



MOTORVEHICLE
UNIVERSITY OF
EMILIA-ROMAGNA

ANTI-ROLL ACTIVE AERODYNAMIC SYSTEM

Applied Automatic Control
Advanced Automotive Electronic Engineering
University of Bologna

A.A. 2022-2023

Mauro Benedos, Andrea Bronconi and Alessandro Chistè

September 15, 2023

Contents

1	Introduction	1
1.1	Motivations	1
1.2	Contributions	2
1.3	State of art and literature comparison	2
1.4	Organisation of the manuscript	3
1.5	List of the symbols	4
2	ANTI-ROLL ACTIVE AERODYNAMIC SYSTEM	6
2.1	Model and Problem Formulation	6
2.1.1	Aerodynamic Force	7
2.1.2	Physical Model	8
2.1.3	Sensors	12
2.1.4	Parameters	16
2.1.5	State-Space Representation	17
2.2	Model Analysis	18
2.2.1	Linearisation	18
2.2.2	Linear system analysis	21
2.2.3	Reachability	25
2.2.4	Integral Action	26
2.2.5	Observability	27
2.3	Proposed Solution	28
2.3.1	Optimal Control Model	28
2.3.2	State Feedback and Integral Action	29
2.3.3	Observer	39
2.3.4	Feed Forward	47

3 Application	48
3.1 Simulator description	48
3.2 Simulation results	51
3.2.1 San Donato	51
3.2.2 Scarperia	55
3.2.3 Arrabbiata 2	59
4 Conclusions and further investigation	64
Bibliography	65

Abstract

As we know, every rigid body in three-dimensional space has six degrees of freedom: three of translational type and three rotational. However, analyzing the dynamics of a vehicle, regardless of its application, the most significant and important movement is the roll. It concerns the rotation around the longitudinal axis of the body. It can be observed very well at first sight when a car runs a turn at high speed: the inner side of the vehicle rises while the outer one lowers towards the ground. As you can imagine this phenomenon is harmful especially for track vehicles where each aerodynamic component is designed to work flat, or rather zero rolling degrees. Despite the various changes that are often made to chassis components or shock absorbers, the roll always occurs and with this project we wanted to develop an active aerodynamics system in order to oppose this phenomenon. Two aerodynamic elements on the rear (right and left) of a vehicle create vertical forces that can counteract variations due to lateral load transfers. In the following report is reported the best solution able to mitigate, as much as physically possible, the longitudinal rotation of the vehicle.

Chapter 1

Introduction

1.1 Motivations

One of the biggest problems related to the driveability of a vehicle concerns rolling. Inevitably, a car along a curve suffer this phenomenon: a load transfer to the outer wheels occurs that damage the balance and geometry of the entire system. Speaking of road cars, the phenomenon is not so influential. Obviously keeping a driving style not suitable for the type of vehicle and road covered can even lead to the rollover, causing considerable damage especially to passengers. It's a complex phenomenon and depends on numerous factors such as the mass of the car (the greater the mass, the greater the roll), the track (proportional to the roll angle), the height of the centre of gravity and the rolling centre. In fact, a sports vehicle (low and light) will suffer less rolling than an SUV (heavy and with a high center of gravity). Shifting attention to vehicles intended for track use, the only elements on which mechanics are able to act are the suspensions and anti-roll bars, which attempt to block the extension of the spring inside the curve to prevent loss of grip. The bars must not be either too soft or too rigid, to avoid balancing errors and the generation of understeer or oversteer, obviously harmful for the performance of the vehicle.



Figure 1.1: Pagani Huayra Active Aerodynamics

Thinking of a sports car immediately comes in mind the image of the Pagani Huayra (Figure 1.1), equipped with four mobile flaps (two front and two rear) regulated by a control unit. With this project we want to adopt a system inspired by the great innovation brought by Pagani, able to generate vertical forces through two aerodynamic components and minimize roll.

1.2 Contributions

In this paper we wanted to develop a control system not too far from real applications and not too difficult to install on sport vehicles intended mainly for track use. Trying to minimize the lap time and make the car as stable as possible in the path of fast curves, the control unit would go to operate the actuator to increase the incidence of the internal flap. Looking in the literature appeared several articles of similar systems already developed. They all used different approaches, depending on the vehicle being examined and the road where it was to be driven. In our case, in order to model the problem in the right way, we have chosen the modeling using the *Quarter-Car* model because it is easier to implement but still gives significant results.

1.3 State of art and literature comparison

Over time, several studies have been proposed regarding active aerodynamics. Its development has been carried out to solve various problems related to braking, acceleration or even vehicle grip. In particular, several articles on roll control have been published by researchers. In 2020, in the magazine "Electronics" released by MDPI, a study was published on roll control and the improvement of the driveability of a vehicle through the use of aerodynamic profiles. Through a half-car model, the authors of the research were able to develop an automatic control system capable of acting even on non-flat roads.



Figure 1.2: Bugatti Aerodynamics System in the straight



Figure 1.3: Bugatti Aerodynamics System in breaking maneuver

The first popular introduction of active aerodynamics concerns Bugatti. With the Veyron, presented to the world in 2001, a system of automatic setting of the incidence of the rear wing was developed in order to increase grip at high speeds and improve performance in braking maneuvers (Figure 1.3). There is also a small spoiler used at very high speeds (over 300 km/h) to improve stability and avoid the use of the large wing that would greatly increase aerodynamic resistance (Figure 1.2). The solution was then adapted on other sports cars but always using a single spoiler. The system developed in this study, however, is inspired by the introduction brought later by Pagani in 2011. On the Pagani Huayra are actually installed four mobile aerodynamic profiles to have a greater (and faster) control of the entire system in both curves and braking. The result is a car in continuous mutation, the aerodynamic profile varies constantly to ensure the least possible friction and maximum downforce for every moment. The aim of this project is to replicate the system of the Huayra but adopting only two rear elements for the control of the roll in the curve maneuvers, especially at high speed, neglecting the possibility of operation in braking.

1.4 Organisation of the manuscript

In Chapter 1 it is reported an introduction to this project with some references to real case already implemented in automotive industry. Then a list of most of the symbols used in this work has been reported to helping the equation reading, the theoretical analysis and the whole control configuration.

In Chapter 2 a physical and mathematical modeling of the real problem is reported. The analysis regards all the properties of the system and then the state-space representation. Afterwards, the model analysis is done including linearisation of the system, the open loop eigenvalues configuration and finally the control part. From reachability and observability to the integral action and then the state feedback stabilizer and the observer design process.

In Chapter 3, all the plants and blocks of *Simulink* environment are shown. Finally all the simulations performed are reported in order to test our control solution of the roll-dynamics problem.

1.5 List of the symbols

Here list all the symbols used in the manuscript:

List of symbols		
Symbol	Meaning	IS Unit
\mathbf{x}	State vector	-
\mathbf{x}_0	State initial vector	-
\mathbf{y}	Measurement vector	-
\mathbf{e}	Controlled output vector	-
\mathbf{d}	Disturbance vector	-
\mathbf{r}	Reference vector	-
\mathbf{u}	Control vector	-
$\mathbf{\nu}$	Noise vector	-
\mathbf{w}	Exogenous vector	-
C_L	Lift coefficient	-
$C_{L\alpha_0}$	Zero-Lift coefficient	-
α	Incident angle of the wing	-
L	Lift Force	[N]
ρ	Air density	[$\frac{kg}{m^3}$]
v	Vehicle speed	[$\frac{m}{s}$]
S	Aerodynamic profile section	[m^2]
m_s	Sprung mass	[kg]
m_u	Unsprung mass	[kg]
g	Gravity acceleration	[$\frac{m}{s^2}$]
k_s	Spring stiffness	[$\frac{N}{m}$]
β_s	Suspension damping coefficient	[$\frac{Ns}{m}$]
k_t	Tire stiffness	[$\frac{N}{m}$]
l_{0s}	Suspension initial length	[m]
l_{0t}	Tire initial length	[m]
θ	Roll angle	[]
p_{ref}	Zero-speed vehicle height	[m]
t	Vehicle track	[m]

a_y	Lateral acceleration	$[\frac{m}{s^2}]$
h_{COM}	Center of mass height	$[m]$
$\tilde{\mathbf{x}}$	Linearised state vector	-
\mathbf{x}^*	State equilibrium vector	-
$\tilde{\mathbf{y}}$	Linearised output vector	-
\mathbf{y}^*	Output equilibrium vector	-
$\tilde{\mathbf{e}}$	Linearised controlled output vector	-
\mathbf{e}^*	Controlled output equilibrium vector	-
$\tilde{\mathbf{u}}$	Linearised control vector	-
\mathbf{u}^*	Control equilibrium vector	-
$\tilde{\mathbf{w}}$	Linearised exogenous vector	-
\mathbf{w}^*	Exogenous equilibrium vector	-
\mathbf{I}	Identity matrix	-
\mathbf{J}	Jordan canonical form	-
\mathbf{V}	Eigenvectors matrix associated to \mathbf{A}	-
\mathbf{z}	Vector of the LTI modes	-
\mathbf{R}	Reachability matrix	-
\mathbf{x}_e	Extended state vector	-
η	Integral of regulated output	-
\mathbf{K}_S	State feedback matrix	-
\mathbf{K}_I	Integral action matrix	-
\mathbf{O}	Observability matrix	-
\mathbf{K}_O	Observer matrix	-
ϵ	State combination to penalize	-
J	Cost function	-
\mathbf{Q}	ϵ weight inside J	-
\mathbf{R}	Control cost matrix	-
σ_{pot}	Precision of the potentiometer	-
σ_{las}	Precision of the laser sensor	-

Chapter 2

ANTI-ROLL ACTIVE AERODYNAMIC SYSTEM

2.1 Model and Problem Formulation

First of all is mandatory to analyse the system model which will be the starting point for the whole control problem formulation. Let's define the main mathematical quantities:

$$\begin{aligned}\dot{\mathbf{x}} &= f(x, u, w) & \mathbf{x}(t_0) &= \mathbf{x}_0 \\ \mathbf{y} &= h(x, u, w) \\ \mathbf{e} &= h_e(x, u, w)\end{aligned}\tag{2.1}$$

and

$$\begin{aligned}\mathbf{x} &= \text{Plant} & \mathbf{x} &\in \mathbb{R}^n \\ \mathbf{d} &= \text{Disturbances} & \mathbf{d} &\in \mathbb{R}^{ld} \\ \mathbf{r} &= \text{References} & \mathbf{r} &\in \mathbb{R}^{lr} \\ \mathbf{u} &= \text{Controls} & \mathbf{u} &\in \mathbb{R}^p \\ \mathbf{y} &= \text{Measurements} & \mathbf{y} &\in \mathbb{R}^q \\ \mathbf{e} &= \text{Goals} & \mathbf{e} &\in \mathbb{R}^{lm} \\ \boldsymbol{\nu} &= \text{Noises} & \boldsymbol{\nu} &\in \mathbb{R}^q\end{aligned}\tag{2.2}$$

where $n, ld, lr, p, q, lm \in \mathbb{N}$.

And finally we can define the *exogenous* vector \mathbf{w} :

$$\mathbf{w} = \begin{bmatrix} \mathbf{d} \\ \boldsymbol{\nu} \\ \mathbf{r} \end{bmatrix}\tag{2.3}$$

To solve in the right way the control problem, we need to state some assumption:

1. w is not observable, so the system needs to be design in order to reach the goals despite the disturbances
2. d has to be bounded
3. r is fixed and known a priori
4. If $\mathbf{u} \in \mathbb{R}^p$ and $\mathbf{e} \in \mathbb{R}^{lm}$, we claim $p \geq m$ in order to be sure to find always a solution for the control problem
5. e is readable from y

2.1.1 Aerodynamic Force

As mentioned in the previous chapter, one of the aims of this project was to choose an appropriate aerodynamic profile to generate a load to counteract the height variation of the car generated by the rolling moment of the car itself while cornering. The profile is ideally mounted on the same axis of the suspension system and by varying the angle of incidence of the flap respect to the air flux, with some sort of electrical actuator, it can generate a variable vertical load.

The modeling of the aerodynamic force was centered around the *thin airfoil theory* [3], which in spite of some seemingly stringent assumptions as regards the airfoil geometry and type of motion, is capable of providing excellent first-order estimates of the lift forces. To be crystal clear, the main assumption and approximation of this theory is to consider that the thickness does not contribute to the load generated by the airfoil which in reality is indeed not true as thickness have a small and non-trivial influence, but for our purpose the results of this theory were enough to describe and analyze the model.

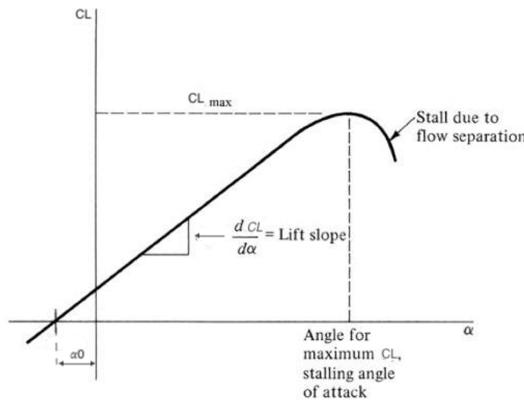


Figure 2.1: $C_L - \alpha$ curve for a generic profile

Therefore, according to the theory, the following linear relation for the lift coefficient is given:

$$C_L(\alpha) = C_{L\alpha_0} + \frac{dC_L}{d\alpha} \cdot \alpha \quad (2.4)$$

where $C_{L\alpha_0}$ is the zero-lift coefficient (i.e the coefficient related to the angle of attack for which the lift force of the airfoil is zero), $\frac{dC_L}{d\alpha}$ is the lift slope that according to the theory is set at 2π and α is the angle of attack which is usually bounded between $[-15^\circ, 15^\circ]$ in order to avoid the stall of the profile and to not exceed the linear region of C_L (2.1). For our purpose we need the lift force which can be expressed as follow:

$$L = \frac{1}{2} \cdot \rho \cdot v^2 \cdot S \cdot (C_{L\alpha_0} + 2 \cdot \pi \cdot \alpha) \quad (2.5)$$

where

- ρ is the air density ($1.225 \frac{kg}{m^3}$)
- v is the speed of the vehicle expressed in $\frac{m}{s}$
- S is the section of the aerodynamic profile (in this case $0.05 m^2$)
- α is the incident angle of the wing

In our case of study, the airfoil is mounted on the car to generate an aerodynamic load only downwards in order to push back to a reference value the suspension while they are extending and not compressing, so in other words we want that our airfoil works between the $[0, 15^\circ]$ region.

The airfoil profile that was selected for this purpose is the NACA0024, that according to the international standard is a symmetric profile for which the $C_{L\alpha_0}$ is null and for which we extrapolated the necessary data to estimate the value of surface S .

The choice of a symmetric profile was driven by the fact that the amount of load required in our application was not that high to justify the use of a cambered profile which is able to provide more lift at the same angle of incidence compared to a symmetric one.

The final consideration that we can make regards the aerodynamic drag generated by the profile: since for our application all the meaningful loads are directed along the vertical axis and since the drag is an horizontal force, his influence can be neglected for this particular case of study. We want to remark the fact that in the overall balance of the parameters that drives the design of a vehicle, drag plays one of the most significant roles and should not be forsaken at all.

2.1.2 Physical Model

This control problem requires the formulation of a physic model able to describe as well as possible the behaviour of the real phenomenon. The system is studied using a simple “Quarter-Car Model” as in Figure 2.2.

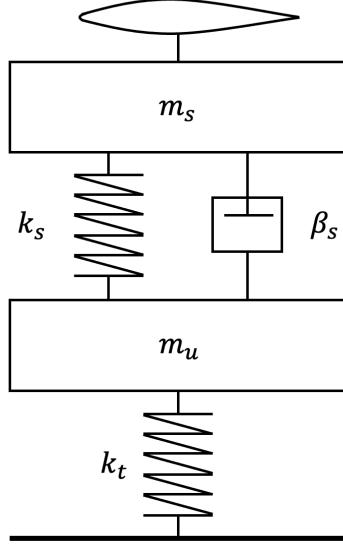


Figure 2.2: *Quarter-Car* Model

The model is composed of: two masses, the upper one m_s ms and the lower one m_u representing, the *sprung* and *unsprung* masses, one suspension, one tire and the aerodynamic profile.

The sprung mass is supported by the suspension and above it has the aerodynamic element used to generate the active force, while the unsprung mass is the portion of mass not supported by the suspension system. This last one is composed of two passive component, a spring and a damper (which are characterised by a stiffness k_s and a damping coefficient β_s). The tire is assumed as a purely-spring system simply for the great stiffness k_t that characterizes it.

It is also important to mention the two sizes l_{0s} and l_{0t} that represent, respectively, the lengths of the spring at rest and the height of the tire side.

The whole dynamics of this system can be represented by writing its equation of motion, which are:

$$\begin{cases} m_s \cdot \ddot{z}_s = -L - m_s \cdot g + F_s + F_d + F_{roll} \\ m_u \cdot \ddot{z}_u = -m_u \cdot g + F_t - F_s - F_d \end{cases} \quad (2.6)$$

where

- $F_s = -k_s \cdot (z_s - z_u - l_{0s})$ is the reacting force of the spring
- $F_d = -\beta_s \cdot (\dot{z}_s - \dot{z}_u)$ is the reacting force of the dumper
- $F_t = -k_t \cdot (z_u - z_r - l_{0t})$ is the reacting force of the tire, which is modelled as a spring

where L is the aerodynamic force, which formulation has already been described in the Section 2.1.1, and the meaning of F_{roll} will be described later.

Considering this model, the two masses are free to move along the vertical axis (*z-axis*) producing a 2-DOF system as we can see in Figure 2.3, where z_s and z_u represent, respectively, the vertical position of the sprung and unsprung masses. An important point regards the height of the road profile z_r which is reported in the whole control analysis (also its speed \dot{z}_r and acceleration \ddot{z}_r) but it will be set to zero considering a flat road surface. This is only due to a simplification of simulations and calculation but this doesn't mean that the system is not able to work even with irregular road profiles.

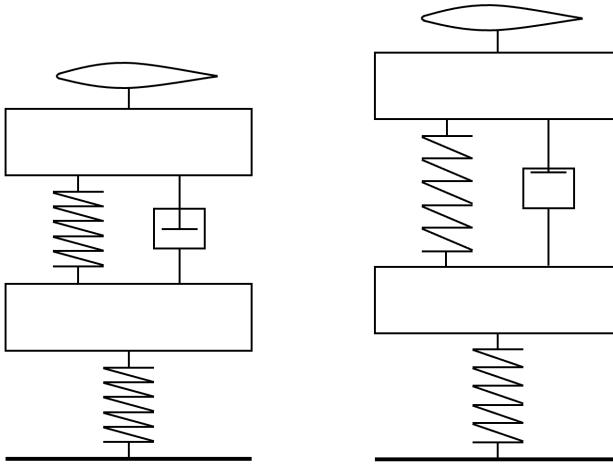


Figure 2.3: Model at rest on the left and model in extension on the right

The choice to implement a *Quartet-Car* model was not trivial. Logically, the most plausible idea would be to analyze the system using a *Half-Car* model. But, as a result of the complexity of the rotations due to the roll, it was more useful and precise the model that does not include the entire rear axle but only one suspension-wheel system.

But it was necessary to find a mathematical relationship that linked the angle of roll (interest of this project) to a measurable value in the *Quarter-Car* model. Looking at the complete model during a turn (Figure 2.4), the solution appeared immediately clear.

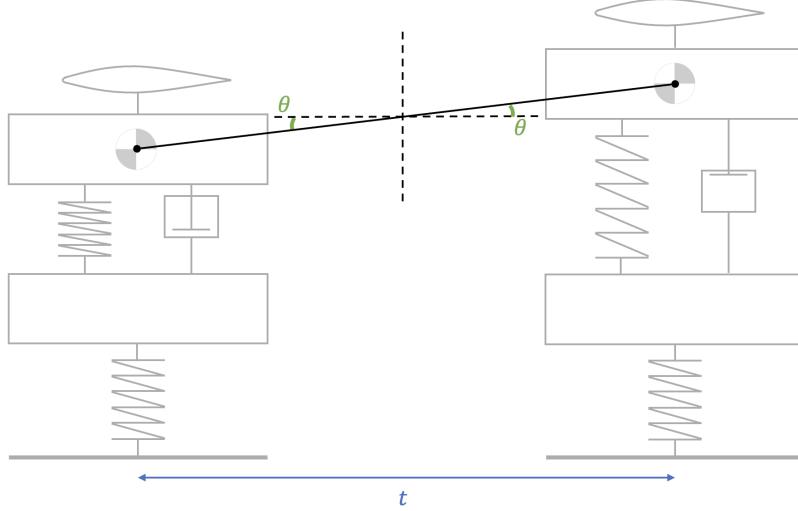


Figure 2.4: Coupling of two *Quarter-Car* performing a right turn

Knowing before all the data related to the vehicle system, and measuring in real time the height of the mass m_s (z_s) you can estimate the angle of roll also from the *Quarter-Car* model using the following equations:

$$\tan \theta = \frac{z_s - z_u - p_{ref}}{\frac{t}{2}} \quad (2.7)$$

where θ is the roll angle, t is the track of the vehicle and p_{ref} is the height of the load vehicle in a zero-speed situation calculated according to a simple balance of forces. Starting from Equations (2.6):

$$p_{ref} = l_{0s} - \frac{m_s \cdot g}{k_s} + \frac{L_0}{k_s} \quad (2.8)$$

A clarification on the $\tan \theta$ term. To greatly simplify the calculations and the development of the control system, it was chosen to adopt the hypothesis of small roll angles. Based on the developments of Taylor-Maclaurin we can say that:

$$\tan \theta = \sin \theta = \theta \quad (2.9)$$

where θ is the angle in radians. Considering also that in vehicle maneuver the maximum roll angle is around 10 degrees, this assumption is fairly compelling.

As regards F_{roll} , which appears in the balance of forces, it comes when the vehicle perform a curve. In a turn, the car undergoes the effect of the centrifugal acceleration a_y generating a force F_y according to the *Second Newton Principle*, which, acting on the center of gravity, generates a moment of roll over the entire system. This moment develops two forces on the wheels of equal

modulus but in the opposite direction, one tries to lift the wheel inside the turn, while the other to push the outside one. This force is called F_{roll} and regarding this project it has always a positive upward direction, as, as mentioned above in the Section 2.1.1, the aerodynamic element is capable only of generating a vertical downward force, it cannot produce a vehicle lift.

The formulation is reported here:

$$F_y = m_s \cdot a_y = m_s \cdot \frac{v^2}{R} \quad (2.10)$$

$$F_{roll} = \frac{F_y \cdot h_{COM}}{t} \quad (2.11)$$

where R is curvature radius of the turn and h_{COM} the height of center of mass.

2.1.3 Sensors

Before going on with the whole analysis, we have to shift the focus on the sensors to be installed in order to complete the plant system. The choice is to put:

- A **POTENTIOMETER**, installed in parallel to the spring, to measure the suspension deflection in order to achieve the distance between m_s and m_u
- A **LASER**, mounted ideally at the center of gravity of the sprung mass, which provides the distance of m_s from the ground

By equipping our *Quarter-Car* model with these two sensors, we are able to measure all the physical quantities needed to develop this automatic control project. The output signals registered are given as follows:

$$y_1 = \text{Output of the Potentiometer} = z_s - z_u - l_s \quad (2.12)$$

$$y_2 = \text{Output of the Laser} = z_s - z_r \quad (2.13)$$

where l_s is the length of the suspension calculate in a static-load condition and it is equal to p_{ref} determined in (2.8).

But, it's important to say that these are real sensors, so of course they are characterised by a certain level of mathematical instability and inaccuracy. This fact is reflected by adding some noise to the measurements:

$$y_1 = z_s - z_u - p_{ref} + \nu_1 \quad (2.14)$$

$$y_2 = z_s - z_r + \nu_2 \quad (2.15)$$

where ν_1 and ν_2 are respectively the noises of the potentiometer and the laser.

Potentiometer

The first quantity to be determined is the distance between the sprung mass and the unsprung one. The device chosen is an **AVIO RACE** potentiometer (model **DIA13 – 100** in Figure 2.5 and 2.6), which is a linear electronic device composed of a linear resistor and a movable slider.

The name is related to its diameter (13 mm). The digits after, in the name, represent the maximum measurable extension expressed in mm. We choose 100 mm due to the fact that in our simulations the system goes, in open loop, near to 50 mm, so, to avoid any mechanical issue in different real conditions we decided to install one with a longer extension.



Figure 2.5: AVIO RACE DIA 13-XX potentiometer

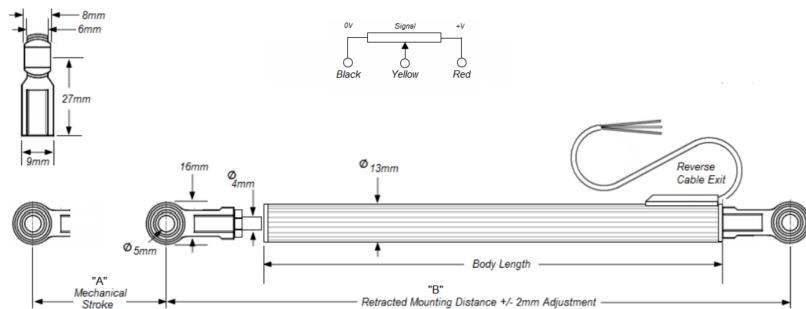


Figure 2.6: AVIO RACE DIA 13-XX potentiometer blueprint

Code	”A” Measurement range mm (± 0.5 mm)	”B” Retracted mount- ing distance mm	Resistance ($\pm 20\%$) k Ω
DIA13 – 100	100	248	6.7

Table 2.1: AVIO RACE DIA-100 technical specifications

This potentiometer is ideal for car suspensions due to its material composition. All technical data are reported below:

- Maximum supply voltage: 40 V DC
- Resolution: infinite
- Repeatability: ≤ 0.01 mm
- Operational speed: 10 m/s max
- Mechanical life: ≥ 25 milion cycles
- Temperature range: -40°C to +125°C
- Protection class: IP65
- Independent linearity: $\leq \pm 0.5\%$
- Cable type: Raychem 55 A 24 AWG
- Cable length: 500 mm
- Housing: aluiminium
- Spherical bearing: $\phi 5$ mm
- Exit cable: STD → reverse cable exit, optional → forward cable exit

Laser

Having now a way to determine the distance between the masses, it is mandatory to find a way to obtain the overall height of the system. The solution adopted is to install a laser sensor on the sprung mass (as said before ideally mounted in the COG) that look to the ground. With this we are able to measure the distance $z_s - z_r$ and also, by simply mathematical steps the height of the unsprung mass from the road.

The device is an **MICRO – EPSILON** laser sensor (model **optoNCDT 1750-750** in Figure 2.7 and 2.8), which is a distance sensor designed specifically for motorsport application.



Figure 2.7: MICRO-EPSILON optoNCDT 1750-750 laser

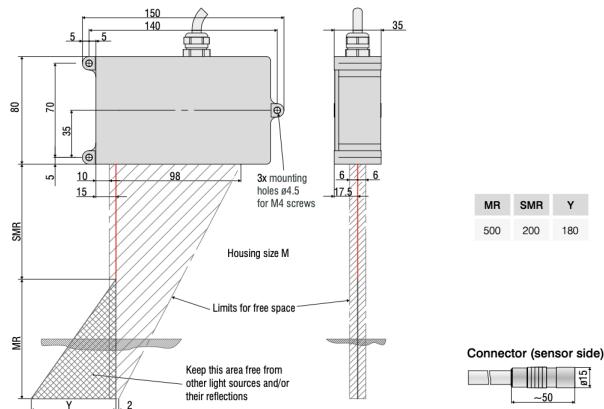


Figure 2.8: MICRO-EPSILON optoNCDT 1750-750 laser blueprint

This is a powerful laser triangulation sensor which is used at high speed characterised by precise measurements and innovative algorithms which provide high accuracy. The exposure time is optimally matched to the reflection characteristics of the target surface. All technical data are reported below:

- Measuring range: 750 mm
- Start of measuring range: 200 mm
- Mid of measuring range: 575 mm
- End of measuring range: 950 mm
- Repeatability: 30 μm
- Measuring rate: 0.3-7.5 kHz

- Linearity: $< \pm 670 \mu\text{m}$
- Light source: semiconductor laser $< 1 \text{ mW}$, 670 nm
- Permissible ambient light: 10.000 lx
- Power consumption: $< 3 \text{ W}$ (24 V)
- Temperature range: 0 to $+50^\circ\text{C}$
- Protection class: IP65
- Weight: 600 g

The maximum value of *Measuring distance* is 950 mm. This is not a problem, because under load condition and maximum extension of the spring, the sprung mass height is however slightly less than 650 mm.

Regarding both devices, all specifications meet the requirement of the project in terms of temperature range, precision and accuracy.

2.1.4 Parameters

The numerical parameters selected for this project are reported below in the Table 2.2:

Parameter	Value	Unit
g	9.81	m/s^2
m_s	350	kg
m_u	35	kg
k_s	50000	N/m
β_s	3000	$(\text{N} * \text{s})/\text{m}$
k_t	500000	N/m
l_{0s}	0.50	m
l_{0t}	0.10	m

Table 2.2: Quarter-car numerical parameters

One of the biggest challenge of the project was the lack of reliable data about all the parameters needed for the simulations. Having developed the project with reference to sports cars, the interested car manufacturers are very reserved about the diffusion of technical data. The only values almost certain are the masses, in fact, starting from values of real masses and using some catalogs of cars components, it was possible to find acceptable and physically consistent

values of stiffness and damping coefficient. By looking at the simulation graphs and applying a reverse-engineering process, it was possible to find all the data needed to carry out all the analyses required by the project.

2.1.5 State-Space Representation

Finally, the goal of this project is to implement an automated system able to reduce the roll angle generating a vertical force thanks to an aerodynamic profile. In fact, we can see in the system of equations of motion (2.6) that the control α appears (in the lift force L), and also the main disturbances, represented by the F_{roll} , the gravity acceleration g and the speed v appear in the equations.

The sensors which are installed on our *Quarter-Car* model, which are described in detail in Section 2.1.3 measure the suspension lenght stroke and the height z_s . The sensors outputs, as we'll see later in detail in Section 2.1.3, can be described by the following equations:

$$\begin{cases} y_1 = z_s - z_u - p_{ref} + \nu_1 \\ y_2 = z_s - z_r + \nu_2 \end{cases} \quad (2.16)$$

and also the regulated output can be defined, using Equation (2.7), as:

$$e_1 = \theta = \frac{z_s - z_u - p_{ref}}{\frac{t}{2}} = \frac{y_1}{\frac{t}{2}} \quad (2.17)$$

For control purpose, the non-linear system describing the dynamics of the plant, the measured output and the regulated output need to be written in the *State-Space Representation*. The whole dynamics must be described using a set of linear first order differential equation.

The vectors required for this representation are the following:

$$\mathbf{x} = \begin{bmatrix} z_s - z_u \\ \dot{z}_s - \dot{z}_u \\ z_u - z_r \\ \dot{z}_u - \dot{z}_r \end{bmatrix} \quad (2.18)$$

$$\mathbf{u} = [\alpha] \quad (2.19)$$

$$\mathbf{w} = \begin{bmatrix} \mathbf{d} \\ \boldsymbol{\nu} \\ \mathbf{r} \end{bmatrix} \quad \mathbf{d} = \begin{bmatrix} \ddot{z}_r \\ F_{roll} \\ g \\ v \end{bmatrix} \quad \boldsymbol{\nu} = \begin{bmatrix} \nu_1 \\ \nu_2 \end{bmatrix} \quad \mathbf{r} = [p_{ref}] \quad (2.20)$$

where

- \mathbf{x} is the State Vector
- \mathbf{u} is the Control Vector
- \mathbf{w} is the Exogenous Vector already seen in Section 2.1

Then, by substituting (2.19) and (2.20) into (2.6), (2.16) and (2.17), it is possible to obtain the form of the plant:

$$\begin{cases} \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, \mathbf{w}) & \mathbf{x}(t_0) = \mathbf{x}_0 \\ \mathbf{y} = h(\mathbf{x}, \mathbf{u}, \mathbf{w}) \\ \mathbf{e} = h_e(\mathbf{x}, \mathbf{u}, \mathbf{w}) \end{cases} \quad (2.21)$$

Finally, thanks to the previous substitutions within the function $f(x, u, w)$, we can achieve the new system composed of four linear ordinary differential equations:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\frac{m_s+m_u}{m_u \cdot m_s} \cdot k_s \cdot (x_1 - l_{0s}) - \frac{m_s+m_u}{m_u \cdot m_s} \cdot \beta_s \cdot x_2 - \frac{L}{m_s} + \frac{k_t}{m_u} \cdot (x_3 - l_{0t}) + \frac{F_{roll}}{m_s} \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = -g + \frac{k_s}{m_u} \cdot (x_1 - l_{0s}) + \frac{\beta_s}{m_u} \cdot (x_2) - \frac{k_t}{m_u} \cdot (x_3 - l_{0t}) - \ddot{z}_r \end{cases} \quad (2.22)$$

And also, it is possible to obtain for the measured output function $h(x, u, w)$ the following expression:

$$\begin{cases} y_1 = x_1 - r + \nu_1 \\ y_2 = x_1 + x_3 + \nu_2 \end{cases} \quad (2.23)$$

Finally, for the regulated output function $h_e(x, u, w)$ it is derived:

$$e_1 = \frac{y_1}{\frac{t}{2}} = \frac{x_1 - r}{\frac{t}{2}} \quad (2.24)$$

2.2 Model Analysis

The all model equations can be linearised and then analysed in order to be able to design a robust optimal control solution.

2.2.1 Linearisation

As first step, the model analysis starts with the linearisation of the system because, thanks to this procedure, we will be able to control better the system. The goal is to find a *linearisation point* $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{w}^*)$ in which the system is supposed to operate, in order to approximate the non-linear equations (2.21) using new linear differential equations. This point needs to be an equilibrium point, in fact

we will call it *equilibrium triplet* and the choice can be made arbitrarily. But since there is the possibility that the two plants (non-linear and linearised) behave differently, far from the linearisation point, the equilibrium triplet should be chosen near the operating point.

Now, an important feature is to choose an operating point in which the perturbations (committed during the linearisation process) remain as low as possible. These perturbations around the operating point can be calculated as:

$$\begin{cases} \dot{\tilde{\mathbf{x}}} = \dot{\mathbf{x}} - \dot{\mathbf{x}}^* = f(x, u, w) - f(x^*, u^*, w^*) & \tilde{\mathbf{x}}(t_0) = \mathbf{x}_0 - \mathbf{x}_0^* \\ \tilde{\mathbf{y}} = \mathbf{y} - \mathbf{y}^* = h(x, u, w) - h(x^*, u^*, w^*) \\ \tilde{\mathbf{e}} = \mathbf{e} - \mathbf{e}^* = h_e(x, u, w) - h_e(x^*, u^*, w^*) \end{cases} \quad (2.25)$$

Knowing also that:

$$\begin{aligned} \tilde{\mathbf{u}} &= \mathbf{u} - \mathbf{u}^* \\ \tilde{\mathbf{w}} &= \mathbf{w} - \mathbf{w}^* \end{aligned} \quad (2.26)$$

$\dot{\tilde{\mathbf{x}}}$ can be written, using Taylor's series, as:

$$\begin{aligned} \dot{\tilde{\mathbf{x}}} &= \dot{\mathbf{x}} - \dot{\mathbf{x}}^* \\ &= f(\tilde{x} + x^*, \tilde{u} + u^*, \tilde{w} + w^*) - f(x^*, u^*, w^*) \\ &= \frac{\partial f}{\partial \mathbf{x}} \Bigg|_{\substack{x=x^* \\ u=u^* \\ w=w^*}} \cdot \tilde{\mathbf{x}} + \frac{\partial f}{\partial \mathbf{u}} \Bigg|_{\substack{x=x^* \\ u=u^* \\ w=w^*}} \cdot \tilde{\mathbf{u}} + \frac{\partial f}{\partial \mathbf{w}} \Bigg|_{\substack{x=x^* \\ u=u^* \\ w=w^*}} \cdot \tilde{\mathbf{w}} + o(||\tilde{\mathbf{x}}^2||, ||\tilde{\mathbf{u}}^2||, ||\tilde{\mathbf{w}}^2||) \end{aligned} \quad (2.27)$$

We can also rewrite $f(x, u, w)$ using always Taylor's series:

$$\begin{aligned} f(x, u, w) &= f(x^*, u^*, w^*) + \frac{\partial f}{\partial \mathbf{x}} \cdot (\mathbf{x} - \mathbf{x}^*) + \frac{\partial f}{\partial \mathbf{u}} \cdot (\mathbf{u} - \mathbf{u}^*) + \frac{\partial f}{\partial \mathbf{w}} \cdot (\mathbf{w} - \mathbf{w}^*) + \\ &\quad + \frac{\partial^2 f}{\partial \mathbf{x}^2} \cdot (\mathbf{x} - \mathbf{x}^*)^2 + \frac{\partial^2 f}{\partial \mathbf{u}^2} \cdot (\mathbf{u} - \mathbf{u}^*)^2 + \frac{\partial^2 f}{\partial \mathbf{w}^2} \cdot (\mathbf{w} - \mathbf{w}^*)^2 + \dots \end{aligned} \quad (2.28)$$

then, taking advantage of neglecting the terms upper the 2nd order, substituting (2.28) in the first equation of (2.25) to obtain (2.27). The same procedure can be applied on the measured outputs $\tilde{\mathbf{y}}$ and regulated outputs $\tilde{\mathbf{e}}$:

$$\begin{aligned} \tilde{\mathbf{y}} &= \frac{\partial h}{\partial \mathbf{x}} \cdot \tilde{\mathbf{x}} + \frac{\partial h}{\partial \mathbf{u}} \cdot \tilde{\mathbf{u}} + \frac{\partial h}{\partial \mathbf{w}} \cdot \tilde{\mathbf{w}} + o(||\tilde{\mathbf{x}}^2||, ||\tilde{\mathbf{u}}^2||, ||\tilde{\mathbf{w}}^2||) \\ \tilde{\mathbf{e}} &= \frac{\partial h_e}{\partial \mathbf{x}} \cdot \tilde{\mathbf{x}} + \frac{\partial h_e}{\partial \mathbf{u}} \cdot \tilde{\mathbf{u}} + \frac{\partial h_e}{\partial \mathbf{w}} \cdot \tilde{\mathbf{w}} + o(||\tilde{\mathbf{x}}^2||, ||\tilde{\mathbf{u}}^2||, ||\tilde{\mathbf{w}}^2||) \end{aligned} \quad (2.29)$$

Finally, it is possible to simplify equations (2.27) and (2.29) thanks to the definition of the following matrices:

$$\begin{aligned}
\mathbf{A} &= \frac{\partial f}{\partial \mathbf{x}} \Bigg|_{\substack{x=x^* \\ u=u^* \\ w=w^*}} & \mathbf{B}_1 &= \frac{\partial f}{\partial \mathbf{u}} \Bigg|_{\substack{x=x^* \\ u=u^* \\ w=w^*}} & \mathbf{B}_2 &= \frac{\partial f}{\partial \mathbf{w}} \Bigg|_{\substack{x=x^* \\ u=u^* \\ w=w^*}} \\
\mathbf{C} &= \frac{\partial h}{\partial \mathbf{x}} \Bigg|_{\substack{x=x^* \\ u=u^* \\ w=w^*}} & \mathbf{D}_1 &= \frac{\partial h}{\partial \mathbf{u}} \Bigg|_{\substack{x=x^* \\ u=u^* \\ w=w^*}} & \mathbf{D}_2 &= \frac{\partial h}{\partial \mathbf{w}} \Bigg|_{\substack{x=x^* \\ u=u^* \\ w=w^*}} \\
\mathbf{C}_e &= \frac{\partial h_e}{\partial \mathbf{x}} \Bigg|_{\substack{x=x^* \\ u=u^* \\ w=w^*}} & \mathbf{D}_{1e} &= \frac{\partial h_e}{\partial \mathbf{u}} \Bigg|_{\substack{x=x^* \\ u=u^* \\ w=w^*}} & \mathbf{D}_{2e} &= \frac{\partial h_e}{\partial \mathbf{w}} \Bigg|_{\substack{x=x^* \\ u=u^* \\ w=w^*}}
\end{aligned} \tag{2.30}$$

And then, the final linear model becomes:

$$\begin{cases} \dot{\tilde{\mathbf{x}}} = A \cdot \tilde{\mathbf{x}} + B_1 \cdot \tilde{\mathbf{u}} + B_2 \cdot \tilde{\mathbf{w}} \\ \tilde{\mathbf{y}} = C \cdot \tilde{\mathbf{x}} + D_1 \cdot \tilde{\mathbf{u}} + D_2 \cdot \tilde{\mathbf{w}} \\ \tilde{\mathbf{e}} = C_e \cdot \tilde{\mathbf{x}} + D_{1e} \cdot \tilde{\mathbf{u}} + D_{2e} \cdot \tilde{\mathbf{w}} \end{cases} \tag{2.31}$$

The matrices are calculated on symbolic variables through the Matlab function *Jacobian*:

$$\begin{aligned}
\mathbf{A} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ -k_s \cdot \frac{m_s+m_u}{m_s \cdot m_u} & -\beta_s \cdot \frac{m_s+m_u}{m_s \cdot m_u} & \frac{k_t}{m_u} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k_s}{m_u} & \frac{\beta_s}{m_u} & -\frac{k_t}{m_u} & 0 \end{bmatrix} \\
\mathbf{B}_1 &= \begin{bmatrix} 0 \\ -\frac{\rho \cdot v^2 \cdot S \cdot \pi}{m_s} \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{B}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{m_s} & 0 & -\frac{\rho \cdot v^2 \cdot S \cdot \pi \cdot \alpha}{m_s} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix} \\
\mathbf{C} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \quad \mathbf{D}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mathbf{D}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \\
\mathbf{C}_e &= \begin{bmatrix} \frac{2}{t} & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{D}_{1e} = \begin{bmatrix} 0 \end{bmatrix} \quad \mathbf{D}_{2e} = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{2}{t} & 0 - \frac{2}{t} \end{bmatrix}
\end{aligned} \tag{2.32}$$

In order to find the equilibrium triplet, the step is to impose:

$$\dot{\mathbf{x}} = f(x^*, u^*, w^*) = \mathbf{0} \tag{2.33}$$

and the result is:

$$\begin{cases} x_1^* = \dot{z}_s - \dot{z}_u = l_{0s} - \frac{m_s \cdot g}{k_s} + \frac{L}{k_s} = 0.4313 \text{ m} \\ x_2^* = \ddot{z}_s - \ddot{z}_u = 0 \\ x_3^* = \dot{z}_u - \dot{z}_r = l_{0t} - \frac{(m_s + m_u) \cdot g}{k_t} + \frac{L}{k_t} = 0.0924 \text{ m} \\ x_4^* = \ddot{z}_u - \ddot{z}_r = 0 \end{cases} \quad (2.34)$$

We also impose $e_1^* = 0$ and constant speed ($d_4^* = v_0 = 100 \text{ km/h}$).

Knowing now the equilibrium triplet and substituting parameters found in Section 2.1.4 in matrices (2.32) we can calculate them numerically:

$$\mathbf{A} = 10^4 \cdot \begin{bmatrix} 0 & 0.0001 & 0 & 0 \\ -0.1571 & -0 - 0094 & 1.4286 & 0 \\ 0 & 0 & 0 & 0.0001 \\ 0.1429 & 0.0086 & -1.4286 & 0 \end{bmatrix} \quad (2.35)$$

$$\mathbf{B}_1 = \begin{bmatrix} 0 \\ -0.3436 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{B}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0029 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \quad (2.36)$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \quad \mathbf{D}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mathbf{D}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (2.37)$$

$$\mathbf{C}_e = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{D}_{1e} = \begin{bmatrix} 0 \end{bmatrix} \quad \mathbf{D}_{2e} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix} \quad (2.38)$$

2.2.2 Linear system analysis

Now we want to study the behaviour of the system in absence of any type of control system, this conditions it is named *Open loop*. It is important to understand if the system is self-stable or drift away from equilibrium (if the initial conditions do not match the equilibrium triplet). To do this, it is necessary to study the eigenvalues and eigenvectors of the \mathbf{A} matrix, which are related to the dynamics of the system:

$$\dot{\tilde{\mathbf{x}}} = \mathbf{A}\tilde{\mathbf{x}} \quad (2.39)$$

To make the notation lighter we will neglect the *tilde* symbol referring to the set of linearised coordinates.

First of all we need to change the coordinates in order to transform the original system into an easier one. This goal can be achieved transforming the system in the *Jordan Canonical form*. We will see later that using this transformation it will be possible to obtain new elements that describe the dynamic of our

original system. The first step is to calculate all the **eigenvalues** of the \mathbf{A} matrix looking for the solution (λ) of Equation (2.40). Then, one for each λ_i , it is possible to find all the correspondent **eigenvectors** v_i :

$$\det(\mathbf{A} - \lambda \mathbf{I}) = 0 \quad (2.40)$$

$$(\mathbf{A} - \lambda \mathbf{I})v = 0 \quad (2.41)$$

It is important to mention that our equilibrium conditions have been chosen according to the following hypothesis:

- Constant speed ($v = 100 \text{ km/h}$)
- Flat road ($z_r = 0$)
- System moving on a straight trajectory ($F_{roll} = 0$)
- No aerodynamic force ($\alpha = 0$)

The eigenvalues found are reported below:

$$\begin{cases} \lambda_1 = -43.47 + i \cdot 114.86 \\ \lambda_2 = -43.47 - i \cdot 114.86 \\ \lambda_3 = -3.67 + i \cdot 11.04 \\ \lambda_4 = -3.67 - i \cdot 11.04 \end{cases} \quad (2.42)$$

As can be seen two distinct couples of complex conjugates eigenvalues are obtained. Due to the fact that the number of distinct eigenvalues is equal to the rank of \mathbf{A} (4), each λ must have an algebraic multiplicity equal to 1, like the geometric multiplicity. This implies that the Jordan Canonical form associated is diagonal, but it is reduced to pure real values isolating the real and the imaginary parts and then collecting them in columns:

$$\mathbf{J} = \begin{bmatrix} -43.47 & 114.86 & 0 & 0 \\ -114.86 & -43.47 & 0 & 0 \\ 0 & 0 & -3.67 & 11.04 \\ 0 & 0 & -11.04 & -3.67 \end{bmatrix} \quad (2.43)$$

We can now collect all the eigenvectors in a single matrix called \mathbf{V} (where $\mathbf{V}^{-1}\mathbf{A}\mathbf{V} = \mathbf{J}$) making the same procedure as (2.43). The final matrix is here reported:

$$\mathbf{V} = \begin{bmatrix} 0.0025 & 0.0080 & -0.7152 & -0.4288 \\ -1.0290 & -0.0627 & 7.3576 & -6.3210 \\ -0.0029 & -0.0076 & -0.0271 & -0.0816 \\ 1 & 0 & 1 & 0 \end{bmatrix} \quad (2.44)$$

We can perform now the change of coordinates in order to have a system composed of independent sub-systems. Considering the system in the original coordinates $\dot{\mathbf{x}} = \mathbf{Ax}$ and assuming the transformation matrix $\mathbf{T} = \mathbf{V}^{-1}$, the system can be written as $\dot{\mathbf{z}} = \mathbf{T}\dot{\mathbf{x}}$, and so:

$$\dot{\mathbf{z}} = \mathbf{T}\dot{\mathbf{x}} = \mathbf{T}\mathbf{A}\mathbf{T}^{-1}\mathbf{z} = \mathbf{V}^{-1}\mathbf{AVz} = \mathbf{Jx} \quad (2.45)$$

$$\mathbf{x} = \mathbf{Vz} \quad (2.46)$$

and the set of coupled differential equations written in the new coordinates is reported below:

$$\begin{cases} \dot{\mathbf{z}} = \bar{\mathbf{A}}\mathbf{z} + \bar{\mathbf{B}}\mathbf{u} \\ \mathbf{z}(t_0) = \mathbf{T}\mathbf{x}_0 \end{cases} \quad (2.47)$$

where $\bar{\mathbf{A}} = \mathbf{T}\mathbf{A}\mathbf{T}^{-1}$ and $\bar{\mathbf{B}} = \mathbf{T}\mathbf{B}$.

Now, we can understand how each mode affects the system analysing the product between the normalized eigenvector matrix $\mathbf{V_n}$ and the new coordinates (looking at (2.46)).

$$\mathbf{V_n} = \begin{bmatrix} 0.22 & 0.69 & 61.95 & 37.14 \\ 6.97 & 0.42 & 49.81 & 42.79 \\ 2.42 & 6.39 & 22.75 & 68.44 \\ 50 & 0 & 50 & 0 \end{bmatrix} \quad (2.48)$$

And then the weights of the modes $z_j(t)$ on the state components $x_i(t)$ can be represented through radar charts, reported in Figure 2.9 for each variable:

- x_1 : is the suspension deflection. It is mainly affected by two components: for the 61.9% by z_3 and 37.1% by z_4 . Both of them have negative real parts so x_1 will converge very slow with, of course, some oscillations
- x_2 : is the suspension deflection velocity. It is mainly affected by the same components of x_1 : for the 49.8% by z_3 and 42.8% by z_4 . Compared to the first state here we have an higher incident of z_1 which will change the oscillation behaviour but not the convergence
- x_3 : is the tire deflection. It is mainly affected by z_4 (68.4%). Here we have a fast convergence and less oscillations
- x_4 : is the tire deflection velocity. It is affected in equal measure (50%) by z_1 and z_3 . So the oscillation is very low and the convergence quite fast

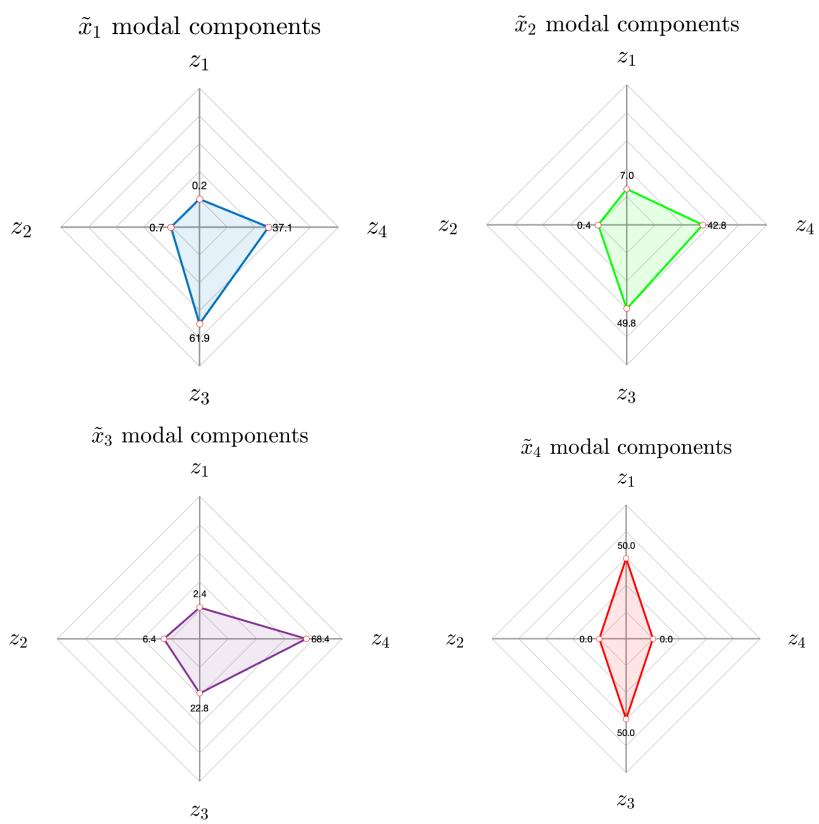


Figure 2.9: Modes radar charts

It is important to highlight that, due to the fact that all the eigenvalues reported in (2.42) are complex numbers with negative real parts, our system is BIBS stable. As we can see in Figure 2.10 all the states, after an initial oscillation, converge to stable values.

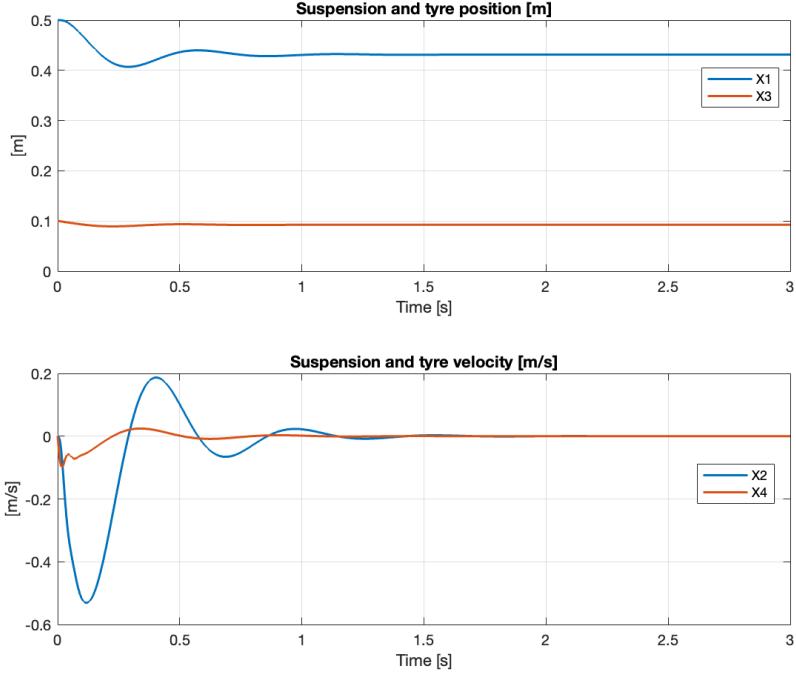


Figure 2.10: System in *Open Loop* simulation

2.2.3 Reachability

The reachability study needs to be performed in order to understand which parts of \mathbf{A} can be modified by the control and which have some limitations to the capability of the control to steer some state back to equilibrium.

We want to find the set of states, called *Reachability set* \mathbf{R} , representing the states we can reach starting from the origin. The dimensions of this matrix are $n \times n \cdot p$, where $n = 4$ is the dimension of the state vector and $p = 1$ the dimension of the control vector. So:

$$\mathbf{R} \in \mathbb{R}^{4 \times 4} \quad (2.49)$$

The \mathbf{R} matrix can now be built:

$$\mathbf{R} = \begin{bmatrix} \mathbf{B}_1 & \mathbf{AB}_1 & \mathbf{A}^2\mathbf{B}_1 & \mathbf{A}^3\mathbf{B}_1 \end{bmatrix} \quad (2.50)$$

And computing in *Matlab*:

$$\mathbf{R} = \begin{bmatrix} 0 & -0.3258 & 30.72 & -2384.30 \\ -0.3258 & 30.72 & -2384.3 & 222400 \\ 0 & 0 & -27.93 & 2167.50 \\ 0 & -27.93 & 2167.50 & 238448 \end{bmatrix} \quad (2.51)$$

The system is said *fully reachable* if \mathbf{R} is full rank. In our study, the system is considered fully reachable if $\text{rank}(\mathbf{R}) = n = 4$. In fact, using *Matlab function* 'ctrb' to determine \mathbf{R} , we can state that the system is fully reachable. This means that we will be able to control all the eigenvalues of \mathbf{A} .

According to this result, we can claim that:

$$\exists \mathbf{K}_R \text{ matrix such that } \mathbf{A} + \mathbf{B}\mathbf{K}_R \text{ is Hurwitz} \quad (2.52)$$

The meaning is that there is a state feedback control law which is able to make the system BIBS stable also in *Closed loop*. This result allows us to write the equation of the state feedback stabilizer in function of K_R :

$$\tilde{\mathbf{u}} = \mathbf{K}_R \tilde{\mathbf{x}} \quad (2.53)$$

Subsequently

$$\dot{\tilde{\mathbf{x}}} = (\mathbf{A} + \mathbf{B}\mathbf{K}_R)\tilde{\mathbf{x}} \quad (2.54)$$

2.2.4 Integral Action

From the previous sections we have understood that the system is BIBS stable by itself and, with the reachability analysis, we have also achieved that the BIBS stability is keep in every moment as the state of the system is bounded in every condition. But another control can be implemented: the *Integral Action*.

The first step is to define an error vector:

$$\tilde{\mathbf{h}}_e = \mathbf{C}_e \tilde{\mathbf{x}} + \mathbf{D}_{1e} \tilde{\mathbf{u}} + \mathbf{D}_{2e} \tilde{\mathbf{w}} \quad (2.55)$$

The goal is to reach the condition in which $\tilde{\mathbf{h}}_e = 0$ despite the unknown exogenous variables $\tilde{\mathbf{w}}$. Knowing that $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{w}}$ are bounded, we can claim, thanks to the BIBS stability analysis, that $\tilde{\mathbf{h}}_e$ is also bounded.

We can define:

$$\dot{\eta} = \mathbf{h}_e \quad (2.56)$$

and then, the new state vector:

$$\dot{\tilde{\mathbf{x}}}_e = \begin{bmatrix} \dot{\tilde{\mathbf{x}}} \\ \dot{\eta} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & 0 \\ \mathbf{C}_e & 0 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}} \\ \eta \end{bmatrix} + \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{D}_1 \end{bmatrix} \tilde{\mathbf{u}} + \begin{bmatrix} \mathbf{B}_2 \\ \mathbf{D}_2 \end{bmatrix} \tilde{\mathbf{w}} \quad (2.57)$$

$$\dot{\tilde{\mathbf{x}}}_e = \overline{\mathbf{A}} \cdot \mathbf{x}_e + \overline{\mathbf{B}_1} \cdot \tilde{\mathbf{u}} + \overline{\mathbf{B}_2} \cdot \tilde{\mathbf{w}} \quad (2.58)$$

Since the couple $(\bar{\mathbf{A}}, \bar{\mathbf{B}}_1)$ is stabilizable, consequently there must exist at least one $\bar{\mathbf{K}}$ such that $(\bar{\mathbf{A}}, \bar{\mathbf{B}}_1)$ is Hurwitz. Knowing that the system is BIBS and $\dot{\mathbf{x}}_e$ is bounded.

To conclude we can state:

$$\begin{aligned}\tilde{\mathbf{u}} &= \bar{\mathbf{K}}\mathbf{x}_e \\ &= \mathbf{K}_s\tilde{\mathbf{x}} + \mathbf{K}_I\eta\end{aligned}\tag{2.59}$$

where $\mathbf{K}_s^{p \times n}$ and $\mathbf{K}_I^{p \times lm}$ are the sub-matrices of $\bar{\mathbf{K}}$, so in our case:

$$\mathbf{K}_s^{1 \times 4} \text{ and } \mathbf{K}_I^{1 \times 1}\tag{2.60}$$

Finally, substituting (2.59) into (2.58):

$$\dot{\mathbf{x}}_e = (\bar{\mathbf{A}} + \bar{\mathbf{B}}_1\bar{\mathbf{K}}) \cdot \mathbf{x}_e + \bar{\mathbf{B}}_2 \cdot \tilde{\mathbf{w}}\tag{2.61}$$

2.2.5 Observability

Another important analysis that has to be performed is referred to the state estimation. In fact, the main assumption that we have made until now is that the state $\tilde{\mathbf{x}}$ is always known. But, in real situations, there is the chance to do not have the state available at any time. So we must introduce a tool that allows us to collect the measurements and the control to make an estimation of the state.

To do so we need to know if all the states of our system are measurable and introduce the concept of *observability*. Remembering that our sensors are measuring these two distinguish parameters:

- The suspension deflection of the spring ($z_s - z_u - p_{ref}$)
- The overall height of the system ($z_s - z_r$)

we can see that the number of measurements is less than the number of states we have to estimate. To solve this problem, we have to build an *observability matrix*:

$$\mathbf{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \mathbf{CA}^3 \end{bmatrix} \in \mathbb{R}^{n \cdot q \times n}\tag{2.62}$$

And computing in *Matlab*:

$$\mathbf{O} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ -1571.4 & -94.286 & 14286 & 0 \\ -142.86 & -8.5714 & 0 & 0 \\ 148163 & 7318.4 & -1346939 & 14286 \\ 13469 & 665.31 & -122449 & 0 \end{bmatrix} \in \mathbb{R}^{8 \times 4} \quad (2.63)$$

Once defined the *observability matrix*, we know from the theory that the concept of observability is strongly related to the $\ker(\mathbf{O})$, in fact if $\dim(\ker(\mathbf{O})) = 0$ the couple (\mathbf{A}, \mathbf{C}) is fully observable. This can be also seen in other algebraic terms: if $\dim(\ker(\mathbf{O})) = 0$, the \mathbf{O} matrix is full rank and the couple (\mathbf{A}, \mathbf{C}) is fully observable.

Other way around if the $\dim(\ker(\mathbf{O})) \neq 0$ the system is not fully observable and this means that some of the states cannot be distinguished by the control from the origin (i.e. the states that belong to the $\ker(\mathbf{C})$ which make $\tilde{\mathbf{y}} = 0$). Using the *Matlab function* 'obsv' to determine \mathbf{O} , we can state that the system is fully observable. This means that $\text{rank}(\mathbf{O}) = n = 4$. Also, if (\mathbf{A}, \mathbf{C}) is fully observable:

$$\exists \mathbf{K}_\mathbf{O} \text{ matrix such that } \mathbf{A} - \mathbf{K}_\mathbf{O} \mathbf{C} \text{ is Hurwitz} \quad (2.64)$$

2.3 Proposed Solution

The plant is controlled through two different control contributions: the *State Feedback* and the *Integral Action*. In this case the Feed Forward is not needed as it will be explained in Section 2.3.4. Moreover, as mentioned before, we need also an *Observer* to track the state of our system and to generate a state estimation that can be sent as an input to the controller. All the tuning and design behind this tools will be covered in the next sections.

2.3.1 Optimal Control Model

In order to design our control system we need to construct in the best way possible K_S, K_I, K_O to solve the *Optimal Control Problem*. The challenge on finding the best solution is based on finding the balance between reaching the goals as fast as possible and, on the other hand, limit the cost of the controls. From the previous sections we have obtained the following equations that can

be used to solve the optimal control problem:

$$\begin{aligned}\dot{\tilde{\mathbf{x}}} &= \mathbf{A} \cdot \tilde{\mathbf{x}} + \mathbf{B}_1 \cdot \tilde{\mathbf{u}} + \mathbf{B}_2 \cdot \tilde{\mathbf{w}} \\ \tilde{\mathbf{h}}_{\mathbf{e}} &= \mathbf{C}_{\mathbf{e}} \tilde{\mathbf{x}} + \mathbf{D}_{1\mathbf{e}} \tilde{\mathbf{u}} + \mathbf{D}_{2\mathbf{e}} \tilde{\mathbf{w}} \\ \dot{\tilde{\mathbf{x}}}_{\mathbf{e}} &= \overline{\mathbf{A}} \cdot \mathbf{x}_{\mathbf{e}} + \overline{\mathbf{B}_1} \cdot \tilde{\mathbf{u}} + \overline{\mathbf{B}_2} \cdot \tilde{\mathbf{w}}\end{aligned}\tag{2.65}$$

2.3.2 State Feedback and Integral Action

In order to design the State Feedback and Integral Action we need to define the shape of K_S and K_I , and to do so $\overline{\mathbf{B}_2}$ in (2.65) can be neglected. The equation becomes:

$$\dot{\tilde{\mathbf{x}}}_{\mathbf{e}} = \overline{\mathbf{A}} \cdot \mathbf{x}_{\mathbf{e}} + \overline{\mathbf{B}_1} \cdot \tilde{\mathbf{u}} + \alpha \cdot \mathbf{I} \cdot \tilde{\mathbf{x}}_{\mathbf{e}}\tag{2.66}$$

Here we can see that an additional parameter ($\alpha \geq 0$) is added that play the role of changing the eigenvalues of the real system in order to make them less negative. Because in this way is possible to find a *virtual system* that is less stable than the real one. The point of that is once we are able to control the less stable virtual system, the robustness of the real one will be higher.

The next step is to define an arbitrary vector:

$$\boldsymbol{\epsilon} = \mathbf{C}_{\epsilon} \cdot \mathbf{x}_{\mathbf{e}} + \mathbf{D}_{\epsilon} \cdot \tilde{\mathbf{u}}\tag{2.67}$$

which is a linear combination of the states and the controls. It does not required to be measurable and it is useful for the design process, as we will see in a short. The matrices \mathbf{C}_{ϵ} and \mathbf{D}_{ϵ} express the relation between the errors vectors and the extended state and control vector, but they are not necessary connected to \mathbf{C} and \mathbf{D} . In our case \mathbf{C}_{ϵ} is built as an identity matrix and \mathbf{D}_{ϵ} as a null matrix. We can also define a *cost function* J :

$$J = \int_0^{\infty} \boldsymbol{\epsilon}^T \cdot \mathbf{Q} \cdot \boldsymbol{\epsilon} + \tilde{\mathbf{u}}^T \cdot \mathbf{R} \cdot \tilde{\mathbf{u}} dt\tag{2.68}$$

where:

- J is a scalar > 0
- \mathbf{Q} is semi-positive defined
- \mathbf{R} is positive defined

Once we found the right \mathbf{u} that solves the optimal control problem, the goal is to minimize J . Moreover the fact that \mathbf{R} is positive defined means that the scope is always to penalise our control because designing a control law with an infinite magnitude cannot be sustained.

In the end, a consideration on \mathbf{Q} that can be zero if we are fine with the error. This are all the consideration to be made to minimize the energy cost function. The next step is to build in the proper way the \mathbf{Q} and \mathbf{R} matrices for our case of study. Theoretically speaking the matrix \mathbf{Q} must have the same dimension

as the \mathbf{x}_ϵ vector ($n_\epsilon \times n_\epsilon$), so considering what we stated previously about \mathbf{C}_ϵ matrix as an identity matrix and \mathbf{D}_ϵ as a null matrix, we can surely say that $\mathbf{Q} \in 5 \times 5$. Going on from these assumptions, \mathbf{Q} and \mathbf{R} are defined by inverting respectively:

$$\mathbf{Q}^{-1} = n_\epsilon \begin{bmatrix} \epsilon_{1_{max}}^2 & 0 & 0 & 0 & 0 \\ 0 & \epsilon_{2_{max}}^2 & 0 & 0 & 0 \\ 0 & 0 & \epsilon_{3_{max}}^2 & 0 & 0 \\ 0 & 0 & 0 & \epsilon_{4_{max}}^2 & 0 \\ 0 & 0 & 0 & 0 & \epsilon_{5_{max}}^2 \end{bmatrix} \quad (2.69)$$

$$\mathbf{R}^{-1} = \begin{bmatrix} u_{max}^2 \end{bmatrix} \quad (2.70)$$

Where $\epsilon_{i_{max}}$ expresses the inverse of the cost of the i^{th} component of $\boldsymbol{\epsilon}$, while u_{max} expresses the inverse of the cost of the control action.

Starting from this, we can make some considerations on \mathbf{Q} and \mathbf{R} : low values of $\epsilon_{i_{max}}$ lead to higher \mathbf{Q} and so we care more about how far we are reaching the zero, while high values of u_{max} lead to lower \mathbf{R} and so we want to use more our control action. Is important to remember that the goal of all this process is not only to find a control law that minimize J , but also to build a close loop that is BIBS stable: in fact the matrices \mathbf{C}_ϵ and \mathbf{D}_ϵ have been chosen like that because in this way we are sure that the couple $(\mathbf{A}, \mathbf{C}_\epsilon)$ is fully observable and also the \mathbf{Q} and \mathbf{R} will be way more easy due to the fact that in this way $\boldsymbol{\epsilon}$ and the control are not directly correlated as we will see in a short.

These considerations have driven the tuning procedure of our system.

In our case of study these are the meaning of every $\epsilon_{i_{max}}$ and u_{max} :

- $\epsilon_{1_{max}}$ refers to the maximum allowable error on suspension deflection
- $\epsilon_{2_{max}}$ refers to the maximum allowable error on suspension deflection speed
- $\epsilon_{3_{max}}$ refers to the maximum allowable error on tire deflection
- $\epsilon_{4_{max}}$ refers to the maximum allowable error on tire deflection speed
- $\epsilon_{5_{max}}$ refers to the integral action on reaching reference position
- u_{max} refers to the control on the airfoil angle

Once is all settled, the control matrices can be obtained from the solution of the *Algebraic Riccati Equation (ARE)*, which is given as:

$$\begin{aligned} & \mathbf{S}\mathbf{B}_e\bar{\mathbf{R}}^{-1}\mathbf{B}_e^T\mathbf{S} - \mathbf{S}(\mathbf{A}_e + \alpha\mathbf{I} - \mathbf{B}_e\bar{\mathbf{R}}^{-1}\mathbf{D}_\epsilon^T\mathbf{Q}\mathbf{C}_\epsilon) + \\ & - (\mathbf{A}_e + \alpha\mathbf{I} - \mathbf{B}_e\bar{\mathbf{R}}^{-1}\mathbf{D}_\epsilon^T\mathbf{Q}\mathbf{C}_\epsilon)^T\mathbf{S} - \mathbf{C}_\epsilon^T\mathbf{Q}(\mathbf{I} - \mathbf{D}_\epsilon\bar{\mathbf{R}}^{-1}\mathbf{D}_\epsilon^T\mathbf{Q})\mathbf{C}_\epsilon = 0 \end{aligned} \quad (2.71)$$

where $\bar{\mathbf{R}} = \mathbf{D}_\epsilon^T \mathbf{Q} \mathbf{D}_\epsilon + \mathbf{R}$ and \mathbf{S} is the solution of (2.71) from which the control law that minimizes the cost function J can be obtained and subsequently the right values for the \mathbf{K}_S and \mathbf{K}_I matrices. From the expression of $\bar{\mathbf{R}}$ we can see what we said before because if \mathbf{D}_ϵ is null there is no direct correlation between \mathbf{Q} and \mathbf{R} . On the other hand, if we have imposed that \mathbf{D}_ϵ was different from a null matrix, the \mathbf{Q} matrix would have influenced directly the control costs resulting in a big constraint in the overall procedure because for example setting \mathbf{Q} too big would lead to limit \mathbf{R} in order to avoid too high values for $\bar{\mathbf{R}}$.

This procedure to find the solution \mathbf{S} of (2.71) can be performed by using the *Matlab function* 'icare', that has a slightly different syntax in defining all the parameters than the Riccati equation reported previously. The syntax of the function is explained as follow:

$$[\mathbf{X}_m, \mathbf{K}_m, \mathbf{L}_m] = \text{icare}(\mathbf{A}_m, \mathbf{B}_m, \mathbf{Q}_m, \mathbf{R}_m, \mathbf{S}_m, \mathbf{E}_m, \mathbf{G}_m) \quad (2.72)$$

where the m pedix refers to Matlab and:

$$\begin{aligned} \mathbf{X}_m &= \mathbf{S} & \mathbf{K}_m &= -[\mathbf{K}_S \mathbf{K}_I] \\ \mathbf{A}_m &= \mathbf{A}_e + \alpha \mathbf{I} & \mathbf{B}_m &= \mathbf{B}_e & \mathbf{Q}_m &= \mathbf{C}_\epsilon^T \mathbf{Q} \mathbf{C}_\epsilon \\ \mathbf{R}_m &= \mathbf{D}_\epsilon^T \mathbf{Q} \mathbf{D}_\epsilon + \mathbf{R} & \mathbf{S}_m &= \mathbf{C}_\epsilon^T \mathbf{Q} \mathbf{D}_\epsilon & \mathbf{E}_m &= \mathbf{I} & \mathbf{G}_m &= 0 \end{aligned}$$

Tuning Procedure

The goal to minimize the cost function J is directly correlated to both \mathbf{Q} and \mathbf{R} matrices and subsequently to the values of $\epsilon_{i_{max}}$ and u_{max} .

In our case we have chosen to develop a test curve, which its details will be given in a short, in order to properly design the tune of the system and obtain the right values for \mathbf{K}_S and \mathbf{K}_I .

The design of the test curve has been performed like this:

- The speed starts from a constant value of 100 km/h entering the curve, decrease to a minimum value 95 km/h in the center of the turn and than speeds up to 100 km/h during the exit
- The radius is kept quite constant as varies from a maximum value of 105 m to a minimum of 80 m at the center
- The simulation is quite slow, performed in a time span of 10 seconds

The speed and the radius are kept quite constant, they vary a little bit because we want to be sure that the system responds well also in a real case simulation. So here we are looking on a reasonable balance within every value of $\epsilon_{i_{max}}$ and u_{max} , taking into consideration that every $\epsilon_{i_{max}}$ affects differently the system.

- $\epsilon_{1_{max}}$: has to be set low in order to maintain a little variation on suspension deflection because we want to reach the reference as fast as possible to keep the car flat

- $\epsilon_{2_{max}}$: has to be in the middle between because keeping the system without oscillations is not our goal in this case but, for safety and comfort reasons, is better to limit this movements
- $\epsilon_{3_{max}}$: is not important to set accurately in this project as the error on deflection of the tire does not affect much the whole dynamics
- $\epsilon_{4_{max}}$: is not important quite as for the reason of $\epsilon_{3_{max}}$
- $\epsilon_{5_{max}}$: has to be set very low due to the fact that we want to keep our system bounded and reach the reference in a fast way
- u_{max} : we do not want to push too much on the control because we prefer to reach a realistic behaviour of the airfoil

Before starting with the first attempt is worth highlighting that the condition $\alpha = 0$ will be maintained during the whole tuning process of \mathbf{K}_S and \mathbf{K}_I . The main reason for that is related to the fact that putting $\alpha \neq 0$ would have increased drastically the \mathbf{K}_S matrix and so, while computing some next simulations, more noise would have been entered into the feedback loop.

As a first attempt we tried roughly just to put all the $\epsilon_{i_{max}}$ and u_{max} equal to the same value and to have an idea on how the system behave.

$$\mathbf{Q} = \frac{1}{5} \begin{bmatrix} \frac{1}{1^2} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{1^2} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{1^2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{1^2} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{1^2} \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} \frac{1}{1^2} \end{bmatrix} \quad (2.73)$$

We tried this first approach without thinking too much about the physical meaning of the values but just to verify that everything was set correctly. In order to be able to understand the meaning of each parameter respect to the assumption made before and to have a starting point for the modification of the values.

As we expected the results were really poor:

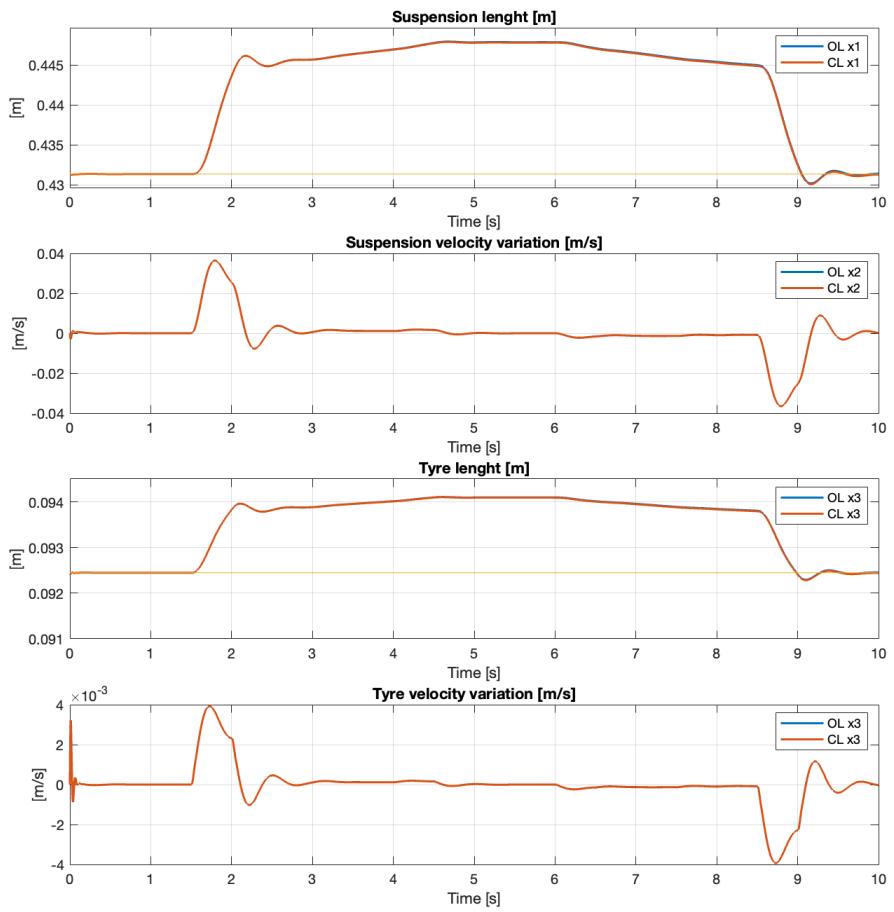


Figure 2.11: First tuning test: OL vs CL

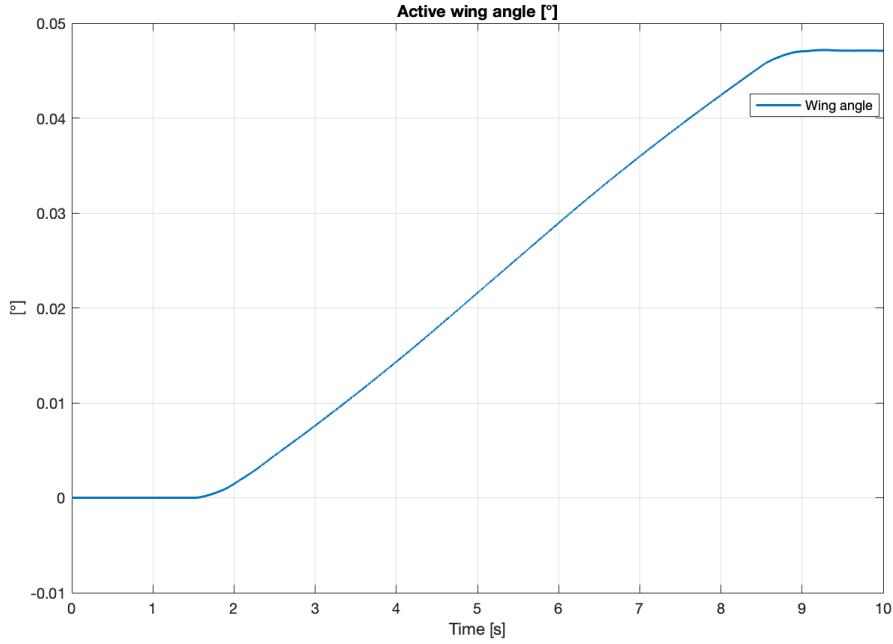


Figure 2.12: First tuning test: control

With these values for $\epsilon_{i_{max}}$ and u_{max} we can see that the control did practically nothing (as there is no difference between the open loop (OL) and the closed loop (CL)). With the next attempt we tried to fix it by varying all $\epsilon_{i_{max}}$ and u_{max} according to what we said previously.

For this second attempt the new matrices \mathbf{Q} and \mathbf{R} become like the following ones:

$$\mathbf{Q} = \frac{1}{5} \begin{bmatrix} \frac{1}{0.01^2} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{0.1^2} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{1^2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{1^2} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{0.001^2} \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} \frac{1}{0.8^2} \end{bmatrix} \quad (2.74)$$

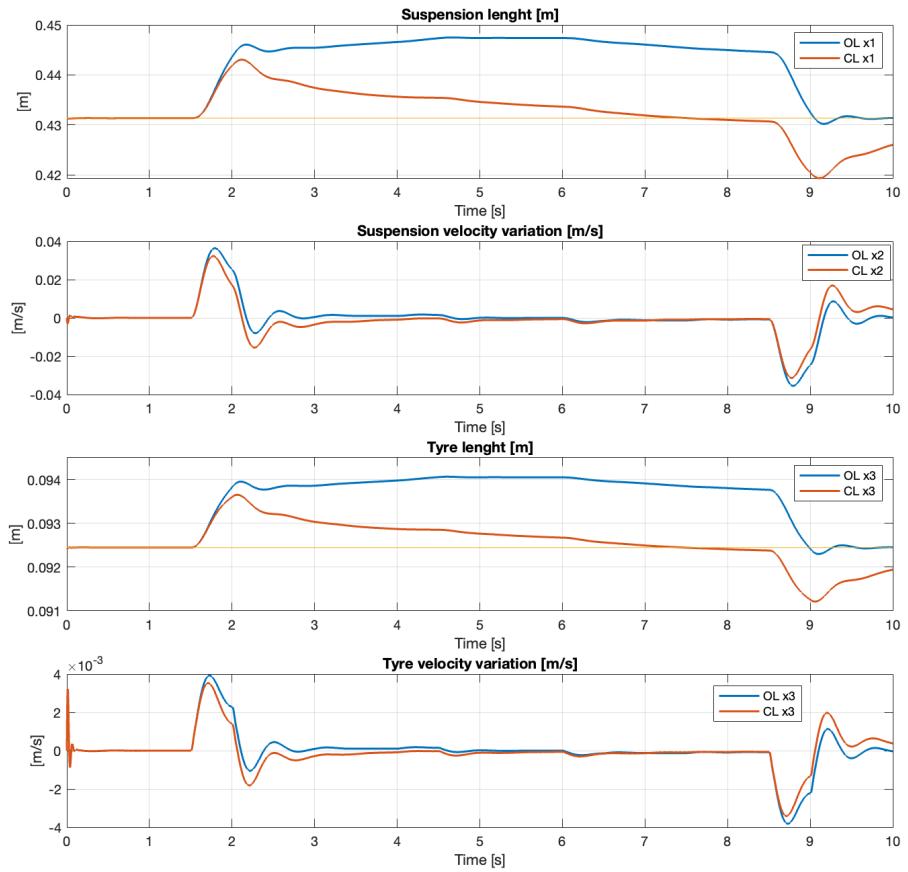


Figure 2.13: Second tuning test: OL vs CL

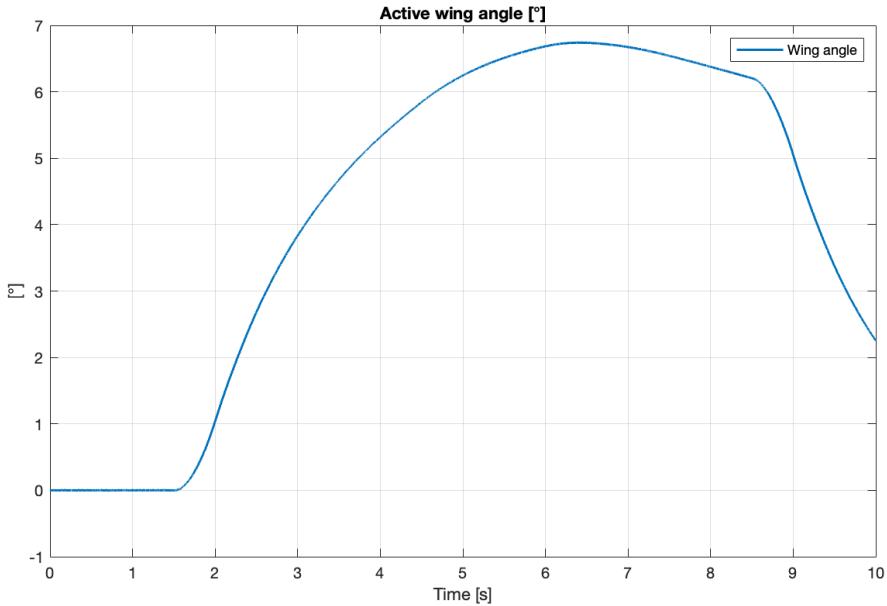


Figure 2.14: Second tuning test: control

In this case the system behaves much better than before. We are closer to the reference but the speed on which we are reaching it is very poor. The control has a smoother curve reaching the goal but it is still too slow.

For the third attempt we tried to push a little bit more on the suspension deflection error and on the suspension deflection speed and in order to speed-up the reaching of the reference $\epsilon_{5,max}$ has been decreased by an order of magnitude and u_{max} was decreased a little bit.

The results we obtain from these modifications are the following:

$$\mathbf{Q} = \frac{1}{5} \begin{bmatrix} \frac{1}{0.001^2} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{0.01^2} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{1^2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{1^2} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{0.0001^2} \end{bmatrix} \quad \mathbf{R} = \left[\frac{1}{0.6^2} \right] \quad (2.75)$$

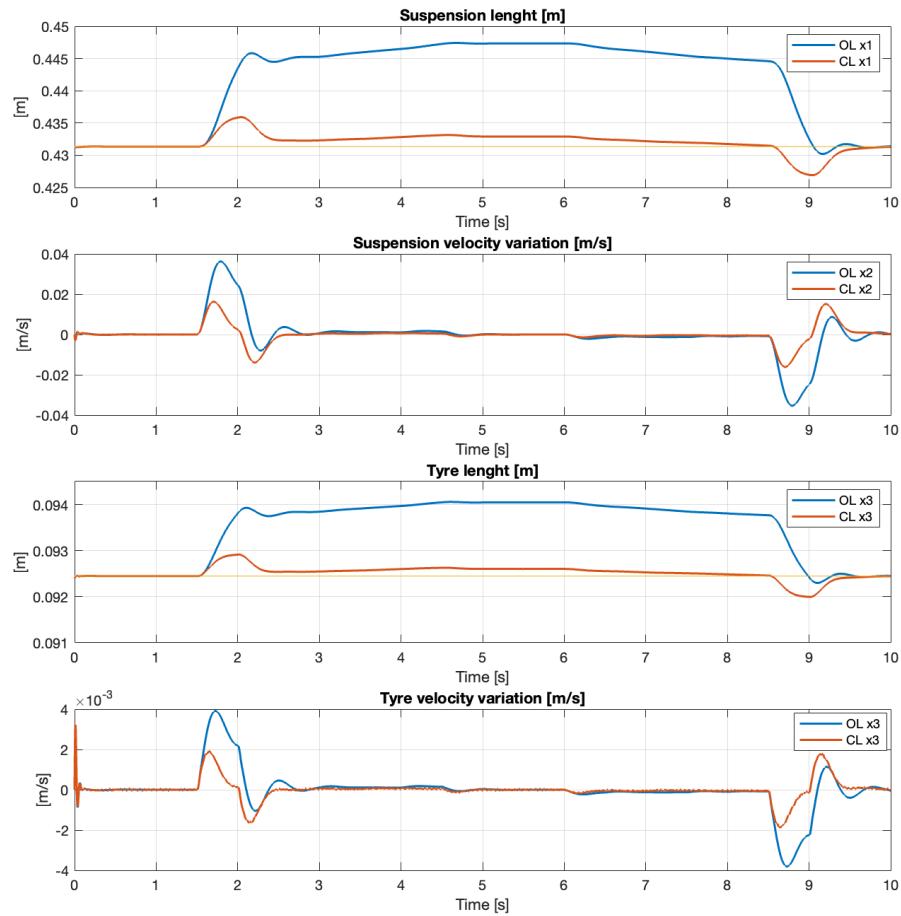


Figure 2.15: Third tuning test: OL vs CL

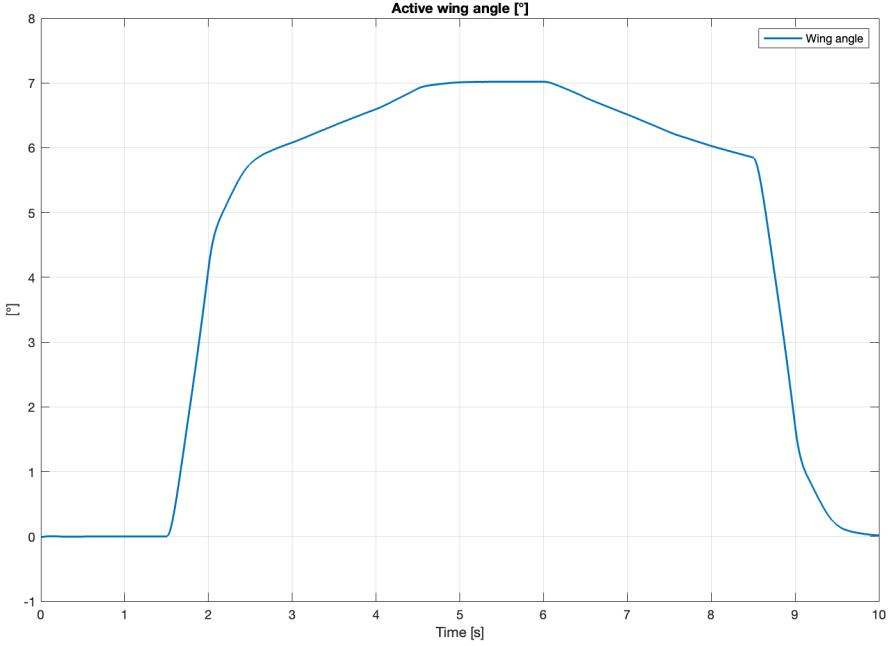


Figure 2.16: Third tuning test: control

Finally we got the result that we wanted as a designer point of view. In the first graph we can see that the suspension length is faster in reaching the reference and has only some errors when the car is entering and exiting the curve. Talking about the dynamics of the suspension the deflection speed is acceptable compared to the OL one, though this behaviour can be improved but for our project is not so crucial. As we said at the beginning we did not care much about the dynamic of the tire, but, looking at the graph, we can see that somehow it follow the dynamic of the suspension with a different order of magnitude.

At the end, the control is faster in reaching the maximum value and has a transient that can resemble a real life situation.

Now, the tuning is completed and all the values are set. So we can compute the matrices \mathbf{K}_s and \mathbf{K}_I :

$$\mathbf{K}_s = \begin{bmatrix} 318.4292 & 31.7314 & 164.3498 & 31.7348 \end{bmatrix} \quad (2.76)$$

$$\mathbf{K}_I = \begin{bmatrix} 2683.3 \end{bmatrix} \quad (2.77)$$

All the tuning process was performed with the *linear system*.

2.3.3 Observer

As mentioned in previous sections, it is not possible to know exactly the state of the system at each time instant, so it is needed to build a new block, the observer, that by collecting all the measurement coming from the sensors try to reconstruct approximately our state. In Section 2.2.5 the analysis on observability was performed and we stated that the couple (\mathbf{A}, \mathbf{C}) is fully observable and so exists a matrix \mathbf{K}_O that make the system $(\mathbf{A} - \mathbf{K}_O \mathbf{C})$ Hurwitz.

The first step is to take into account the dual system, solution of the control problem:

$$\begin{cases} \dot{\chi} = \mathbf{A}^T \chi + \mathbf{C}^T \nu \\ \mu = \mathbf{B}_2^T \chi + \mathbf{D}_2^T \nu \end{cases} \quad (2.78)$$

The strategy here is to find a certain \mathbf{Ks}_d , solution of the dual system, that makes $\mathbf{A}^T + \mathbf{C}^T \mathbf{Ks}_d$ Hurwitz and find the Optimal control set (2.79) to apply on the dual system. From the theory is demonstrated that the solution \mathbf{K}_O of the system will be equal to $-\mathbf{Ks}_d^T$.

$$\begin{cases} \dot{\chi} = (\mathbf{A}^T + \alpha_d \mathbf{I}) \chi + \mathbf{C}^T \nu \\ \mu = \mathbf{B}_2^T \chi + \mathbf{D}_2^T \nu \\ J_d = \int_{t_0}^{\infty} \mu^T \cdot \mathbf{Q}_d \cdot \mu + \nu^T \cdot \mathbf{R}_d \cdot \nu dt \end{cases} \quad (2.79)$$

Same as the state feedback and integral action, here the scope is to minimize the cost function J by applying the Riccati equation (ARE) in 'icare' Matlab function:

$$\begin{aligned} & \mathbf{S} \mathbf{B}_d \bar{\mathbf{R}}_d^{-1} \mathbf{B}_d^T \mathbf{S} - \mathbf{S} (\mathbf{A}_d + \alpha_d \mathbf{I} - \mathbf{B}_d \bar{\mathbf{R}}_d^{-1} \mathbf{D}_d^T \mathbf{Q}_d \mathbf{C}_d) + \\ & - (\mathbf{A}_d + \alpha_d \mathbf{I} - \mathbf{B}_e \bar{\mathbf{R}}_d^{-1} \mathbf{D}_d^T \mathbf{Q}_d \mathbf{C}_d)^T \mathbf{S} - \mathbf{C}_d^T \mathbf{Q}_d (\mathbf{I} - \mathbf{D}_d \bar{\mathbf{R}}_d^{-1} \mathbf{D}_d^T \mathbf{Q}_d) \mathbf{C}_d = 0 \end{aligned} \quad (2.80)$$

The matrices that enter ARE are written as:

$$\mathbf{A}_d = \mathbf{A}^T \quad \mathbf{B}_d = \mathbf{C}^T \quad \mathbf{C}_d = \mathbf{B}_2^T \quad \mathbf{D}_d = \mathbf{D}_2^T \quad (2.81)$$

Similarly, we have to exploit the Matlab syntax which is slightly different from the one from the theory:

$$\begin{aligned} \mathbf{A}_m &= \mathbf{A}_d + \lambda_d \cdot \text{eye}(n) & \mathbf{B}_m &= \mathbf{B}_d \\ \mathbf{Q}_m &= \mathbf{C}_d^T \mathbf{Q}_d \mathbf{C}_d & \mathbf{R}_m &= \bar{\mathbf{R}}_d = \mathbf{R}_d + \mathbf{D}_d^T \mathbf{Q}_d \mathbf{D}_d & \mathbf{S}_m &= \mathbf{C}_d^T \mathbf{Q}_d \mathbf{D}_d \\ \mathbf{E}_m &= \text{eye}(n) & \mathbf{G}_m &= 0 \end{aligned}$$

$$[\mathbf{X}_{mo}, \mathbf{K}_{mo}, \mathbf{L}_{mo}] = \text{icare}(\mathbf{A}_m, \mathbf{B}_m, \mathbf{Q}_m, \mathbf{R}_m, \mathbf{S}_m, \mathbf{E}_m, \mathbf{G}_m) \quad (2.82)$$

From the outputs of 'icare' command we get $\mathbf{K}_m = -\mathbf{Ks}_d$ and thanks to the duality properties we get $\mathbf{K}_O = -\mathbf{Ks}_d^T$ which in Matlab syntax equals to $\mathbf{K}_O =$

$\mathbf{K}^{\mathbf{T}_{mo}}$. The value obtained for the matrix \mathbf{K}_O is used to build the Luenberger observer which also define our observer block.

From (2.79) system we can exploit two important matrices \mathbf{Q}_d and \mathbf{R}_d which represents the two tuning parameters to design the observer. These two matrices share the same properties of \mathbf{Q} and \mathbf{R} of the previous section and are defined as follow:

$$\mathbf{Q}_d = \begin{bmatrix} \omega_{1_{max}}^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \omega_{2_{max}}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \omega_{3_{max}}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \omega_{4_{max}}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{R}_d = \begin{bmatrix} \sigma_{pot}^2 & 0 \\ 0 & \sigma_{las}^2 \end{bmatrix} \quad (2.83)$$

Here $\omega_{i_{max}}$ in \mathbf{Q}_d represents an indication of the i^{th} maximum possible disturbance and σ_{pot} and σ_{las} represent an indication of the magnitude of the error in the measurement performed, but they are fixed by the sensibility of the sensors chosen. Therefore, high magnitude of $\omega_{i_{max}}$ would result in less reliability of the linearised model compared to the measurement performed, while the same can be said for low values of σ_{pot} and σ_{las} .

It is importat to specify that:

- $\omega_{1_{max}}$: is related to the road height
- $\omega_{2_{max}}$: is related to the F_{roll}
- $\omega_{3_{max}}$: is related to the gravity acceleration g
- $\omega_{4_{max}}$: is related to the speed v of the car

Generally speaking, the optimal value \mathbf{K}_O for the observer would result in a trade-off between the reliability of the linearised model and the reliability of the measurement coming from the sensors that will introduce a certain level of noise. So, if we have high noise coming from the sensors is way better to have a low \mathbf{K}_O and subsequently low value of \mathbf{Q}_d to perceive it as low as possible, sacrificing a little bit the reliability and the speed in reaching our goals.

From a designer point of view, we can play only with the values of λ_d and \mathbf{Q}_d because \mathbf{R}_d is fixed by the choice of our sensor and their sensitivity according to Section 2.1.3. So the quality of our observer block will be evaluated in function of how fast we are in reaching the same values of our state, but at the same time generates an acceptable level of noise: that is quite challenging since in our case we have four disturbances coming as an input and to find this balance between noise and speed is not an easy task.

Tuning Procedure

According to what we said previously, to design \mathbf{K}_O we can only work on two tuning parameters: λ_d and \mathbf{Q}_d since \mathbf{R}_d is fixed. Here is important to highlight that we want to evaluate our observer on how fast is reaching the real state, but also generate an acceptable noise.

For our purpose we do not want to insert too much noise in the system, so maybe we would sacrifice an higher convergent speed of the observer if needed.

For every attempt the \mathbf{R}_d matrix is fixed and it is given by:

$$\mathbf{R}_d = \begin{bmatrix} \sigma_{pot}^2 & 0 \\ 0 & \sigma_{las}^2 \end{bmatrix} = \begin{bmatrix} (1 \cdot 10^{-5})^2 & 0 \\ 0 & (30 \cdot 10^{-6})^2 \end{bmatrix} \quad (2.84)$$

The first attempt performed was focused on understanding how much we can push on the values. So, we set some values that do not necessarily resemble some physical meaning, but are useful to understand how the system behaves.

The first λ_d and \mathbf{Q}_d values are:

$$\lambda_d = 0 \quad \mathbf{Q}_d = \begin{bmatrix} 1^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.001^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.85)$$

From now on we refer to the state estimated by the observer using $\hat{\mathbf{x}}$.

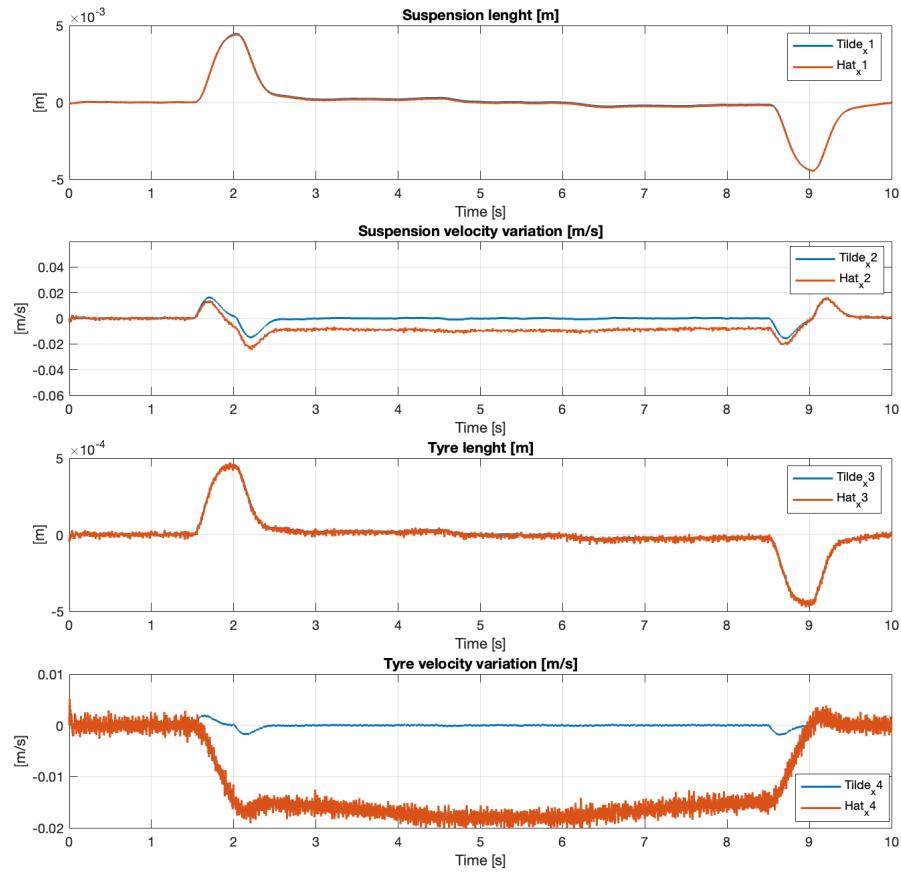


Figure 2.17: First tuning test: $\tilde{x} vs \hat{x}$

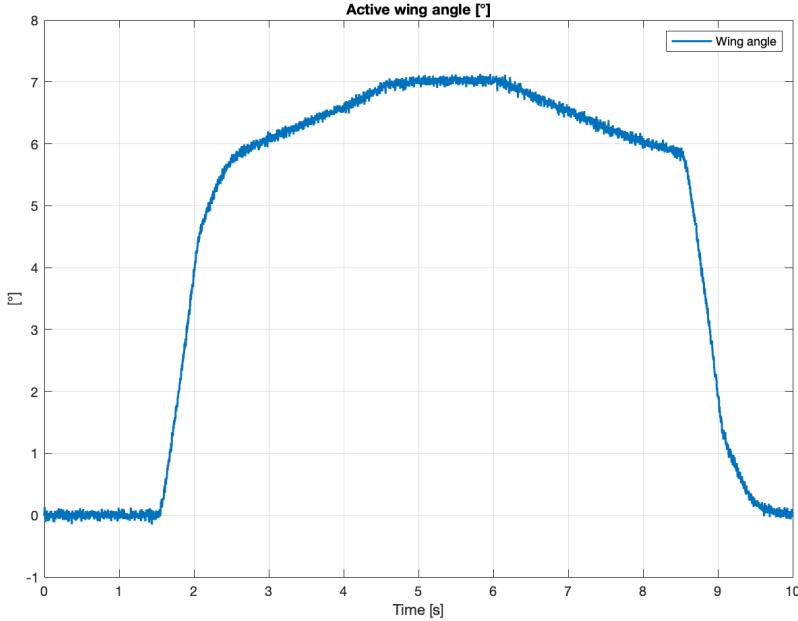


Figure 2.18: First tuning test: control

From the first test we can see that the results are quite poor. The only state that seems to be tracked well, also without noise, is the first one. While, especially the states that are tracking the speed are quite noisy and are diverging a little bit too much. Also \tilde{u} seems to be too noisy and, in a real application it will be not useful.

For the second attempt we want to see how much we can push on λ_d and \mathbf{Q}_d and quantify the magnitude of the noise that we obtain. So, we exaggerate our tuning parameters and the λ_d and \mathbf{Q}_d are the following:

$$\lambda_d = 10 \quad \mathbf{Q}_d = \begin{bmatrix} 10^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2000^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.001^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.86)$$

The only one that we kept constant was $\omega_{3_{max}}$ because it is related to the gravity acceleration and it make no sense to set an high value. We expect to have \tilde{x} and \hat{x} more close to each other than before but introducing an higher level of noise.

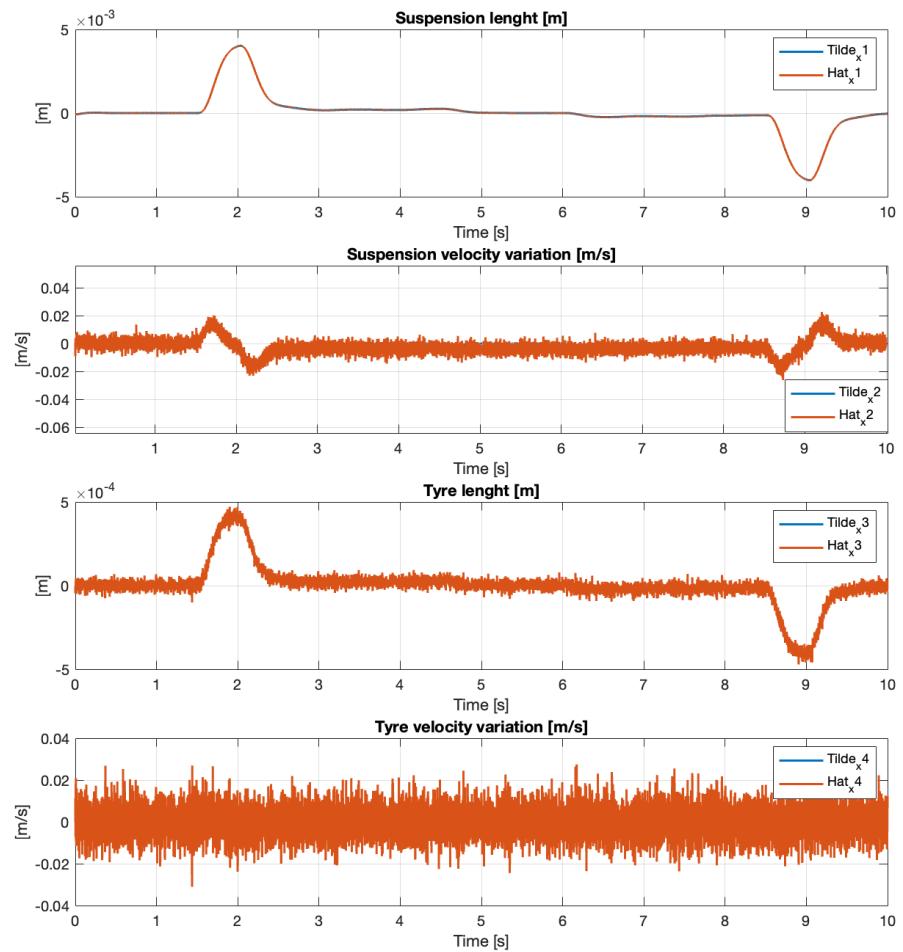


Figure 2.19: Second tuning test: $\tilde{\mathbf{x}} vs \hat{\mathbf{x}}$

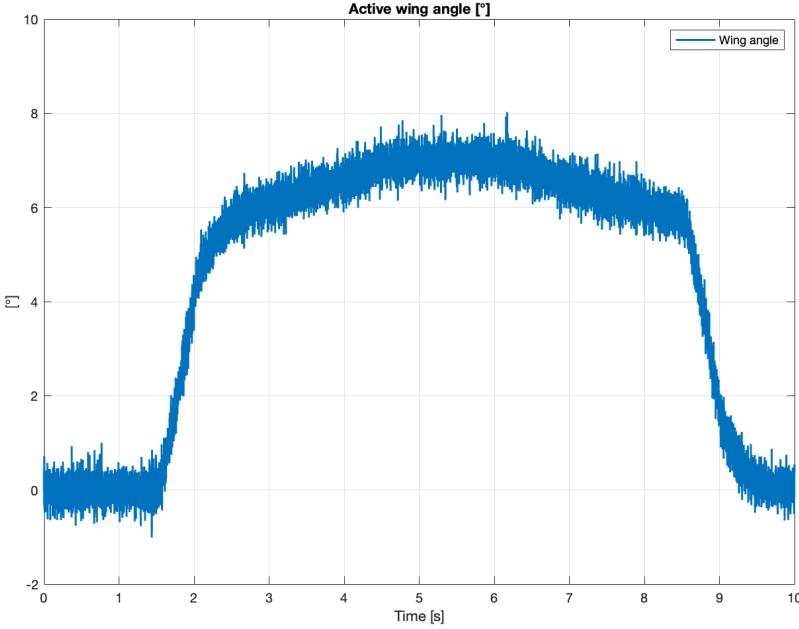


Figure 2.20: Second tuning test: control

In this attempt our expectation has been confirmed: it is possible to see a really noisy signal but the two curves are getting closer. The only state that does not seem to be affected by the variation of the values is the first one, this is related to the fact that the measure of this state is directly performed by the potentiometer (which is affected by less disturbances respect to the laser). Also for the control chart we can see a lot of noise which makes its use totally useless.

The last tuning attempt was focused on balancing all the parameters basing on what we learned in the previous ones. So, we want to focus on not introduce much noise in the measure and keep especially the tracking of the first and second states as close as possible to the real ones.

Finally, the λ_d and \mathbf{Q}_d values are reported:

$$\lambda_d = 3.5 \quad \mathbf{Q}_d = \begin{bmatrix} 0.01^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 80^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.001^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.87)$$

A consideration should be made on the value of λ_d , we chose this specific value to keep the eigenvalues of \mathbf{A}_m matrix with negative real parts so not disturb the stability of the system too much.

All the other values were chosen to have some sort of physical meaning, but always keeping in mind that we do not want to introduce too much noise in the system.

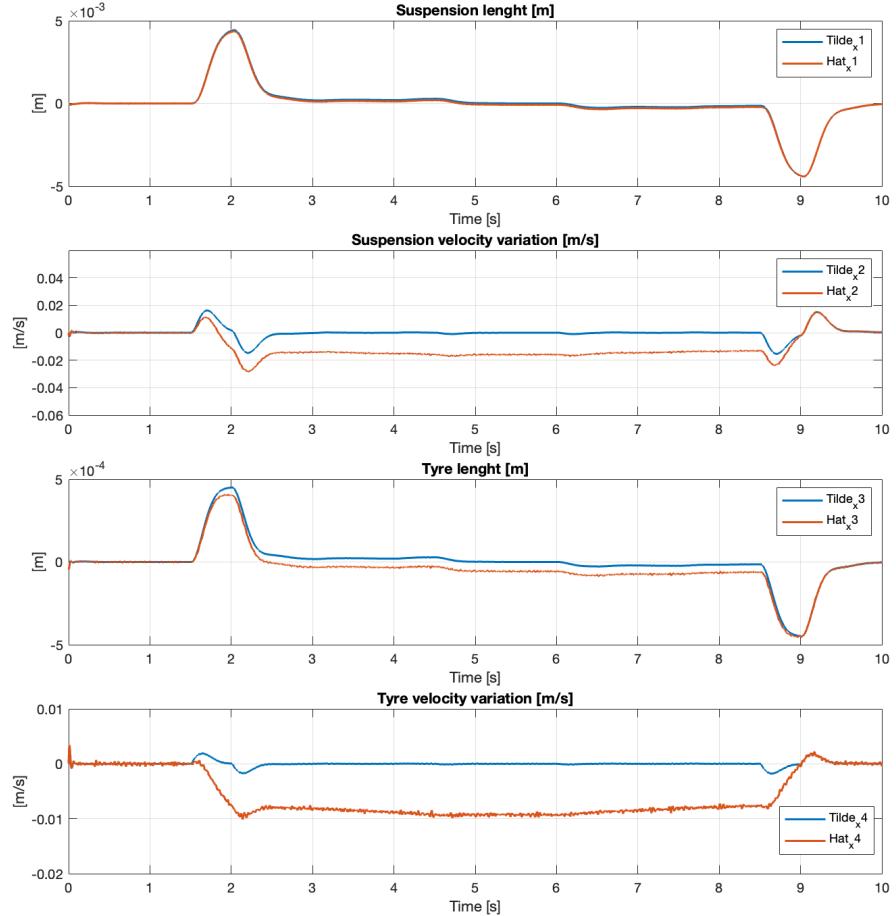


Figure 2.21: Second tuning test: $\tilde{\mathbf{x}} vs \hat{\mathbf{x}}$

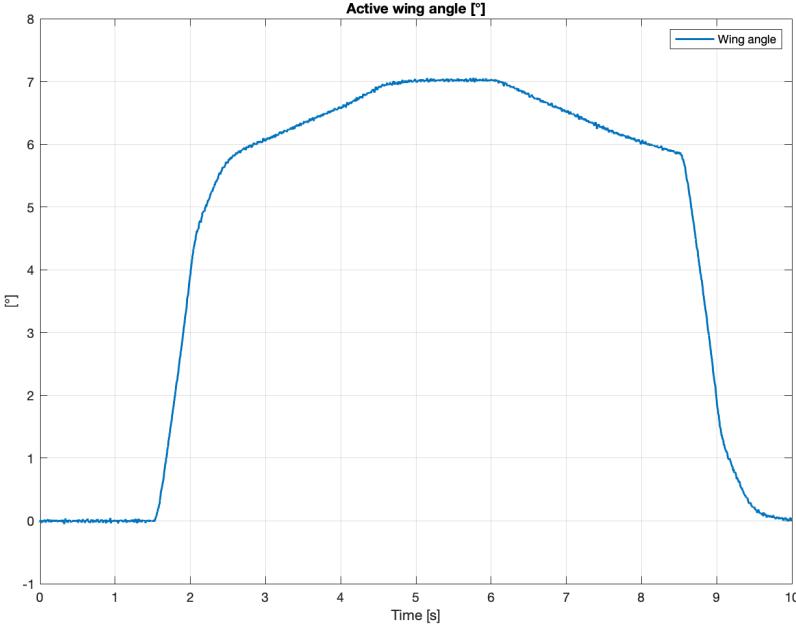


Figure 2.22: Second tuning test: control

Finally, we get a more precise calibration of the observer reaching a compromise between noise and performance. The only states that seems to present some uncertainty are the two related to the speed, we accepted this because we are focusing more on the positions and this compromise let us have less global noise. Now the control is more smooth, it has a better shape and it has a low magnitude of noise.

As before, all the tuning procedure has been performed with the *linear system*.

2.3.4 Feed Forward

In our case of study, a feedforward control contribution is not needed. In this project, the system has to follow a constant reference (not a time-variant one) and so a control design that works only when the system has to be steered to a time-variable reference is useless. Differently, if we had to adapt our reference to a dynamic variation problem for some particular application, in this case we should implement a feedforward control to stabilise the system.

Chapter 3

Application

In this section will be reported the entire *Simulink* scheme and the most relevant subsystems in order to get a complete view of the environment in which all the simulations are performed.

3.1 Simulator description

Simulation environment

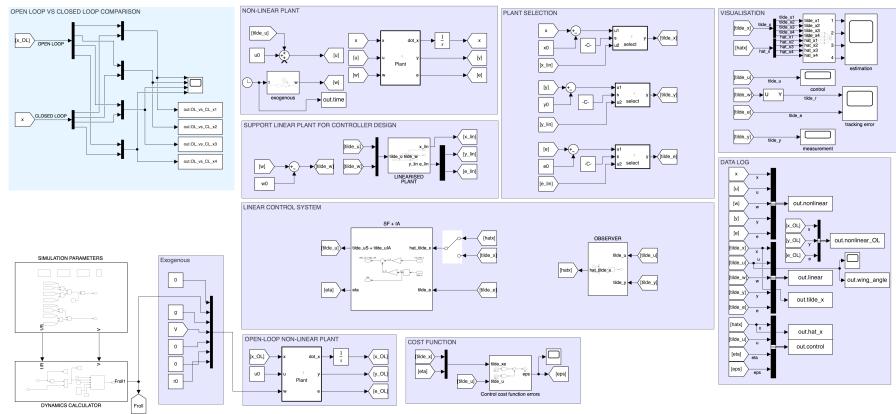


Figure 3.1: Simulation environment

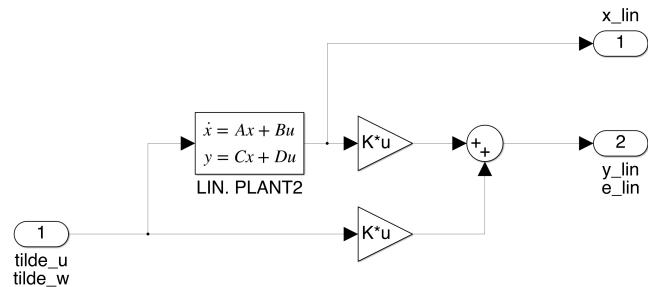


Figure 3.2: Linear plant

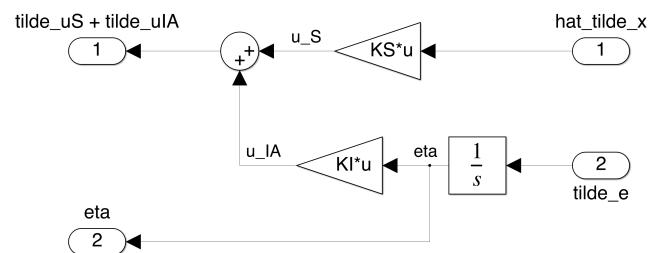


Figure 3.3: State feedback and integral action

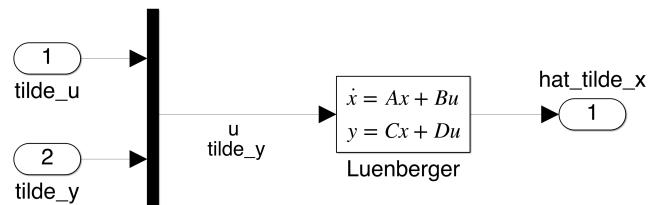


Figure 3.4: Observer

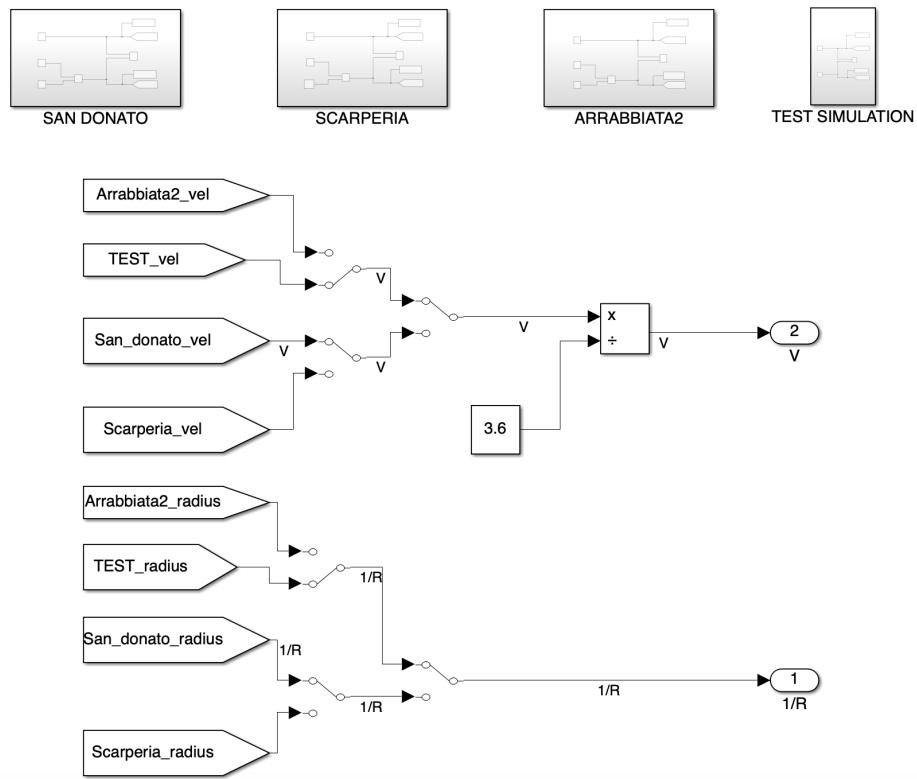


Figure 3.5: Simulation scenario

Inside the simulation scenario is possible to chose the desired track or test conditions, both speed and curvature radius. These blocks are useful to compute F_{roll} during a corner, according to the formulas written in Section 2.1.

3.2 Simulation results

In this section we want to test the system in real application conditions on the non linear plant, with noises and disturbances taken into account.

As the reference model could be implemented on a GT car, it has been decided to perform the simulations through three Mugello's specific curves. Thanks to the real telemetry we had the opportunity to interpolate and to copy the speed and radius profiles as well as possible. The two references have been replicated trying to get the profiles as continuous as possible, in order to avoid sever discontinuities. Due to some lack of information's availability about the circuit, road grade and banking have not been considered and set to zero.

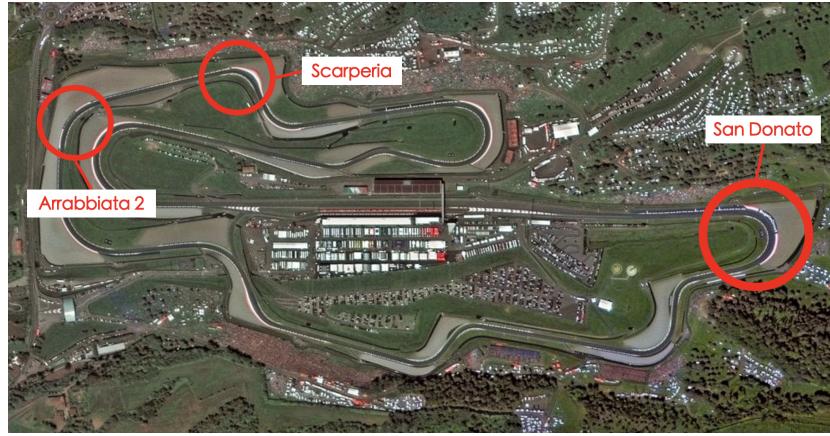


Figure 3.6: Mugello circuit

The three curves, *San Donato*, *Scarperia* and *Arrabbiata 2* have been chosen because of their different speed and radius characteristics, this in order to have quite various simulations to stress the system and to verify its robustness.

3.2.1 San Donato

The first scenario is related to the *San Donato* turn in which there is a speed reduction from 130 km/h to a minimum speed of 97 km/h in the centre of the turn. The curvature profile shows that the radius of the turn is decreasing getting closer to the centre, this will lead to a higher steering angle and faster speed reduction. To sum up it's possible to say that *San Donato* is a medium speed corner with small variation in the radius, these characteristics will lead to a medium stress on the control system.

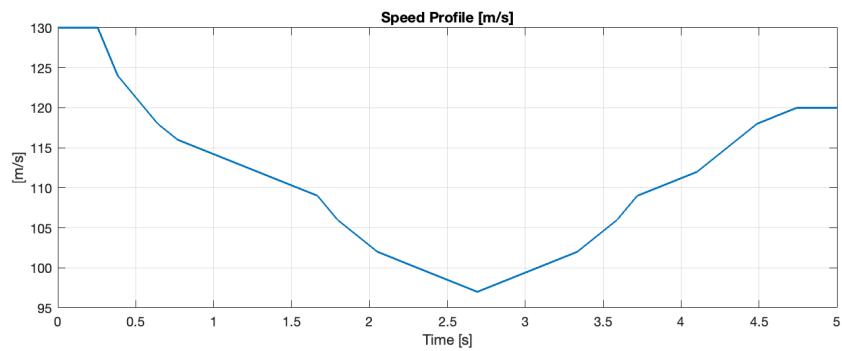


Figure 3.7: San Donato: speed profile

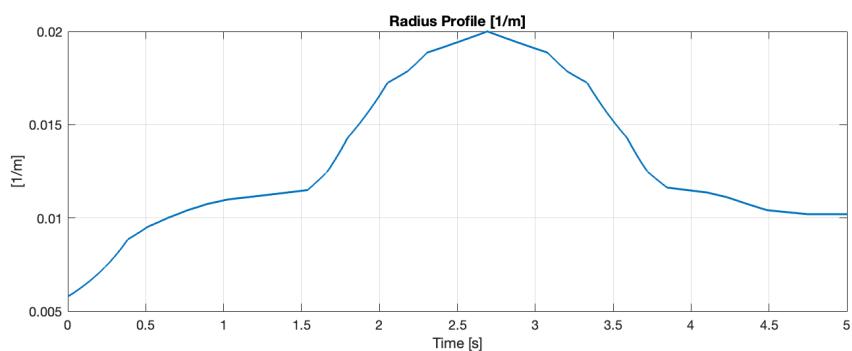


Figure 3.8: San Donato: radius profile

The results we get are the following:

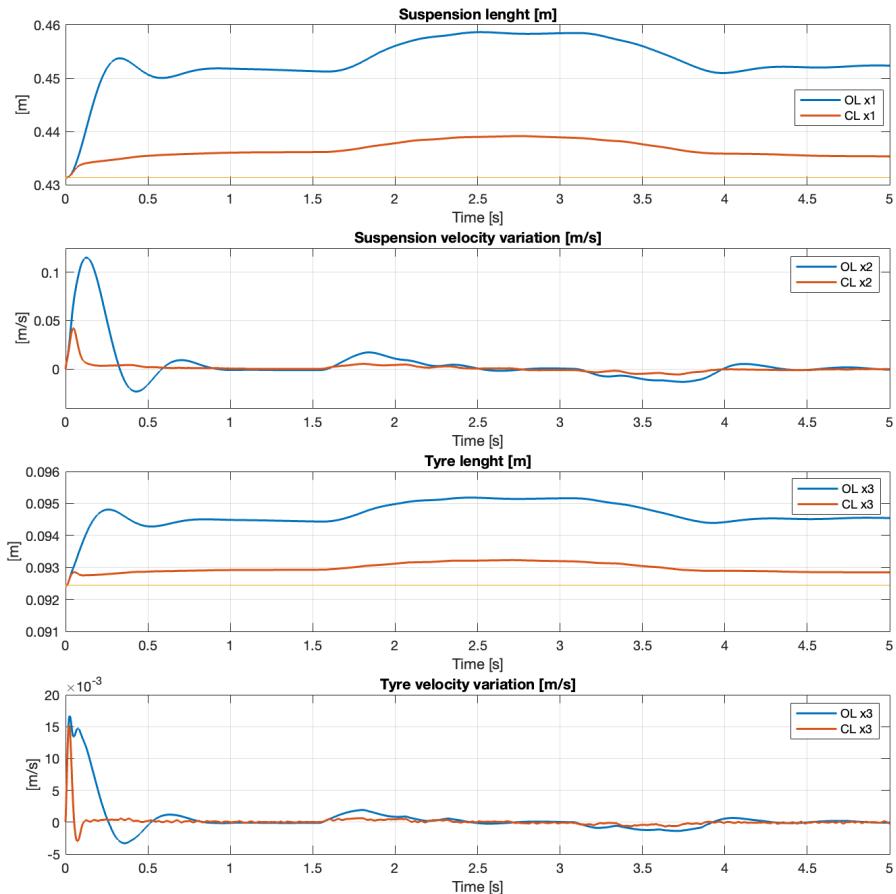


Figure 3.9: San Donato: states (OL vs CL)

From the comparison between open and closed loop, it's possible to appreciate a consistent reduction in the suspension length variation. The closed loop system is significantly closer to the reference value, this means that the vehicle height variation is lower than the open loop one and the roll phenomenon is consistently mitigated. This goal is considered reached for the actual application since the error is limited to an acceptable value.

Also the dynamics behaviour of the suspension is better in the closed loop system, where a big part of the oscillations has been reduced or cancelled. Even if our focus is not on the tyre behaviour, an improvement in the performance can be easily detected.

The maximum angle reached by the flap is around 9° , this means that it does not need to operate close to the aerodynamic stall region. A consideration about the start of the simulation must be done, due to the initial velocity different from the linearised one, the system will try to get closer to the correct value in the faster way possible, generating the initial peak.

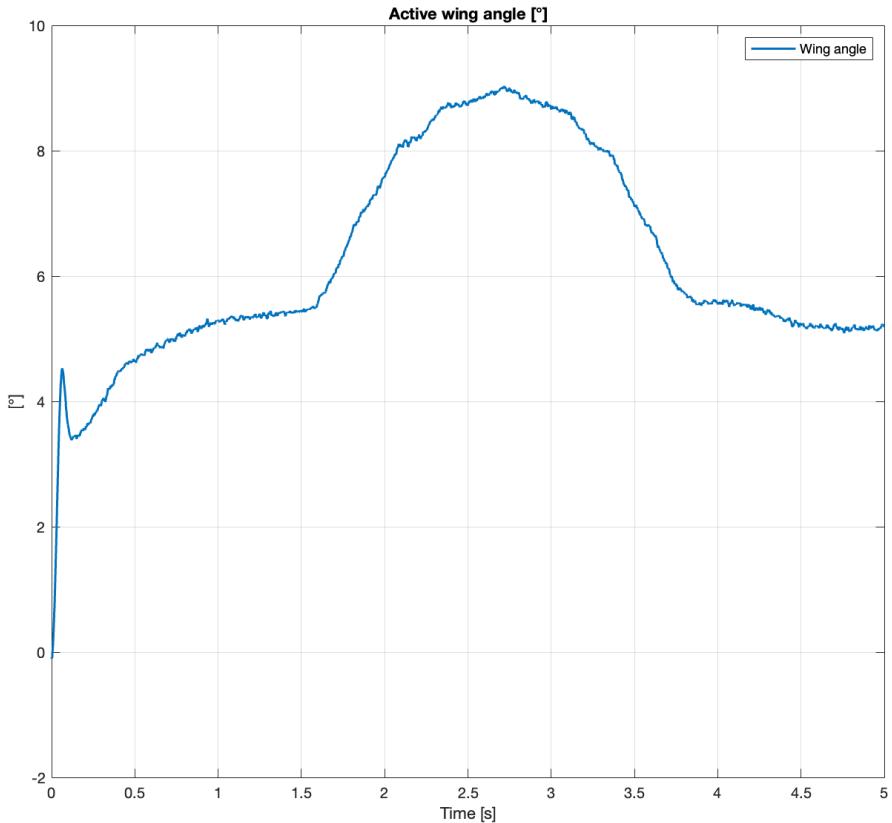


Figure 3.10: San Donato: control

Here, the roll angle comparison graph, between open loop and closed loop, in order to understand in a easier way the contribution of the control system:

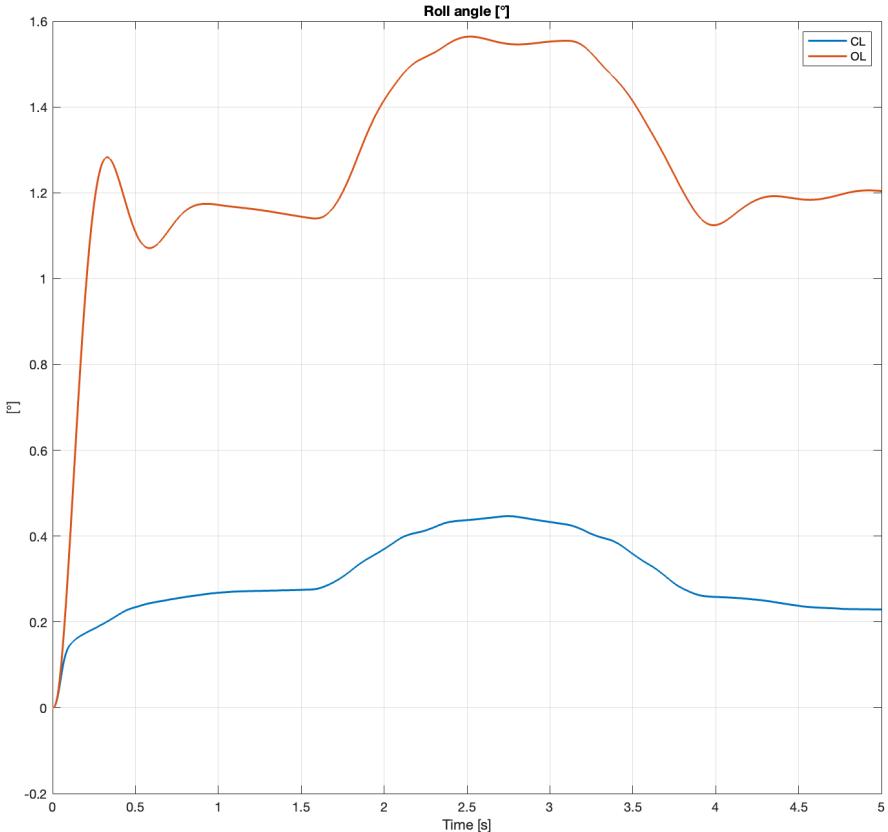


Figure 3.11: San Donato: roll angle (OL vs CL)

3.2.2 Scarperia

The second scenario is related to the *Scarpaia* corner in which the speed reduction is lower than before, because the speed in the center of the corner is higher (115 km/h). Due to this behaviour we expect more difficult conditions to be managed from the control in the center of the turn. A stress relief for the control system is the fact that the curvature is less aggressive, as we can see from the profile, with respect to the *San Donato*.

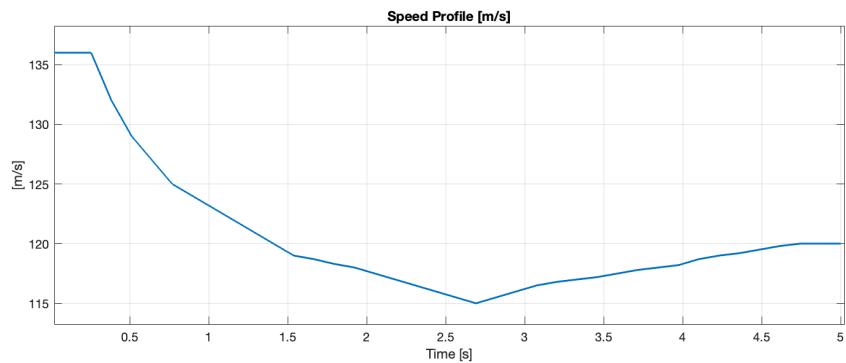


Figure 3.12: Scarperia: speed profile

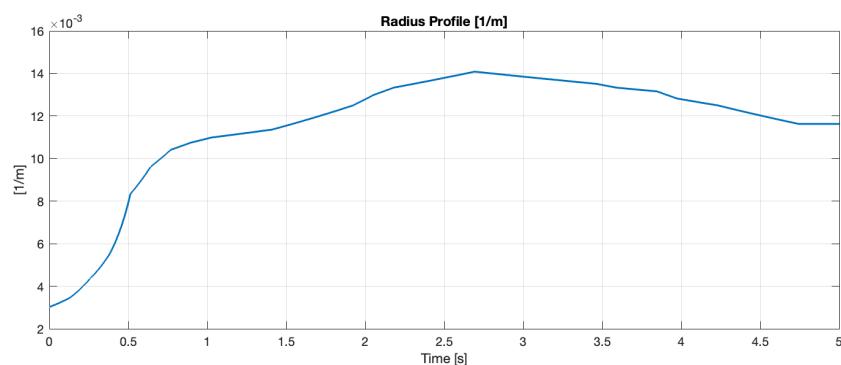


Figure 3.13: Scarperia: radius profile

The results we get are the following:

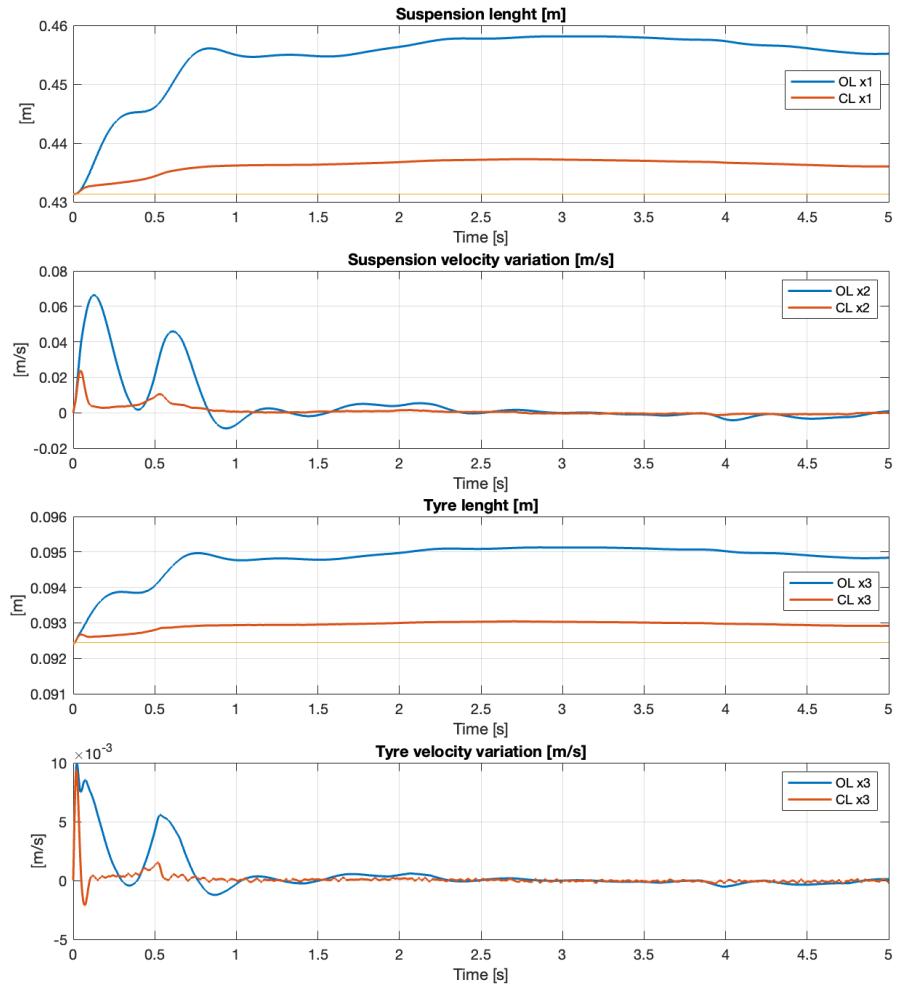


Figure 3.14: Scarperia: states (OL vs CL)

As before, it is possible to appreciate a good improvement in terms of vehicle height and consequently in roll angle. Also the dynamic of the suspension is better in the closed loop, where the control system works.

In this case we achieved a double success: the roll angle control and a big oscillation reduction of the vehicle. The states related to the tires reproduce the first and second states behaviour with a lower magnitude.

The maximum angle reached by the flap is around 7° , also in this scenario the stall limit is quite far, so this means that the control system could respond well also in a higher stress environment.

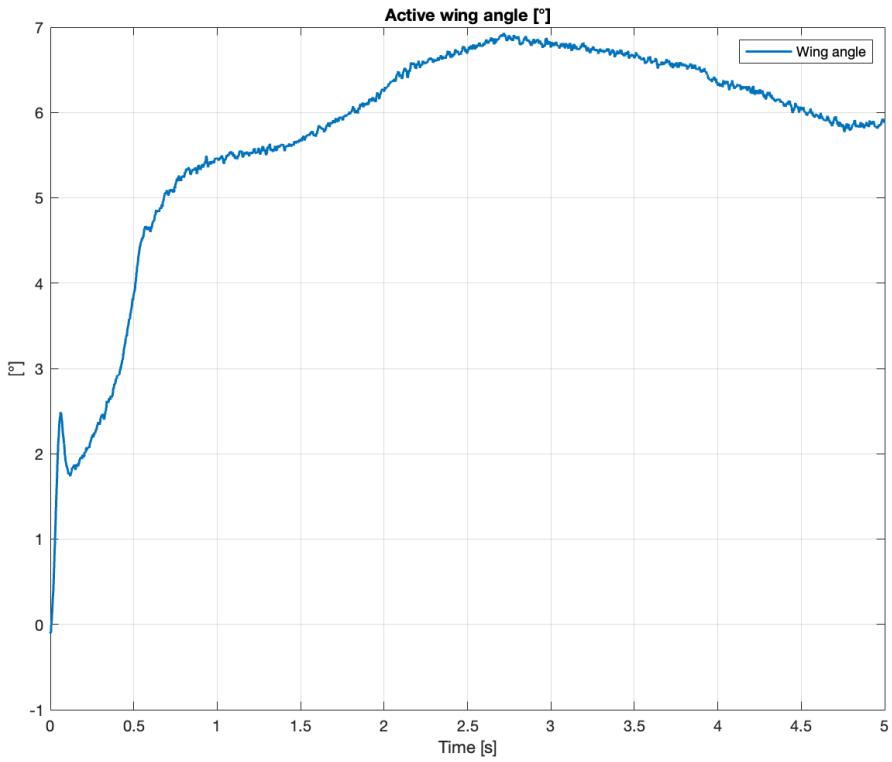


Figure 3.15: Scarperia: control

Here, the roll angle comparison graph, between open loop and closed loop, in order to understand in a easier way the contribution of the control system:

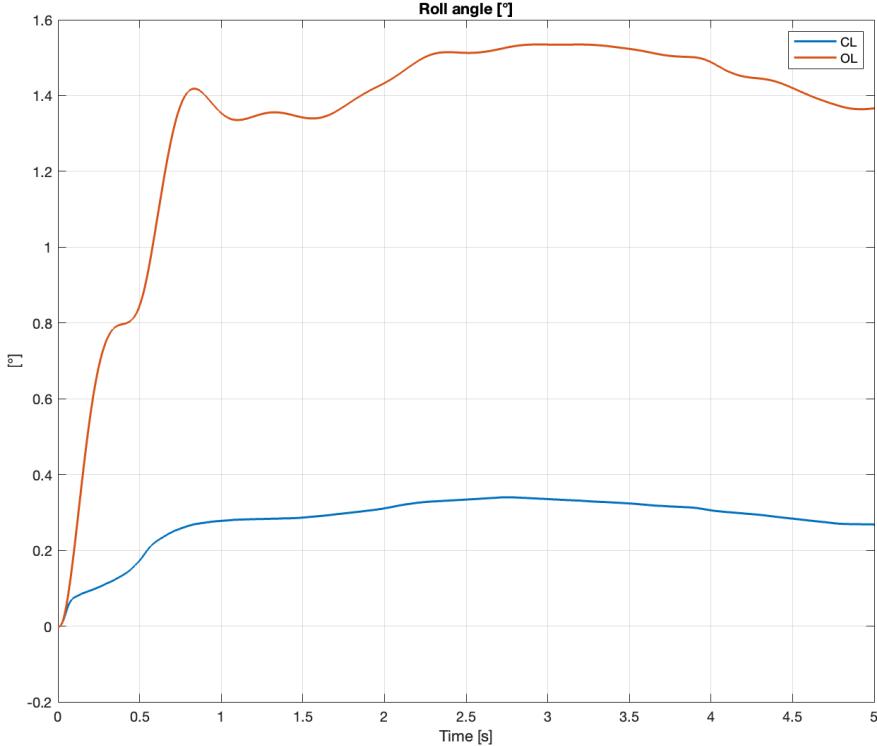


Figure 3.16: Scarperia: roll angle (OL vs CL)

3.2.3 Arrabbiata 2

The third scenario is related to the *Arrabbiata 2* turn, we decided to consider this corner due to the different characteristics with respect to the others. Here the initial speed is quite higher (184 km/h) and the vehicle follows the center of the curve with a minimum speed of 165 km/h. This profile speed is possible thanks to a lower radius which define this curve as a high speed turn, so different from the previous two.

Due to the different nature of this corner, the first simulation was not acceptable; so a change in the control parameters was needed. In particular we decided to have a lower permissible error for the control, due to the high speed in which the corner is performed, in order to avoid significant variation in the force applied

by the wing. The new matrices are here reported:

$$\mathbf{Q} = \frac{1}{5} \begin{bmatrix} \frac{1}{0.0001^2} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{0.1^2} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{1^2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{1^2} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{0.005^2} \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} \frac{1}{0.1^2} \end{bmatrix} \quad (3.1)$$

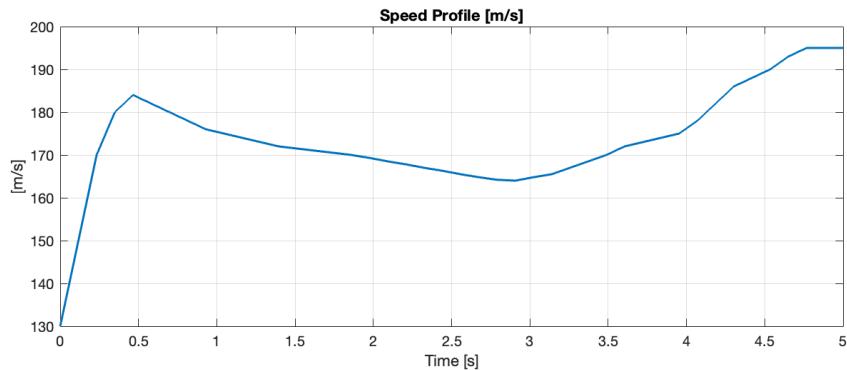


Figure 3.17: Arrabbiata 2: speed profile

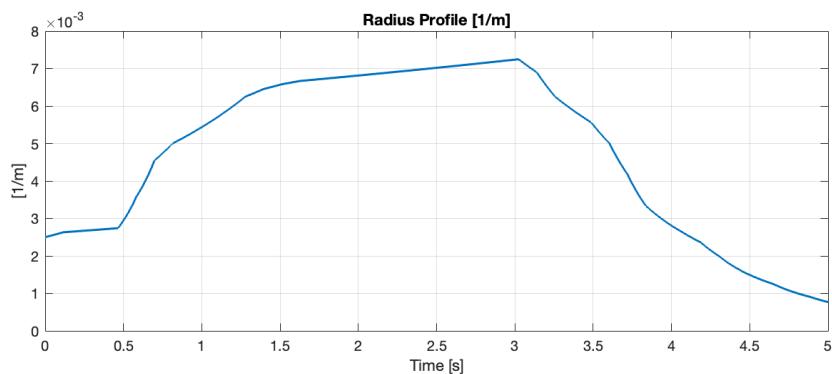


Figure 3.18: Arrabbiata 2: radius profile

The results we get are the following:

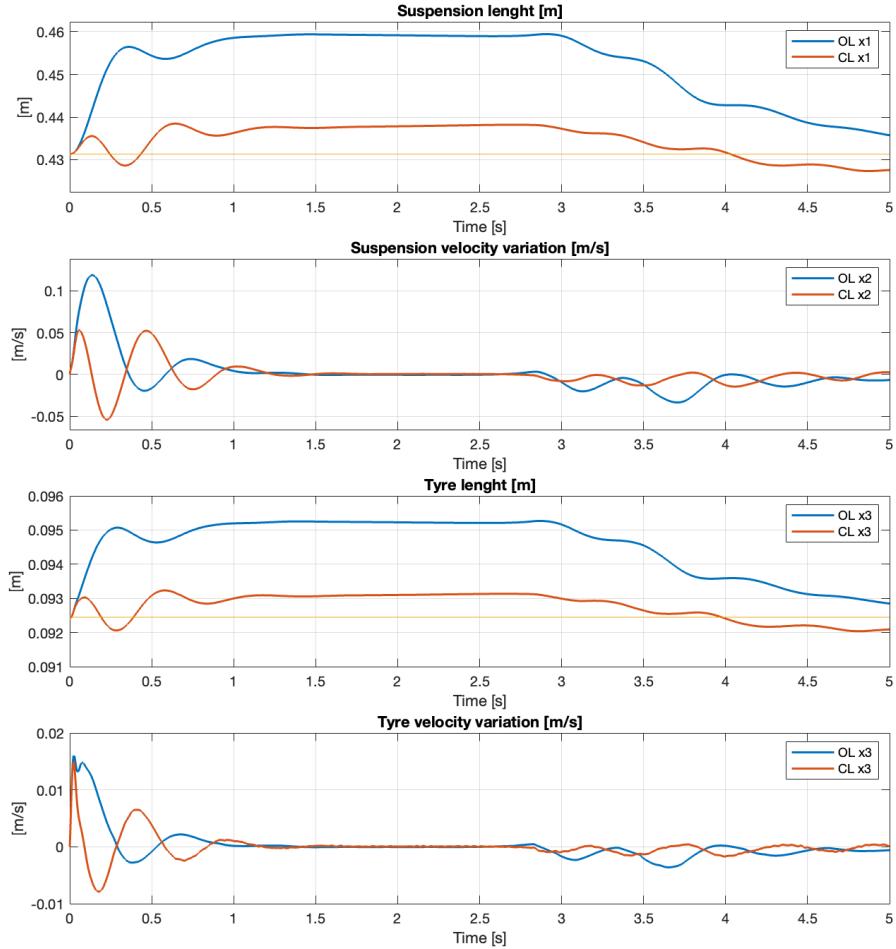


Figure 3.19: Arrabbiata 2: states (OL vs CL)

Also in this third test, the results confirmed that the control works in a good way, in different operating condition with a small variation in the setting. From the first state is possible to appreciate a significant reduction in the suspension length, while in the second one the magnitude of the oscillation is reduced. As before, thanks to this control architecture, is possible to control in a better way the suspension, both in length and velocity, improving on-board comfort and safety.

The maximum angle reached by the flap is around 2.4° , this because, with respect to the other simulations, the speed is increased a lot and the force generate by the wing is achieved with a lower incident angle.

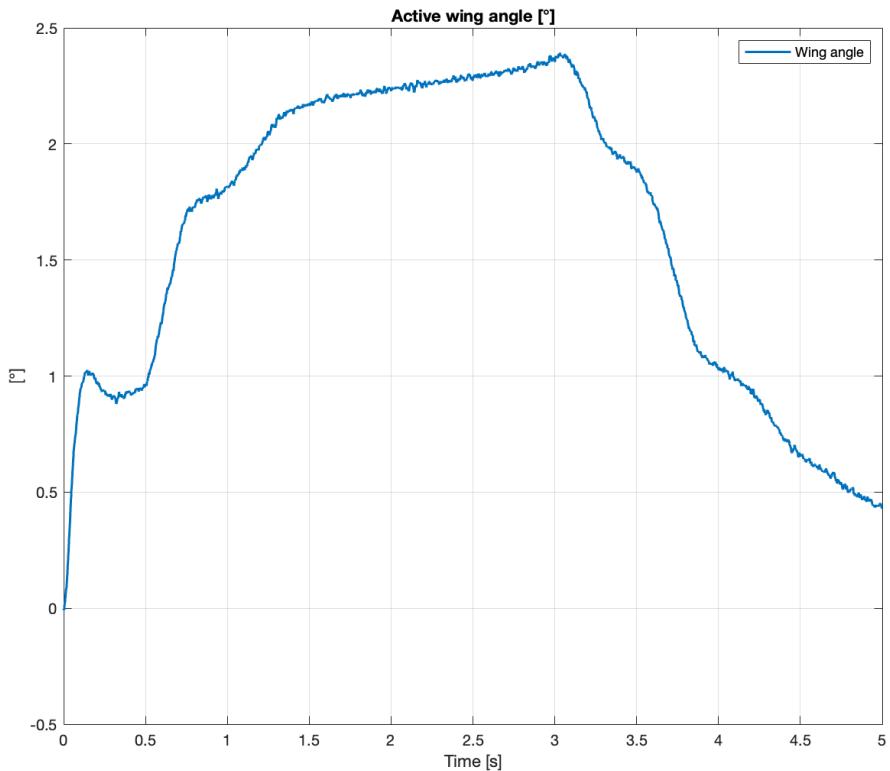


Figure 3.20: Arrabbiata 2: control

Here, the roll angle comparison graph, between open loop and closed loop, in order to understand in a easier way the contribution of the control system:

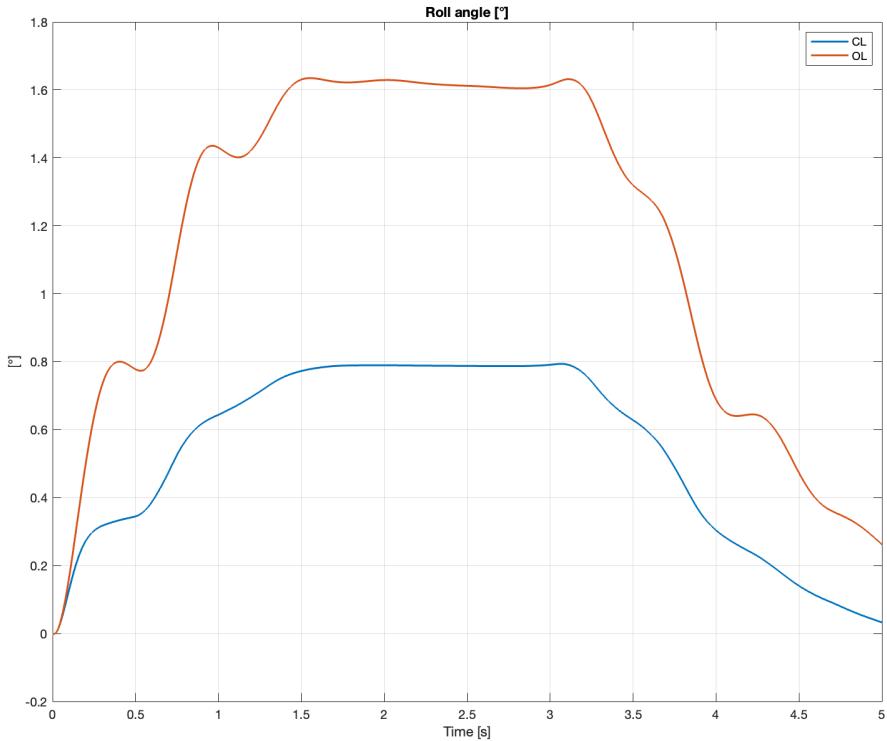


Figure 3.21: Arrabbiata 2: roll angle (OL vs CL)

Chapter 4

Conclusions and further investigation

The object of this manuscript was to propose a possible solution for reducing the roll angle acting on a GT car performing a corner with the introduction of an aerodynamic active system. The way we organise our solution proposal has been focused on finding a compromise between a real application model and some needed simplified hypothesis.

It is important to highlight that the work has been performed considering a flat surface, as it is expect from most of the racing circuit application. This assumption will lead to a zero-slope road, but the possibility to introduce a disturbance on the road behaviour is allowable.

The results we expect from the simulation environment were met, in accordance with the theory discovered during the classes. It has been a challenging project, but, thanks to it, we improved our control system skills and sensibility.

Bibliography

- [1] Nicola Mimmo (1977), *Analysis and Design of Control Laws for Advanced Driver-Assistance Systems Theory and Applications*, University of Bologna.
- [2] Ejaz Ahmad and Iljoong Youn (2022), *Performance Improvement of a Vehicle Equipped with Active Aerodynamic Surfaces Using Anti-Jerk Preview Control Strategy*, Sensors
- [3] Guido Buresti (2012), *Elements of fluid dynamics*, Imperial College Press
- [4] Jorge Segers (2014), *Analysis Techniques for Racecar Data Acquisition*, SAE-International
- [5] Ejaz Ahmad, Jamshed Iqbal, Muhammad Arshad Khan, Wu Liang and Iljoong Youn (2020), *Predictive Control Using Active Aerodynamic Surfaces to Improve Ride Quality of a Vehicle*, Electronics