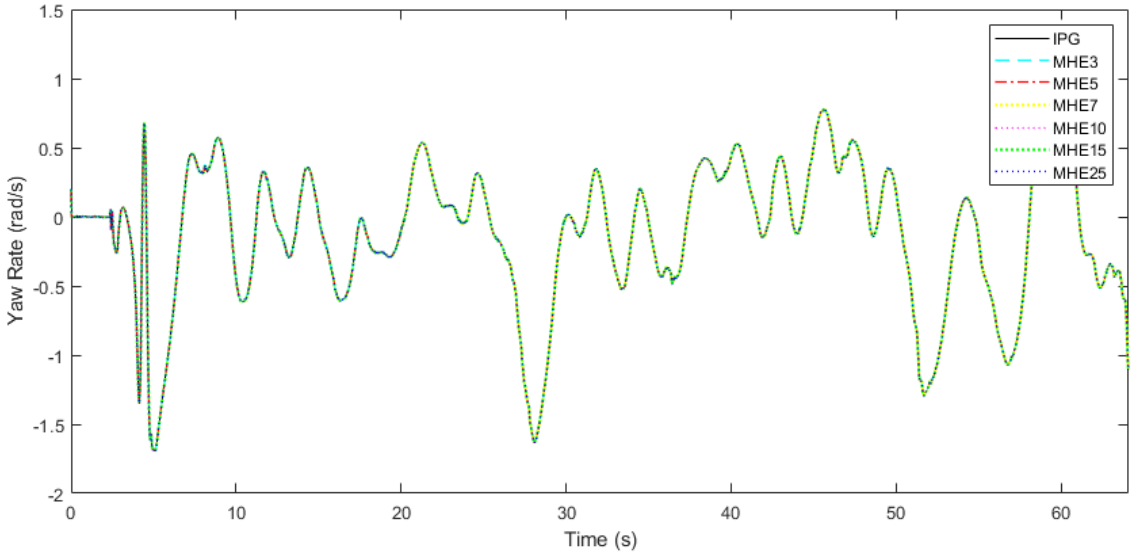

Figure 4.6: Yaw rate for different estimation horizon lengths.

The yaw rate is represented in Figure 4.6. The MHE estimates this variable properly during all the autocross run with a mean relative error between the 0.2391% and the 0.2621%. Both error bounds have a low average error and it can be considered that the estimation is appropriate.

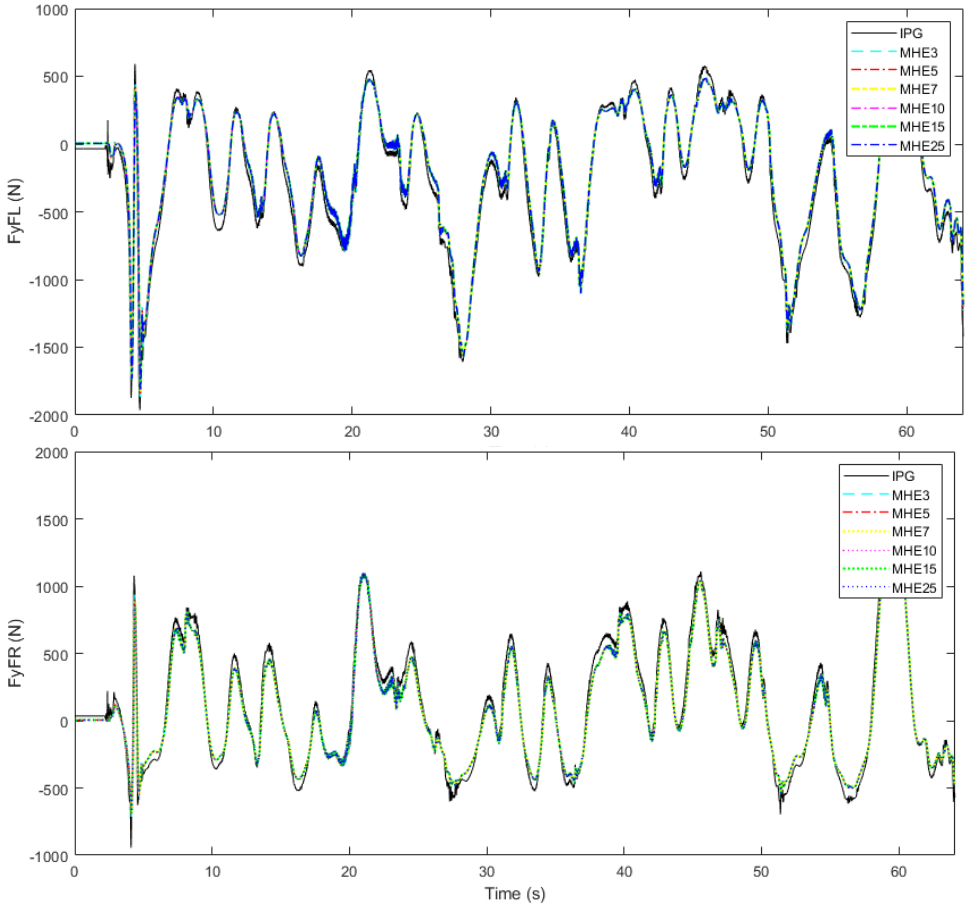

Figure 4.7: Front left (up) and right (down) lateral forces for different estimation horizon lengths.

Figure 4.7 represents the front left and right tire lateral forces for the horizons defined previously. The front relative errors are between the 3.8% and the 4%, which are similar values for all the options considered. These errors are higher in the peaks of the curve due to an abrupt steering wheel angle change and, consequently, a slip angle error. An expanded image (Figure 4.9) shows more clearly the difference estimation for all the N values.

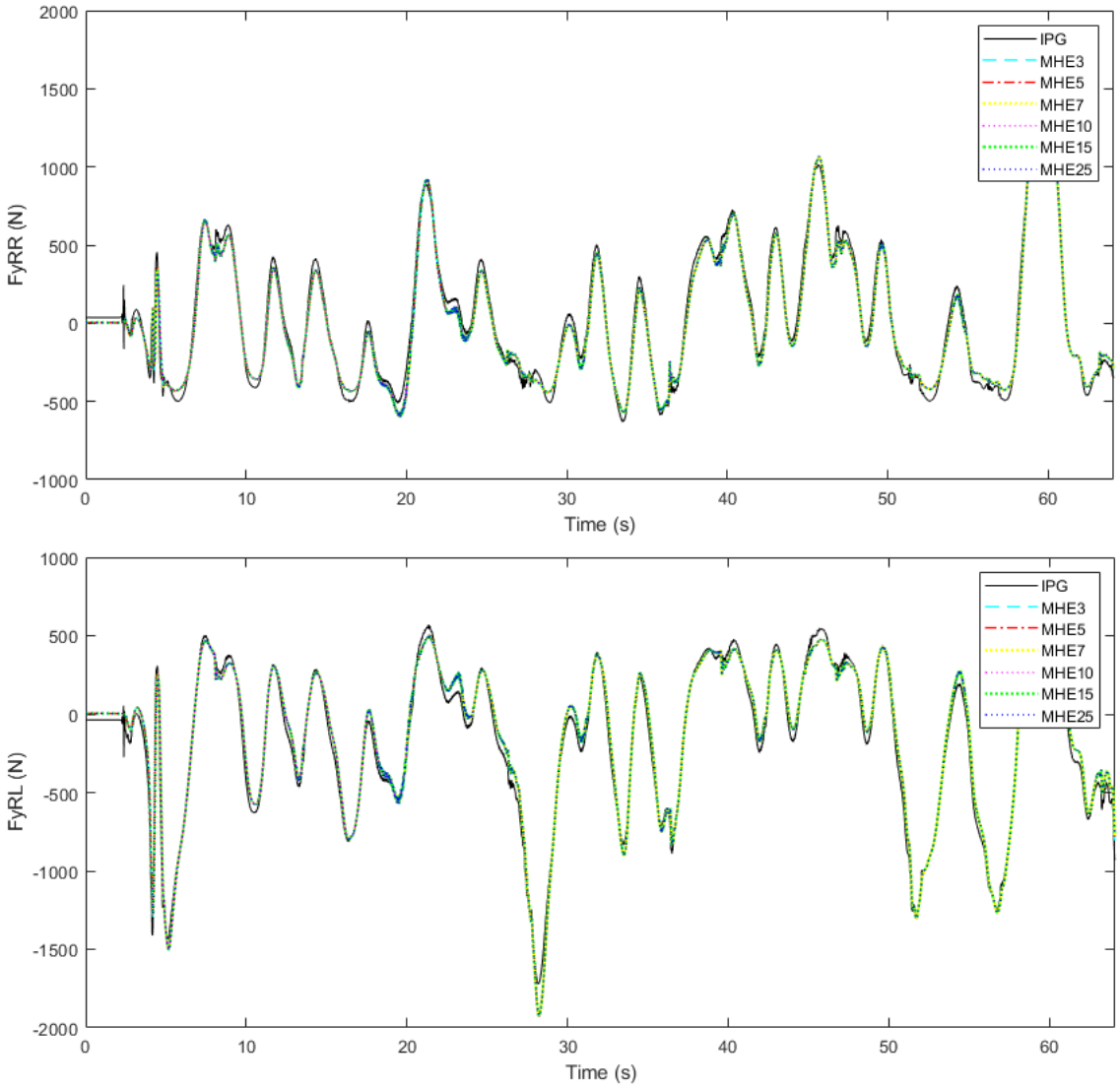

Figure 4.8: Rear right (up) and left (down) lateral forces for different estimation horizon lengths.

Figure 4.8 shows the tire lateral forces for the rear left and right wheels. As it happened when validating the model (Section 2.4), the errors obtained for the rear axle wheels are lower than for the front wheels. With the MHE, they vary between a 3.24% and a 3.28%. Both cases have small errors to consider both forces to be suitably estimated.

Figure 4.9 also represents an expanded image for these forces so the difference between the six estimation horizon proposals can be properly observed. They are figured in a period of time reduced so the difference can be noticed.

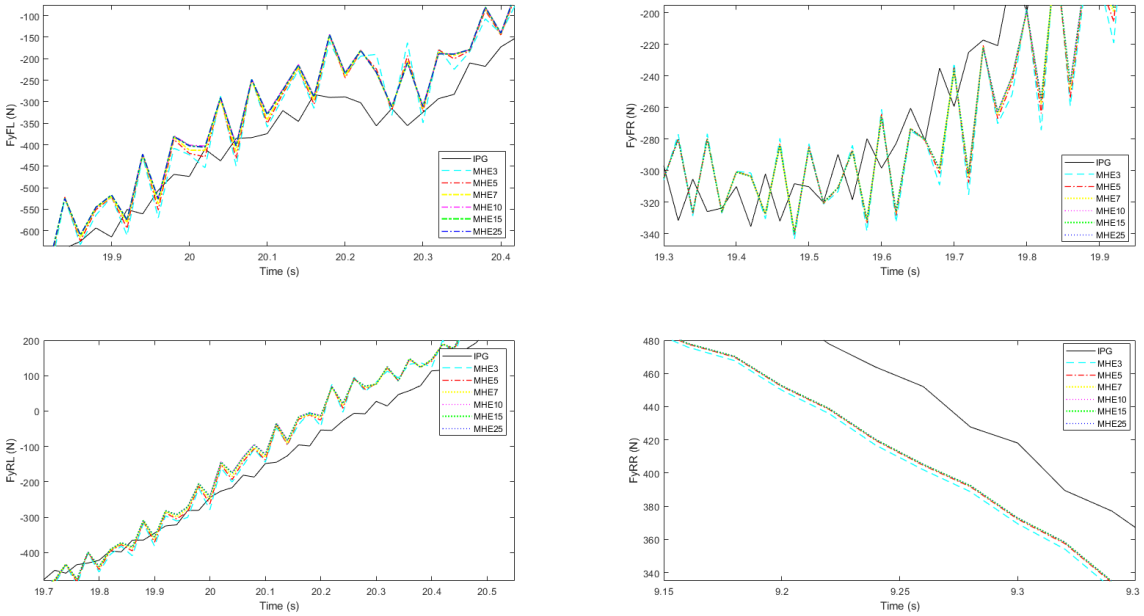

Figure 4.9: Expanded lateral forces estimation.

In order to choose the best option, the KPIs should be analized, starting with the computational cost. As mentioned before, if the CPU time is higher than the 20ms sampling time it cannot be incorporated in the car, so it is a proper way to start discarding options. Figure 4.10 represents the CPU time required to solve (3.8) for each sampling instant. It has been mentioned in Section 4.2 that some solver options have been modified to optimize the algorithm. However, there are more options that could be changed in the future in order to decrease the CPU time required. This possibility will be taken into account and if the computational time barely exceeds the 20ms for any estimation horizon length, it will be also considered as a valid option.

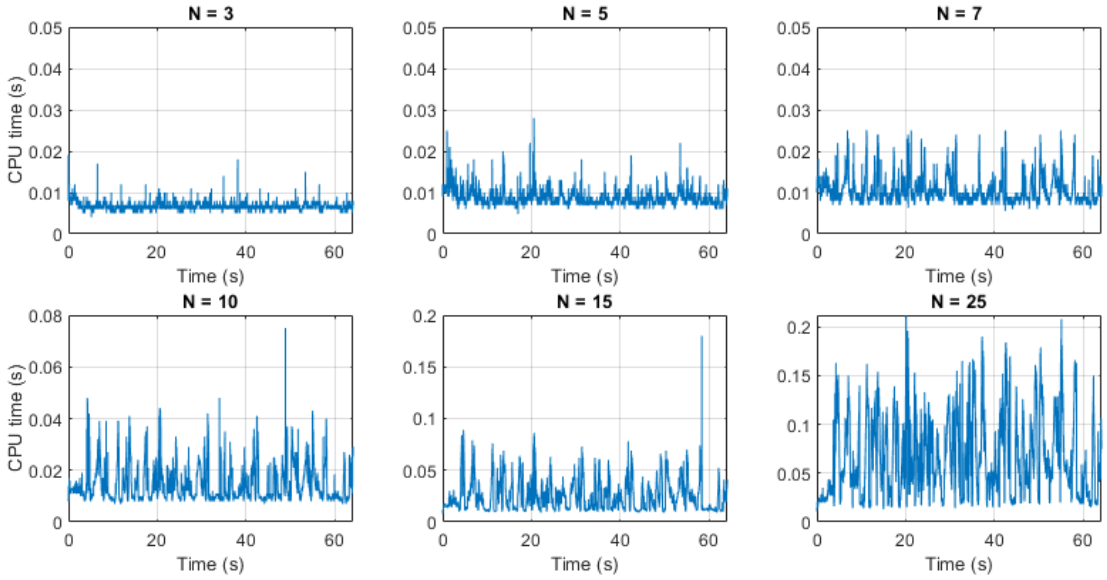

Figure 4.10: CPU time for the different estimation horizon lengths.

It can be seen in Figure 4.10 how for $N = \{10, 15, 25\}$ the CPU time exceeds excessively the sampling period. For $N = 5$, the time cost presents some peaks that barely surpass the

limit specified. Considering the previous conditions defined, this estimation horizon length will be accepted in terms of the computational cost. Finally, for $N = 7$, the number of peaks exceeding the sample period is higher than for $N = 5$, but it will be further analyzed according to the other KPIs to see if it is worth to further improve the MHE algorithm and use this estimation horizon length in the observer. Table 4.1 shows the relative error between the state and the IPG software variables. To obtain these results, a set of 25 simulations with different noises has been done.

Table 4.1: State variable mean errors (in %) according to the IPG response.

Variable	$N = 3$	$N = 5$	$N = 7$
Longitudinal velocity	0.0189	0.0194	0.0196
Lateral velocity	0.5105	0.5132	0.5177
Yaw rate	0.2391	0.2433	0.2485
Front left F_y	3.8694	3.8882	3.8974
Front right F_y	3.8694	3.8882	3.8974
Rear left F_y	3.2471	3.2629	3.2686
Rear right F_y	3.2471	3.2629	3.2686

It can be seen how all of the normalized errors are below the limits considered in Section 4.2 and it can be considered the variables are properly estimated. The variables with the lowest mean errors are the kinematic ones, especially the longitudinal speed and the yaw rate. There is not a significant difference between the errors for the three estimation horizon lengths considered, but it can be noticed how they slightly increase when N gets bigger. The same happens for the dynamic variables, whose mean errors are below the 5% for all of them. The front wheels present a bigger estimation error than the rear ones, probably caused due to a tire slip angle error, as seen when modeling the vehicle in Section 2.4. Although the lowest error obtained is for $N = 3$, the difference is in the order of a hundredth percentage, so it is not a significant difference.

Table 4.2: State variable mean errors (in %) according to the sensor input.

Variable	$N = 3$	$N = 5$	$N = 7$
Longitudinal velocity	0.0038	0.0038	0.0039
Lateral velocity	0.0402	0.0405	0.0407
Yaw rate	0.1303	0.1335	0.1357

It has been also commented that the error between the sensor inputs will be considered. These errors are summarized in Table 4.2, where it can be seen how the error for all the variables also increases as the estimation horizon length gets bigger (although the

difference between them is almost null). The difference between the estimated variables and the sensor inputs mean errors are much lower than the difference with the IPG results. The smallest errors are obtained for the longitudinal and lateral speed, but all of them are lower than the 0.5% values specified in Section 4.2. Taking into account all the KPIs, the most appropriate horizon length for the MHE is $N = 3$, as it presents the lowest computational cost and relative errors.

4.5 Skidpad Analysis

Now that the MHE has been completely defined, the four estimators that will be compared and analyzed are finally decided. They are the nonlinear Luenberger (NLO), EKF, UKF and the MHE (with a estimation horizon length of $N = 3$). As commented in Section 4.1, one of the Formula Student events used to test and compare the estimators is the Skidpad. Figures 4.11 to 4.14 represent all the state variables with the corresponding estimation of each observer.

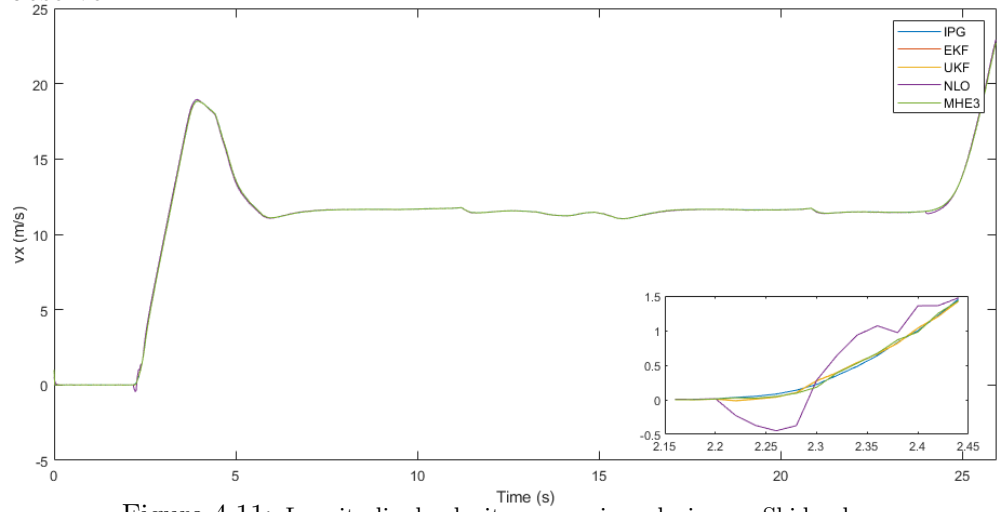

Figure 4.11: Longitudinal velocity comparison during an Skidpad run.

In Figure 4.11 the longitudinal speed during an Skidpad event is represented. It can be seen how all the observers estimate the simulated values with small errors. In Table 4.3, the difference between the errors obtained is summarized numerically with their mean and standard deviation. There is not a significant difference between the four options considered related to the longitudinal velocity estimation.

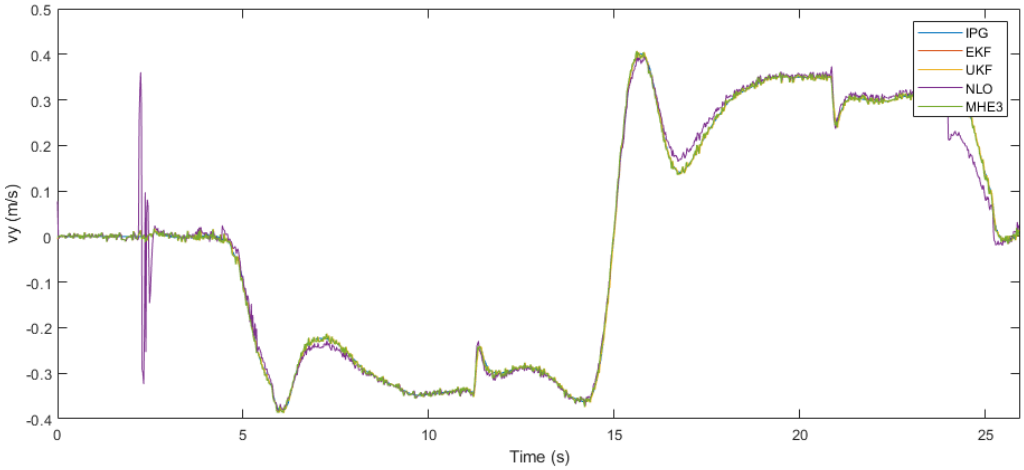

Figure 4.12: Lateral velocity comparison during an Skidpad run.

The lateral velocity is shown in Figure 4.12. Although the errors obtained are slightly higher than for the longitudinal speed, they are also low enough to consider that this variable is properly estimated. The NLO has a higher mean error (around the 3.5%) in comparison to the other observers (which are around the 1%), as shown in Table 4.3. The dispersion of the estimated variables is also noticeable for the lateral speed, especially in the NLO and the MHE.

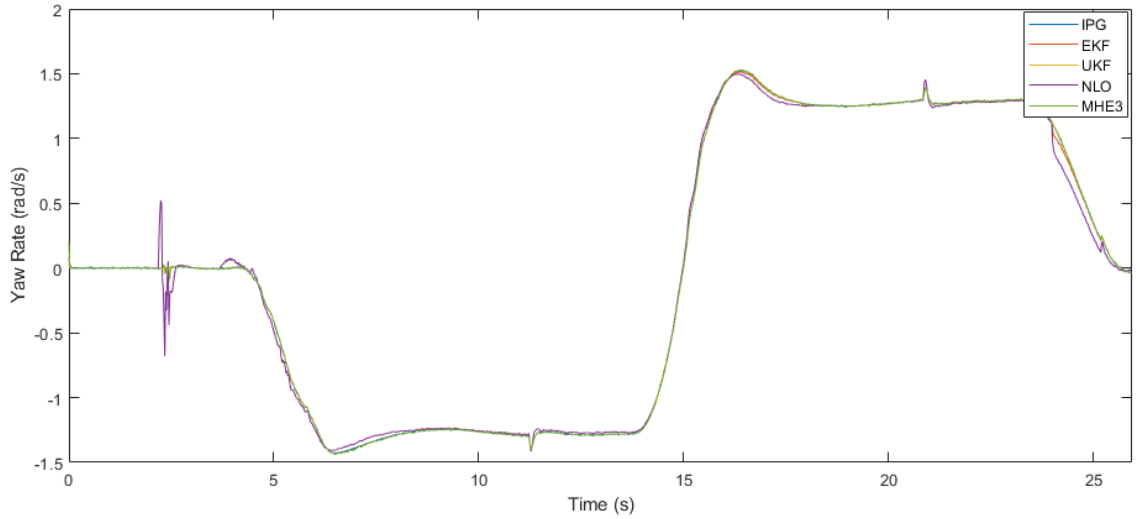

Figure 4.13: Yaw rate comparison during an Skidpad run.

In Figure 4.13, it can be observed how the NLO has a higher error in comparison with the other estimators, as it happened with the lateral velocity. Between the EKF, UKF and MHE, there is not a significant difference. The numerical analysis is summarized in Table 4.3.

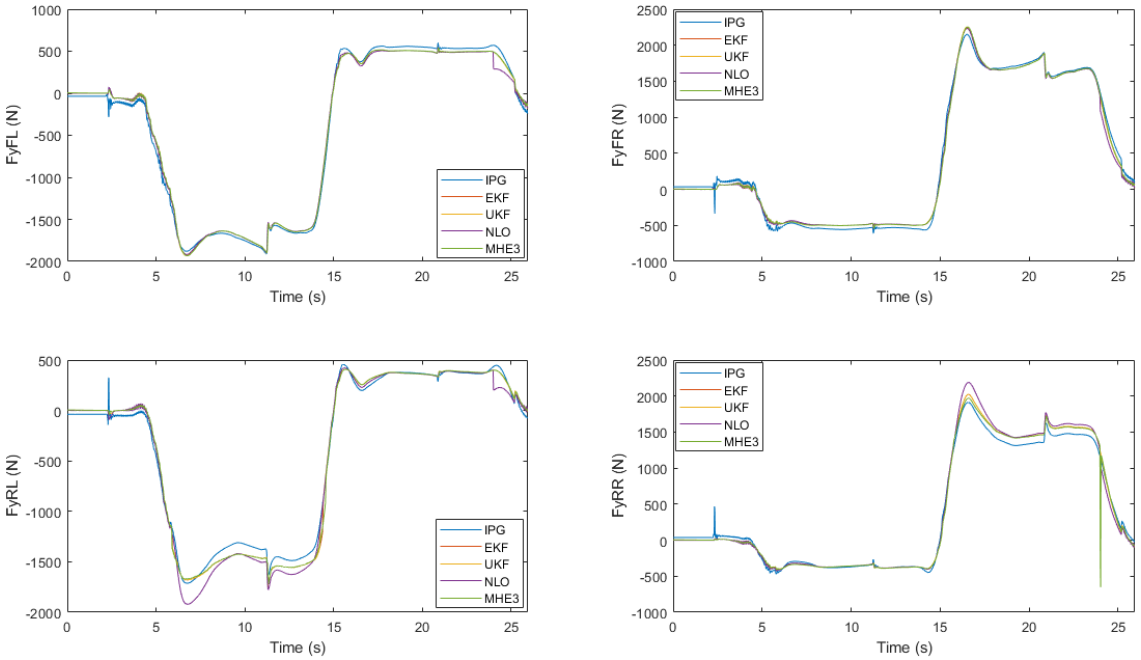

Figure 4.14: Tire lateral forces comparison during an Skidpad run.

The tires lateral forces are represented in Figure 4.14. They all present higher estimation errors when the lateral forces reach high values. It seems that all four observers estimate properly the forces and, in order to get a better perception of the difference between them, the mean and standard deviation of the relative errors are studied. The numerical results can be found in Table 4.3.

Table 4.3: State variable errors (%) for an Skidpad event simulation.

std / **mean**	**EKF**	**UKF**	**NLO**	**MHE**
Long. speed	0.0259 / 0.0279	0.0259 / 0.0279	0.3279 / 0.1762	0.1238 / 0.0303
Lat. speed	0.7511 / 0.9943	0.7513 / 0.9942	7.4525 / 3.6260	6.9380 / 1.1863
Yaw rate	0.4815 / 0.3303	0.2517 / 0.2434	3.1927 / 1.8081	0.4395 / 0.2581
$\mathbf{F_{y11}}$	1.4126 / 3.3515	1.4125 / 2.3515	2.1062 / 2.6358	1.5991 / 2.4068
$\mathbf{F_{y12}}$	1.3756 / 2.1974	1.3906 / 2.1989	1.9462 / 2.4411	1.5711 / 2.2333
$\mathbf{F_{y21}}$	2.3913 / 2.8987	2.2075 / 2.8295	3.5997 / 3.9472	2.5051 / 3.0243
$\mathbf{F_{y22}}$	2.1856 / 2.8177	2.1704 / 2.8071	3.5580 / 3.6890	3.8339 / 2.7572

It can be noticed how the Kalman Filters have similar normalized error values despite the UKF errors are slightly lower. The MHE estimation errors are between the Kalman Filters except for the rear right force, whose estimation is the best of the four estimations. The NLO and the MHE are the observers with higher standard deviation in the velocity errors. Regarding the acceptable values defined in Section 4.2, all the estimators fulfil the requirements for the forces but the NLO exceeds the 1.5% limit designated for the lateral speed and the yaw rate. The longitudinal speed and the yaw rate are again the variables that are better estimated. Regarding these errors, the UKF is the one with better results, followed by the MHE and EKF.

The results have been obtained with 25 iterations with different noised input measurements. To see them in a more graphical way, the errors for one iteration are represented in Figures 4.15 and 4.16.

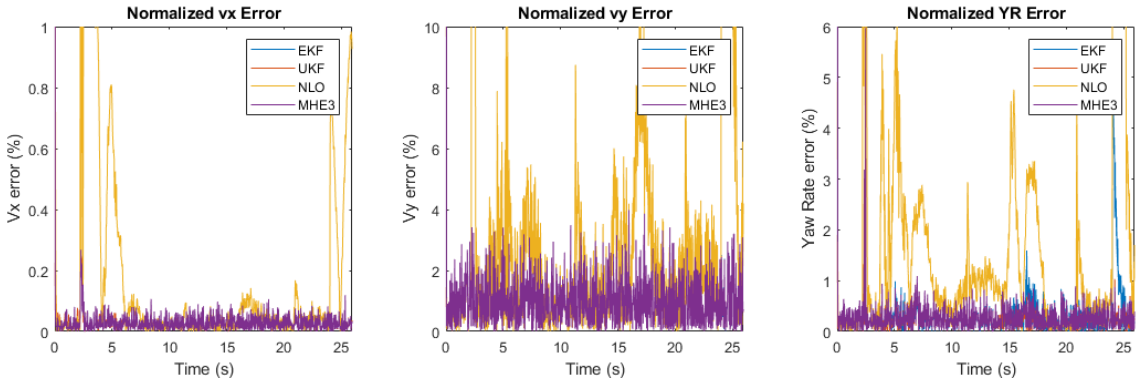

Figure 4.15: State variable relative error (kinematic variables) during an Skidpad run.

In Figure 4.15, the normalized error obtained with (4.1) is represented for the linear velocities and the yaw rate to complement Table 4.3. It can be noticed how the NLO has the higher errors compared with the other observers. In the longitudinal velocity graphic, although the mean and standard deviation are low, the error is relatively high in the beginning and the end of the simulation. It can also be seen the main difference between the Kalman Filters and the MHE, which is basically the dispersion in the error (the standard deviation is higher as shown in Table 4.3).

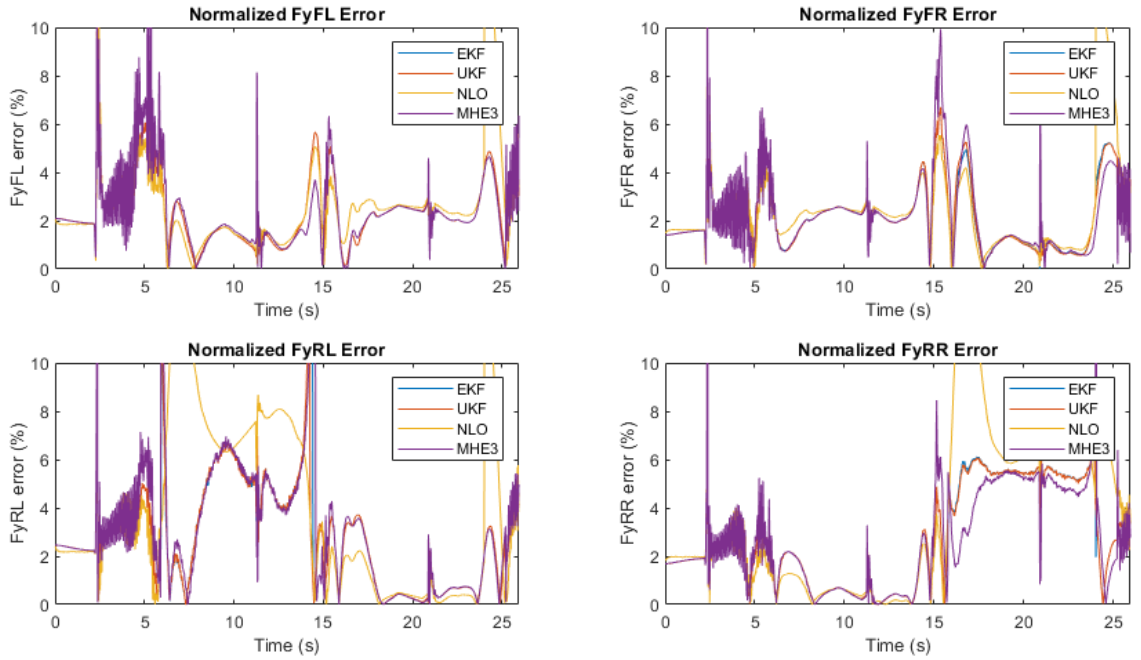

Figure 4.16: State variable relative error (lateral forces) during an Skidpad run.

The normalized error for the lateral forces is shown in Figure 4.16. As mentioned before, these errors reach higher values because they are obtained according to a dynamic model with some simplifications and there is not an input measurement to compare with (the kinematic variables are constantly being compared with the sensor measurements in order to get a better estimation). The errors are generally similar between the four estimators and they are slightly lower for the front tires. The NLO presents a higher error for the

rear axle tires in comparison with the other estimators.

The error with the sensor inputs is also taken into account and summarized in Table 4.4. In here, the NLO errors are much higher than the accepted ones. The rest of the results are proper, being the Kalman Filters the ones with the lowest normalized mean errors.

Table 4.4: State variables mean errors (in %) according to the sensor input.

Variable	EKF	UKF	NLO	MHE
Longitudinal velocity	0.0063	0.0063	0.1763	0.0091
Lateral velocity	0.0051	0.0045	3.2917	0.1916
Yaw rate	0.2082	0.1089	1.7968	0.1230

With the normalized error analyzed, the computational and convergence time remain to be studied. In Figure 4.17, the elapsed time for the four estimators is represented. The main difference noticeable is the time cost of the MHE computational cost, which is three magnitude order higher than the rest of the observers. Although this difference, all of them are below the limit of 20ms. The NLO has the smallest computational cost, followed by the EKF and then the UKF. The UKF elapsed time is more than 10 times higher than the EKF or UKF, but it is still in the order of the microseconds

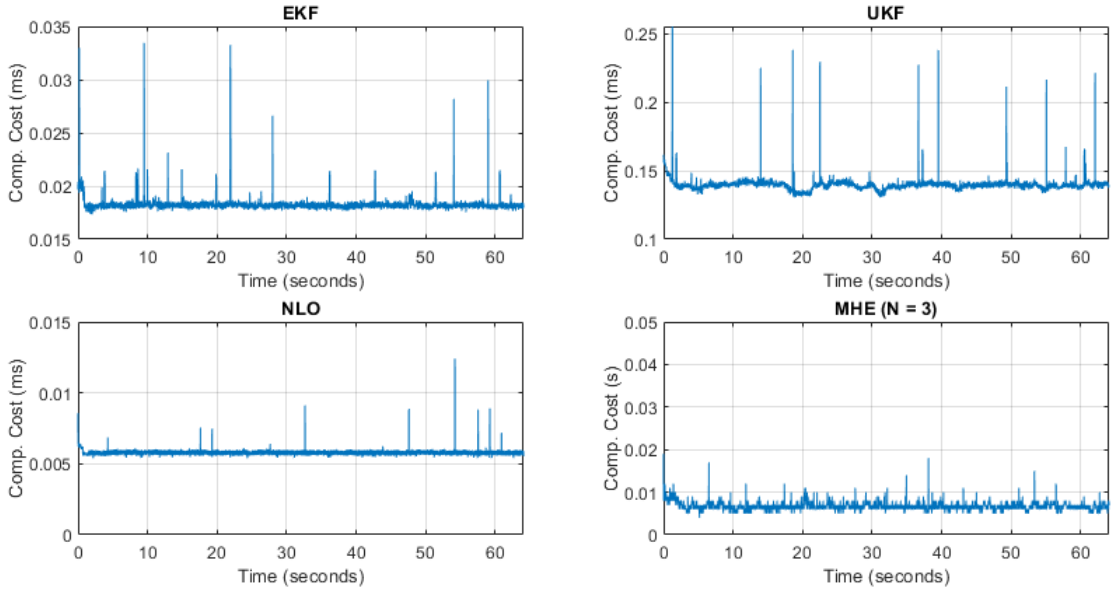

Figure 4.17: Computational time for the NLO, EKF, UKF and MHE observers.

The convergence time is summarized in Table 4.5. Considering that the whole event lasts approximately 30s and the lateral forces values are needed mainly for the torque vectoring while cornering (see Chapter 1), all the different times obtained are proper in a Skidpad event. The NLO is the observer that lasts the most to converge but it does it in 0.14s, which is fast enough.

Table 4.5: Convergence time.

	NLO	EKF	UKF	MHE
Samples	7	3	3	2

The Skidpad test is the most stationary event and the seven variables have almost a constant value. This could lead to errors if the system is working in the nonlinear or linear part of the model most of the time. To get a more rigorous point of view, the Autocross event is also considered.

4.6 Autocross Analysis

As commented previously, before getting a conclusion on which estimator is the best option to use in the car, the Autocross event will also be studied. The values the variables will get during this test will be more diverse than for the Skidpad event and the system will work either in the linear and nonlinear part of the model during the simulation. Because the computational cost is the same obtained in Section 4.5 (as the algorithm used is also the same), just the error and the convergence time will be analyzed in this section. Figures 4.18 to 4.21 show the results obtained during the simulation and then, the errors and convergence time will be analyzed.

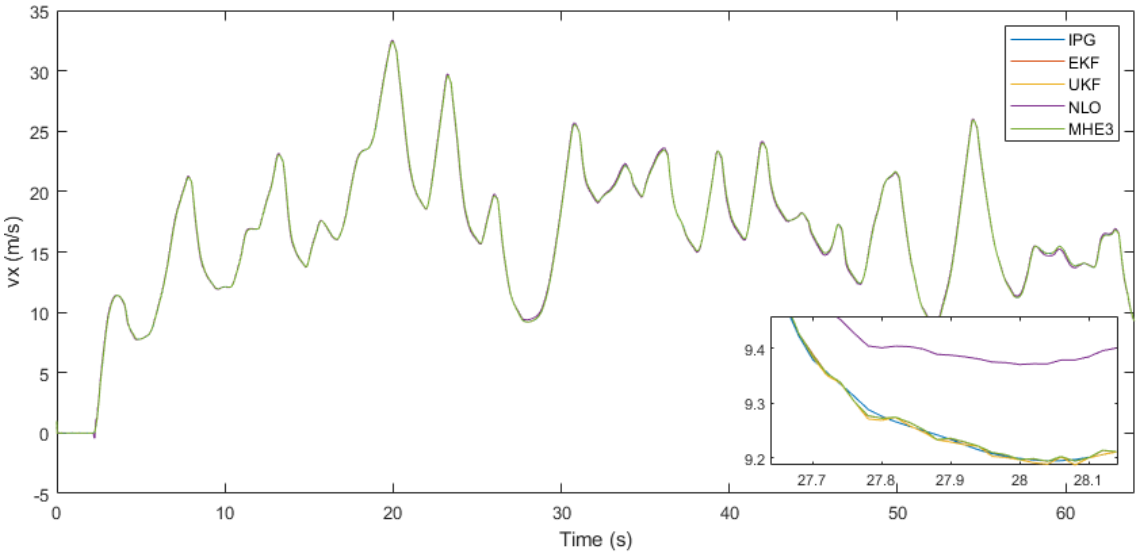

Figure 4.18: Longitudinal velocity comparison during an Autocross run.

In Figure 4.18, the longitudinal speed estimated with the four observers is represented with the IPG result. The four estimations have low error values during all the test and the longitudinal speed is properly estimated. The NLO has a higher error than the Kalmans Filters or the MHE but it is still a low error as it can be seen in Table 4.6, where the numerical results for the errors are summarized.

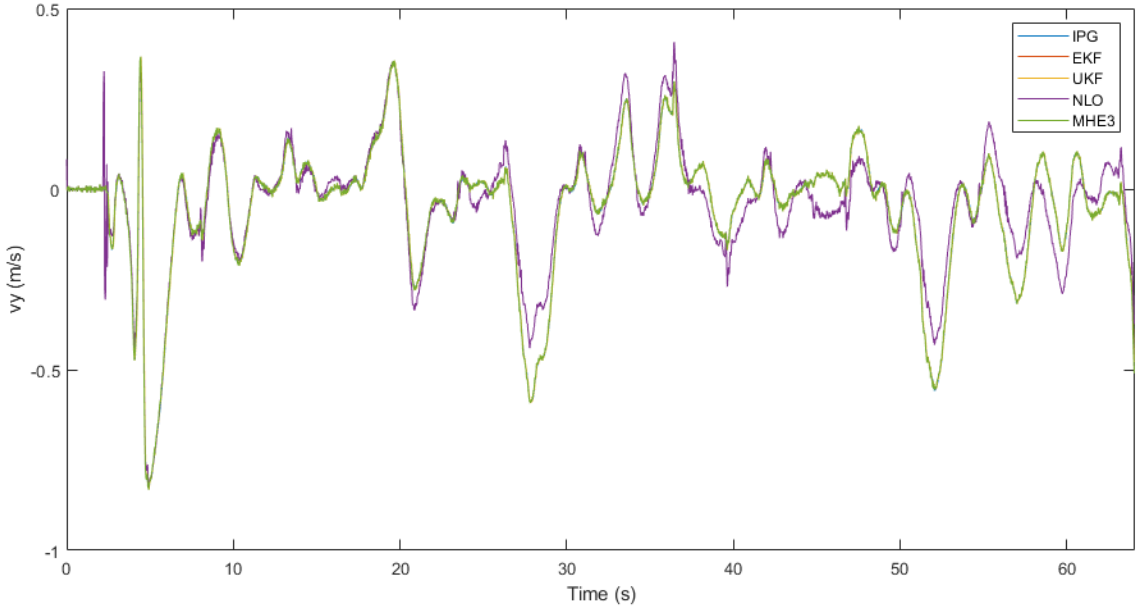

Figure 4.19: Lateral velocity comparison during an Autocross run.

Figure 4.19 shows the lateral speed estimated in an Autocross run. It can be seen how the NLO error is higher and it does not estimate the variable properly. This error could not be noticed properly in the Skidpad simulation (Section 4.5) as the speed remained almost constant during all the test. The other observers estimate the variable suitably during all the run.

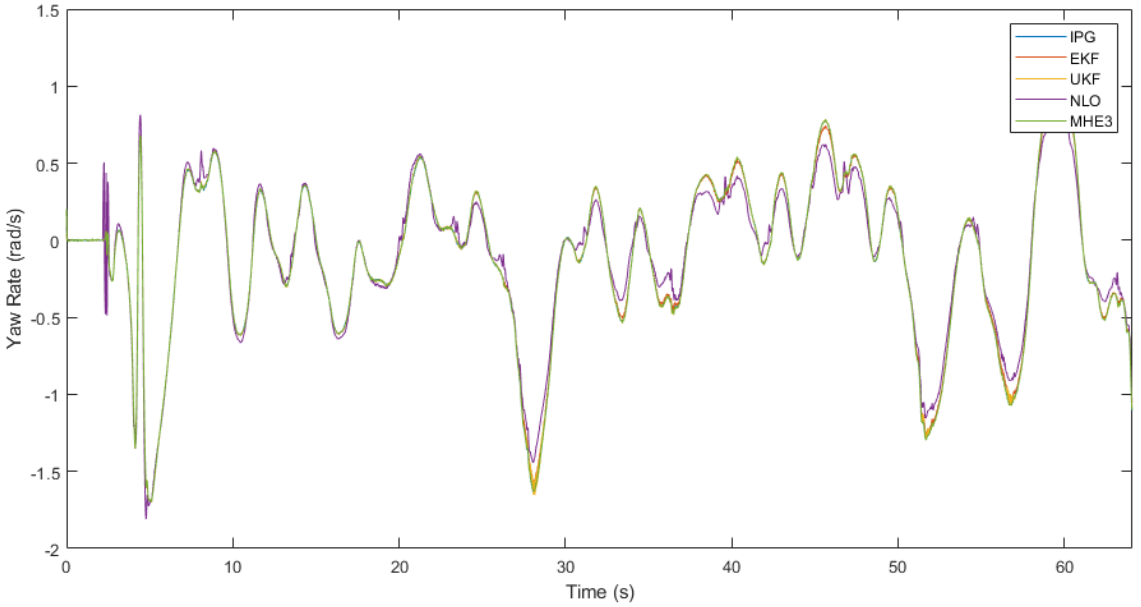

Figure 4.20: Yaw rate comparison during an Autocross run.

The last measurable variable of the state-space vector is the yaw rate, represented in Figure 4.20. It can be noticed how the NLO has a higher error than the other estimators mainly when the car is cornering (which is when the yaw rate presents some peaks). The numerical results are summarized in Table 4.6 as it has been commented previously.

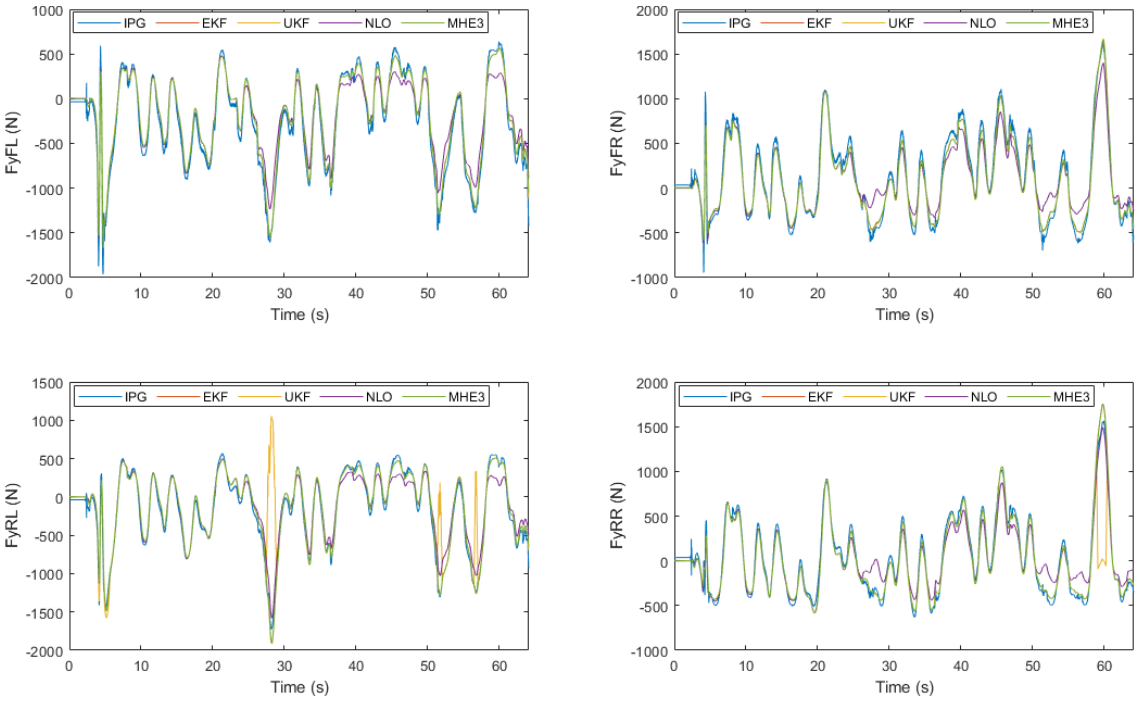

Figure 4.21: Tire lateral forces comparison during an Autocross run.

Finally, the four tire lateral forces are represented in Figure 4.21. The maximum errors when estimating these forces are obtained with the NLO, but with the UKF it can be noticed how there are some periods of time where it does not estimate the cornering forces properly. These specific cases happen for the estimation of the rear left and rear right tires and they take place because of the probabilistic assumptions of the Kalman Filter noises. Although it looks like it is going to diverge, the new set of measurement inputs (obtained with the sensor) redirect the system to the convergence again.

As mentioned before, the numerical results for the errors (mean and standard deviation) are summarized in Table 4.6.

The NLO estimates the variables with the higher mean error and dispersion. The other three observers present, generally, low mean errors, being the MHE the one with a higher standard deviation. In the rear left and rear right rows, it can be noticed the increase of the UKF mean and standard deviation error, which has been commented previously when analyzing Figure 4.21. These increase makes the errors to surpass the limits designated for the forces (5%). In the same lines, the NLO does not accomplish the requirements needed to consider the variables are properly estimated. To get a better perception of the errors (especially on the UKF commented ones), the normalized errors during all the simulation are represented in Figures 4.22 and 4.23.

In Figure 4.22, the kinematic variables errors are represented. It can be noticed how the MHE has a higher dispersion than the Kalman Filters and how the higher errors are obtained with the NLO. In the yaw rate estimation, the EKF and the UKF reach a higher error than the MHE during the final part of the test.

Table 4.6: State variable errors (in %) for an Autocross event simulation.

std / mean	EKF	UKF	NLO	MHE
Long. speed	0.0166 / 0.0197	0.0166 / 0.0198	0.2304 / 0.3457	0.0562 / 0.0200
Lat. speed	0.3671 / 0.4798	0.3671 / 0.4799	5.2272 / 5.7472	2.1554 / 0.5176
Yaw rate	1.0524 / 0.6639	0.5620 / 0.3409	3.4329 / 3.7235	0.3292 / 0.2462
$\mathbf{F_{y11}}$	1.9174 / 3.3419	1.9004 / 3.3375	4.5465 / 6.1700	2.4398 / 3.5033
$\mathbf{F_{y12}}$	2.0661 / 3.7727	2.0736 / 3.7557	5.7745 / 7.1076	2.6023 / 3.8772
$\mathbf{F_{y21}}$	1.7575 / 2.7670	22.3164 / 5.9571	4.2215 / 5.5997	2.1534 / 2.9393
$\mathbf{F_{y22}}$	1.9293 / 3.1320	12.8329 / 4.8222	4.8931 / 6.2176	2.2278 / 3.2589

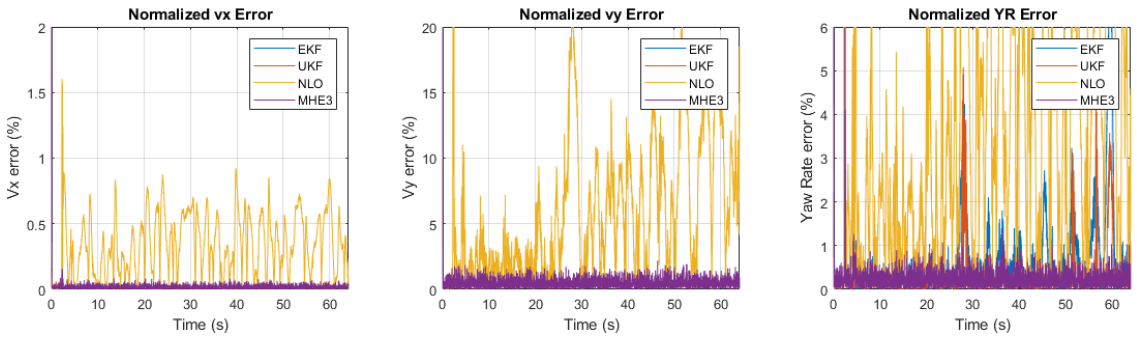

Figure 4.22: State variable relative error (kinematic variables) during an Autocross run.

Figure 4.23 represents the normalized error of the tire cornering forces. It is noticeable how all the observers, except the NLO, have a similar error values during all the Autocross simulation event. The UKF has a small error during all the test until the rear forces diverge a little bit for a low period of time. The high mean error observable in Table 4.6 are caused just because of this divergence period as seen in Figure 4.23.

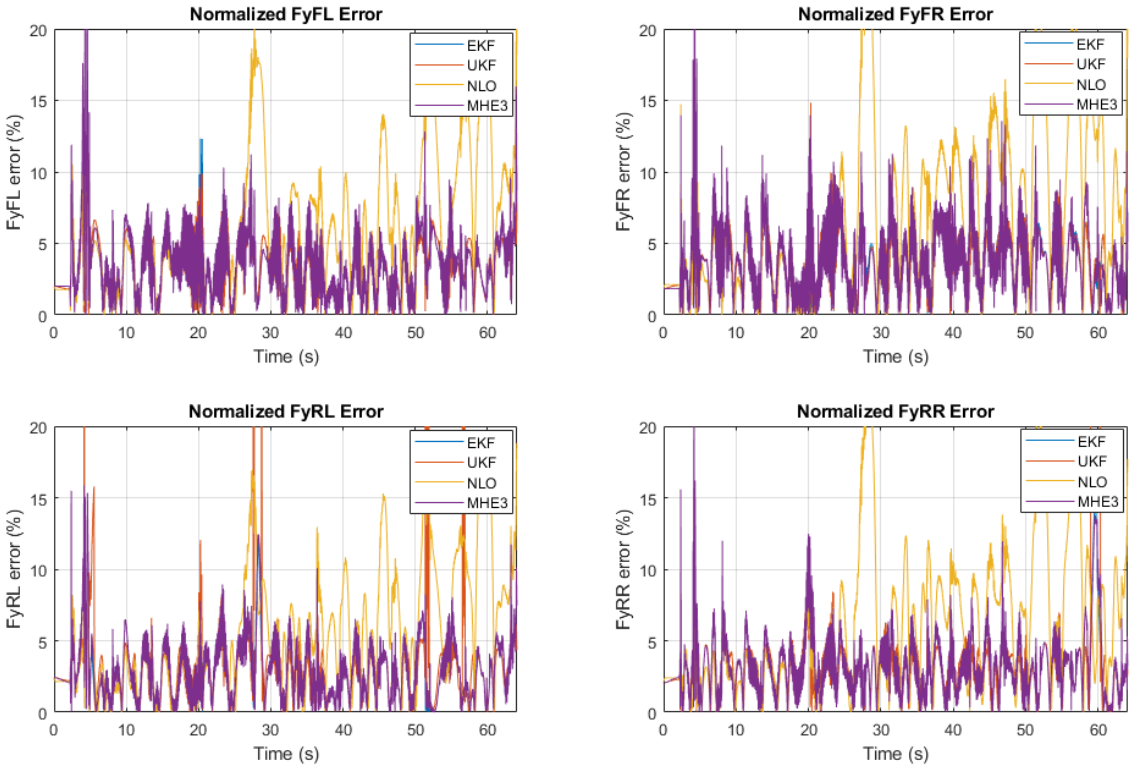

Figure 4.23: State variable relative error (lateral forces) during an Autocross run.

Finally, related to the observers errors, Table 4.7 summarizes the normalized mean error according to the measurement inputs of the linear velocities and yaw rate. The UKF and the MHE have the smallest errors and the NLO the highest ones. It is also important to take into account these results because in the future, when the chosen estimator is integrated in the car, the only way to compare and analyze the results will be comparing them with the measures obtained with the different sensors. Regarding the comparison of the errors in the Skidpad and Autocross, it is important to notice the improvement the MHE has had in comparison with the other estimators, mainly because the variables are all the time in transitory time and they do not remain stationary anytime. This fact could lead to some errors if just the Skidpad event was analyzed.

Table 4.7: State variables mean errors (in %) according to the sensor input.

Variable	EKF	UKF	NLO	MHE
Longitudinal velocity	0.0059	0.0063	0.3460	0.0057
Lateral velocity	0.0043	0.0038	5.6918	0.0403
Yaw rate	0.5951	0.2397	3.7079	0.1342

The last KPI that remains to be analyzed is the convergence time, which is represented in Table 4.8. Considering that the sampling time defined in the work is of 20ms, it can be considered that all four estimators converge in a proper amount of time.

Table 4.8: Convergence time.

	NLO	EKF	UKF	MHE
Samples	4	2	2	2

With the Skidpad and Autocross simulations done and analyzed, the final comparison according to the KPI will be done in Section 4.7.

4.7 Estimators comparison

In order to do the comparison between the four estimators, the three KPI defined in Section 4.2 will be taken into account, but also the assurance of convergence of them. As the estimator will be an input to the torque vectoring system of the car (as commented in Section 1.2), it is important the observer converges during all the time the car is driven and the results obtained with it are appropriate. In Table 4.9, the four observers are compared using the following notation:

- ✓: If the results are appropriate.
- ✗: If the results are inappropriate.
- /: If the results obtained are acceptable.

Table 4.9: Estimators comparison.

KPI	NLO	EKF	UKF	MHE
Assures convergence	✓	✗	✗	✓
Normalized error	✗	✓	✓	✓
Convergence time	✓	✓	✓	✓
Computational cost	✓	✓	✓	/

It has been seen in Sections 4.5 and 4.6 how the error obtained with the NLO is excessively high and it cannot be said it estimates the variables properly according to the limitations mentioned in Section 4.2. Between the other three estimators, the error results and convergence time are similar and proper. The main differences between them are the assurance of convergence and the computational cost. Related to the convergence, the Kalman Filters cannot assure it although the input measurements may redirect the estimator towards convergence (fast enough as seen with the convergence time of both, the EKF and the UKF). During the period of time they might diverge, the torque vectoring system will get an inappropriate input of the cornering forces and the response of the car will not be the expected and optimal. It has been already commented in Section 1.1 the importance of the assurance of convergence in terms of performance and security. Related to computational cost, the MHE requires a CPU time much higher than the EKF or the UKF, but it is still below the 20ms limit (sampling time of the main ECU).

It is also important to comment the differences between the Skidpad and Autocross simulations. It has been already mentioned that during a Skidpad run, the variables remain almost constant as it is a quasi-stationary event and so do the estimated variables that cannot be measured such as the lateral forces. This fact implies that the system is going to work most of the time in the linear or nonlinear part (depending on the tire and the vertical force) and the error related to the forces is going to remain almost constant during all the test. This condition could lead to a bad decision if just the Skidpad event is considered. This is why the Autocross simulation is also studied and it should be considered even more important. During the Autocross run, the variables are constantly changing its values alternating from the linear and the nonlinear part of the model. Regarding the errors with the IPG values, all of them have similar mean and standard deviation errors (without considering the divergence period of the UKF) and there is not a significant difference between them (see Tables 4.3 and 4.6). Regarding the normalized errors with the measured variables (which will be important when integrating the observer in the car), it can be seen in Tables 4.4 and 4.7 how the better results are obtained with the UKF and the MHE. Considering the condition mentioned previously about the relevance of the Autocross test, the MHE is the observer with the lowest errors. The error of the MHE with the sensor inputs decreases significantly between the Skidpad event and the Autocross run, specially for the longitudinal and the lateral velocity.

Taking into account these considerations, also represented in Table 4.9, and the importance of the assurance of convergence mentioned in Section 1.1, it is concluded that the estimator that best fits the requirements for the Formula Student car studied during this project is the MHE with a estimation horizon length of $N = 3$.

4.8 Summary

In this chapter, the discretization method (Euler and Tustin) has been analyzed. Some different estimation horizon lengths has been studied and compared in order to choose the one that bests fulfils the requirements exposed during all the work and Section 4.2. With the MHE completely defined, it has been compared with the NLO, the EKF and the UKF in two different track layouts and one of them has been chosen as the best option to integrate in the Torque Vectoring control system of the car. In the next chapters, the budget and the environmental impact are analyzed, respectively.

Chapter 5

Economical analysis

This chapter presents an economical analysis and the budget of this project. The work carried out is an analysis and design of a tire lateral force estimator based on the simulation software IPG and the hardware needed to implement it in the car will not be considered in the budget. The IPG software license will not be added to the economical analysis as it is provided to the team free of charge by the enterprise. Finally, the cost corresponding to external expenses such as the space and energetic consumption (light or water) used during the project will not be included in the economical study. The budget is broken down into parts in Tables 5.1 and 5.2, which correspond to the personal cost and material cost, respectively. It is also important to add some comments when analyzing the economical expenses of the project: the cost referred to software (Table 5.2) is null because the student status licenses and a future work related with state estimation would require less hours of learning and previous research.

Table 5.1: Economical analysis of the different tasks of the project.

Personal costs	Hours spent	Cost per hour	Total [€]
Research	240	20	4800
Vehicle and tire modeling	30	20	600
Model validation with the IPG CarMaker	30	20	600
Design and tuning parameters for the NLO	35	20	700
Design and tuning parameters for the EKF	45	20	900
Design and tuning parameters for the UKF	60	20	1200
Design and tuning parameters for the MHE	180	20	3600
Observers simulation and comparison	35	20	700
Total	655	-	13100

Table 5.2: Economical analysis of the material and software required for the project.

Material costs	Cost per unit	Total [€]
MATLAB student license	-	0
IPG CarMaker annual license	-	0
Computer cost	800	800
Total	-	800

Then, the total budget of the project is 13900€.

Chapter 6

Environmental impact

In this chapter, the environmental impact is presented. Although the work exposed during this project is a software design based on simulation and it does not have a direct environmental impact until it is integrated in the car, the indirect impact will commented briefly in this section.

The most important aspect to consider related to the environmental impact is the car where it is going to be implemented. As mentioned before, it is a completely electric vehicle, and as it is exposed in the work [Buekers et al., 2014], there are a lot of benefits instead of the combustion engine cars. In fact, the European Environmental Agency [AEMA, 2018] assures that during the vehicle life-cycle, the pollutant emissions are reduced between the $17 - 30\%$ if using an electric car instead than a combustion one.

The worst aspect according to the environmental impact is the construction of the vehicle. However, this topic is not related to the case this project is studying. However, there are some interesting aspects related to the use of the Torque Vectoring system control that should be commented, as the estimator is used as an input of it. The use of this type of system has a direct impact on the tire wear, which should be smaller than without using the Torque Vectoring control as it helps the car to control the undesired wheel spinning. This tire wear reduction has a direct environmental impact that could be studied in a further work. In the same lines, the implementation of these power distribution systems allow to reduce the consumption as it does just provide the optimal torque to the wheels and battery discharges in a more optimal way.

Chapter 7

Conclusions

All the work done during the realisation of this project and the accomplishment of the objectives defined are analyzed in this chapter, including its main contribution and the possibilities of future work from now on. Then, the conclusions are divided in three sections: Main contribution, conclusions on the project and further work and proposals.

7.1 Main Contribution

As mentioned in Chapter 1, the design of the estimator is mainly to incorporate it in the Torque Vectoring control system of the ETSEIB Motorsport car so the torque per wheel that it outputs is the optimal one at each time. The observer designed allows to estimate the forces properly and it assures the convergence during all the driving maneuver. This work not only provides the team with the lateral forces values for the Torque Vectoring control, but also for all the controllers or dynamic behavior analysis of the car. The current estimators can also be improved with the techniques presented in this project.

The work is also a basis in estimation principles and techniques for the future members of the team, especially for the Vehicle Dynamics Controls Department. Currently, there are not any optimal estimation techniques added in the control system of the car. With this work, a lot of options regarding the estimation of immeasurable variables are introduced and a lot of new future dynamic controls can be designed and added from now on.

7.2 Conclusions on the project

The main objective of this project has been accomplished as a tire cornering force estimator for a Formula Student vehicle has been designed and it does fulfil all the requirements to be implemented in the car. A discrete-time vehicle and lateral tire model have been presented and validated with the IPG CarMaker software. It has been modeled with a nonlinear state-space representation, defining all the vector fields and state and input vectors needed. The measurement vectors have been added according to the sensors available in the car.

Four different observers, based on different estimation techniques, have been designed and analyzed via the simulation of two different Formula Student competition tracks. All the observers converge and estimate properly the lateral forces. Defining some KPIs and acceptable errors, the MHE has been proposed as the most suitable estimator to be integrated in the car. It also assures convergence, which was one of the important topics mentioned in Chapter 1. This observer can now be added to the Torque Vectoring control

system as an input in the torque distribution algorithm in order to get the optimal force per tire and improve the vehicle performance, specially while cornering.

Regarding the normalized error analysis, the estimator proposed estimates the kinematic variables of the state vector with low mean errors. For the longitudinal velocity and the yaw rate, these mean errors are of the 0.025% and the 0.25%, approximately. The lateral speed is the kinematic variable estimated with higher errors, being them of the 1.186% in the Skidpad event and the 0.518% in the Autocross run. Comparing them with the sensor inputs, the normalized errors are even lower, being the highest error for the yaw rate input, which is of the 0.13%. Considering that the sensors provide a good value of the real variable, the MHE estimates them suitably. For the dynamic variables analysis, the normalized errors obtained are lower for the Skidpad event than for the Autocross run. They vary between the 2.2% and the 3% for the Skidpad simulation and between the 2.9% and the 3.9% for the Autocross test. All the errors are appropriate, as well as the convergence time.

7.3 Further work and proposals

This work leaves some future aspects to be done in the future, starting from the integration of the estimator in the current ETSEIB Motorsport car. The observer should be incorporated in the vehicle and validated with some testing in order to get more data and study the actual estimation results. With the real data measured, there should be a study and tuning of the parameters defined for the simulation and a adaptation them to the real vehicle conditions. These parameters include the noise matrices Q and R and the covariance matrix P_0 used to obtain the arrival cost of the MHE. The forgetting factor added in the estimator should also be analyzed with the real data measurements to study which λ value allows to get the better estimation for the cornering forces. Finally, another study of the estimation horizon length could be done as the results may be different with real testing conditions than the results obtained with simulation, and maybe, the estimation error decreases when the estimation horizon length gets bigger. All of this work should be considered and compared with the simulation results.

Another important aspect to consider as a future work is the improvement of the MHE algorithm, especially in terms of decreasing the computational cost. Although the main ECU should be able to compute the algorithm every 20ms as determined in Chapter 4, it has to be tested out and analyzed. These improvements can be done by changing some options of the solver used (*ipopt*) or optimizing the code with functions which require a less computational cost than the current ones.

According to the results obtained, the main difference between the UKF and the MHE was the assurance of convergence of the estimators. The normalized errors were similar and the elapsed time was lower for the Kalman Filter and a future addition of a UKF might be done if an error detection is also added with it. This would not provide the optimal solution but it is a proper alternative to the MHE in case it is impossible to integrate it in the car. Finally, the nonlinear model proposed can be extended considering the roll rate for the calculation of the vertical loads or the actual relaxation length used in (2.4) could be further analyzed.

Bibliography

[Abe, 2015] Abe, M. (2015). *Vehicle handling dynamics: theory and application*. Butterworth-Heinemann.

[AEMA, 2018] AEMA (2018). Electric vehicles from life cycle and circular economy perspectives. Available in: `https://www.eea.europa.eu/publications/electric-vehicles-from-life-cycle`.

[Andersson et al., 2019] Andersson, J. A. E., Gillis, J., Horn, G., Rawlings, J. B., and Diehl, M. (2019). CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36.

[Ascher and Petzold, 1998] Ascher, U. M. and Petzold, L. R. (1998). *Computer methods for ordinary differential equations and differential-algebraic equations*, volume 61, pages 51–56. Siam.

[Bishop et al., 2001] Bishop, G., Welch, G., et al. (2001). An introduction to the Kalman filter. *Proc of SIGGRAPH, Course*, 8(27599-23175):41.

[Buekers et al., 2014] Buekers, J., Van Holderbeke, M., Bierkens, J., and Panis, L. I. (2014). Health and environmental benefits related to electric vehicle introduction in eu countries. *Transportation Research Part D: Transport and Environment*, 33:26–38.

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

[Doumiati et al., 2010] Doumiati, M., Victorino, A. C., Charara, A., and Lechner, D. (2010). Onboard real-time estimation of vehicle lateral tire–road forces and sideslip angle. *IEEE/ASME Transactions on Mechatronics*, 16(4):601–614.

[Ghoneim and Chin, 2001] Ghoneim, Y. A. and Chin, Y.-K. (2001). Active brake control having yaw rate estimation. US Patent 6,169,951.

[GmbH, 2020] GmbH, I. A. (2020). Official website. `https://ipg-automotive.com/`.

[Hamann et al., 2014] Hamann, H., Hedrick, J. K., Rhode, S., and Gauterin, F. (2014). Tire force estimation for a passenger vehicle with the unscented kalman filter. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 814–819. IEEE.

[Imsland et al., 2006] Imsland, L., Johansen, T. A., Fossen, T. I., Grip, H. F., Kalkkuhl, J. C., and Suissa, A. (2006). Vehicle velocity estimation using nonlinear observers. *Automatica*, 42(12):2091–2103.

[Iserles, 2009] Iserles, A. (2009). *A first course in the numerical analysis of differential equations*, pages 8–10. Number 44. Cambridge university press.

[Kang et al., 2005] Kang, H., Jalil, M. A., and Mailah, M. (2005). Development of a vehicle

dynamic model (VDM) for virtual driving simulation.

[Kuster et al., 2015] Kuster, J., Toscano, B., Curtosi, G. N., Arechaga, T., and Tires-Argentina, F. (2015). Numerical simulation applied to the tire industry. In *Pan-American Congress on Computational Mechanics*.

[Luenberger, 1971] Luenberger, D. (1971). An introduction to observers. *IEEE Transactions on Automatic Control*, 16(6):596–602.

[Ma et al., 2010] Ma, Y., Wang, Z., Zhao, X., Han, J., and He, Y. (2010). A UKF algorithm based on the singular value decomposition of state covariance. In *2010 8th World Congress on Intelligent Control and Automation*, pages 5830–5835. IEEE.

[Milliken et al., 1995] Milliken, W. F., Milliken, D. L., et al. (1995). *Race car vehicle dynamics*, volume 400. Society of Automotive Engineers Warrendale, PA.

[Morera Torres, 2020] Morera Torres, E. (2020). *Real-Data-Based modeling and torque vectoring algorithm for a 4-Wheel-Drive Formula Student vehicle*. Treball final de grau, UPC, Escola Tècnica Superior d'Enginyeria Industrial de Barcelona.

[Nam et al., 2012] Nam, K., Fujimoto, H., and Hori, Y. (2012). Lateral stability control of in-wheel-motor-driven electric vehicles based on sideslip angle estimation using lateral tire force sensors. *IEEE Transactions on Vehicular Technology*, 61(5):1972–1985.

[OldBastardRacing, 2018] OldBastardRacing (2018). Available in:. `https://oldbastardsracing.com/tire-management/slip-angle/`.

[OptimumG, 2012] OptimumG (2012). Available in. `https://optimumg.com/asymmetric-tire-data-and-optimumtire/`.

[Pacejka, 2005] Pacejka, H. (2005). *Tire and vehicle dynamics*. Elsevier.

[Rajamani, 2011] Rajamani, R. (2011). *Vehicle dynamics and control*, pages 392–393. Springer Science & Business Media.

[Rao et al., 2003] Rao, C. V., Rawlings, J. B., and Mayne, D. Q. (2003). Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations. *IEEE transactions on Automatic Control*, 48(2):246–258.

[Ray, 1995] Ray, L. R. (1995). Nonlinear state and tire force estimation for advanced vehicle control. *IEEE Transactions on Control Systems Technology*, 3(1):117–124.

[Ray, 1997] Ray, L. R. (1997). Nonlinear tire force estimation and road friction identification: Simulation and experiments. *Automatica*, 33(10):1819–1833.

[Seward, 2014] Seward, D. (2014). *Race car design*. Macmillan International Higher Education.

[Troupe, 2011] Troupe, N. W. (2011). Optimization of steering geometry for ultra-high-mileage vehicles. Master's thesis, Pennsylvania State University.

[Trunas Bruguera, 2019] Trunas Bruguera, F. (2019). *Disseny i implementació d'un algoritme de torque vectoring per un vehicle Formula Student*. Treball final de grau, UPC, Escola Tècnica Superior d'Enginyeria Industrial de Barcelona.

[Wan and Van Der Merwe, 2000] Wan, E. A. and Van Der Merwe, R. (2000). The Unscented Kalman Filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*, pages 153–158. IEEE.

[Wan et al., 2001] Wan, E. A., Van Der Merwe, R., and Haykin, S. (2001). The Unscented

Kalman Filter. *Kalman filtering and neural networks*, 5(2007):221–280.

[Watkins, 2004] Watkins, D. S. (2004). *Fundamentals of matrix computations*, volume 64, pages 32–53. John Wiley & Sons.

[Wilkin et al., 2005] Wilkin, M. A., Crolla, D. C., Levesley, M. C., and Manning, W. J. (2005). Design of a robust tyre force estimator using an extended kalman filter. Technical report, SAE Technical Paper.