



MOTORVEHICLE
UNIVERSITY OF
EMILIA-ROMAGNA

ANTI-PORPOISING ACTIVE SUSPENSION SYSTEM FOR A FORMULA ONE CAR

Authors: Guido Matteo
Mora Salazar Manuel Alejandro
Pedroni Alberto
Pelaez Rodriguez Luis Eduardo

Abstract

After forty years, Formula 1 cars with “*ground effect*” have been reintroduced: the new rules adopted in 2022 [3], in fact, revolved around a different approach to the way such race-cars produce aerodynamic load: indeed, the “*downforce*” is generated through the floor of the car, renouncing to the complex aerodynamic devices that characterised single-seaters until 2021 and reducing the complexities of both wings and sidepods, thus making them less sensitive to the wake [5] and less prone to lose downforce when following closely another car.

However, together with this radical change, a problem has arisen since the early stages of the 2022 season: it is the so-called “*Porpoising effect*”, which is the effect that makes the cars bouncing and pitching aggressively while driving along a straight at high speeds. This phenomenon significantly affects the performance of the vehicle since such oscillations can cause loss of grip as a result of downforce variation reducing car performance, stability and driver comfort [4].

In Formula 1, where every fraction of a second can make the difference, porpoising has become a significant challenge for teams, that want to find the right balance between the car’s downforce and its stability.

This report proposes the design of an active suspension control system for a Formula 1 car to smoothen porpoising, reaching the best achievable performance.

Contents

1	Introduction	3
1.1	Preamble	3
1.2	Porpoising Effect	6
1.3	Motivations and Approach to the Issue	8
1.3.1	Motivations	8
1.3.2	Approach to the Issue	9
1.4	Sensors	10
1.5	Organisation of the Manuscript	13
1.6	List of the Symbols	15
2	Main Body	16
2.1	Model and Problem Formulation	16
2.1.1	Aerodynamic Force	18
2.1.2	Comparison between open-loop simulation results and real world porpoising data	21
2.2	Model Analysis	24
2.2.1	Linearisation	24
2.2.2	Plant Self-Stability Analysis	25
2.2.3	Reachability Study	28
2.2.4	Observability Analysis	29
2.3	Proposed Solution	31
2.3.1	State Feedback and Integral Action	31
2.3.2	Observer	33
3	Application	35
3.1	Simulator Description	35
3.1.1	Non-linear Plant, Closed-Loop	35
3.1.2	Non-Linear Plant, Open-Loop	36
3.1.3	Linear Plant	37
3.1.4	Coordinates Linearisation	38
3.1.5	Exogenous vector Generator	39
3.1.6	Non-linear Control Implementation	40
3.1.7	Observer Block	40
3.1.8	Cost Function Block	41
3.1.9	Visualisation Block	41
3.1.10	Data Log	42
3.2	Simulation Results	43
3.2.1	Controller Tuning	43
3.2.2	“Sweep Test” for the Controller Tuning	44
3.2.3	Observer Tuning	47
3.2.4	Final Optimisation	49
3.2.5	Robustness Tests	50
3.2.6	Performance Analysis of some relevant testing scenarios	53
4	Conclusions and Further Investigations	62
4.1	Conclusions	62
4.2	Further Investigations	62

5 Appendix	64
5.1 Matlab Launch code	64
5.2 Matlab Non-Linear Plant function	69
5.3 Matlab Noise generation function	70

Chapter 1

Introduction

1.1 Preamble



Figure 1.1: Three-quarter view of the 2022 Formula One vehicle concept

Before moving on to the real problem that will be dealt with in this report, it is interesting to note that there has been a sort of “revolution” in the Formula 1 regulation [3] between the 2021 and 2022 seasons which led to a radical modification of the racing cars.

More specifically, nine are the “key things” that have been introduced or modified through this significant change in the regulation:

1. **“Better Racing”**

The 2022 regulations, originally planned to arrive in 2021 but delayed by Covid-19, had one guiding principle: to both allow closer racing and enhance the show by increasing overtaking.

What prevented closer racing, up to 2021, was the “catastrophic” effect of downforce loss resulting from the “dirty air” being churned chaotically off a leading car [5]:



Figure 1.2: Loss of Downforce comparison between 2021 and 2022 Formula 1 single-seaters

Indeed, just looking at the pictures reported in fig.1.2, researches proved that 2021 F1 machines lost:

- 35% of their downforce when running three-car-lengths behind a leading car (approximately 20 metres).
- 46% of their downforce when running one-car-length behind a leading car (around 10 metres).

Instead, the 2022 car (developed by Formula 1's in-house Motorsports team in collaboration with the FIA) reduces those figures to 4% at 20 metres, rising to just 18% at 10 metres, which are visibly important improvements achieved.

2. “Over-Wheel Winglets and Wheel Covers”

Two of the striking features introduced on 2022 cars are its over-wheel winglets and the wheel covers (which were abandoned after 2009).

Starting from the first feature, the over-wheel winglets, their job is basically to control the wake coming off the front tyre and direct it away from the rear wing: this task was previously performed on 2021 cars through dedicated aerodynamic appendixes, whose drawback was that they lost in functionality when the vehicle was found in “following car” condition. So, such winglets achieve the same task but in a more aerodynamically “resilient” way during close racing.

Otherwise, the inclusion of the wheel covers, even if it is not the best solution in terms of overheating of the braking system, allowed to clean up the airflow coming from the wheels and so helped increasing overtaking.



Figure 1.3: Rendering: Three-quarter view of the Formula 1 vehicle concept

3. “Tyre Diameter from 13 to 18 inches and Lowered-Profile”

The new Pirelli compounds and constructions for the 18-inch tyres have been designed with the main purpose of reducing the amount the tyres overheat when sliding, and this should help even more in the occurrence of closer racing.

Instead, lowering the tyre’s profile added the benefit of reducing the sidewall deflection changes and the resulting aerodynamic wake effect.

4. “New Front Wing and Nose concepts”

Keeping with the philosophy of the 2022 car, the front wing, pictured in fig. 1.4, aims at generating consistent downforce when running closely behind another car and ensures that the front wheel wake is well controlled and directed down the car in the least disruptive possible way.



Figure 1.4: Rendering: Front view of the Formula 1 vehicle concept

5. “More emphasised Ground Effect”

Ground effect came to prominence in F1 in the late 1970s, with cars effectively designed in the shape of upside-down airplane wings, creating huge amounts of downforce as they were pushed into the track [10].

Full ground effect cars were subsequently outlawed at the end of 1982 for the danger posed while cornering but, thanks to the work of the Venturi ducts that keep the cars “glued” to the ground, more grip was provided to such cars and so the drivers will be consequently facilitated in overtaking, safely.

6. “*Rolled tips* of the Rear Wing”

While current cars’ rear wings direct airflow upwards, they are also designed to send flow outwards, leaving the “dirty air” sitting there for the following car to drive through. By contrast, the shape and position of the 2022 car’s rear wing, pictured in fig. 1.5, creates a rotational airflow that collects the rear wheel wake and rolls it into the flow exiting the diffuser.

This narrower wake is then thrown high up into the air, allowing a following car to drive through less disrupted “cleaner air”.



Figure 1.5: Rendering: Rear view of the Formula 1 vehicle concept

7. “Same Power Unit”

Many, things were new on the 2022 car, but the power unit is not one of them, meaning that on Formula 1 cars are still installed 1.6-litre turbo-hybrid units introduced in 2014.

8. “Sustainable Fuel”

A major change occurred for the 2022 season, concerning the power unit, was the mandate to have an internal combustion engine running on fuel containing 10% bio-components, due to a move to “E10 fuel”, where “E” stands for “Ethanol” and “10” refers to its percentage in the mixture.

This choice has been implemented to reduce carbon footprint, which also helps the sport aligning with the current road car fuel regulations.

9. “Safety as the ‘forefront’ of the design”

It almost goes without saying that the main objective of Formula 1 cars is safety: therefore, all the innovations introduced must not only guarantee the current level of safety for the driver and the spectators but, if it is possible, they should increase the overall level of safety in the sport.

1.2 Porpoising Effect

In addition to the positive changes made to the single-seaters previously listed in terms of “entertainment” and safety, some knots “have come to a head”: in particular, the effect that has begun to glimpse starting from the early stages of the season was the so-called “Porpoising”, namely the aerodynamic phenomenon in which the car starts to pitch and bounce significantly once high speeds are reached [4].

This is probably one of the most influencing result that the new regulations, imposed by FIA [3], generated on the new prototypes, which are “*ground effect-based*” cars.

Ground effect is the phenomenon that occurs when air is sucked underneath the car to pull it down onto the track at high-speed, producing downforce as a result and so a better manoeuvrability of the vehicle, especially when cornering [10].

Indeed, the porpoising issue occurs when the floor gets too close to the ground: in such a case what happens is that the airflow stalls and forces the car to spring upwards. However, once the car’s floor is clear of the ground, the airflow gets in again and the car is sucked downwards again causing a bouncing

phenomenon known as “porpoising” [4].

To understand a little bit more about such effect, the most representative “imaginary picture” about porpoising is that of the movement of a porpoise in the sea: in a Formula 1 car, porpoising may be described as “bouncing” or “hopping”.

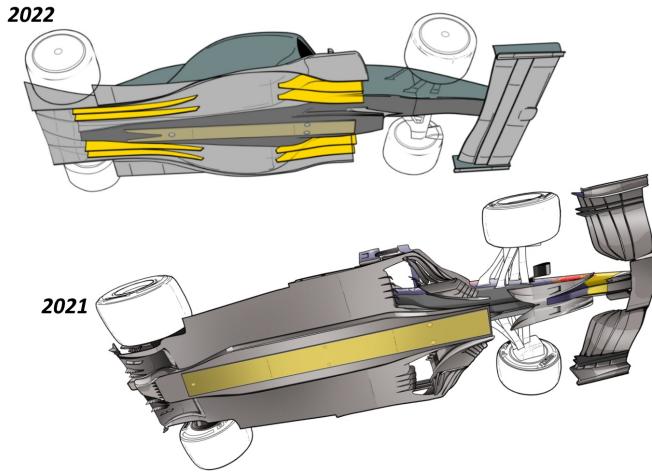


Figure 1.6: Comparison between 2021 and 2022 undertrays

Unlike the 2021 F1 car, as it is easy to see in fig. 1.6, the underbody of the 2022 F1 car is not flat: this was the consequence of the fact that, as already mentioned previously, due to the removal of the aerodynamic appendixes and devices that were previously installed on 2021 cars, there was a downforce loss which was compensated through the introduction of “Venturi ducts” on the underbody of the car, infact such devices are capable of generating a huge amount of downforce [10] by the depression formed by the air accelerating through such ducts.

As the car cuts through the air, the air is propelled through the ducts in the underbody. This air forms a negative pressure sucking the car towards the ground: this is the “ground effect” that creates the downforce, which is basically the mean through which the car is helped to take corners at great speeds and race across straights faster.

Then, due to the reaching of the critical height of the undertray, the airflow detaches itself from the upper edge of the diffuser reducing the suction effect: this reduction of downforce unloads the suspension raising the car upwards. The easing of load on the suspension is reversed a while later causing the re-loading of the suspension. Because of the speed at which the car is travelling, the cycle of aerodynamic loading and unloading of the car is rapid and so what the driver feels is like a travel across a series of bumps: higher is the speed and higher is also the harshness of oscillation experienced by the car and driver.



Figure 1.7: Ferrari SF21 experiencing porpoising during the pre-season tests in Barcelona

It is intuitive that porpoising can be very dangerous for the driver's safety as grip between the tyres and the ground can be lost on fast corners leading to higher risk of accidents where the bumping experienced by the driver can cause him to lose control of the car.

In addition to the reduction in both performance [4] and safety terms, porpoising has potential long-term health implications.



Figure 1.8: Lewis Hamilton suffering backache after the Azerbaijan Grand Prix, 2021

Indeed, due to the repeated hopping of the car, the driver turns out to be particularly tossed and put under physical stress especially in the long run and more critically for older drivers.

1.3 Motivations and Approach to the Issue

1.3.1 Motivations

The motivation behind this project can be gathered in a single keyword: “performance”. As well known, Formula 1 cars are the most performing prototypes in the world, characterised by the highest level of engineering knowledge, with everything aimed at the reduction in lap time.

More specifically, during the years the “attention on the details” shifted from the macro to the micro, meaning that each component, also the smaller one, is nowadays optimised since it provides, even if in a small way, a contribution that is, at the end of the day, non-negligible in terms of lap time, meaning hundreds of second over a single lap that, in the end, translates in gaining seconds, which then means victory and therefore notoriety and more funds that can be invested for the study and improvement of

every aspect of the car.

However, as already stated in the first lines of this paper, due of the changes in regulations introduced by the FIA in 2022 [3], the cars have been transformed into “ground-effect cars”, which basically exploits the floor of the car as the main contribution for the generation of the aerodynamic load, so all those appendixes and wings have a much lower impact in terms of downforce generated than they did during the previous years.

In general, when substantial changes are made to the vehicle, especially if limited opportunity of “on-ground testing” is provided, to reduce the costs as much as possible, there is an increase in the risk of appearance of undesired and negative effects on both the performance and the safety of the race car.

The desire for performance of F1 teams led them to run their cars very close to the ground in order to extract the maximum potential from the newly allowed ground effect floor [10], but this means that at a certain speed the critical height of the floor (which is the minimum height of the floor from the ground below which such aerodynamic device stalls) is reached, and this leads to a loss in functionality due to the detachment of the air fluid dynamic boundary layer [6], resulting so in a massive reduction in downforce terms.

Due to the reduction in aerodynamic load production, the consequence is that the floor and so the vehicle rises back and then, after a while, once a new height is reached, the attachment of the air boundary layer to the floor is restored. Then the floor starts again to work in its nominal condition generating such an aerodynamic load, that once again compresses the suspension. At a sufficiently high speed a continuous attachment-detachment of the air boundary layer from the floor occurs, resulting in a high frequency oscillation of the ride height and downforce generated [4].

As already stated, this result does not only translate into a loss in terms of performance, as both stability and maximum speed reached are affected, the latter because a larger aerodynamic drag is consequently generated, but also in terms safety and health of the driver.

So this work proposes the design of an automatic control system that, given the suspension configuration, the geometry and mass distribution of the vehicle, being known the “optimal” condition for the vehicle to be set in terms of aerodynamic load generation, is capable to act on an active suspension system that imposes the desired floor-height and ensures that the floor, during the movement of the car, does not stall, guaranteeing steady aerodynamic load generation without allowing any bouncing behaviour of the vehicle.

It has to be noted that the kind of approach proposed by this paper for what concerns the issue of porpoising is in some way “parallel” to the ones tried and actuated by actual F1 teams [4] as the use of active suspension has been banned in F1 by the FIA in 1994 [9]. Nevertheless, it is thought that an active suspension system could help massively to resolve this issue and this work aims at proving this widespread belief [8].

1.3.2 Approach to the Issue

At the start of the project the main focus has been on the investigation of the porpoising issue and the design of possible physical models to accurately represent it. This has taken some time due to the very limited presence of similar content in the literature [4].

Then the aerodynamic force model has been implemented in an active suspension quarter-car model which, on the contrary, has been very much a part of the active suspension control literature [1], [2].

Then, according to *State-Space based control theory* [7], a theoretical investigation of the main relevant proprieties of this model has been conducted for control purpose and then, in the end, our system has been implemented in Matlab and Simulink environments to run some real time simulations.

During the first stage of the project a great deal of emphasis has been placed on writing a realistic aerodynamic model that was simple enough to be controllable with both the tools and the theory that were available for this project [7]. Then many Open-Loop simulations have been performed in order

to verify that the results were consistent when compared to the real world measured porpoising data. Minor adjustments were made to all those numerical parameters that were not available to the public in order to best fit real world behaviour in the widest set of conditions.

The theoretical analysis has been exploited to fully understand which were the system properties and based on those how the set control goals and the final results that could be achieved by creating a robust system with great performance. The most important control theory tools exploited for those purposes are: State Feedback, Integral Action and Observer. No Feedforward was deemed necessary as the goal of this system is to keep the car at an optimal and time-constant ride height.

The control system parameters were then tuned starting from first attempt values that were deemed to be the most logical values by using a *Trial & Error* approach aimed at creating an effective and robust control policy.

A final performance and robustness evaluation has been carried out to verify the optimisation process, by comparing the optimised system to other proposals and to the Open-Loop system and by verifying that the tuned one allows to obtain the best outcome when accounting for all the different scenarios the system may be subjected to.

1.4 Sensors

Before proceeding with the next step of the project, up to now the attention was focused on the dynamics of the plant, but what is required to be done is to “complete the model”, and this can be done through the introduction of the useful sensors that could be installed on the plant under investigation.

More specifically, the sensors that the plant is equipped with are:

- **Accelerometer**, which provides the acceleration of the body without considering the gravity effect (meaning that it provides the real inertial acceleration minus the gravity).
- **Potentiometer**, which measures the deflection of the suspension depending on the heights at which are located both sprung and unsprung masses, respectively z_s and z_u .

Since these sensors are real, they of course are characterised by a certain level of inaccuracy, meaning that some noise will affect the measurements given by both, respectively ν_1 and ν_2 .

So, installing these two sensors onboard the vehicle, the output signals y_1 and y_2 that are consequently registered are given as follows:

$$y_1 = \text{Output of the Accelerometer} = \ddot{z}_s - g + \nu_1 \quad (1.1)$$

(looking at this expression, it is worth understanding that the accelerometer is installed at the COG of the sprung mass and measures its relative vertical acceleration with respect the gravity one)

$$y_2 = \text{Output of the Potentiometer} = z_s - z_u - l_s + \nu_2 \quad (1.2)$$

(the potentiometer is installed on the suspension in order to measure its deflection, meaning the stroke of the suspension, which measurement of course ranges between zero to a maximum value)

Going more in depth on such devices, by researching in the internet, the following models of accelerometer and potentiometer have been selected:

- The ACCELEROMETER chosen is a TE Connectivity model 4203 triaxial motorsport accelerometer (fig. 1.9 and fig. 1.10), which is a high-performance sensor designed for harsh installations that is capable of providing exceptional performance for demanding racing applications. The version selected for this application is the 4203-10-10-A1-C.

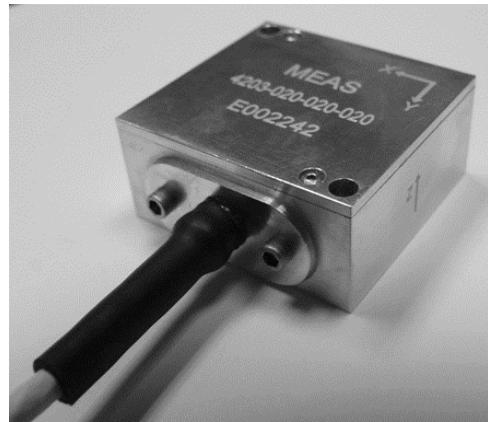


Figure 1.9: TE Connectivity Model 4203 Triaxial Motorsport Accelerometer

This high-performance accelerometer is designed with performance and reliability in mind.

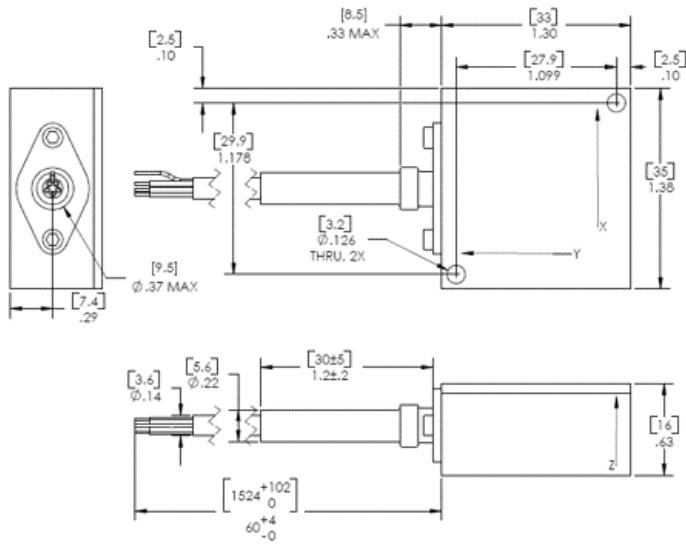


Figure 1.10: TE Connectivity Model 4203 Accelerometer blueprint

Having a closer look at its key features it is found what follows:

- ▶ $\pm 10 G$ measurement range.
- ▶ $-40^\circ C$ to $+125^\circ C$ operating temperature.
- ▶ Measures static and dynamic accelerations.

These features clearly respect the requirements for this application as nor the control system nor the porpoising by itself will lead to accelerations larger than $\pm 10 G$, the same can be said for the operating temperature range which in this application would never fall outside of the range guaranteed by the sensor's manufacturer. Moreover, here below is reported table 1.1 listing all the technical data of such sensor:

Environmental	
Operating Temperature range	-40°C to 125°C
Max shock acceleration	5000g
Electrical	
Input voltage	8 V _{DC} to 16 V _{DC}
Input current	< 30 mA max
Insulation resistance	> 100 MΩ min @ 100 V _{DC}
Mechanical	
Weight	< 60 grams max
Material	Anodized Aluminium
Performance	
Measurement range	±10 g
Sensitivity	200 mV/g ±10 %
Residual noise	0.5 mg rms
Linearity	±1 % FSO
Frequency response (bandwidth)	60±10 Hz

Table 1.1: Technical Data of the Accelerometer

- The POTENTIOMETER chosen is an AVIO RACE model DIA 13-XX in fig.1.11 and fig.1.12, which is a linear electronic device composed of a linear resistor and a movable slider that can be moved along the transistor. Its name comes from its dimensions, as the diameter of the potentiometer body is approximately 13 mm.



Figure 1.11: AVIO RACE DIA 13-XX potentiometer

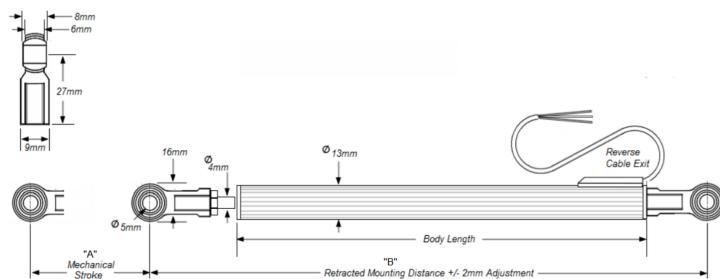


Figure 1.12: AVIO RACE DIA 13-XX potentiometer blueprint

The “XX” digit present inside the product name indicates the resistance value of the potentiometer, which can vary depending on the requirements of the specific application as can be seen in table 1.2.

Code	“A” Measurement range mm (± 0.5 mm)	“B” Retracted mounting distance mm	Resistance ($\pm 20\%$) k Ω
DIA 13-25	25	173	1.7
DIA 13-50	50	198	3.4
DIA 13-75	75	223	5
DIA 13-100	100	248	6.7
DIA 13-125	125	271	8.4
DIA 13-150	150	298	10
DIA 13-200	200	348	13.4
DIA 13-250	250	398	16.7

Table 1.2: Available versions of the potentiometer AVIO RACE DIA 13-XX

The precision of the DIA 13-XX linear potentiometer depends on the quality of the materials used in its construction, as well as the degree of accuracy required for the application in which it is used. This specific kind of potentiometer is based on “conductive plastic” technology, highly reliable and ideal for the following applications: car suspensions, motorcycle forks and shock absorbers. The technical data of such sensor are reported here below:

- ▶ Maximum supply voltage: 40 V DC
- ▶ Resolution: essentially infinite
- ▶ Repeatability: ≤ 0.01 mm
- ▶ Operational Speed: 10 m/s max
- ▶ Mechanical life: ≥ 25 million cycles
- ▶ Temperature range: $-40^\circ C$ to $+125^\circ C$
- ▶ Environmental sealing: IP65
- ▶ Independent linearity: $\leq \pm 0.5\%$
- ▶ Cable type: Raychem 55 A 24 AWG
- ▶ Cable length: 500 mm
- ▶ Housing: Aluminium
- ▶ Spherical bearing: $\varnothing 5$ mm
- ▶ Exit cable: STD → reverse cable exit, optional → forward cable exit

The version of the accelerometer that best fits the space available in the car is the one characterised by a stroke of 75 mm: consequently, its code-name is “DIA 13-75”.

As observed for the accelerometer, all technical specifications meet the requirements of the project in terms of precision, accuracy and operating temperature range. The maximum operational speed should not be exceeded in this application, too.

1.5 Organisation of the Manuscript

Chapter 1 is an introductory section that describes:

- the regulation changes that led to porpoising and their original purposes.
- how porpoising works and why it is an issue.
- why it may be beneficial to solve porpoising, how could be done and the methodological approach to the issue used for this work.

- the sensors used in the system.

Finally a list of most of the mathematical symbols used has been given to help the reader when looking at the equations describing the model, the state-space plant and the theoretical analysis of Chapter 2.

In Chapter 2, a mathematical formulation of the system is presented and analysed in depth. The analysis involves all the relevant properties for control purpose and outlines the specific properties of this system and their most relevant implications [7].

Then a solution for the control problem is proposed according to optimal control theory.

In Chapter 3, initially all the components of the simulator are described and both their working principle and motivations for introduction into the simulator are outlined to the reader.

Then, the control parameter tuning process is described and finally the control system results are shown in order to demonstrate the achievement of the prescribed goals in terms of both robustness performance in a wide range of different and proving external conditions.

In Chapter 4, conclusions about the system performance and limitations are discussed. Then some future possible relevant modifications and ways to improve the model are presented.

1.6 List of the Symbols

Symbol	Meaning	Symbol	Meaning
\mathbf{x}	State vector	m_s	Sprung mass
\mathbf{u}	Control vector	M	Total mass
\mathbf{w}	Exogenous vector	m_u	Unsprung mass
\mathbf{x}^*	State equilibrium vector	g	Gravity acceleration
\mathbf{u}^*	Control equilibrium vector	k_s	Suspension stiffness
\mathbf{w}^*	Exogenous equilibrium vector	c_s	Suspension damping
$\tilde{\mathbf{x}}$	Linearized state vector	k_t	Tire equivalent stiffness
$\tilde{\mathbf{u}}$	Linearized control vector	f_a	Hydraulic actuator force
$\tilde{\mathbf{w}}$	Linearized exogenous vector	F_z	Aerodynamic force
\mathbf{d}	Disturbance vector	h	Height delta between m_s COG and floor edge
ν_1	Accelerometer noise	C_L	Floor downforce coefficient
ν_2	Potentiometer noise	ρ	Air density
\mathbf{r}	Reference vector	v	Longitudinal relative airspeed
y_1	Sprung mass acceleration measurement	S	Floor surface
y_2	Suspension length measurement	S_W	Wing surface
\mathbf{e}	Controlled Output vector	C_{LW}	Wing downforce coefficient
$\tilde{\mathbf{e}}$	Linearized controlled output vector	z_{max}	Optimal floor ground height
$\tilde{\mathbf{y}}$	Linearized output vector	$C_{L_{max}}$	Maximum floor downforce coefficient
\mathbf{x}_0	State initial vector	z_s	Sprung mass COG height
n	State vector dimension	z_u	Unsprung mass COG height
p	Control vector dimension	l_s	Suspension initial length
q	Output vector dimension	l_t	Tire initial length
m	Regulated output vector dimension	z_{s0}	Initial sprung mass COG height
nd	Disturbance vector dimension	z_{u0}	Unsprung mass COG height
r	Reference vector dimension	v_{s0}	Initial sprung mass vertical speed
p	Control vector dimension	v_{u0}	Initial unsprung mass vertical speed
\mathbf{d}^*	Disturbance equilibrium vector	m_r	Reduced 2-body system mass
\mathbf{r}^*	Reference equilibrium vector	$C_{L_{stall}}$	Residual downforce coefficient after stall
ν^*	Noise equilibrium vector	z_{stall}	Floor-height where airflow stalls
\mathbf{y}^*	Output equilibrium vector	$z_{recovery}$	Floor-height when airflow get reattached
\mathbf{e}^*	Regulated output equilibrium vector	\mathbf{A}	\mathbf{A} matrix in Jordan canonical form coordinates
\mathbf{J}	Jordan canonical form	\mathbf{B}	\mathbf{B} matrix in Jordan canonical form coordinates
\mathbf{V}	Eigenvectors matrix associated to \mathbf{A}	\mathbf{A}_χ	Closed loop dynamics matrix
λ	Vector of eigenvalues associated to \mathbf{A}	\mathbf{T}_R	Kalman reachability trasformation matrix
\mathbf{J}	Real form of \mathbf{J}	\mathbf{T}_O	Kalman observability trasformation matrix
$\mathbf{\bar{V}}$	Real form of \mathbf{V}	\mathbf{z}	Vector of the LTI modes
\mathbf{T}	Inverse of \mathbf{V}	\mathbf{z}_0	Initial conditions of the modes
\mathbf{R}	Reachability matrix	\mathbf{O}	Observability matrix
σ_{pot}	Precision of potentiometer	σ_{acc}	Precision of accelerometer
η	Integral of regulated output	\mathbf{x}_e	Extended state vector
ϵ	State combination to penalize	\mathbf{x}_{e_0}	Extended state vector initial condition
J	Cost function	α	Parameter to induce instability in the model
\mathbf{Q}	ϵ weight inside \mathbf{J}	\mathbf{R}	Control cost
\mathbf{K}_S	State feedback matrix	\mathbf{K}_I	Integral action matrix
\mathbf{K}_O	Observer matrix	\mathbf{S}	Matrix that solve the optimal control problem

Table 1.3: List of the Symbols

Chapter 2

Main Body

2.1 Model and Problem Formulation

The control problem requires the formulation of a mathematical model able to describe as well as possible the behaviour of the real system. The F1 car affected by porpoising is studied at an individual wheel-level by means of a “*Quarter-Car Model*” [2] as in fig. 2.1.

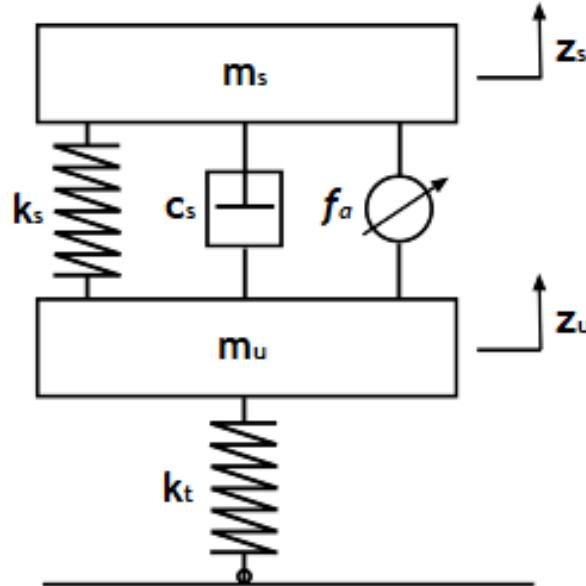


Figure 2.1: Active Suspension System: *Quarter-Car Model*

The model consists of: two masses, the upper one m_s and the lower one m_u representing, respectively, the *sprung* and *unsprung masses* acting on a single wheel, one suspension system and one tire. More specifically:

- The sprung mass of the vehicle is the mass supported by the suspension, while the unsprung mass is basically “the rest of the mass”, meaning the portion of mass not supported by the suspension system.
- The suspension system is composed of the two passive components, a spring and a damper (which are respectively characterised by a stiffness equal to k_s and a damping coefficient equal to c_s), and of an hydraulic actuator, responsible for the application of a time-varying force to the system that actively influences the ride-height.
- The tire which, for the sake of simplicity and without losing too much in accuracy, is assumed as a purely-spring model and characterised by a stiffness equal to k_t .

Moreover, each mass can move along the vertical axis, *z-axis*, producing in such a way a 2-DOF system for this specific case, where z_s and z_u represent, respectively, the vertical positions along the *z-axis* of the COG of sprung and unsprung masses.

Compared to other quarter-car models that can be found in literature [1] [2], the one proposed in this project assumes a perfectly flat-ground surface, this sounds reasonable since in those racetracks in which F1 Grand Prix are hosted the tarmac is usually well-maintained.

Therefore, the only loads acting on the system are, **divided by four**, the weight of the vehicle and the aerodynamic load that the single-seater can generate. The dynamics of this 2-DOF system can be described by writing its equations of motion, which are given as follows:

$$\begin{cases} m_s \cdot \ddot{z}_s = -m_s \cdot g - k_s \cdot (z_s - z_u - l_s) - c_s \cdot (\dot{z}_s - \dot{z}_u) + f_a - F_z \\ m_u \cdot \ddot{z}_u = -m_u \cdot g + k_s \cdot (z_s - z_u - l_s) + c_s \cdot (\dot{z}_s - \dot{z}_u) - f_a - k_t \cdot (z_u - l_t) \end{cases} \quad (2.1)$$

The numerical parameters chosen for the quarter-car model are reported below in table 2.1:

Parameter	Value	Unit
g	9.81	m/s^2
M	200	kg
m_s	180	kg
m_u	20	kg
k_s	165000	N/m
c_s	3500	$(N \cdot s)/m$
k_t	270000	N/m
l_s	0.1107	m
l_t	0.1073	m

Table 2.1: Quarter-car model numerical parameters

The masses, which are prescribed by the regulation [3], are the only parameters that are known with great accuracy. The other numerical parameters, as well as those in section 2.1.1, have been chosen, when in absence of available data, to best reproduce the observed oscillating behaviour of 2022 F1 cars. The value of k_s may seem especially high and this is probably due to the attempt to model the suspension as composed by a linear spring even if, in reality, during such condition it would presumably present a highly non-linear operating behaviour.

The *aerodynamic force model* for F_z will be described accurately in the next section, 2.1.1.

The main limitations of this model depend on the lack of data for what concerns the modelling of the aerodynamic force, the impossibility to properly control pitching-oscillations and the supposed linearity of the suspension. Another challenge encountered during the project is the lack of certain numerical data useful in the modelling stage of the components, as F1 teams do not release such information to the public and only certain data can be retrieved with a good degree of accuracy from the regulation [3].

The system of equations of motion 2.1 clearly shows the control f_a while the disturbance, represented by the longitudinal speed of the vehicle, is “hidden” inside the aerodynamic load term, F_z .

The goal of this control system is to keep the car as close as possible to the ride-height at which is associated the generation of the maximum downforce and maintaining it across the widest possible longitudinal speed range, to improve aerodynamic performance [6].

The sensors that are installed in our system, to achieve these goals, are described in section 1.4 and measure both vehicle vertical acceleration (accelerometer, 1.9) and suspension length stroke (potentiometer, 1.11). The sensor outputs, as it will be shown later on, can be described by the following set of equations:

$$\begin{cases} y_1 = \ddot{z}_s + g + \nu_1 \\ y_2 = z_s - z_u - l_s + \nu_2 \end{cases} \quad (2.2)$$

while, the regulated output can be defined as follows:

$$e_1 = z_s - r_1 \quad (2.3)$$

For control purpose, the non-linear system describing the dynamics of both masses involved in the plant (namely the sprung and unsprung masses, m_s and m_u respectively), the measured output \mathbf{y} and the regulated output \mathbf{e} need to be rewritten in the *State-Space Representation* [7].

More precisely, this kind of representation requires the definition of:

- a State Vector \mathbf{x} .
- a Control Vector \mathbf{u} .
- an Exogenous Vector \mathbf{w} .

The system dynamics in the state-space representation must be described by a set of linear first order differential equations, so the state vector needs to be designed in order to respect such constraint [7].

The vectors required for the state-space representation of the system are defined as:

$$\mathbf{x} = \begin{bmatrix} z_s - z_u \\ \dot{z}_s - \dot{z}_u \\ z_u \\ \dot{z}_u \end{bmatrix} \quad \mathbf{u} = [f_a] \quad \mathbf{w} = \begin{bmatrix} \mathbf{d} \\ \boldsymbol{\nu} \\ \mathbf{r} \end{bmatrix} \quad \mathbf{d} = [v] \quad \boldsymbol{\nu} = \begin{bmatrix} \nu_1 \\ \nu_2 \end{bmatrix} \quad \mathbf{r} = [z_{max} + h] \quad (2.4)$$

By substituting 2.4 into 2.1, 2.2, 2.3 respectively, it is possible to retrieve the following form of the plant model:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{w}) & \mathbf{x}(t_0) = \mathbf{x}_0 \\ \mathbf{y} = \mathbf{h}(\mathbf{x}, \mathbf{u}, \mathbf{w}) \\ \mathbf{e} = \mathbf{h}_e(\mathbf{x}, \mathbf{u}, \mathbf{w}) \end{cases} \quad (2.5)$$

Finally, the original system of two ordinary non-linear differential equations can be expressed, thanks to the substitutions just mentioned within the function $\mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{w})$, into a new system of four linear ordinary differential equations:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -k_s \cdot m_r \cdot (x_1 - l_s) - c_s \cdot m_r \cdot x_2 + f_a \cdot m_r - \frac{F_z}{m_s} \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = -g + \frac{k_s}{m_u} \cdot (x_1 - l_s) + \frac{c_s}{m_u} \cdot x_2 - \frac{f_a}{m_u} - \frac{k_t}{m_u} \cdot (x_3 - l_t) \end{cases} \quad (2.6)$$

where $m_r = \frac{m_u + m_s}{m_u \cdot m_s}$.

More than that, always thanks to the substitutions already mentioned, it is possible to retrieve, for the measured output function $\mathbf{h}(\mathbf{x}, \mathbf{u}, \mathbf{w})$, what follows:

$$\begin{cases} y_1 = -\frac{k_s}{m_s} \cdot (x_1 - l_s) - \frac{c_s}{m_s} \cdot x_2 + \frac{f_a}{m_s} - \frac{F_z}{m_s} + \nu_1 \\ y_2 = x_1 - l_s + \nu_2 \end{cases} \quad (2.7)$$

while, for the regulated output function $\mathbf{h}_e(\mathbf{x}, \mathbf{u}, \mathbf{w})$, it is obtained that:

$$e_1 = x_1 + x_3 - r_1 \quad (2.8)$$

The model written in the form 2.5 can be furthermore analysed according to what is prescribed by the control theory [7].

2.1.1 Aerodynamic Force

Since no experimental data describing the way in which the *lift* (“downforce”) *coefficient* C_L of the floor varies with its height (indicated with “ z ” in this paragraph) was available for 2022 F1 cars [4], its behaviour was modelled with the following analytical expression, which can approximate its expected real behaviour according to the relevant literature on ground effect cars [10]:

$$C_L(z) = z \cdot \frac{C_{L_{max}}}{z_{max}} \cdot e^{\frac{z_{max}-z}{z_{max}}} \quad (2.9)$$

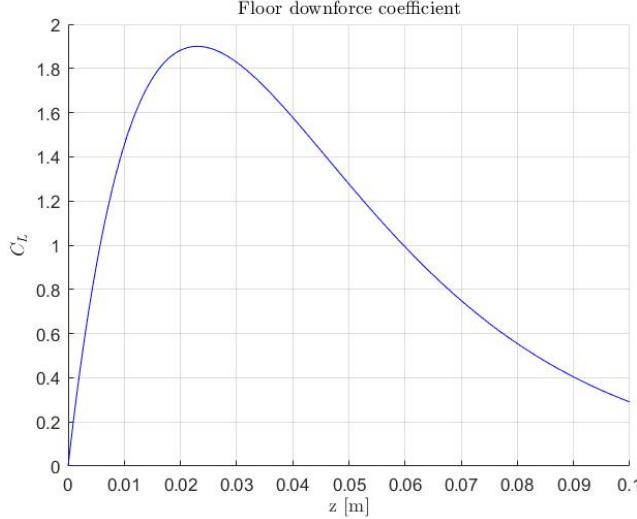


Figure 2.2: C_L variation with floor-height using model 2.9

Indeed, as observed experimentally [6], the downforce coefficient experiences a peak value in z_{max} and decreases gently if the floor-height increases, while decreases very rapidly if z approaches the ground, meaning for $z \rightarrow 0$, mimicking a stall.

The overall aerodynamic vertical load acting on the vehicle can be expressed as the summation of the contributions given by the floor and by both front and rear wings, whose downforce production is reasonably unaffected by the ride-height [6]. As a result of this consideration, it is possible to express that:

$$F_z(z) = \frac{1}{2} \cdot \rho \cdot v^2 \cdot [S_W \cdot C_{LW} + S \cdot C_L(z)] \quad (2.10)$$

which expression can be simply developed, by substituting 2.9 into 2.10, as follows:

$$F_z(z) = \frac{1}{2} \cdot \rho \cdot v^2 \cdot \left(S_W \cdot C_{LW} + S \cdot z \cdot \frac{C_{L_{max}}}{z_{max}} \cdot e^{\frac{z_{max}-z}{z_{max}}} \right) \quad (2.11)$$

This model of the downforce coefficient, in fig. 2.2, is fairly realistic, but lacks some key characteristics that can be seen from the experimental data in “*Ground effect aerodynamics of race cars*”, pictured in fig. 2.4, and that are crucial to induce porpoising in an F1 car:

- The stall occurrence, as can be seen in 2.2, is too gradual as compared to what happens in reality, namely a sort of staircase [6].
- During the stall recovery, the lift coefficient assumes values that are lower than its nominal ones, until the airflow boundary layer is fully attached once again to the floor: the “full-recovery” condition is regained once the ride-height reaches a sufficiently high value [10].

To improve the model and add this sort of *hysteresis behaviour*, the expression of F_z was changed to:

$$\begin{cases} F_z(z) = \frac{1}{2} \cdot \rho \cdot v^2 \cdot (S_W \cdot C_{LW} + S \cdot C_{L_{stall}}) & \text{for } z \leq z_{stall} \\ F_z(z) = a \cdot z^2 + b \cdot z + c & \text{for } z_{stall} < z < z_{recovery} \wedge \dot{z} > 0 \quad \text{after the stall of the floor} \\ F_z(z) = \frac{1}{2} \cdot \rho \cdot v^2 \cdot \left(S_W \cdot C_{LW} + S \cdot z \cdot \frac{C_{L_{max}}}{z_{max}} \cdot e^{\frac{z_{max}-z}{z_{max}}} \right) & \text{for all other conditions} \end{cases} \quad (2.12)$$

Comparing figures 2.3 and 2.4, it can be seen that this newly developed model shows a trend of C_L as function of the ride height which, at least from a qualitative point of view, matches quite well what has been found by Zhang et al. (2006) [10]. As detailed in section 2.1.2, this new addition to the aerodynamic model makes it more realistic and therefore allows to better reproduce the floor porpoising behaviour in high-speed conditions.

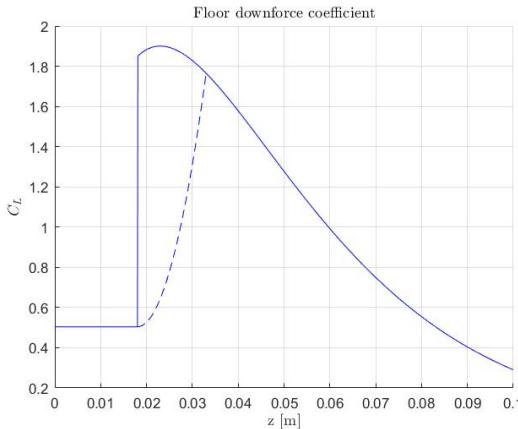


Figure 2.3: C_L variation with floor-height using model 2.12

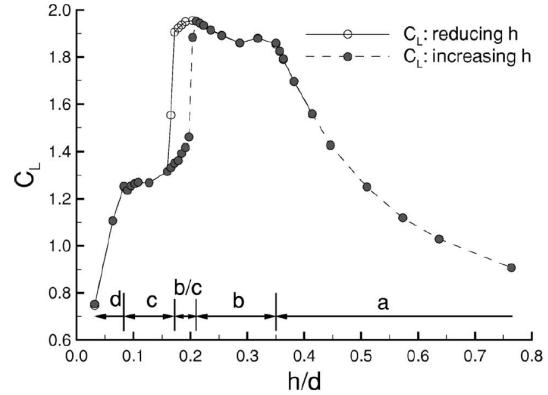


Figure 2.4: C_L variation with floor-height for ground effect cars according to “Ground effect aerodynamics of race cars” by Zhang et al. (2006)

The model for F_z in 2.12 is capable of reproducing the stalling behaviour experienced by F1 car’s floor in 2022: indeed, for $z < z_{stall}$, the downforce coefficient associated with the floor ($C_L(z)$) is made equal to $C_{L_{stall}} = 0.5$, and so from there on the contribution to the generation of the downforce from the floor is massively reduced [6].

More specifically, as can be seen in 2.3, the downforce coefficient during stall recovery is modelled as a parabolic function, whose coefficients a , b and c are determined by imposing that:

- When $z = z_{stall}$, it is assumed that the aerodynamic load $F_z(z)$ assumes value $F_z(z_{stall})$.
- When $z = z_{recovery}$, it is assumed that $F_z(z_{recovery}) = \frac{1}{2} \cdot \rho \cdot v^2 \cdot [S_W \cdot C_{LW} + S \cdot C_L(z_{recovery})]$.
- When $z = z_{stall}$, it is assumed that the derivative of the force function $F_z(z)$ is null, namely $\frac{dF_z(z_{stall})}{dz} = 0$.

Finally, by imposing these conditions, it is possible to derive a , b and c coefficients as follows:

$$\begin{cases} a = \frac{\frac{1}{2} \cdot \rho \cdot S \cdot v^2 \cdot \left(z_{recovery} \cdot \frac{C_{L_{max}}}{z_{max}} \cdot e^{\frac{z_{max} - z_{recovery}}{z_{max}}} - C_{L_{stall}} \right)}{z_{recovery}^2 + z_{stall}^2 - 2 \cdot z_{recovery} \cdot z_{stall}} \\ b = -2 \cdot a \cdot z_{stall} \\ c = \frac{1}{2} \cdot \rho \cdot v^2 \cdot \left(S_W \cdot C_{LW} + S \cdot z_{recovery} \cdot \frac{C_{L_{max}}}{z_{max}} \cdot e^{\frac{z_{max} - z_{recovery}}{z_{max}}} \right) + a \cdot (-z_{recovery}^2 + 2 \cdot z_{recovery} \cdot z_{stall}) \end{cases}$$

The last model of the aerodynamic force mentioned (2.12) is, in such a way, more complete, and this is the reason why it has been implemented in the simulator. For what concerns the “control-side”, the simplified model 2.11 was instead taken into consideration. Given this mismatch, great care was used to ensure that the closed-loop system never crossed z_{stall} , making the control system operating only within the domain of ride-heights where robustness can be guaranteed also in reality.

To implement the simplified aerodynamic force expressed by 2.11 into the plant equations (2.5), proper substitutions are required to be performed accordingly to 2.4 and, considering that in this section z represents the floor-height ($z = z_s - h$), it is retrieved that:

$$F_z(\mathbf{x}, \mathbf{d}) = \frac{1}{2} \cdot \rho \cdot v^2 \cdot \left[S_W \cdot C_{LW} + S \cdot (x_1 + x_3 - h) \cdot \frac{C_{L_{max}}}{z_{max}} \cdot e^{\frac{z_{max} - x_1 - x_3 + h}{z_{max}}} \right] \quad (2.13)$$

which can be implemented into both 2.6 and 2.7 sets of equations.

The numerical parameters that have been chosen for model 2.12 are listed inside the table 2.2 reported here below:

Parameter	Value	Unit
z_{max}	0.023	m
$C_{L_{max}}$	1.9	-
ρ	1.225	kg/m^3
$S_W \cdot C_{L_W}$	0.75	m^2
h	0.125	m
z_{stall}	0.018	m
$z_{recovery}$	0.033	m
S	0.4	m^2
$C_{L_{stall}}$	0.5	-

Table 2.2: Aerodynamic model numerical parameters

These parameters, once again due to the lack of official data, have been calculated by assuming, quite reasonably, that:

1. The downforce produced by the single-seater at 250 km/h is 18000 N.
2. The downforce production at the optimal ground-height is divided equally between the Venturi ducts and both front and rear wings.
3. The car is at sea-level.

The other data have been retrieved by integrating information coming from:

- The regulation [3].
- The data available in “*Aerodynamics of race cars*” by Katz Joseph [6].
- The data available in “*Ground effect aerodynamics of race cars*” by Xin Zhang et al. (2006) [10].

2.1.2 Comparison between open-loop simulation results and real world porpoising data

Since the model used for this application is much more simple than those present in the literature to describe porpoising [4], a great deal of emphasis has been posed on verifying that the outputs of this model, when fed with existing telemetry data, matched the behaviour observed in F1 single-seaters in the same scenario.

Therefore a comparison has been carried out between the behaviour of:

- the simplified model proposed in this work.
- the model used in “*Analyzing Porpoising on High Downforce Race Cars: Causes and Possible Setup Adjustments to Avoid It*” by Marco Gadola et al. (2022) [4].
- the relevant data released by the FIA during the 2022 F1 season.

It is to be mentioned that this comparison is only qualitative because, even though both this model and the one by Gadola et al. (2022) are tested using real telemetry data, those telemetry comes from different tracks; on top of that Gadola uses data from an F2 car so it does not aim at obtaining quantitative results matching those released by the FIA unlike this analysis.

The data released by F1 and the FIA during the Spanish Grand Prix in 2022 in fig. 2.5 indicate that at the beginning of the season, during F1 testing calendar, the cars that showed most aggressive porpoising, vertical accelerations had oscillations with amplitude in the order of $1 - 1.5 g$ and frequency $\approx 5 Hz$.

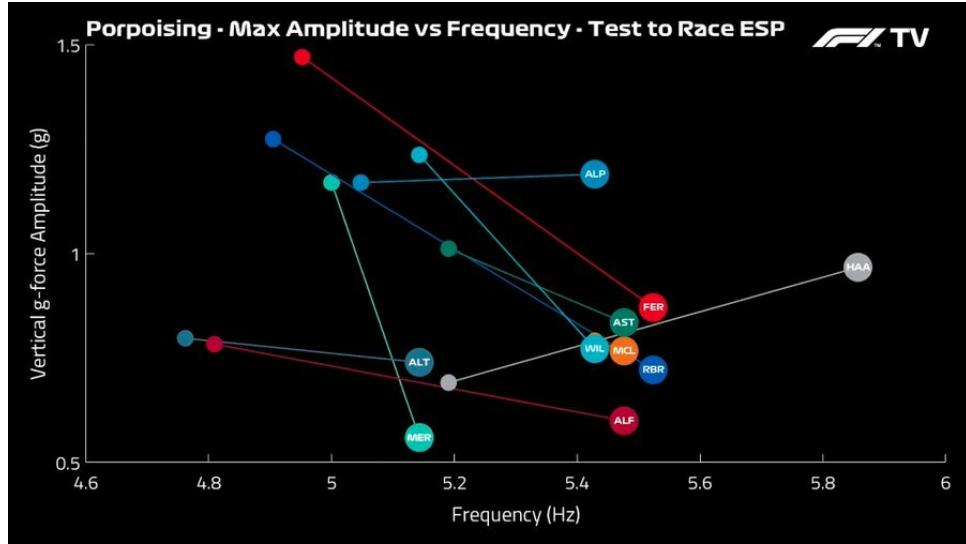


Figure 2.5: F1 porpoising amplitude vs frequency in Barcelona: a comparison between pre-season testing and Grand Prix weekend

Testing the model detailed in this work, using telemetry data from Barcelona F1 2022 qualifying session during one of the main straights, the model outputs, as can be seen in figures 2.6 and 2.7 a maximum vertical acceleration amplitude $\ddot{z}_{s_{max}} \approx 1.6 g$ and a frequency of circa $5.26 Hz$. These fall respectively a little above and in line with what observed in real F1 cars.

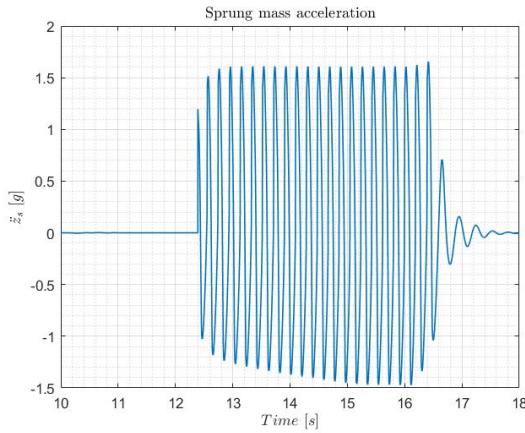


Figure 2.6: Sprung mass acceleration during porpoising using the OL model in 2.1

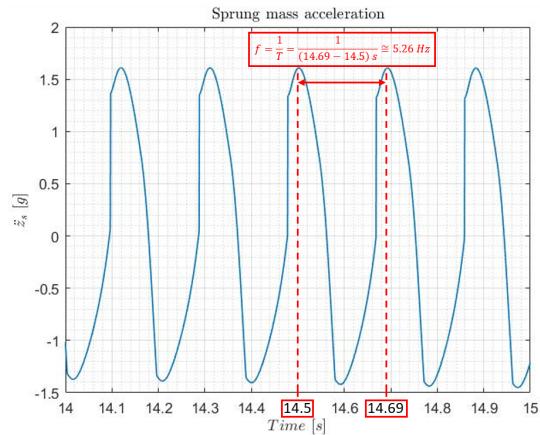


Figure 2.7: Zoomed-view of sprung mass acceleration during porpoising reported in 2.6

In the figures 2.8, 2.9 2.10 and 2.11 it is possible to compare the simulation done with the Open-Loop system described in 3.1.2 to those performed in “*Analyzing Porpoising on High Downforce Race Cars: Causes and Possible Setup Adjustments to Avoid It*” during a portions of different tracks with similar characteristics (straight sections in **Baku City Circuit** and **Bahrain International Circuit** respectively).

Since the comparison being made is between a quarter car model and an half car model, for the sake of brevity, only the front end behaviour of the vehicle in “*Analyzing Porpoising on High Downforce Race Cars: Causes and Possible Setup Adjustments to Avoid It*” is reported in the figures as the rear shows a similar behaviour at least from a qualitative point of view.

The main differences that can be observed in figs. 2.8 and 2.10 are:

- The speed required to initiate the oscillating behaviour is much lower is “*Analyzing Porpoising on High Downforce Race Cars: Causes and Possible Setup Adjustments to Avoid It*” compared to the model used for this work ($\approx 220 km/h$ vs $\approx 290 km/h$). It was chosen to have this difference to

be closer to what appeared to be the triggering speed for F1 cars, at least from the video evidence available.

- The oscillating frequency variation with speed is much more pronounced in the model from “*Analyzing Porpoising on High Downforce Race Cars: Causes and Possible Setup Adjustments to Avoid It*”.

Looking at the downforce production comparison in figures 2.9 and 2.11 no additional differences between the two model are shown. In terms of raw downforce production values the large difference is easily explained by the choice of Gadola et al. (2022) [4] to use F2 aerodynamic maps as those cars produce much less downforce than F1 cars.

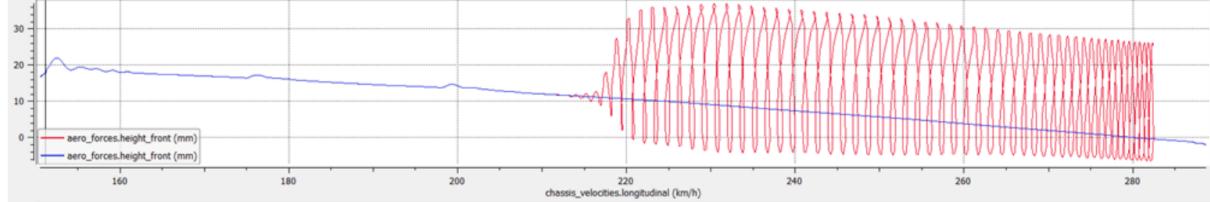


Figure 2.8: Floor-height from ground during porpoising using in “*Analyzing Porpoising on High Downforce Race Cars: Causes and Possible Setup Adjustments to Avoid It*” [4]

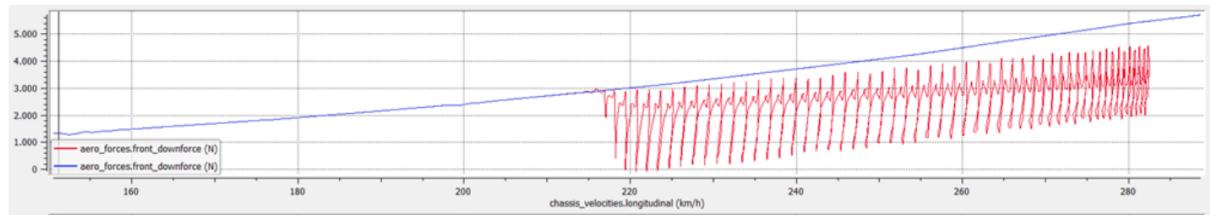


Figure 2.9: Downforce production during porpoising using in “*Analyzing Porpoising on High Downforce Race Cars: Causes and Possible Setup Adjustments to Avoid It*” [4]

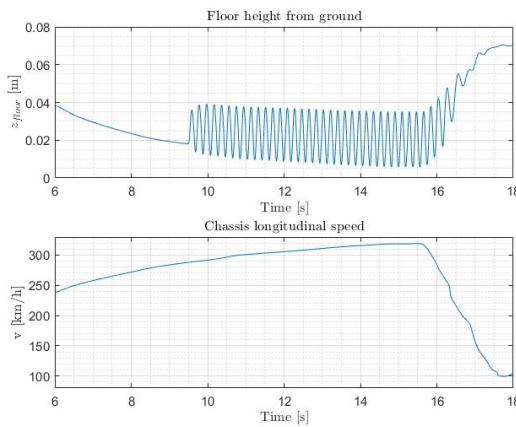


Figure 2.10: Floor-height from ground during porpoising using the OL model in 2.1

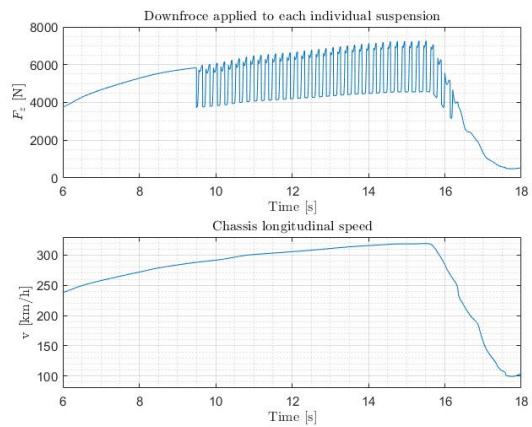


Figure 2.11: Downforce production during porpoising using the OL model in 2.1

From figures 2.10 and 2.11 it is also possible to observe how the breaking manoeuvre affects porpoising, and, similarly to what could be observed in video during the 2022 F1 season, the vehicle stops to oscillate almost immediately finding a new equilibrium position.

2.2 Model Analysis

The plant model equations, as written in 2.5 and detailed in 2.6, 2.7 and 2.8, can then be linearised and further analysed to better understand their properties and be able to design a robust and performant optimal control solution.

2.2.1 Linearisation

The model analysis starts with the linearisation of the system. Linearisation gives relevant information about how the system behaves in the proximity of the selected linearisation point. To obtain not only a linear system but also a **Linear Time Invariant (LTI)** system, the chosen linearization point needs to be an equilibrium point for the system (which, in such a specific case, is commonly named “Equilibrium Triplet”). The choice of the equilibrium triplet (\mathbf{x}^* , \mathbf{u}^* , \mathbf{w}^*) is arbitrary and done by the control designer but, as a rule of thumb, since the non-linear plant may behave much different than the linearised plant far from the linearisation point, it may be a good idea to select an equilibrium state configuration not too far from the desired working point of the plant.

Linearisation allows to approximate the non-linear differential equations that govern the behaviour of the system in 2.5 making use of linear differential equations [7].

Then it was possible to perform further analyses to understand whether the linearisation point is stable or unstable, which will be detailed in section 2.2.2.

To guarantee the validity of the linearisation procedure, which requires \mathbf{f} to be smooth and locally Lipschitz [7] in \mathbf{x} , the control system was designed using the “simplified” aerodynamic model 2.13 and not the more complex one 2.12 since the latter is non-continuous and, therefore non-Lipschitz. As previously mentioned, to ensure the stability of the control system designed, with this simplification of the problem great deal of emphasis was aimed at avoiding that the state \mathbf{x} could assume values that would make the simulator of the aerodynamic model diverge from the model implemented for control purposes.

The resulting set of linear ODEs is fairly accurate in the proximity of the equilibrium triplet (\mathbf{x}^* , \mathbf{u}^* , \mathbf{w}^*) and, unlike the original non-linear set of ODEs, can be solved. The selection of the equilibrium triplet has been made by imposing:

$$\begin{cases} \mathbf{0} = \mathbf{f}(\mathbf{x}^*, \mathbf{u}^*, \mathbf{w}^*) & \mathbf{x}^*(t_0) = \mathbf{x}^*_0 \\ \mathbf{y}^* = \mathbf{h}(\mathbf{x}^*, \mathbf{u}^*, \mathbf{w}^*) \\ \mathbf{0} = \mathbf{h}_e(\mathbf{x}^*, \mathbf{u}^*, \mathbf{w}^*) \end{cases} \quad (2.14)$$

and:

$$\mathbf{w}^* = \begin{bmatrix} \mathbf{d}^* \\ \boldsymbol{\nu}^* \\ \mathbf{r}^* \end{bmatrix} \quad \mathbf{d}^* = [v^*] \quad \boldsymbol{\nu}^* = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mathbf{r}^* = [z_{max} + h] \quad (2.15)$$

which results in:

$$\begin{cases} x_1^* = z_{max} + h - l_t + \frac{M \cdot g + \frac{1}{2} \cdot \rho \cdot v^{*2} \cdot (S_W \cdot C_{LW} + S \cdot C_{Lmax})}{k_t} \\ x_2^* = 0 \\ x_3^* = -\frac{M \cdot g + \frac{1}{2} \cdot \rho \cdot v^{*2} \cdot (S_W \cdot C_{LW} + S \cdot C_{Lmax})}{k_t} + l_t \\ x_4^* = 0 \\ f_a^* = -m_u \cdot g + k_s \cdot (x_1^* - l_s) - k_t \cdot (x_3^* - l_t) \end{cases} \quad (2.16)$$

The original set of ordinary non-linear differential equations of the plant can be linearised exploiting the Taylor Polynomial Expansion [7] writing:

$$\begin{aligned} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{w}) &= \mathbf{f}(\mathbf{x}^*, \mathbf{u}^*, \mathbf{w}^*) + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\substack{\mathbf{x}=\mathbf{x}^* \\ \mathbf{u}=\mathbf{u}^* \\ \mathbf{w}=\mathbf{w}^*}} (\mathbf{x} - \mathbf{x}^*) + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right|_{\substack{\mathbf{x}=\mathbf{x}^* \\ \mathbf{u}=\mathbf{u}^* \\ \mathbf{w}=\mathbf{w}^*}} (\mathbf{u} - \mathbf{u}^*) + \\ &+ \left. \frac{\partial \mathbf{f}}{\partial \mathbf{w}} \right|_{\substack{\mathbf{x}=\mathbf{x}^* \\ \mathbf{u}=\mathbf{u}^* \\ \mathbf{w}=\mathbf{w}^*}} (\mathbf{w} - \mathbf{w}^*) + \sigma(||(\mathbf{x} - \mathbf{x}^*)||^2, ||(\mathbf{u} - \mathbf{u}^*)||^2, ||(\mathbf{w} - \mathbf{w}^*)||^2) \end{aligned} \quad (2.17)$$

Using the equilibrium triplet $(\mathbf{x}^*, \mathbf{u}^*, \mathbf{w}^*)$ as a reference it is possible to define a set of linearised coordinates for: the state, the control and the exogenous as:

$$\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}^* \quad \tilde{\mathbf{u}} = \mathbf{u} - \mathbf{u}^* \quad \tilde{\mathbf{w}} = \mathbf{w} - \mathbf{w}^* \quad (2.18)$$

Substituting 2.18 into 2.17 one gets:

$$\dot{\tilde{\mathbf{x}}} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Big|_{\substack{\mathbf{x}=\mathbf{x}^* \\ \mathbf{u}=\mathbf{u}^* \\ \mathbf{w}=\mathbf{w}^*}} \tilde{\mathbf{x}} + \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \Big|_{\substack{\mathbf{x}=\mathbf{x}^* \\ \mathbf{u}=\mathbf{u}^* \\ \mathbf{w}=\mathbf{w}^*}} \tilde{\mathbf{u}} + \frac{\partial \mathbf{f}}{\partial \mathbf{w}} \Big|_{\substack{\mathbf{x}=\mathbf{x}^* \\ \mathbf{u}=\mathbf{u}^* \\ \mathbf{w}=\mathbf{w}^*}} \tilde{\mathbf{w}} + \sigma(\|\tilde{\mathbf{x}}\|^2, \|\tilde{\mathbf{u}}\|^2, \|\tilde{\mathbf{w}}\|^2) \quad (2.19)$$

and, similarly, it is possible to use Taylor series to approximate also the measured and regulated outputs [7] in the vicinity of the equilibrium triplet, indeed:

$$\begin{aligned} \tilde{\mathbf{y}} = \mathbf{y} - \mathbf{y}^* &= \mathbf{h}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}, \tilde{\mathbf{w}}) = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \Big|_{\substack{\mathbf{x}=\mathbf{x}^* \\ \mathbf{u}=\mathbf{u}^* \\ \mathbf{w}=\mathbf{w}^*}} \tilde{\mathbf{x}} + \frac{\partial \mathbf{h}}{\partial \mathbf{u}} \Big|_{\substack{\mathbf{x}=\mathbf{x}^* \\ \mathbf{u}=\mathbf{u}^* \\ \mathbf{w}=\mathbf{w}^*}} \tilde{\mathbf{u}} + \frac{\partial \mathbf{h}}{\partial \mathbf{w}} \Big|_{\substack{\mathbf{x}=\mathbf{x}^* \\ \mathbf{u}=\mathbf{u}^* \\ \mathbf{w}=\mathbf{w}^*}} \tilde{\mathbf{w}} + \sigma(\|\tilde{\mathbf{x}}\|^2, \|\tilde{\mathbf{u}}\|^2, \|\tilde{\mathbf{w}}\|^2) \\ \tilde{\mathbf{e}} = \mathbf{e} - \mathbf{e}^* &= \mathbf{h}_e(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}, \tilde{\mathbf{w}}) = \frac{\partial \mathbf{h}_e}{\partial \mathbf{x}} \Big|_{\substack{\mathbf{x}=\mathbf{x}^* \\ \mathbf{u}=\mathbf{u}^* \\ \mathbf{w}=\mathbf{w}^*}} \tilde{\mathbf{x}} + \frac{\partial \mathbf{h}_e}{\partial \mathbf{u}} \Big|_{\substack{\mathbf{x}=\mathbf{x}^* \\ \mathbf{u}=\mathbf{u}^* \\ \mathbf{w}=\mathbf{w}^*}} \tilde{\mathbf{u}} + \frac{\partial \mathbf{h}_e}{\partial \mathbf{w}} \Big|_{\substack{\mathbf{x}=\mathbf{x}^* \\ \mathbf{u}=\mathbf{u}^* \\ \mathbf{w}=\mathbf{w}^*}} \tilde{\mathbf{w}} + \sigma(\|\tilde{\mathbf{x}}\|^2, \|\tilde{\mathbf{u}}\|^2, \|\tilde{\mathbf{w}}\|^2) \end{aligned} \quad (2.20)$$

Then it is possible to approximate the equations in 2.19 and in 2.20 and further simplify them into:

$$\begin{aligned} \dot{\tilde{\mathbf{x}}} &= \mathbf{A}\tilde{\mathbf{x}} + \mathbf{B}_1\tilde{\mathbf{u}} + \mathbf{B}_2\tilde{\mathbf{w}} \\ \tilde{\mathbf{y}} &= \mathbf{C}\tilde{\mathbf{x}} + \mathbf{D}_1\tilde{\mathbf{u}} + \mathbf{D}_2\tilde{\mathbf{w}} \\ \tilde{\mathbf{e}} &= \mathbf{C}_e\tilde{\mathbf{x}} + \mathbf{D}_{1_e}\tilde{\mathbf{u}} + \mathbf{D}_{2_e}\tilde{\mathbf{w}} \end{aligned} \quad (2.21)$$

where:

$$\begin{aligned} \mathbf{A} &= \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Big|_{\substack{\mathbf{x}=\mathbf{x}^* \\ \mathbf{u}=\mathbf{u}^* \\ \mathbf{w}=\mathbf{w}^*}} & \mathbf{B}_1 &= \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \Big|_{\substack{\mathbf{x}=\mathbf{x}^* \\ \mathbf{u}=\mathbf{u}^* \\ \mathbf{w}=\mathbf{w}^*}} & \mathbf{B}_2 &= \frac{\partial \mathbf{f}}{\partial \mathbf{w}} \Big|_{\substack{\mathbf{x}=\mathbf{x}^* \\ \mathbf{u}=\mathbf{u}^* \\ \mathbf{w}=\mathbf{w}^*}} \\ \mathbf{C} &= \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \Big|_{\substack{\mathbf{x}=\mathbf{x}^* \\ \mathbf{u}=\mathbf{u}^* \\ \mathbf{w}=\mathbf{w}^*}} & \mathbf{D}_1 &= \frac{\partial \mathbf{h}}{\partial \mathbf{u}} \Big|_{\substack{\mathbf{x}=\mathbf{x}^* \\ \mathbf{u}=\mathbf{u}^* \\ \mathbf{w}=\mathbf{w}^*}} & \mathbf{D}_2 &= \frac{\partial \mathbf{h}}{\partial \mathbf{w}} \Big|_{\substack{\mathbf{x}=\mathbf{x}^* \\ \mathbf{u}=\mathbf{u}^* \\ \mathbf{w}=\mathbf{w}^*}} \\ \mathbf{C}_e &= \frac{\partial \mathbf{h}_e}{\partial \mathbf{x}} \Big|_{\substack{\mathbf{x}=\mathbf{x}^* \\ \mathbf{u}=\mathbf{u}^* \\ \mathbf{w}=\mathbf{w}^*}} & \mathbf{D}_{1_e} &= \frac{\partial \mathbf{h}_e}{\partial \mathbf{u}} \Big|_{\substack{\mathbf{x}=\mathbf{x}^* \\ \mathbf{u}=\mathbf{u}^* \\ \mathbf{w}=\mathbf{w}^*}} & \mathbf{D}_{2_e} &= \frac{\partial \mathbf{h}_e}{\partial \mathbf{w}} \Big|_{\substack{\mathbf{x}=\mathbf{x}^* \\ \mathbf{u}=\mathbf{u}^* \\ \mathbf{w}=\mathbf{w}^*}} \end{aligned} \quad (2.22)$$

and each matrix defined in 2.22 can be evaluated and the following are the results:

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ -k_s \cdot m_r & -c_s \cdot m_r & \frac{k_t}{m_u} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k_s}{m_u} & \frac{c_s}{m_u} & -\frac{k_t}{m_u} & 0 \end{bmatrix} & \mathbf{B}_1 &= \begin{bmatrix} 0 \\ m_r \\ 0 \\ -\frac{1}{m_u} \end{bmatrix} & \mathbf{B}_2 &= \begin{bmatrix} 0 \\ -\frac{\rho \cdot v^*}{m_s} \cdot (S_W \cdot C_{LW} + S \cdot C_{Lmax}) \\ 0 \\ 0 \end{bmatrix} & 0 & 0 & 0 \\ \mathbf{C} &= \begin{bmatrix} -\frac{k_s}{m_s} & -\frac{c_s}{m_s} & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} & \mathbf{D}_1 &= \begin{bmatrix} \frac{1}{m_s} \\ 0 \end{bmatrix} & \mathbf{D}_2 &= \begin{bmatrix} 0 \\ -\frac{\rho \cdot v^*}{m_s} \cdot (S_W \cdot C_{LW} + S \cdot C_{Lmax}) \\ 0 \\ 0 \end{bmatrix} & 1 & 0 & 0 \\ \mathbf{C}_e &= [1 \ 0 \ 1 \ 0] & \mathbf{D}_{1_e} &= [0] & \mathbf{D}_{2_e} &= [0 \ 0 \ 0 \ -1] & 0 & 1 & 0 \end{aligned} \quad (2.23)$$

2.2.2 Plant Self-Stability Analysis

To evaluate the linearised plant 2.21 asymptotic behaviour and assess whether it was self-stable or could drift away from equilibrium, if the initial condition did not match the equilibrium triplet, a self-stability analysis has been carried out on the system:

$$\dot{\tilde{\mathbf{x}}} = \mathbf{A}\tilde{\mathbf{x}} \quad (2.24)$$

To make the notation lighter from this chapter onward we will refer to the set of linearised coordinates neglecting the “tilde” symbol.

To study the asymptotic behaviour of a system, an eigenvalue analysis is required for the \mathbf{A} matrix. Computing the eigenvalues of \mathbf{A} one obtains:

$$\begin{cases} \lambda_1 = -93.77 + i \cdot 111.9 \\ \lambda_2 = -93.77 - i \cdot 111.9 \\ \lambda_3 = -3.45 + i \cdot 24.575 \\ \lambda_4 = -3.45 - i \cdot 24.575 \end{cases} \quad (2.25)$$

As can be seen two distinct couples of complex conjugates eigenvalues are obtained, as the number of distinct eigenvalues is equal to the rank of \mathbf{A} each eigenvalue must have an algebraic multiplicity of 1 and therefore a geometric multiplicity of 1 [7].

This implies that the Jordan canonical form associated to \mathbf{A} is diagonal and equal to:

$$\mathbf{J} = \begin{bmatrix} -93.77 + i \cdot 111.9 & 0 & 0 & 0 \\ 0 & -93.77 - i \cdot 111.9 & 0 & 0 \\ 0 & 0 & -3.45 + i \cdot 24.575 & 0 \\ 0 & 0 & 0 & -3.45 - i \cdot 24.575 \end{bmatrix}$$

The matrix \mathbf{V} , representing the compositions of the eigenvectors associated to each eigenvalue, as one could have predicted, is composed of two couples of complex conjugates eigenvectors:

$$\mathbf{V} = \begin{bmatrix} 0.0045 + i \cdot 0.0061 & 0.0045 - i \cdot 0.0061 & -0.0336 - i \cdot 0.0454 & -0.0336 + i \cdot 0.0454 \\ -1.1 - i \cdot 0.0693 & -1.1 + i \cdot 0.0693 & 1.23 - i \cdot 0.6704 & 1.23 + i \cdot 0.6704 \\ -0.0044 - i \cdot 0.0052 & -0.0044 + i \cdot 0.0052 & -0.0056 - i \cdot 0.0399 & -0.0056 + i \cdot 0.0399 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Then according to the procedure described in section 3.1.3 of “Analysis and Design of Control Laws for Advanced Driver-Assistance Systems: Theory and Applications” [7], the matrices $\mathbf{J}, \mathbf{V} \in \mathbb{C}^n$, have been reduced to their pure real valued equivalents obtaining:

$$\bar{\mathbf{J}} = \begin{bmatrix} -93.77 & 111.9 & 0 & 0 \\ -111.9 & -93.77 & 0 & 0 \\ 0 & 0 & -3.45 & 24.575 \\ 0 & 0 & -24.575 & -3.45 \end{bmatrix} \quad \bar{\mathbf{V}} = \begin{bmatrix} 0.0045 & 0.0061 & -0.0336 & -0.0454 \\ -1.1 & -0.0693 & 1.23 & -0.6704 \\ -0.0044 & -0.0052 & -0.0056 & -0.0399 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

Then it is possible to compute the dynamics of the LTI system:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{x}(t_0) = \mathbf{x}_0 \end{cases} \quad (2.26)$$

by solving the system of uncoupled first order differential equations that can be obtained by a change of coordinates from \mathbf{x} to \mathbf{z} where $\mathbf{z} = \mathbf{T}\mathbf{x}$ and $\mathbf{T} = \bar{\mathbf{V}}^{-1}$.

The set of coupled differential equations in 2.26 written in the new coordinates is:

$$\begin{cases} \dot{\mathbf{z}} = \bar{\mathbf{A}}\mathbf{z} + \bar{\mathbf{B}}\mathbf{u} \\ \mathbf{z}(t_0) = \mathbf{T}\mathbf{x}_0 \end{cases} \quad (2.27)$$

Once solved 2.27, and found $\mathbf{z}(t)$ the solution of 2.26 is obtained as $\mathbf{x}(t) = \bar{\mathbf{V}}\mathbf{z}(t)$ for all time instants after t_0 . The i^{th} component of the state \mathbf{x} can be expressed as a weighted mean of \mathbf{z} with weights $v_{i,j}$, as can be seen below:

$$x_i(t) = \sum_{j=1}^n \bar{v}_j z_j(t)$$

and the the weights of the modes $z_j(t)$ on the state components $x_i(t)$ can be represented through spider plots as can be seen in figures: 2.12, 2.13, 2.14 and 2.15.

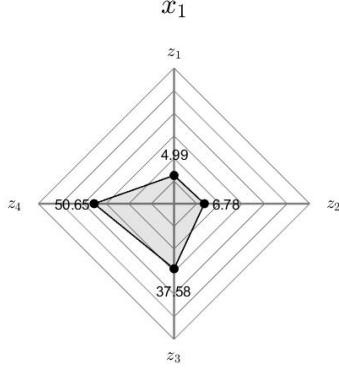


Figure 2.12: Weight of the uncoupled modes on the dynamics of x_1

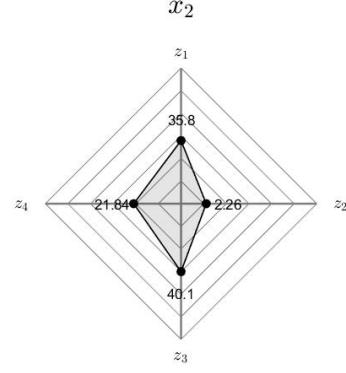


Figure 2.13: Weight of the uncoupled modes on the dynamics of x_2

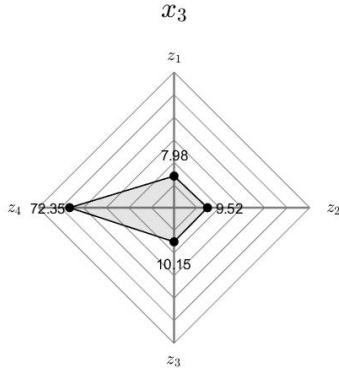


Figure 2.14: Weight of the uncoupled modes on the dynamics of x_3

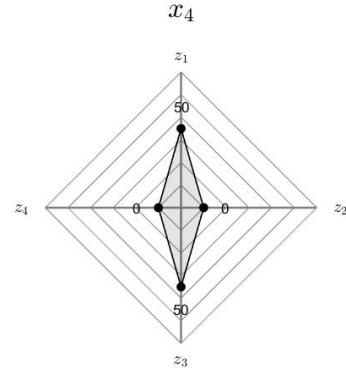


Figure 2.15: Weight of the uncoupled modes on the dynamics of x_4

The set of uncoupled first-order differential equations in 2.27 can be solved and for each component as follows:

$$\mathbf{z}_{i,j}(t) = \exp(\bar{\mathbf{J}}_{i,j}(t - t_0)) \cdot \mathbf{z}_{i,j}(t_0) + \int_t^{t_0} \exp(\bar{\mathbf{J}}_{i,j}(t - \tau)) \bar{\mathbf{B}}_{i,j} \mathbf{u}(\tau) d\tau$$

where two contributions can be defined:

$$\mathbf{z}_{i,j}^{free}(t) = \exp(\bar{\mathbf{J}}_{i,j}(t - t_0)) \cdot \mathbf{z}_{i,j}(t_0) \quad \mathbf{z}_{i,j}^{forced}(t) = \int_t^{t_0} \exp(\bar{\mathbf{J}}_{i,j}(t - \tau)) \bar{\mathbf{B}}_{i,j} \mathbf{u}(\tau) d\tau$$

$\mathbf{z}_{i,j}^{free}(t)$ refers to the free evolution of the system and depends only on the perturbation provided by the initial condition, while $\mathbf{z}_{i,j}^{forced}(t)$ refers to the forced time evolution of the system and depends only on the input time history [7].

To study the self-stability of the plant $\mathbf{u} = \mathbf{0}$ is assumed and so $\mathbf{z}_{i,j}(t)$ coincides with $\mathbf{z}_{i,j}^{free}(t)$ which can be calculated knowing $\bar{\mathbf{J}}$ and the initial conditions.

LTI system may have different behaviours depending on the eigenvalues associated to \mathbf{A} , the possible behaviours are:

$$\begin{cases} \mathbf{z}_{i,j}(t) \text{ exponentially convergent to 0} \iff \alpha_i < 0, q_{i,j} \geq 1 \\ \mathbf{z}_{i,j}(t) \text{ constant} \iff \alpha_i = 0, q_{i,j} = 1 \\ \mathbf{z}_{i,j}(t) \text{ polynomially divergent from } \mathbf{z}_{i,j}(t_0) \iff \alpha_i = 0, q_{i,j} > 1 \\ \mathbf{z}_{i,j}(t) \text{ exponentially divergent from } \mathbf{z}_{i,j}(t_0) \iff \alpha_i > 0, q_{i,j} \geq 1 \end{cases}$$

As can be derived from 2.25 and from the values of algebraic and geometric multiplicity associated to each eigenvalue, all the modes of this system are exponentially convergent to 0 and therefore this system is BIBS stable [7].

The time evolution of the state components, given an arbitrary perturbation of the initial condition, as could be expected by the eigenvalues being complex numbers with negative real parts, is a exponentially decreasing cosinusoidal function, depicted in fig. 2.16.

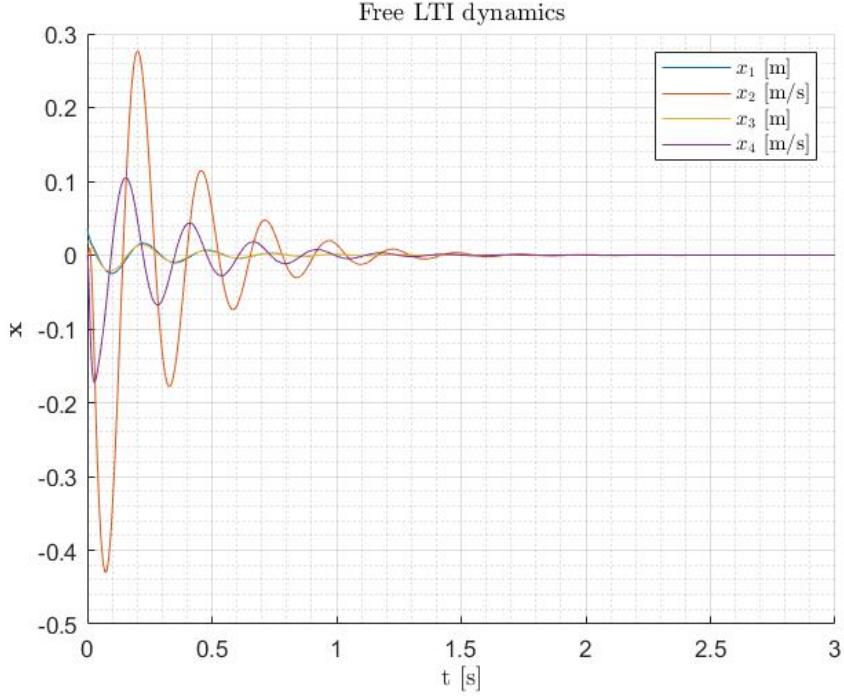


Figure 2.16: Free-evolution of the LTI system when perturbed with non-zero initial conditions

2.2.3 Reachability Study

To assess whether the whole possible set of states is reachable from the control law to bring back the system to equilibrium from whatever starting condition, or to assess whether there exist some limitations to the capability of the control law to steer some state sets back to equilibrium, a reachability study was performed on the system.

The first step consisted in computing the reachability matrix \mathbf{R} to assess the properties of the linearized system [7]. The matrix \mathbf{R} can be computed as:

$$\mathbf{R} = [\mathbf{B}_1 \quad \mathbf{AB}_1 \quad \mathbf{A}^2\mathbf{B}_1 \quad \mathbf{A}^3\mathbf{B}_1] \in \mathbb{R}^{n \times n \cdot p} \quad (2.28)$$

resulting in:

$$\mathbf{R} = \begin{bmatrix} 0 & m_r & -c_s \cdot m_r^2 & m_r^2 \cdot (c_s^2 \cdot m_r - k_s) - \frac{k_t}{m_u^2} \\ m_r & -c_s \cdot m_r^2 & m_r^2 \cdot (c_s^2 \cdot m_r - k_s) - \frac{k_t}{m_u^2} & m_r \cdot c_s \cdot (-c_s^2 \cdot m_r^3 + 2 \cdot k_s \cdot m_r^2 + 2 \cdot \frac{k_t}{m_u^2}) \\ 0 & -\frac{1}{m_u} & \frac{c_s \cdot m_r}{m_u} & \frac{1}{m_u} \cdot \left[m_r \cdot (-c_s^2 \cdot m_r + k_s) + \frac{k_t}{m_u} \right] \\ -\frac{1}{m_u} & \frac{c_s \cdot m_r}{m_u} & \frac{1}{m_u} \cdot \left[m_r \cdot (-c_s^2 \cdot m_r + k_s) + \frac{k_t}{m_u} \right] & \frac{c_s}{m_u} \cdot \left[m_r^2 \cdot (c_s^2 \cdot m_r - 2 \cdot k_s) - \frac{k_t}{m_u} \cdot (m_r + \frac{1}{m_u}) \right] \end{bmatrix}$$

According to control theory [7], the system is defined as “fully reachable” if and only if it has a “full rank” reachability matrix \mathbf{R} , meaning if it is composed by linearly independent vectors and so if its determinant is non-null.

Computing the rank of \mathbf{R} in MATLAB environment, it turns out that the system is fully reachable. This means that no *Kalman Reachability Decomposition* is needed to study the dynamics of our system since

\mathbf{R} being full rank results in: $\mathbf{T}_R^{-1} = \mathbf{I}$ and so in $\overline{\mathbf{A}}_{11} = \mathbf{A}$ and $\overline{\mathbf{A}}_{12}, \overline{\mathbf{A}}_{22} = \mathbf{0}$ [7].

This, accordingly to the theorem of existence of a stabilising state feedback [7], implies that:

$$\exists \mathbf{K}_S : \mathbf{A} + \mathbf{B}\mathbf{K}_S \text{ is Hurwitz.}$$

Due to the definition of the relative longitudinal airspeed v as a disturbance, an Integral Action contribution must be included in the control architecture in order to obtain a control able to steer the system to a zero error even under constant disturbance [7].

To do so, the previously studied system needs to be expanded as follows:

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\boldsymbol{\eta}} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C}_e & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\eta} \end{bmatrix} + \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{D}_{1e} \end{bmatrix} \mathbf{u} + \begin{bmatrix} \mathbf{B}_2 \\ \mathbf{D}_{2e} \end{bmatrix} \mathbf{w}$$

where it is possible to define:

$$\mathbf{A}_e = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C}_e & \mathbf{0} \end{bmatrix} \quad \mathbf{B}_{1e} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{D}_{1e} \end{bmatrix} \quad (2.29)$$

The new reachability matrix \mathbf{R}_e , for the expanded system, can therefore be obtained, similarly to 2.28, as:

$$\mathbf{R}_e = [\mathbf{B}_{1e} \quad \mathbf{A}_e \mathbf{B}_{1e} \quad \mathbf{A}_e^2 \mathbf{B}_{1e} \quad \mathbf{A}_e^3 \mathbf{B}_{1e} \quad \mathbf{A}_e^4 \mathbf{B}_{1e}] \in \mathbb{R}^{(n+m) \times (n+m) \cdot p}$$

resulting in:

$$\mathbf{R}_e = \left[\begin{array}{c|c} \mathbf{R} & \begin{array}{c} R_e(1,5) \\ R_e(2,5) \\ R_e(3,5) \\ R_e(4,5) \end{array} \\ \hline \begin{array}{cccc} 0 & 0 & m_r - \frac{1}{m_u} & m_r \cdot c_s \cdot (m_r - \frac{1}{m_u}) \end{array} & R_e(5,5) \end{array} \right]$$

where:

$$\begin{cases} R_e(1,5) = m_r \cdot c_s \cdot (-c_s^2 \cdot m_r^3 + 2 \cdot k_s \cdot m_r^2 + 2 \cdot \frac{k_t}{m_u^2}) \\ R_e(2,5) = \frac{m_r \cdot k_t}{m_u^2} \cdot (2 \cdot k_s - 3 \cdot c_s^2 \cdot m_r) + m_r^3 \cdot (c_s^4 \cdot m_r^2 - 3 \cdot c_s^2 \cdot k_s \cdot m_r + k_s^2) + \frac{k_t^2}{m_u^3} \\ R_e(3,5) = \frac{c_s}{m_u} \cdot \left[m_r^2 \cdot (c_s^2 \cdot m_r - 2 \cdot k_s) - \frac{k_t}{m_u} \cdot (m_r + \frac{1}{m_u}) \right] \\ R_e(4,5) = \frac{m_r \cdot k_t}{m_u^2} \cdot (c_s^2 \cdot m_r - k_s) - \frac{k_t}{m_u^3} \cdot (k_s + k_t - 2 \cdot c_s^2 \cdot m_r) - \frac{m_r^2}{m_u} \cdot (k_s^2 - 3 \cdot c_s^2 \cdot m_r \cdot k_s + c_s^4 \cdot m_r^2) \\ R_e(5,5) = m_r \cdot (m_r - \frac{1}{m_u}) \cdot (c_s^2 \cdot m_r - k_s) \end{cases}$$

Computing once again the rank of \mathbf{R}_e in MATLAB, it was retrieved that the reachability matrix \mathbf{R}_e of the extended system is “full rank”, meaning that the system is “fully reachable” and so that its determinant is non-null.

Then, it can be proven that if the extended system is fully reachable, then:

$$\exists [\mathbf{K}_S, \mathbf{K}_I] : \begin{bmatrix} \mathbf{A} + \mathbf{B}_1 \mathbf{K}_S & \mathbf{B}_1 \mathbf{K}_I \\ \mathbf{C}_e + \mathbf{D}_{1e} \mathbf{K}_S & \mathbf{D}_{1e} \mathbf{K}_I \end{bmatrix} \text{ is Hurwitz.} \quad (2.30)$$

This ensures that a control law such $\mathbf{u} = \mathbf{K}_S \mathbf{x} + \mathbf{K}_I \boldsymbol{\eta}$, composed of both state feedback and integral action, can:

- steer the regulated output to 0 in presence of a constant exogenous vector.
- lead to a bounded trajectory of the state for any bounded bounded exogenous vector [7].

Being both the original linearised plant 2.21 and the extended plant 2.29 fully reachable, they are also *stabilisable*, which is the minimum requirement to obtain the necessary aforementioned condition which guarantees the controlled plant to be BIBS [7].

2.2.4 Observability Analysis

In parallel to what has been done for the reachability analysis, another crucial point was to assess which set of states of the system can be estimated from the plant at all times and from whatever starting condition and which ones do not. More in depth, the first step of this analysis consisted in computing the observability matrix \mathbf{O} to assess the properties of the linearised system in 2.21 [7].

To do so, the observability matrix \mathbf{O} can be computed as:

$$\mathbf{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \mathbf{CA}^3 \end{bmatrix} \in \mathbb{R}^{n \cdot q \times n} \quad (2.31)$$

which for this system results in:

$$\mathbf{O} = \begin{bmatrix} -\frac{k_s}{m_s} & -\frac{c_s}{m_s} & 0 & 0 \\ 1 & 0 & 0 & 0 \\ \frac{c_s \cdot k_s \cdot m_r}{m_s} & \frac{c_s^2 \cdot m_r - k_s}{m_s} & \frac{-c_s \cdot k_t}{m_s \cdot m_u} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{m_r \cdot k_s}{m_s} \cdot (k_s - c_s^2 \cdot m_r) & \frac{m_r \cdot c_s}{m_s} \cdot (2 \cdot k_s - c_s^2 \cdot m_r) & \frac{k_t}{m_s \cdot m_u} \cdot (c_s^2 \cdot m_r - k_s) & -\frac{c_s \cdot k_t}{m_s \cdot m_u} \\ -k_s \cdot m_r & -c_s \cdot m_r & \frac{k_t}{m_u} & 0 \\ O(7, 1) & O(7, 2) & O(7, 3) & O(7, 4) \\ c_s \cdot k_s \cdot m_r^2 & m_r \cdot (c_s^2 \cdot m_r - k_s) & -\frac{c_s \cdot k_t \cdot m_r}{m_u} & \frac{k_t}{m_u} \end{bmatrix}$$

where:

$$\begin{cases} O(7, 1) = -\frac{c_s \cdot m_r^2 \cdot k_s}{m_s} \cdot (2 \cdot k_s - c_s^2 \cdot m_r + \frac{k_t}{m^2 \cdot u \cdot m_r^2}) \\ O(7, 2) = \frac{1}{m_s} \cdot (k_s^2 \cdot m_r - 3 \cdot c_s^2 \cdot m_r^2 \cdot k_s + c_s^4 \cdot m_r^3 - \frac{c_s^2 \cdot k_t}{m_u^2}) \\ O(7, 3) = \frac{k_t \cdot c_s}{m_s \cdot m_u} \cdot (\frac{k_t}{m_u} - c_s^2 \cdot m_r^2 + 2 \cdot k_s \cdot m_r) \\ O(7, 4) = -\frac{k_t}{m_s \cdot m_u} \cdot (c_s^2 \cdot m_r - k_s) \end{cases}$$

Accordingly to control theory [7], the system is defined as “fully observable” if and only if its kernel is null (namely $\ker \mathbf{O} = \mathbf{0}$), meaning that \mathbf{O} is full rank.

Computing the dimension of the kernel of \mathbf{O} in MATLAB environment, it turns out that the system is fully observable. This means that, as it was for the reachability analysis, no *Kalman Decomposition of Observability* is needed to study this system since $\ker \mathbf{O} = \mathbf{0}$ results in: $\mathbf{T}_O^{-1} = \mathbf{I}$ and so in $\bar{\mathbf{A}}_{22} = \mathbf{A}$ and $\bar{\mathbf{A}}_{11}, \bar{\mathbf{A}}_{12} = \mathbf{0}$.

This, according to the theorem of existence of a stabilising output feedback [7], implies that:

$$\exists \mathbf{K}_O : \mathbf{A} - \mathbf{K}_O \mathbf{C} \text{ is Hurwitz.} \quad (2.32)$$

Moreover, this means that it is possible to design a BIBS stable dynamic system which provides an estimation of the state dynamics [7].

A further analysis has been made regarding the observability of the system removing either the potentiometer or the accelerometer, which led to two modified versions of C that can be defined as follows:

$$\mathbf{C}_{pot} = [1 \ 0 \ 0 \ 0] \quad \mathbf{C}_{acc} = \left[-\frac{k_s}{m_s} \ -\frac{c_s}{m_s} \ 0 \ 0 \right]$$

which, following 2.31 leads to the following observability matrices:

$$\mathbf{O}_{pot} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -k_s \cdot m_r & -c_s \cdot m_r & \frac{k_t}{m_u} & 0 \\ c_s \cdot k_s \cdot m_r^2 & m_r \cdot (c_s^2 \cdot m_r - k_s) & -\frac{c_s \cdot k_t \cdot m_r}{m_u} & \frac{k_t}{m_u} \end{bmatrix}$$

$$\mathbf{O}_{acc} = \begin{bmatrix} -\frac{k_s}{m_s} & -\frac{c_s}{m_s} & 0 & 0 \\ \frac{c_s \cdot k_s \cdot m_r}{m_s} & \frac{c_s^2 \cdot m_r - k_s}{m_s} & -\frac{c_s \cdot k_t}{m_s \cdot m_u} & 0 \\ \frac{m_r \cdot k_s}{m_s} \cdot (k_s - c_s^2 \cdot m_r) & -\frac{c_s \cdot m_r}{m_s} \cdot (2 \cdot k_s - c_s^2 \cdot m_r) & \frac{k_t}{m_s \cdot m_u} \cdot (m_r \cdot c_s^2 - k_s) & -\frac{c_s \cdot k_t}{m_s \cdot m_u} \\ O_{acc}(4, 1) & O_{acc}(4, 2) & O_{acc}(4, 3) & O_{acc}(4, 4) \end{bmatrix}$$

where:

$$\begin{cases} O_{acc}(4, 1) = \frac{c_s \cdot k_s}{m_s} \cdot (\frac{k_t}{m_u} - 2 \cdot k_s \cdot m_r^2 - c_s^2 \cdot m_r^3) \\ O_{acc}(4, 2) = -\frac{k_s^2 \cdot m_r + c_s^4 \cdot m_r^3 - \frac{c_s \cdot k_t}{m_u^2} - 3 \cdot k_s \cdot c_s^2 \cdot m_r^2}{m_s} \\ O_{acc}(4, 3) = \frac{c_s \cdot k_t}{m_s \cdot m_u} \cdot (\frac{k_t}{m_u} + 2 \cdot k_s \cdot m_r - c_s \cdot k_s \cdot m_r^2) \\ O_{acc}(4, 4) = \frac{k_t}{m_s \cdot m_u} \cdot (c_s^2 \cdot m_r - k_s) \end{cases}$$

By calculating the kernel of both the alternative systems, without either one of the sensors, it can be seen that removing one of the two sensors does not change the system properties, as both alternatives are still fully observable.

In most applications this could be seen as a push to remove either the accelerometer or the potentiometer to save money, while retaining a functional control; for this application, though, we already have both sensors installed for other purposes, so it is best to retain both to improve the observer performance, as both installed sensors are expensive, highly precise and reliable.

2.3 Proposed Solution

The plant is controlled through two different control contributions: the State Feedback and the Integral Action (a Feed Forward contribution is not needed since the reference is constant [7]).

To generate the plant control input \mathbf{u} , since it is not possible to know the exact state \mathbf{x} at each time instant, as \mathbf{x} is an internal plant variable, an Observer is needed in order to generate a state estimation $\hat{\mathbf{x}}$ that can be sent as input to the controller.

To track the state \mathbf{x} , the Observer need as input both linearised control vector $\tilde{\mathbf{u}}$ and linearised measurement vector $\tilde{\mathbf{y}}$, with the latter being different from real parameters due to limited sensor precision and accuracy, which cannot be avoided even in high-quality sensors and which is modelled by the sensor's noise vector ν .

The more accurate the state estimation is and the better the controller will be able to steer the regulated output to the reference no matter the magnitude of the disturbance.

As proved in 2.2.3 and in 2.2.4 the system is fully reachable and fully observable, leading to the possibility of modifying the closed-loop dynamic, expressed by the \mathbf{A}_x matrix, through $\mathbf{K}_S, \mathbf{K}_I$ and \mathbf{K}_O in order to make it Hurwitz [7]. From a physical point of view, the controller will generate a control input \mathbf{u} that, by acting on the system, will make it BIBS [7].

The aforementioned properties of the analysed system allow to compute the closed-loop dynamics as follows:

$$\mathbf{A}_x = \begin{bmatrix} \mathbf{A} + \mathbf{B}_1 \mathbf{K}_S & \mathbf{B}_1 \mathbf{K}_I & \mathbf{B}_1 \mathbf{K}_S \\ \mathbf{C}_e + \mathbf{D}_{e_1} \mathbf{K}_S & \mathbf{D}_{e_1} \mathbf{K}_I & \mathbf{D}_{e_1} \mathbf{K}_S \\ \mathbf{A}_O + \mathbf{K}_O \mathbf{C} - \mathbf{A} + \mathbf{M} \mathbf{K}_S & \mathbf{M} \mathbf{K}_I & \mathbf{A}_O + \mathbf{M} \mathbf{K}_S \end{bmatrix}$$

where:

$$\mathbf{A}_O = \mathbf{A} - \mathbf{K}_O \mathbf{C} \quad \mathbf{M} = \mathbf{B}_O + \mathbf{K}_O \mathbf{D}_1 - \mathbf{B}_1$$

Now, since $\mathbf{B}_O = \mathbf{B}_1 - \mathbf{K}_O \mathbf{D}_1$, one gets $\mathbf{M} = \mathbf{0}$, allowing to simplify \mathbf{A}_x as

$$\mathbf{A}_x = \left[\begin{array}{cc|c} \mathbf{A} + \mathbf{B}_1 \mathbf{K}_S & \mathbf{B}_1 \mathbf{K}_I & \mathbf{B}_1 \mathbf{K}_S \\ \mathbf{C}_e + \mathbf{D}_{e_1} \mathbf{K}_S & \mathbf{D}_{e_1} \mathbf{K}_I & \mathbf{D}_{e_1} \mathbf{K}_S \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{A} - \mathbf{K}_O \mathbf{C} \end{array} \right]$$

which, thanks to 2.32 and 2.30, can be proved to be Hurwitz [7].

More specifically, in the case in which the exogenous vector \mathbf{w} is kept constant, also the control goal G2 is satisfied, meaning that the reference would be tracked perfectly at time infinite [7].

State Feedback, Integral Action and Observer mathematical characterisation and tuning will be presented in following subsections.

2.3.1 State Feedback and Integral Action

By designing the control action it is possible to guarantee the satisfaction of goal G2: more specifically, the design of the state feedback needs to be done alongside the one of the integral action on the extended-state plant described in 2.29.

The control law that is obtained as a result of the application of Optimal Control Theory to the extended-state plant is:

$$\mathbf{u} = \mathbf{K}_S \mathbf{x} + \mathbf{K}_I \boldsymbol{\eta} \tag{2.33}$$

where \mathbf{K}_S and \mathbf{K}_I must be obtained by solving the optimal control problem on the extended plant as calculating firstly \mathbf{K}_S on the non-extended-state and then $\mathbf{K}_S, \mathbf{K}_I$ on the extended plant, would lead to

two different values for the state feedback with only the latter providing the required control properties [7].

To solve the optimal control problem and obtain \mathbf{K}_S , \mathbf{K}_I one must follow these steps:

1. Define a modified extended plant:

$$\dot{\mathbf{x}}_e = (\mathbf{A}_e + \alpha \mathbf{I})\mathbf{x}_e + \mathbf{B}_e \mathbf{u}$$

where $\alpha \geq 0$ is a parameter that increasing in magnitude makes the controller look at the extended plant as less stable than it actually is by fictionally increasing its eigenvalues [7].

2. Define an arbitrary error vector $\boldsymbol{\varepsilon}$:

$$\boldsymbol{\varepsilon} = \mathbf{C}_\varepsilon \mathbf{x}_e + \mathbf{D}_\varepsilon \mathbf{u}$$

which denotes all the linear combination of states and controls the designer want to penalise.

\mathbf{C}_ε , \mathbf{D}_ε are matrices expressing the relation between the errors vector an the extended state and control vector and are not necessarily connected to \mathbf{C} , \mathbf{D} [7].

For this system $\boldsymbol{\varepsilon}$ has been defined as:

$$\begin{cases} \varepsilon_1 := \text{vehicle ride - height} \\ \varepsilon_2 := \text{tire COG height} \\ \varepsilon_3 := \text{sprung mass acceleration} \\ \varepsilon_4 := \text{tire COG acceleration} \\ \varepsilon_5 := \text{vehicle ride - height error integral} \end{cases}$$

Consequently the matrices \mathbf{C}_ε , \mathbf{D}_ε are defined as:

$$\mathbf{C}_\varepsilon = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -\frac{k_s}{m_s} & -\frac{c_s}{m_s} & 0 & 0 & 0 \\ \frac{k_s}{m_u} & \frac{c_s}{m_u} & -\frac{k_t}{m_u} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{D}_\varepsilon = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{m_{\ddot{s}}} \\ -\frac{1}{m_u} \\ 0 \end{bmatrix}$$

3. Define a cost function $J > 0$ as:

$$J = \int_{t_0}^{\infty} \boldsymbol{\varepsilon}^T \mathbf{Q} \boldsymbol{\varepsilon} + \mathbf{u}^T \mathbf{R} \mathbf{u} dt$$

where \mathbf{Q} is a symmetric semi-positive definite matrix defining the cost to pay for $\boldsymbol{\varepsilon}(t) \neq 0$ and, similarly, \mathbf{R} is a symmetric positive definite matrix constituting the cost to pay if $\mathbf{u}(t) \neq 0$ [7].

\mathbf{Q} , \mathbf{R} are defined by inverting respectively:

$$\mathbf{Q}^{-1} = q \begin{bmatrix} \varepsilon_{1_{max}}^2 & 0 & 0 & 0 & 0 \\ 0 & \varepsilon_{2_{max}}^2 & 0 & 0 & 0 \\ 0 & 0 & \varepsilon_{3_{max}}^2 & 0 & 0 \\ 0 & 0 & 0 & \varepsilon_{4_{max}}^2 & 0 \\ 0 & 0 & 0 & 0 & \varepsilon_{5_{max}}^2 \end{bmatrix} \quad \mathbf{R}^{-1} = p [u_{max}^2]$$

where $\varepsilon_{i_{max}}$ expresses the inverse of the cost of the i^{th} component of $\boldsymbol{\varepsilon}$ and u_{max} expresses the inverse of the cost of the control action.

Therefore, low values of $\varepsilon_{i_{max}}$ lead to very high cost for that component of $\boldsymbol{\varepsilon}$ not being equal to 0, while very high values of u_{max} lead to extremely low cost of the control action [7]: this is the reason why, when pushing u_{max} to very high values, one should verify that the real actuator is capable of delivering that kind of control action.

Once the optimal control problem is defined, the control matrices can be obtained from the solution of the *Algebraic Riccati Equation* (ARE), which is given as follows:

$$\begin{aligned} & \mathbf{S} \mathbf{B}_e \bar{\mathbf{R}}^{-1} \mathbf{B}_e^T \mathbf{S} - \mathbf{S}(\mathbf{A}_e + \alpha \mathbf{I} - \mathbf{B}_e \bar{\mathbf{R}}^{-1} \mathbf{D}_\varepsilon^T \mathbf{Q} \mathbf{C}_\varepsilon) + \\ & - (\mathbf{A}_e + \alpha \mathbf{I} - \mathbf{B}_e \bar{\mathbf{R}}^{-1} \mathbf{D}_\varepsilon^T \mathbf{Q} \mathbf{C}_\varepsilon)^T \mathbf{S} - \mathbf{C}_\varepsilon^T \mathbf{Q} (\mathbf{I} - \mathbf{D}_\varepsilon \bar{\mathbf{R}}^{-1} \mathbf{D}_\varepsilon^T \mathbf{Q}) \mathbf{C}_\varepsilon = 0 \end{aligned} \quad (2.34)$$

where $\bar{\mathbf{R}}^{-1} = \mathbf{D}_\varepsilon^T \mathbf{Q} \mathbf{D}_\varepsilon + \mathbf{R}$ and \mathbf{S} is the solution of the ARE from which one can obtain the control law that minimises J and therefore the optimal values for \mathbf{K}_S , \mathbf{K}_I [7].

The optimal control solution \mathbf{S} can be obtained in Matlab through the function *icare*, whose syntax is:

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

where:

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

While the stability of the control law obtained from 2.34 is guaranteed by the aforementioned properties of the system and its optimal nature by the aforementioned theory [7], its performance can be influenced by appropriately tuning the control matrices \mathbf{Q} and \mathbf{R} as discussed in section 3.2.

2.3.2 Observer

Since it is not possible to know exactly the state \mathbf{x} at each time instant, we need to be able to estimate it somehow to be capable to compute the regulated output and therefore to obtain the control law at each time instant.

As the measurements, coming from the sensors installed in the plant, may be incomplete or too inaccurate or even both, when trying to estimate the state of the plant the observer combines the measurements and the linearised model to be able to fulfil this requirement.

Starting from 2.21, which is a fully observable LTI system, it is possible to define an *Identity Observer* [7] as follows:

$$\begin{cases} \dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}_1\mathbf{u} & \hat{\mathbf{x}}(t_0) = \hat{\mathbf{x}}_0 \\ \hat{\mathbf{y}} = \mathbf{C}\hat{\mathbf{x}} + \mathbf{D}_1\mathbf{u} \end{cases}$$

In addition to this, it is then possible to define also the estimation error $\mathbf{e}_x = \hat{\mathbf{x}} - \mathbf{x}$, whose dynamics can be obtained as follows:

$$\begin{cases} \dot{\mathbf{e}}_x = \mathbf{A}\mathbf{e}_x - \mathbf{B}_2\mathbf{w} & \mathbf{e}_x(t_0) = \mathbf{e}_{x0} \\ \hat{\mathbf{y}} - \mathbf{y} = \mathbf{C}\hat{\mathbf{x}} + \mathbf{D}_2\mathbf{w} \end{cases} \quad (2.35)$$

Then, by exploiting the concept of duality [7], the dual model associated to 2.35 can be defined as:

$$\begin{cases} \dot{\chi} = \mathbf{A}_d\chi - \mathbf{B}_d\nu & \chi(t_f) = \chi_f \\ \mu = \mathbf{C}_d\chi + \mathbf{D}_d\nu \end{cases} \quad (2.36)$$

where:

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

Being 2.21 fully observable, due to the definition of duality, the dual plant 2.36 is fully reachable, and this enables the possibility of finding the optimal observer matrix \mathbf{K}_O by deploying on the dual plant the same strategy used in section 2.3.1 [7].

The only relevant difference is that $\boldsymbol{\mu}$, \mathbf{C}_d , \mathbf{D}_d are defined in the plant and not by the designer unlike $\boldsymbol{\varepsilon}$, \mathbf{C}_ε , \mathbf{D}_ε .

So by making the same steps of section 2.3.1:

1. The modified plant is:

$$\dot{\chi} = (\mathbf{A}_d + \alpha_d \mathbf{I})\chi - \mathbf{B}_d\nu$$

with α_d being the dual of α .

2. The cost function is defined as:

$$J_d = \int_{t_0}^{\infty} \boldsymbol{\mu}^T \mathbf{Q}_d \boldsymbol{\mu} + \boldsymbol{\nu}^T \mathbf{R}_d \boldsymbol{\nu} dt$$

where \mathbf{Q}_d , \mathbf{R}_d have the same properties of \mathbf{Q} , \mathbf{R} of section 2.3.1. The matrices \mathbf{Q}_d , \mathbf{R}_d are defined as:

$$\mathbf{Q}_d = \begin{bmatrix} w_{max}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{R}_d = \begin{bmatrix} \sigma_{acc}^2 & 0 \\ 0 & \sigma_{pot}^2 \end{bmatrix}$$

where w_{max} represents an indication of the maximum possible disturbance and σ_{acc} , σ_{pot} indication of the magnitude of the error in the measurements performed.

Therefore, high magnitude of w_{max} indicates that the linearised model is less reliable than the measurement in the estimation of the state, and the same can be said for low values of σ_{acc} and σ_{pot} [7].

More in general, the optimal gain for the observer represents a trade-off between the reliability of the linearised plant and the reliability of the measurement coming from the sensors: this compromise needs to be tuned by the designer depending on the sensors' properties and the possible disturbance values during operation.

For this model the sensors' standard deviations have been derived from the characteristics reported in section 1.4.

Once again, the solution of the dual optimal control problem can be found by solving the ARE in 2.37 through the Matlab function *icare*, whose syntax is detailed in section 2.3.1.

$$\begin{aligned} & \mathbf{S}\mathbf{B}_d\bar{\mathbf{R}}_d^{-1}\mathbf{B}_d^T\mathbf{S} - \mathbf{S}(\mathbf{A}_d + \alpha_d\mathbf{I} - \mathbf{B}_d\bar{\mathbf{R}}_d^{-1}\mathbf{D}_d^T\mathbf{Q}_d\mathbf{C}_d) + \\ & - (\mathbf{A}_d + \alpha_d\mathbf{I} - \mathbf{B}_d\bar{\mathbf{R}}_d^{-1}\mathbf{D}_d^T\mathbf{Q}_d\mathbf{C}_d)^T\mathbf{S} - \mathbf{C}_d^T\mathbf{Q}_d(\mathbf{I} - \mathbf{D}_d\bar{\mathbf{R}}_d^{-1}\mathbf{D}_d^T\mathbf{Q}_d)\mathbf{C}_d = 0 \end{aligned} \quad (2.37)$$

From the outputs of *icare*, one gets $\mathbf{K}_m = -\mathbf{K}_{S_d}$ and, due to duality [7], one can obtain $\mathbf{K}_O = -\mathbf{K}_{S_d}^T$ [7], which implies in the end that $\mathbf{K}_O = \mathbf{K}_m^T$.

Chapter 3

Application

3.1 Simulator Description

The next step of the project consists in the description of the system simulator implemented on Simulink, and shown in fig. 3.1, which was the software chosen to design and perform simulations to:

- tune the control system parameters.
- prove the control system robustness in all possible operating conditions.
- verify that the control performance met the design targets.

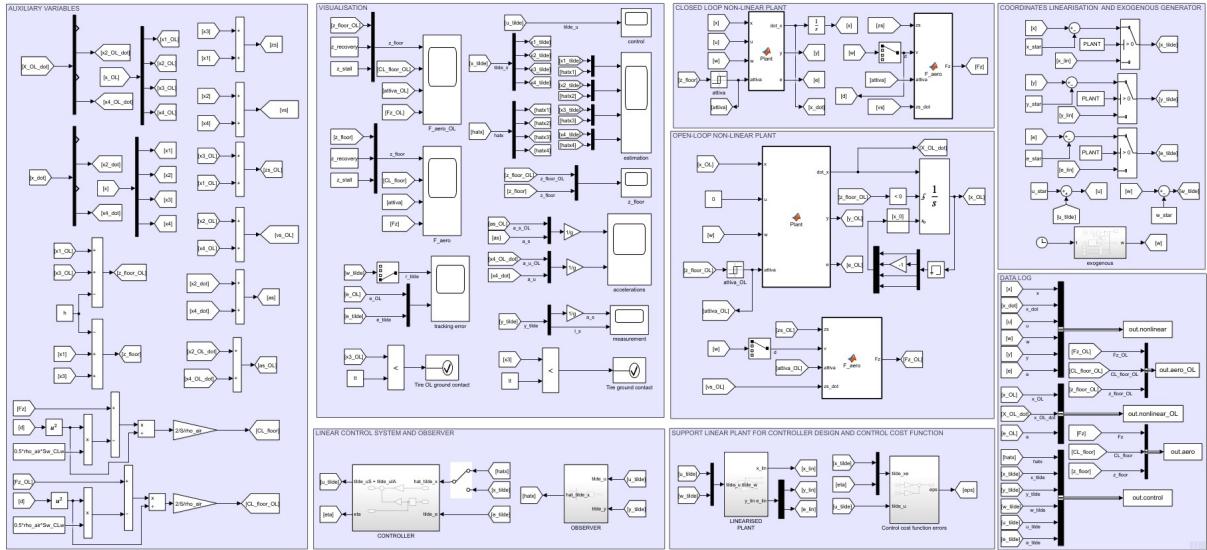


Figure 3.1: Full-Simulator scheme

The simulator description that will be reported in the following pages of this paper is provided to help better understanding the logic behind its functioning. Each subsystem will be discussed in detail but for the “Auxiliary variables” which simply calculate some quantities that are required by other subsystems. The in-depth analysis and description of all the simulations performed in order to study the system and validate the assumptions previously proposed are presented in Section 3.2.

For the sake of clearness, in this section, the symbol “tilde” will be made always explicit in order to differentiate the various parameters and avoiding, so, misunderstandings.

3.1.1 Non-linear Plant, Closed-Loop

Figure 3.2 represents the non-linear plant in Closed-Loop implemented in Simulink through a Matlab function, called “*Plant*”, reported in Appendix 5.2, whose inputs are:

1. the state vector \mathbf{x} .
2. the control vector \mathbf{u} .
3. the exogenous vector \mathbf{w} .
4. the “attiva” parameter that describes if the fluid boundary layer is stalled or attached. This variable is necessary to implement 2.12.

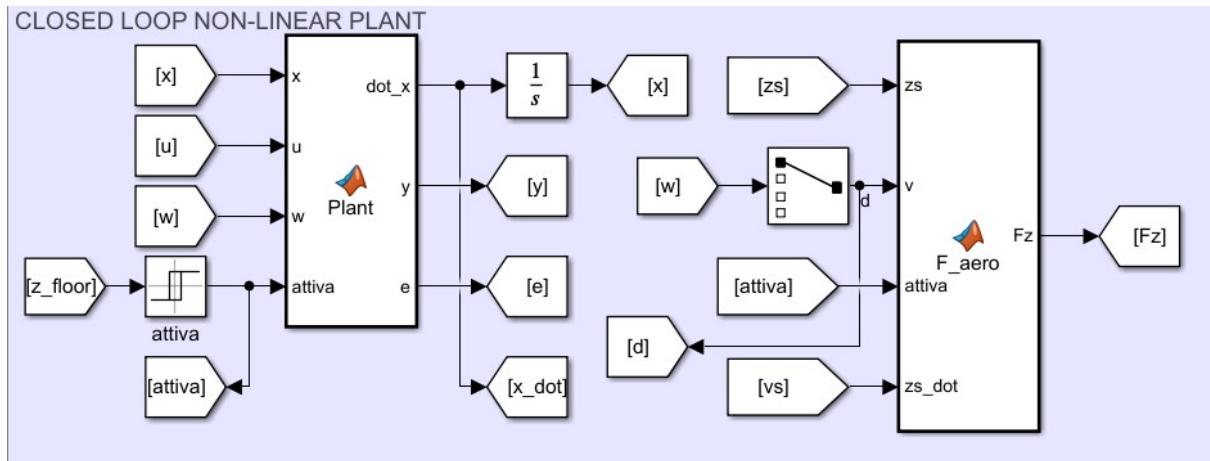


Figure 3.2: Closed-Loop Non-Linear Plant subsystem

The exogenous vector \mathbf{w} is composed of three distinct components: disturbance d , noise ν and reference r , and its generation is described in more detail in subsection 3.1.5.

The outputs of the *Plant* function are:

1. the state vector derivative $\dot{\mathbf{x}}$
2. the output vector \mathbf{y}
3. the regulated output vector \mathbf{e}

As can be seen in fig. 3.2 there is also reported another Matlab function, F_{aero} , which implements the aerodynamic load F_z within the model as expressed in 2.12 and through which it is possible to plot its time history.

3.1.2 Non-Linear Plant, Open-Loop

The non-linear plant in Open-Loop configuration, in fig. 3.3, for the most part, works in the same way of the system described in the subsection 3.1.1.

There are two main differences:

1. The control vector \mathbf{u} is set to **0**.
2. Since there is no control to set and influence the ride height, there is the risk that at high-speed the car may hit the ground. Therefore an hybrid model simulating a perfectly elastic impact of the sprung mass with the ground has been added to avoid the unreal scenario of the car floor located below ground surface.

The choice of a perfectly elastic collision has been made as both materials involved, carbon fibre reinforced plastic (CFRP) and tarmac, are very stiff and therefore this assumption makes the model simpler but not so far from reality.

There are no other differences between the two models.

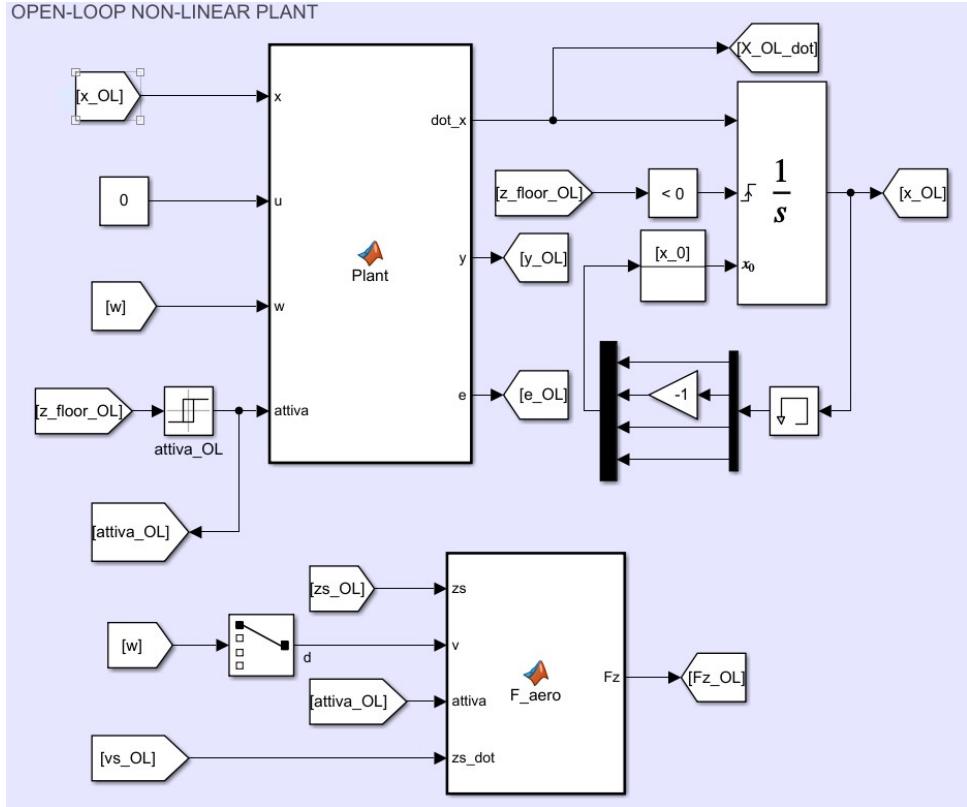


Figure 3.3: Open-Loop Non-Linear Plant subsystem

3.1.3 Linear Plant

The implementation of the linear plant is shown in fig. 3.4 and makes use of the Simulink block “*State-Space*”. By definition, all the input coordinates are linearised.

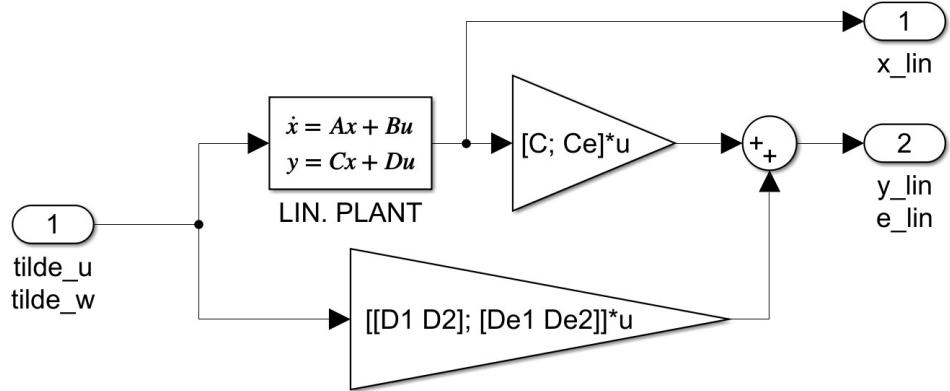


Figure 3.4: Linearised Plant subsystem

The input is a columnar vector stacking both linearised control and linearised exogenous. Then, the *State-Space* block outputs:

- the state linearised coordinates \tilde{x} .
- the sensors linearised signal \tilde{y} .
- the linearised regulated output \tilde{e} .

The matrices in the *State-Space* block, marked below by the subscript “block”, are defined as follows:

$$\mathbf{A}_{block} = \mathbf{A} \quad \mathbf{B}_{block} = [\mathbf{B}_1 \ \mathbf{B}_2] \quad \mathbf{C}_{block} = \mathbf{I} \wedge \mathbf{C}_{block} \in \mathbb{R}^{n \times n} \quad \mathbf{D}_{block} = \mathbf{0} \wedge \mathbf{D}_{block} \in \mathbb{R}^{n \times (p+r)}$$

More specifically, this block helps during the tuning stage, namely when the attention is focused on the research of a control law that works on the linearised plant at first and then, secondly, on the non-linear one.

3.1.4 Coordinates Linearisation

The *Coordinates Linearisation* block aims at obtaining the linearised coordinates.

This can be achieved in two possible ways, one that works on the linearised plant while the other on the non-linear plant:

1. In the latter case it subtracts the equilibrium triplet $(\mathbf{x}^*, \mathbf{u}^*, \mathbf{w}^*)$ coordinates from the non-linear ones to obtain the linearised coordinates.
2. It equals the linearised coordinates calculated by the subsystem described in section 3.1.3 to the “tilde” coordinates.

This operation, as can be seen in fig. 3.5, is performed on:

- State coordinates.
- Output vector.
- Regulated output error.
- Control vector.
- Exogenous Vector

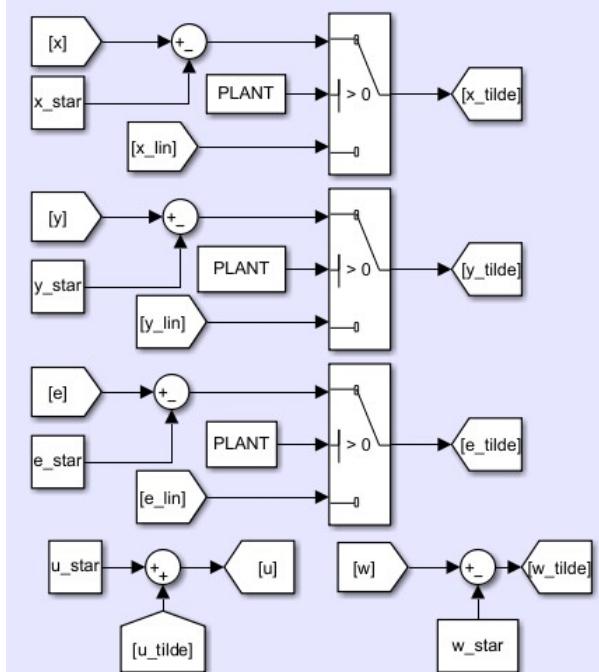


Figure 3.5: Coordinates linearisation subsystem

The variable *PLANT* initialised by the Matlab file required to launch the simulator allows to switch between the two possible operation modes.

3.1.5 Exogenous vector Generator

The exogenous vector \mathbf{w} , as previously mentioned, is composed of:

- the disturbance \mathbf{d} .
- the noise vector ν .
- the reference vector \mathbf{r} .

and it is generated from the subsystem in fig. 3.6.

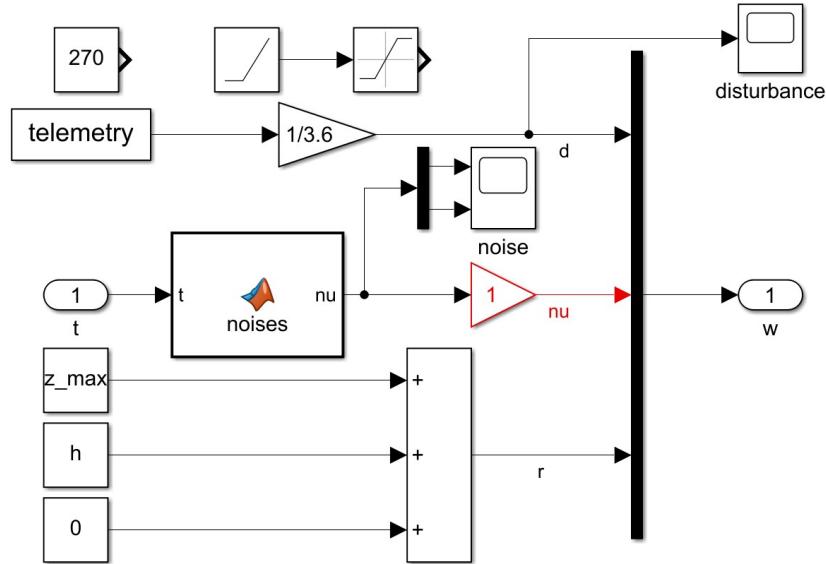


Figure 3.6: Exogenous generator subsystem

The disturbance vector is fed from the timeseries block “telemetry”, which contains the longitudinal speed values at each time instant in several scenarios: the scenario can be selected in the launch Matlab code needed to initialise the variables present in this Simulink scheme and reported in Appendix 5.1.

The scenarios are real world telemetry data from tracks where many F1 cars have experienced severe porpoising issue during the F1 2022 season. This data is available to the public and has been sampled and then interpolated to obtain a speed value for each time instant of the simulation.

One additional scenario simulates the longitudinal speed of an F1 car during maximum acceleration condition: this scenario is particularly relevant since it represented the most challenging possible condition for this system. The data implemented is accurate until 220 km/h and after that the car acceleration has been overestimated to ensure robustness in this specific case, which coincides with the worst possible case scenario.

The tracks chosen for the scenarios are:

- The **Baku City Circuit** in Azerbaijan.
- The **Circuit de Barcelona-Catalunya** in Spain.
- The **Jeddah Corniche Circuit** in Saudi Arabia.
- The **Albert Park Circuit** in Melbourne, Australia.

For each track two disturbance scenarios have been generated: one for the whole lap telemetry and another one featuring a high-speed straight which may be subjected to porpoising.

It is also possible to perform simulations with simpler disturbances given by constant acceleration from a certain speed to another, or at constant speed.

The noise vector ν is generated by the Matlab function “noises”, reported in Appendix 5.3, which, at each time instant, outputs the noise produced by the accelerometer and the potentiometer as a band limited white noise with amplitude proportional to the values of precision reported in 1.4 for each sensor. Noise may be switched-off by setting the gain in fig. 3.6 to 0.

The reference is set as a constant as per 2.4.

3.1.6 Non-linear Control Implementation

In order to obtain the control law \tilde{u} it is necessary to input to the controller, represented in figures 3.7 and 3.8, both system state and regulated output coming from the Observer and from the Closed-Loop Non-linear Plant, described in subsections 3.1.7 and 3.1.1 respectively.

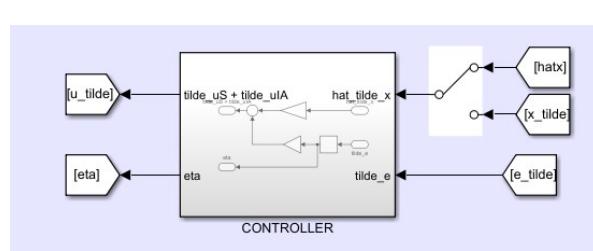


Figure 3.7: Inputs and outputs of the controller

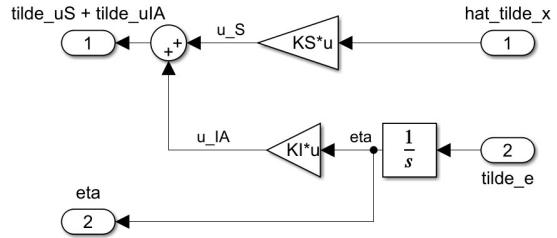


Figure 3.8: Controller block

The $\hat{\tilde{x}}$ vector is multiplied by the \mathbf{K}_S matrix to obtain the state feedback control law \tilde{u}_S , while the η vector, obtained by integrating \tilde{e} , is multiplied by the \mathbf{K}_I matrix to obtain the integral action control law \tilde{u}_{IA} .

Afterwards the two contributions to the control law are added up to find the linearised control law \tilde{u} . The latter is then summed-up with \mathbf{u}^* in order to obtain the actual control law \mathbf{u} .

To be able to tune the control matrices \mathbf{K}_S and \mathbf{K}_I , independently from the Observer performance, and to take advantage of the separation principle [7] it is possible, as can be seen in fig. 3.7, to use a *manual switch* to feed the subsystem with \tilde{x} instead of $\hat{\tilde{x}}$, by assuming to be capable to know the exact state at each time instant.

3.1.7 Observer Block

The *Observer* block shown in fig. 3.9 requires as input the linearised control vector \tilde{u} and the linearised sensor output vector \tilde{y} to build the estimated state vector $\hat{\tilde{x}}$.

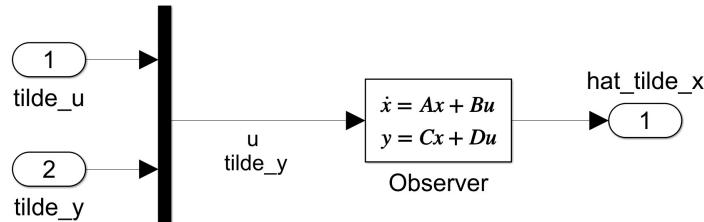


Figure 3.9: Observer subsystem

It has been modelled through the use of a *State-Space* block, in which:

- \mathbf{A} is the matrix \mathbf{A}_O .
- \mathbf{B} is the matrix $[\mathbf{B}_O \quad \mathbf{K}_O]$.

- \mathbf{C} is the matrix \mathbf{C}_O .
- $\mathbf{D} \in \mathbb{R}^{n \times (p+q)}$ is a null-matrix, namely a matrix composed by all zeros as components.

where:

$$\begin{cases} \mathbf{A}_O = \mathbf{A} - \mathbf{K}_O \mathbf{C} \\ \mathbf{B}_O = \mathbf{B}_1 - \mathbf{K}_O \mathbf{D}_1 \end{cases}$$

and \mathbf{C}_O is an identity matrix of dimension n .

3.1.8 Cost Function Block

The *Cost Function* block, represented in fig. 3.10, is meant to calculate the cost function errors for the non-linear plant. This block requires inputting the linearised extended-state $\tilde{\mathbf{x}}_e$ and control input $\tilde{\mathbf{u}}$.

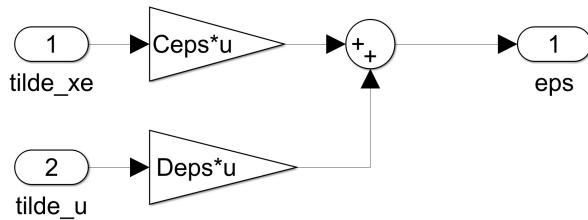


Figure 3.10: Cost function subsystem

The vector $\boldsymbol{\varepsilon}$ is directly obtained by the following definition:

$$\boldsymbol{\varepsilon} = \mathbf{C}_\varepsilon \tilde{\mathbf{x}}_e + \mathbf{D}_\varepsilon \tilde{\mathbf{u}}$$

where \mathbf{C}_ε and \mathbf{D}_ε express the dependence of the arbitrarily designed errors vector $\boldsymbol{\varepsilon}$ with the extended-state $\tilde{\mathbf{x}}_e$ components and with the control vector $\tilde{\mathbf{u}}$, respectively.

3.1.9 Visualisation Block

The *Visualisation* block (fig. 3.11) allows to monitor the time-history of the most relevant system quantities during the simulation and tuning stages.

This block also contains two “assertion” blocks that stop the simulation as soon as the tire is lifted from ground, this is done as the model does not account for this situation as it was considered very unlikely due to the high-downforce levels a racecar is subjected to.

The many scopes present allow to monitor:

- The linearised control $\tilde{\mathbf{u}}$.
- The observer performance by comparing $\hat{\mathbf{x}}$ to $\tilde{\mathbf{x}}$ component by component.
- The racecar ride-height in both Open-Loop and Closed-Loop.
- The sprung and unsprung mass accelerations in both Open-Loop and Closed-Loop configurations.
- The linearised output vector $\tilde{\mathbf{y}}$.
- The linearised reference $\tilde{\mathbf{r}}$ and tracking error $\tilde{\mathbf{e}}$ in both Open-Loop and Closed-Loop configurations.
- The aerodynamic situation of the vehicle plotting: F_z , C_L , the boundary-layer state and the ride-height in both Open-Loop and Closed-Loop configurations.

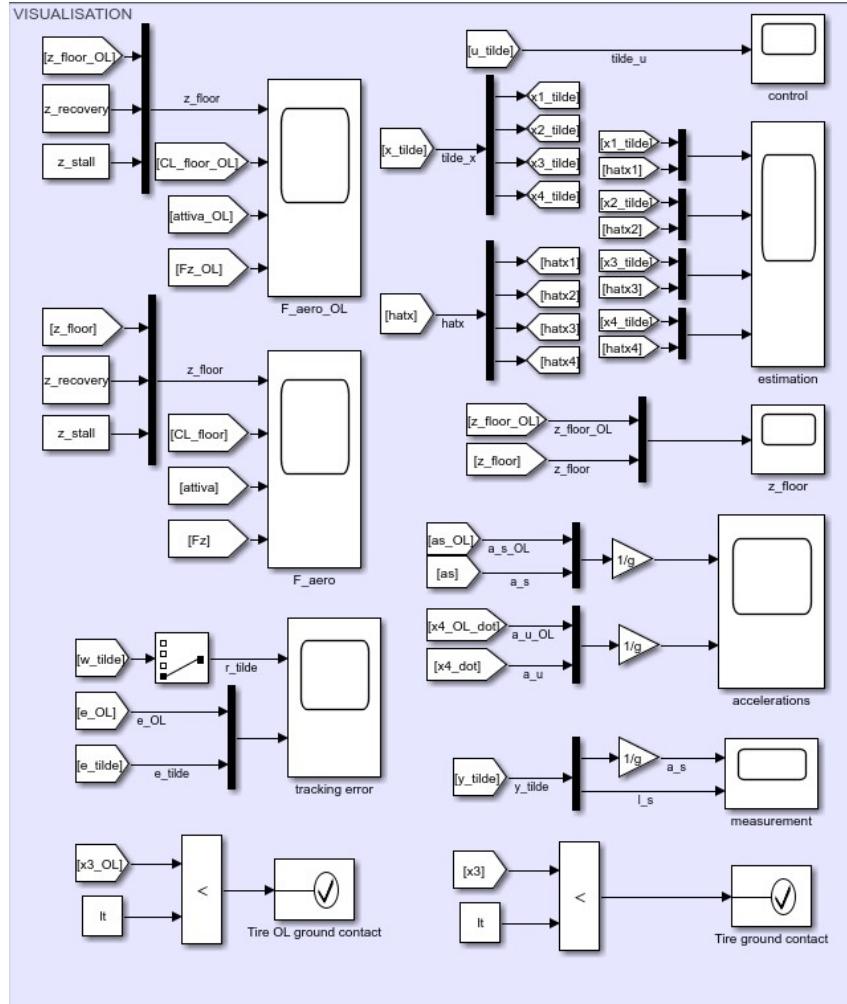


Figure 3.11: Visualisation subsystem

All these quantities allow to understand the way in which the control is operating and its effectiveness in achieving the desired performance.

3.1.10 Data Log

Several *Bus Creator* blocks have been used to send to Matlab workspace the most relevant variables coming from the simulation: this allowed to perform in Matlab environment the post-processing and plotting data required to obtain the figures shown in section 3.2.

The quantities collected from the simulation, as can be seen in fig. 3.12, include:

- The most relevant quantities of the non-linear system in 2.5.
- The most relevant quantities of the Open-Loop non-linear system from 3.1.2.
- The most relevant quantities of the linearised system in 2.21 and the Observer state vector \hat{x} .
- The most relevant quantities from the Open-Loop aerodynamic model according to 2.12.
- The most relevant quantities from the Closed-Loop aerodynamic model according to 2.12.

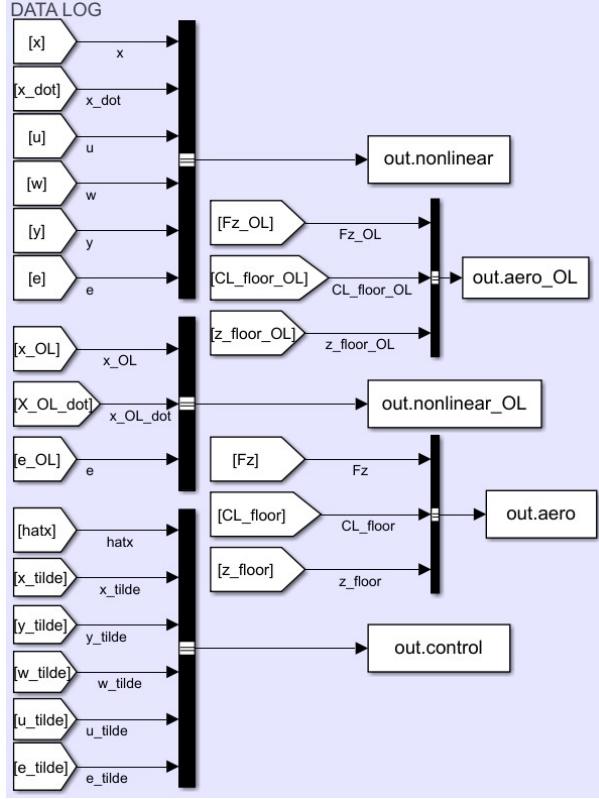


Figure 3.12: Data Log subsystem

3.2 Simulation Results

The optimal control technique relies on the proper settings of the matrices \mathbf{Q} , \mathbf{R} , \mathbf{Q}_d and \mathbf{R}_d . In this section, several tests have been performed in order to find the best combination of values for α , $\varepsilon_{i_{max}}$, u_{max} , α_d , σ_{acc} , σ_{pot} and w_{max} to tune both controller and Observer.

The tests were performed on some of the several telemetry data traces available and detailed in section 3.1.5.

The goal of this section is to show that the tuning obtained in sections 3.2.1 and 3.2.3 can guarantee:

1. Robustness in the toughest testing scenario ensuring that the Closed-Loop system never reaches the stall condition which would make the system used for control in 2.5 different from the aerodynamic model 2.12 implemented in the simulator. This would hinder the ability of the controller to retain all the properties, detailed in chapter 2, that guarantee its robustness.
2. Performance across a selected range of tracks where the porpoising issue was presented to some F1 teams. This capacity will be assessed as the controller ability to track the reference while not hindering driver's health subjecting the vehicle to extreme vertical accelerations. The FIA recommends in a directive that the driver should not be exposed to repeated accelerations of amplitude larger than 0.6 g: this threshold is taken as a guideline to evaluate the harshness of the control impact on the driver's health.

3.2.1 Controller Tuning

In this section, the matrices \mathbf{Q} and \mathbf{R} are tuned by trying out different values for α , $\varepsilon_{i_{max}}$ and u_{max} . In order to take advantage of the *separation principle* and be able to tune \mathbf{K}_S and \mathbf{K}_I independently from the Observer performance, the manual switch in fig. 3.7 is switched to input $\tilde{\mathbf{x}}$ into the controller.

Intuitively, due to the aforementioned performance requirements, both $\varepsilon_{1_{max}}$ and $\varepsilon_{5_{max}}$ need to be very close to zero, imposing in such a way a high-cost associated to sub-optimal downforce production; $\varepsilon_{3_{max}}$ can limit vertical acceleration to avoid affecting driver's health; $\varepsilon_{2_{max}}$ and $\varepsilon_{4_{max}}$ need to be properly

calibrated to avoid both tire-lifting from the ground and excessive tire-vibrations.

Many tests have been run to obtain the best combination of the cost parameters: more specifically, in 3.1 is reported the optimal combination that has been achieved, which achieves the goal as well as another positive aspect, namely the one of limiting vertical accelerations experienced by the driver. For the sake of both simplicity and used for the tuning since it coincides with the most “challenging” one for the vehicle from the porpoising perspective.

$$\left\{ \begin{array}{l} \varepsilon_{1_{max}} = 1.4 \times 10^{-4} \\ \varepsilon_{2_{max}} = 0.5 \\ \varepsilon_{3_{max}} = 7 \times 10^{-3} \cdot g \\ \varepsilon_{4_{max}} = 1.8 \times 10^{-2} \cdot g \\ \varepsilon_{5_{max}} = 3 \times 10^{-4} \\ \alpha = 6 \\ u_{max} = 5000 \end{array} \right. \quad (3.1)$$

The system behaviour in terms of both vertical mass acceleration and of tracking error are given as follows:

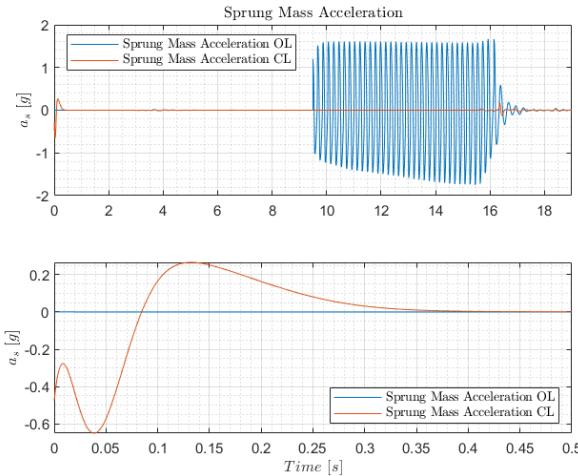


Figure 3.13: Vertical mass acceleration experienced with the optimal combination of the five cost parameters

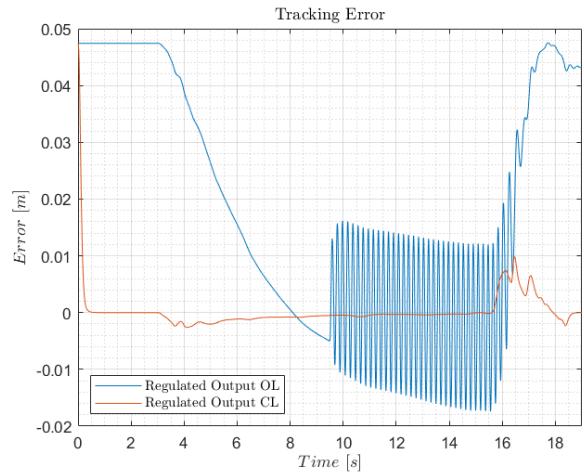


Figure 3.14: Tracking error experienced with the optimal combination of the five cost parameters

Intuitively, just looking at the results obtained here above (figures 3.13 and 3.14), it can be seen that both the objectives set and mentioned at the beginning of this chapter have been achieved simultaneously.

In order to fully understand how each cost parameter affects the performance of the controller, a “sweep test” was conducted for each of them in section 3.2.2. The sensitivity analysis consists of observing the variation in behaviour of the controller, and therefore of the Closed-Loop plant, when increasing and decreasing each of the cost parameters by one order of magnitude.

3.2.2 “Sweep Test” for the Controller Tuning

- $\varepsilon_{1_{max}}$ sensitivity analysis:



Figure 3.15: Simulation Results due to a decrease of one order of magnitude of $\varepsilon_{1_{max}}$ cost parameter

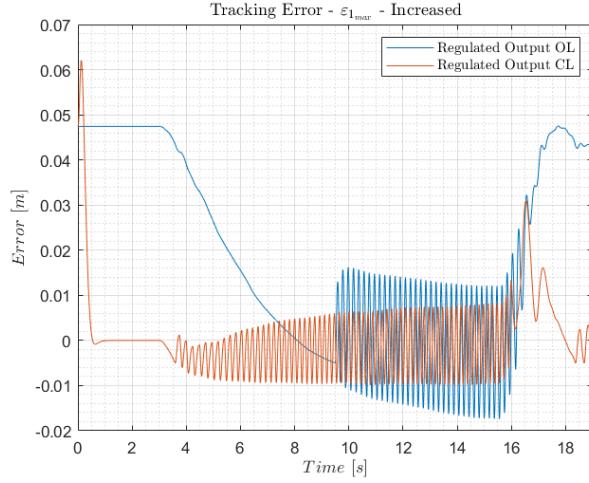


Figure 3.16: Simulation Result due to an increase of one order of magnitude of $\varepsilon_{1_{max}}$ cost parameter

Comparing the figures 3.15 and 3.16 it is evident that the parameter $\varepsilon_{1_{max}}$ has a very large effect on the behaviour of the Closed-Loop plant.

Indeed, by reducing of one order of magnitude $\varepsilon_{1_{max}}$, at the start of the simulation, when the error on the ride-height is at its largest, the control imposes such a high cost that the actuator ends up lifting the tire from the ground while trying to bring the sprung mass down.

On the other hand, by increasing of one order of magnitude $\varepsilon_{1_{max}}$ the control becomes too soft and unresponsive to ride-height error leading to the floor falling below the stall-height.

Both behaviours are clearly unacceptable and highlight the necessity of this parameter being present and properly tuned to obtain a satisfactory control action.

- $\varepsilon_{2_{max}}$ sensitivity analysis:

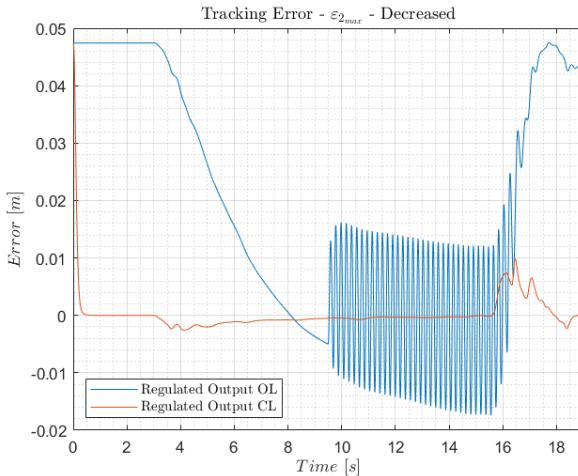


Figure 3.17: Simulation Result due to a decrease of one order of magnitude of $\varepsilon_{2_{max}}$ cost parameter

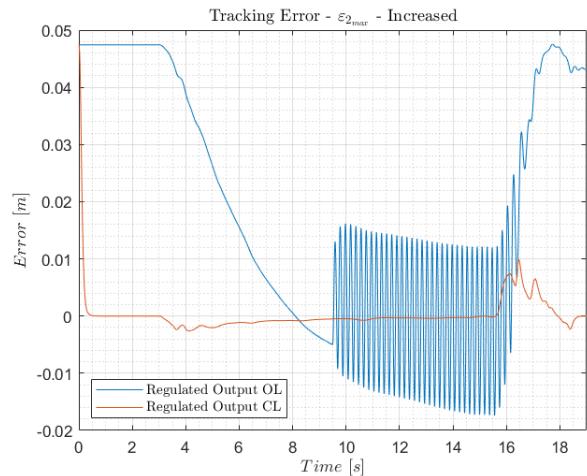


Figure 3.18: Simulation Result due to an increase of one order of magnitude of $\varepsilon_{2_{max}}$ cost parameter

Comparing the figures 3.17 and 3.18 it is evident that the parameter $\varepsilon_{2_{max}}$ has arguably no effect on the behaviour of the Closed-Loop plant. This means that one could think of simplifying the arbitrarily designed error vector ε by removing this component.

- $\varepsilon_{3_{max}}$ sensitivity analysis:

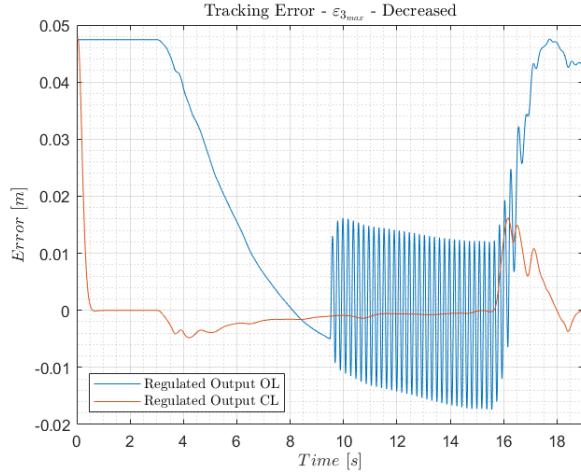


Figure 3.19: Simulation Result due to a decrease of one order of magnitude of $\varepsilon_{3_{max}}$ cost parameter

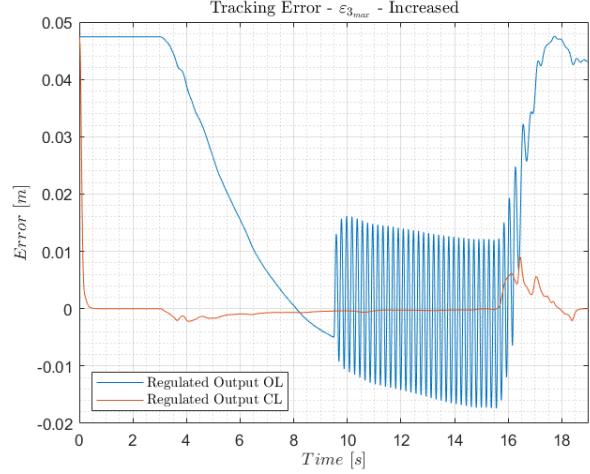


Figure 3.20: Simulation Result due to an increase of one order of magnitude of $\varepsilon_{3_{max}}$ cost parameter

Comparing the figures 3.19 and 3.20 one can see that the parameter $\varepsilon_{3_{max}}$ has some effect on the behaviour of the Closed-Loop plant.

Indeed, by reducing of one order of magnitude $\varepsilon_{3_{max}}$ the control imposes a tighter margin for sprung mass acceleration, which is reflected in a smoother decrease from the initial ride-height to the optimal ride-height. The downside is the lack of responsiveness of the control when dealing with large longitudinal accelerations, leading to a worsening in performance especially during breaking. On the other hand, by increasing of one order of magnitude $\varepsilon_{3_{max}}$ the control becomes too careless about vertical acceleration amplitude, which can be seen especially when the controller is switched on.

Therefore, even though the effects of $\varepsilon_{3_{max}}$ do not look as massive as those of $\varepsilon_{1_{max}}$ or $\varepsilon_{4_{max}}$, this parameter is still useful to provide the proper balance between an excessively aggressive control action and one that is too lenient.

- $\varepsilon_{4_{max}}$ sensitivity analysis:

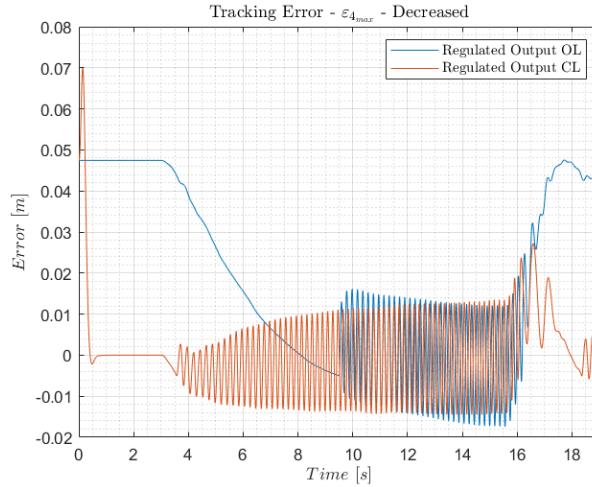


Figure 3.21: Simulation Result due to a decrease of one order of magnitude of $\varepsilon_{4_{max}}$ cost parameter

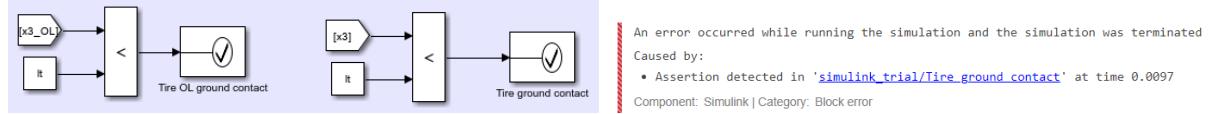


Figure 3.22: Simulation Results due to an increase of one order of magnitude of $\varepsilon_{4_{max}}$ cost parameter

Comparing the figures 3.21 and 3.22 it is evident that the parameter $\varepsilon_{4_{max}}$ has a very large effect on the behaviour of the Closed-Loop plant.

Indeed, by decreasing of one order of magnitude $\varepsilon_{4_{max}}$ the control becomes too soft and unresponsive to ride-height error leading to the floor falling below the stall-height.

On the other hand, by decreasing of one order of magnitude $\varepsilon_{4_{max}}$ the control, at the start of the simulation, when the error on the ride-height is at its largest, imposes such a high cost that the actuator ends up lifting the tire from the ground while trying to bring the sprung mass down.

Both behaviours are clearly unacceptable and highlight the necessity of this parameter being present and properly tuned to obtain a satisfactory control action.

- $\varepsilon_{5_{max}}$ sensitivity analysis:

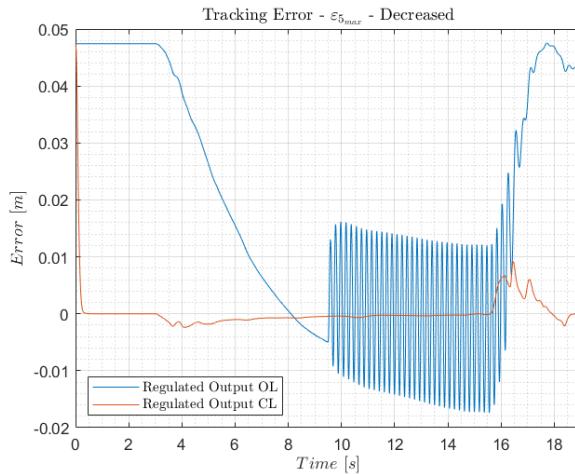


Figure 3.23: Simulation Result due to a decrease of one order of magnitude of $\varepsilon_{5_{max}}$ cost parameter

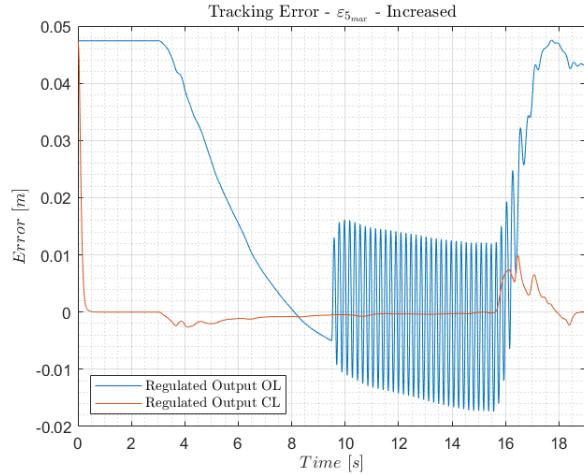


Figure 3.24: Simulation Result due to an increase of one order of magnitude of $\varepsilon_{5_{max}}$ cost parameter

Comparing the figures 3.23 and 3.24 it is evident that the parameter $\varepsilon_{5_{max}}$ has arguably no effect on the behaviour of the Closed-Loop plant. This means that one could think of simplifying the arbitrarily designed error vector ε by removing this component.

3.2.3 Observer Tuning

In this section, in analogy to what just done in section 3.2.1, the matrices \mathbf{Q}_d and \mathbf{R}_d have been tuned by trying out different values for α_d , σ_{acc} , σ_{pot} and w_{max} .

The desired goal to be checked is to have $\hat{\mathbf{x}}$ tracking as well as possible $\tilde{\mathbf{x}}$ so that the controller performance is seemingly unchanged when $\hat{\mathbf{x}}$ is set as input in the controller in place of $\tilde{\mathbf{x}}$.

Here the values for σ_{acc} and σ_{pot} are indicated by the sensors' properties in 1.4 and the value of w_{max} has been dictated by the possible operating conditions, therefore the tuning procedure is mainly limited to α_d and somewhat more straightforward. To be more accurate, the values for the standard deviations σ_{acc} and σ_{pot} has been taken directly from the sensors' datasheets and not modified hoping that they would ensure a good approximation of the linearised system by the Observer.

A further consideration that should be introduced concerns the reliability of the sensors: more specifically,

being the sensors chosen and installed onboard the F1 racecars (1.9 and 1.11) of high-end quality, it is advisable to trust the sensor outputs more than the linearised model when estimating the state. Therefore the values selected for the Observer tuning are:

$$\begin{cases} \sigma_{acc} = 5 \times 10^{-4} \cdot g \\ \sigma_{pot} = 1 \times 10^{-5} \\ w_{max} = 70 \\ \alpha_d = 0 \end{cases}$$

As can be noticed directly from figures reported in fig. 3.25 the Observer is reliable in tracking the system state.

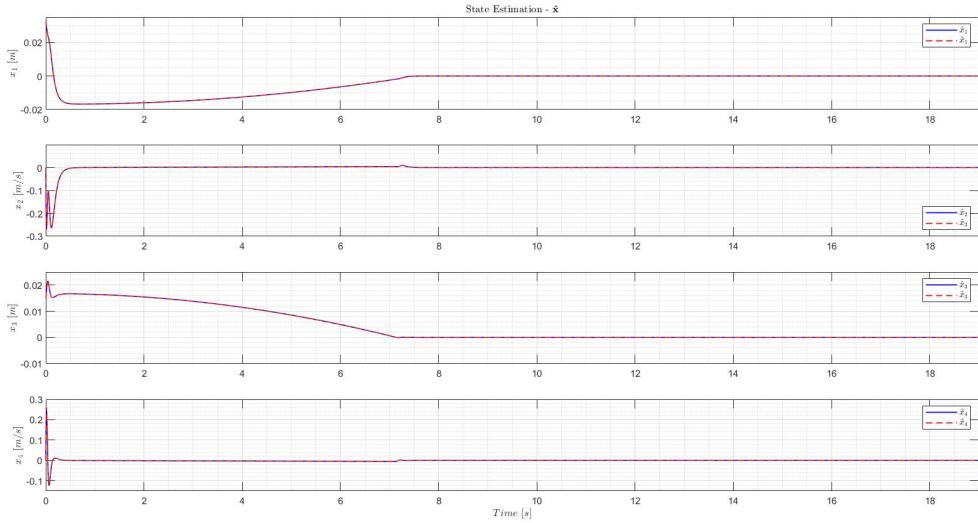


Figure 3.25: State Estimation Tracking: $\tilde{\mathbf{x}}$ vs $\hat{\mathbf{x}}$

More than this, another interesting comparison that can be shown is that one between the control input $\tilde{\mathbf{u}}$ generated by using the estimated state coming from the Observer $\hat{\mathbf{x}}$ as input and the one that would be, theoretically, given by using the state of the linearised plant $\tilde{\mathbf{x}}$ as the input of the same controller (3.26 vs 3.27), indeed:

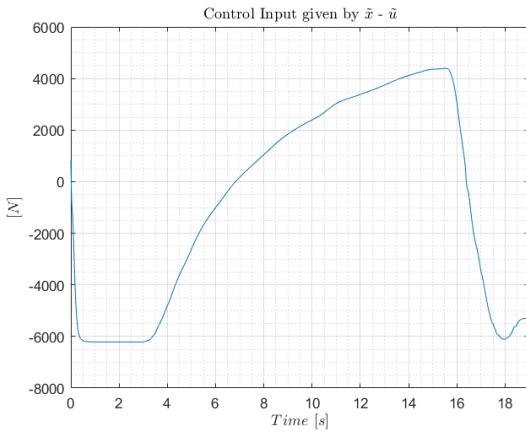


Figure 3.26: Control Law $\tilde{\mathbf{u}}$ given by $\tilde{\mathbf{x}}$

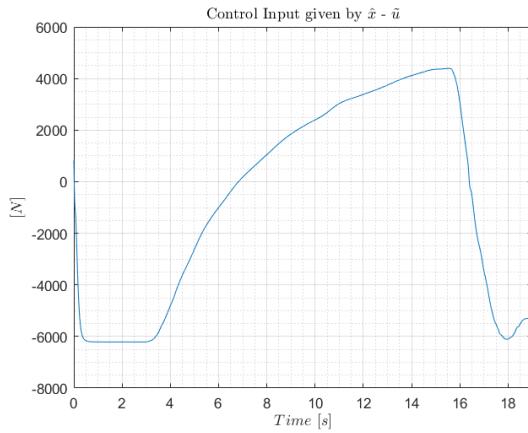


Figure 3.27: Control Law $\tilde{\mathbf{u}}$ given by $\hat{\mathbf{x}}$

As can be clearly noticed, the behaviours shown in figures 3.26 and 3.27 are quite similar: this means that the Observer is performing well in estimating the state of the plant.

3.2.4 Final Optimisation

Due to the results from the *Sweep Test* performed in section 3.2.2 it has been chosen to “cut” the number of cost parameters, reducing them to the ones which have been found to be the most relevant and influencing, namely $\varepsilon_{1_{max}}$, $\varepsilon_{3_{max}}$, $\varepsilon_{4_{max}}$. These parameters have also been slightly changed in magnitude to optimise the system behaviour with the newly designed error vector ε .

Therefore, the newly defined vector ε is constituted as follows:

$$\begin{cases} \varepsilon_1 := \text{vehicle ride - height} \\ \varepsilon_3 := \text{sprung mass acceleration} \\ \varepsilon_4 := \text{tire COG acceleration} \end{cases}$$

and then, consequently, the matrices \mathbf{C}_ε , \mathbf{D}_ε are defined as:

$$\mathbf{C}_\varepsilon = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ -\frac{k_s}{m_s} & -\frac{c_s}{m_s} & 0 & 0 & 0 \\ \frac{k_s}{m_u} & \frac{c_s}{m_u} & -\frac{k_t}{m_u} & 0 & 0 \end{bmatrix} \quad \mathbf{D}_\varepsilon = \begin{bmatrix} 0 \\ \frac{1}{m_s} \\ -\frac{1}{m_u} \end{bmatrix}$$

So, in the end, the final numerical parameters chosen for the control system are:

$$\begin{aligned} \text{Controller} \rightarrow & \begin{cases} \varepsilon_{1_{max}} = 1.3 \times 10^{-4} \\ \varepsilon_{3_{max}} = 8 \times 10^{-3} \cdot g \\ \varepsilon_{4_{max}} = 1.5 \times 10^{-2} \cdot g \\ \alpha = 6 \\ u_{max} = 5000 \end{cases} \\ \text{Observer} \rightarrow & \begin{cases} \alpha_d = 0 \\ \sigma_{acc} = 5 \times 10^{-4} \cdot g \\ \sigma_{pot} = 1 \times 10^{-5} \\ w_{max} = 70 \end{cases} \end{aligned}$$

The behaviour of the system in terms of both vertical mass acceleration and of tracking error are given as follows:

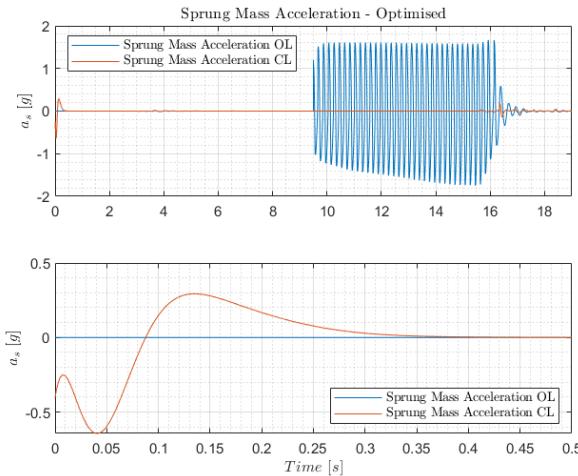


Figure 3.28: Vertical mass acceleration considering the final optimised cost parameters

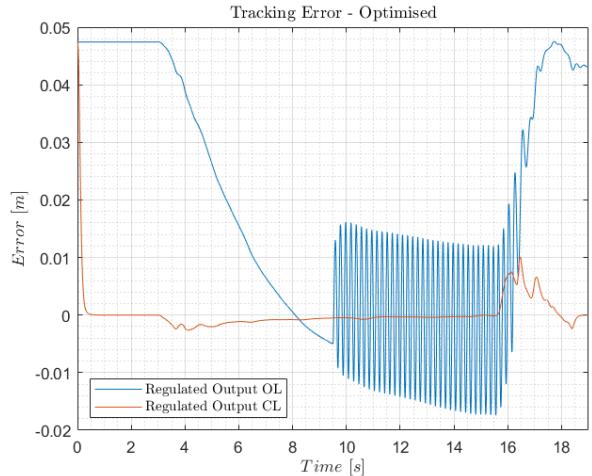


Figure 3.29: Tracking error considering the final optimised cost parameters

As can be seen comparing the results obtained in figures 3.28 and 3.29 with those in figures 3.13 and 3.14, the removal of the cost parameters $\varepsilon_{2_{max}}$ and $\varepsilon_{5_{max}}$ has not affected significantly the behaviour of both the vertical sprung mass acceleration \ddot{z}_s and the tracking error \tilde{e} . This was to be expected and is

in line with the observations made in section 3.2.2 when performing the *sweep test*.

The final tuning parameters lead to our control cost matrices assuming the following values:

$$\mathbf{Q} = \begin{bmatrix} 2.56 \times 10^3 & 0 & 0 \\ 0 & 4.25 & 0 \\ 0 & 0 & 2.26 \end{bmatrix} \quad \mathbf{R} = [4 \times 10^{-8}]$$

$$\mathbf{Q}_d = \begin{bmatrix} 4900 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{R}_d = \begin{bmatrix} 2.5 \times 10^{-7} & 0 \\ 0 & 1 \times 10^{-10} \end{bmatrix}$$

3.2.5 Robustness Tests

The toughest scenario for this system is represented by fast varying exogenous vector as control theory guarantees null regulated output after a transient time provided that \mathbf{w} is constant. This can happen in the form of high horizontal accelerations.

Despite being higher in magnitude, negative accelerations pose a performance but not a robustness problem as they tend to raise the floor and bring the system away from the stall height. Positive accelerations on the other hand are lower in magnitude being limited by: engine power, drag and available traction but tend to bring the system to the stall condition and trigger porpoising.

To assess the robustness of the control a maximum acceleration scenario has been implemented as described in 3.1.5. The scenario is on the safer side as it grossly overestimates the vehicle acceleration after 220 km/h by keeping the value of maximum acceleration constant from such a value to the top speed. The top speed was assumed to be 350 km/h , which is also a slight overestimation as the highest recorded speed in the 2022 F1 season was 351.7 km/h , but this was reached at 2285 m of altitude in Mexico City which implies a drag reduction.

The obtained telemetry data for maximum acceleration at each longitudinal speed and velocity evolution over time are shown below:

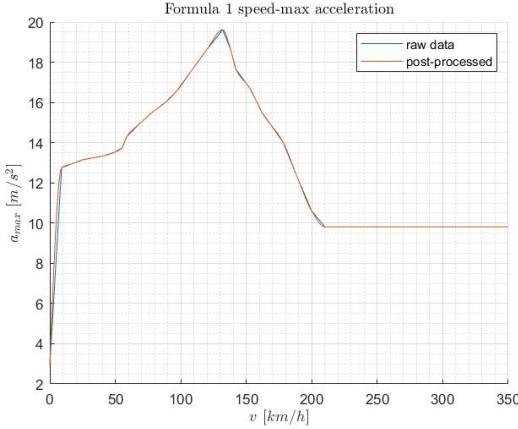


Figure 3.30: Maximum acceleration feasible at each given longitudinal speed

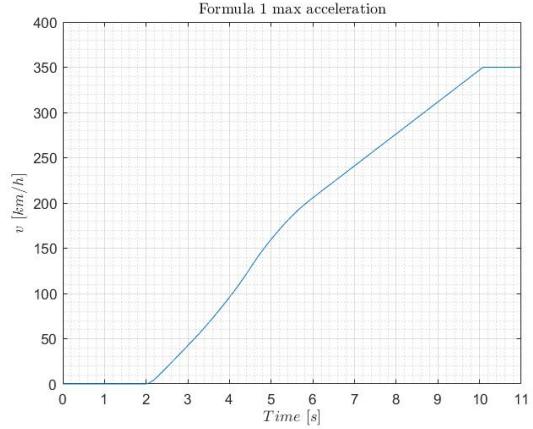


Figure 3.31: Longitudinal speed over time during maximum acceleration manoeuvre

As can be seen from fig. 3.31 the car accelerates from $0 - 100 \text{ km/h}$ in around 2 seconds and from $0 - 200 \text{ km/h}$ in around 3.85 seconds , which is slightly exaggerated but still fairly accurate.

To further boost the safety of the control system and be sure that the final tuning could avoid reaching stall height in any real life scenario an even harsher test was designed.

The test subjected the control system to an hypothetical vehicle accelerating twice as fast as a F1 single-seater with a maximum acceleration manoeuvre shown below:

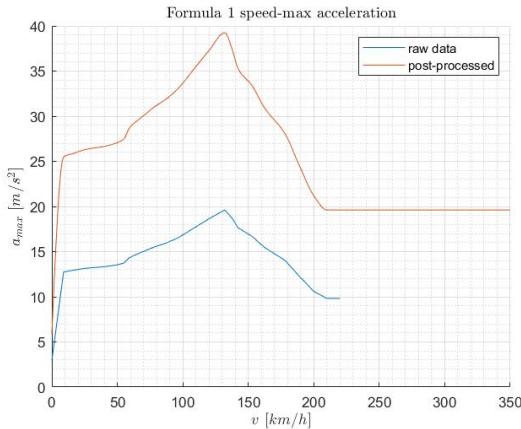


Figure 3.32: Doubled maximum acceleration feasible at each given longitudinal speed

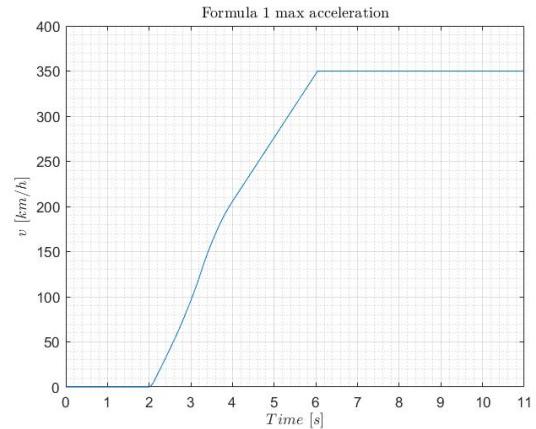


Figure 3.33: Longitudinal speed over time during double maximum acceleration manoeuvre

In this test the vehicle accelerates from 0 – 100 km/h in around 1 second and from 0 – 300 km/h in around 3.3 seconds making this test much more challenging than any possible real life condition.

To make this extreme robustness test a true worst case scenario, the precision of the two sensors has been also reduced by two orders of magnitude just for simulating some possible malfunctioning conditions. This can be appreciated by comparing figures 3.34 and 3.35.

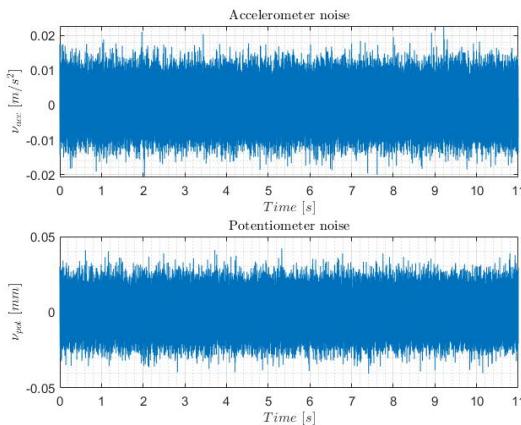


Figure 3.34: Standard sensors' noise amplitude

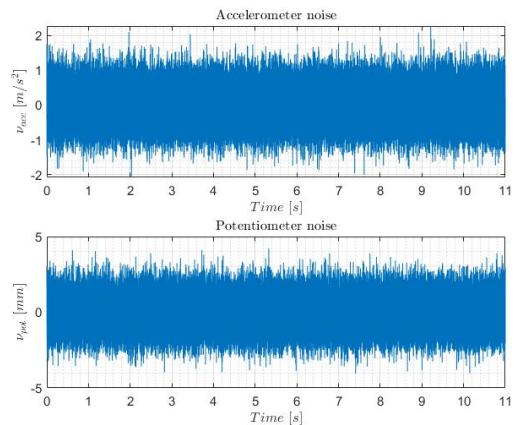


Figure 3.35: Worst case sensors' noise amplitude

By using the tuning parameters defined in section 3.2.4 and by testing the control system in the scenario highlighted in fig. 3.31, it can be noticed that the control algorithm is “reactive enough” to avoid the floor falling below the stall-height as can be seen in fig. 3.36:

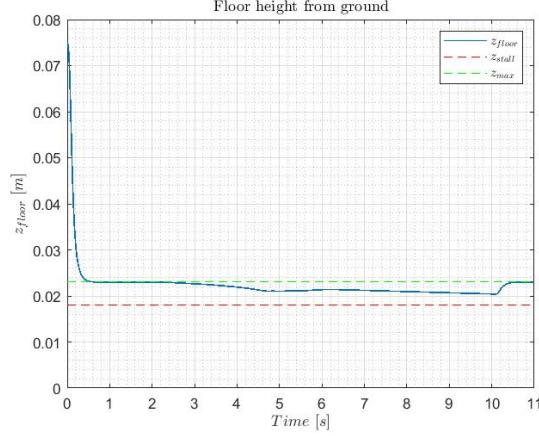


Figure 3.36: Closed-Loop floor-height during maximum acceleration test

If the system is subjected to the extreme acceleration test, as reported in fig. 3.33, the response of the Closed-Loop still manages to prevent the stall of the underbody airflow as can be seen from figures 3.37 and 3.38.

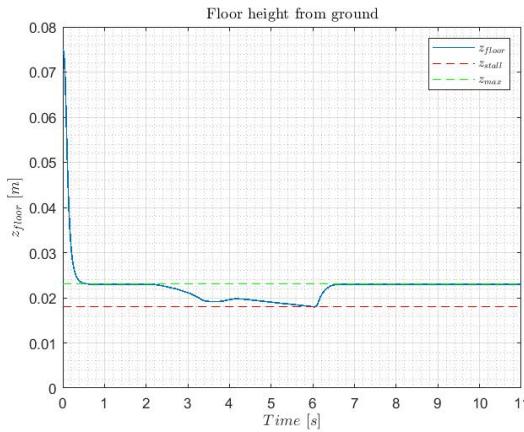


Figure 3.37: Closed-Loop floor-height during doubled maximum acceleration test

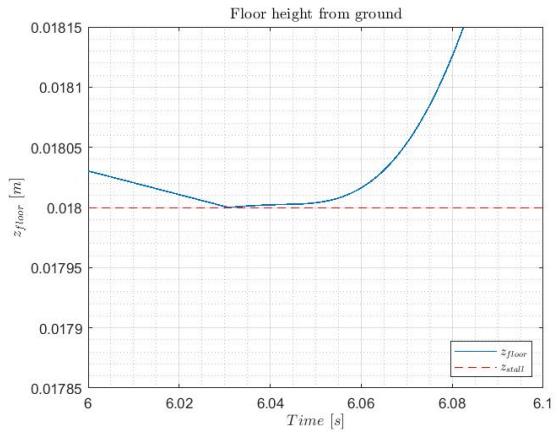


Figure 3.38: Zoomed-in view of the most critical section of fig. 3.37

As can be seen from fig. 3.38, by doubling the vehicle maximum acceleration at each longitudinal speed the control can barely avoid the stall and without changing any parameter the airflow would certainly stall, bringing the vehicle in oscillation if subjected to even more extreme accelerations.

If one desires to further boost robustness to account for a possibly destabilising effect of some unpredictable or neglected variable it is possible to slightly increase the reference that is desired to track: this will massively increase robustness at the cost of relatively little loss in performance.

For example, by simply rising the reference height by only 4 mm it is possible to avoid the floor stall condition even in the hypothetical, but not so reasonable, condition of a maximum acceleration manoeuvre performed by the vehicle which acceleration is three times higher than the one depicted in fig. 3.31.

During the aforementioned raised reference floor test, as one can see in fig. 3.40, the steady-state $C_{L_{floor}}$ is approximately equal to 1.875: this implies that the floor is generating 98.7 % of its possible maximum downforce and therefore the vehicle as a whole is generating 99.3 % of its possible maximum downforce due to wings' downforce generation being unaffected by the raise in floor-height.

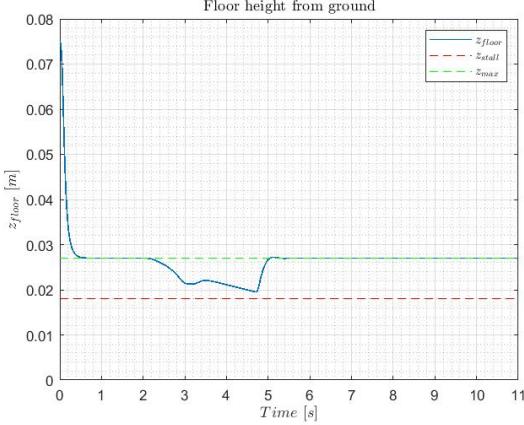


Figure 3.39: Closed-Loop floor-height during tripled maximum acceleration test with raised reference

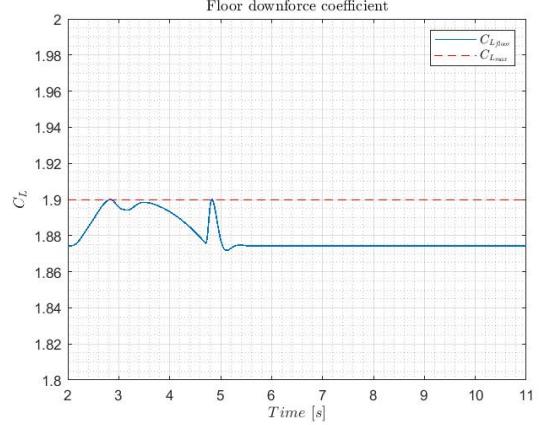


Figure 3.40: Floor downforce coefficient during tripled maximum acceleration test with raised reference

3.2.6 Performance Analysis of some relevant testing scenarios

All the scenarios analysed in this subsection share three main properties:

- The initial condition for the simulation is the OL equilibrium point at the starting speed.
- The simulation starts with three seconds at constant speed to allow the Closed-Loop system to also find an equilibrium: this is done as the control would be switched on at 0-speed when turning the car on so during a flying lap it should start from an equilibrium position.
- All tracks are located at sea level to respect the hypothesis in 2.1.1.

The behaviour of the control system when switched-on when the car is standing still can be retrieved from the simulation of the maximum acceleration scenario detailed in section 3.2.5, which starts from the OL 0-speed equilibrium condition.

Scenario 1: Flying lap at Baku Circuit

In scenario 1, the simulation was performed exposing the system to a time-varying disturbance obtained from telemetry data, pictured in fig. 3.41, recorded at **Baku City Circuit**.

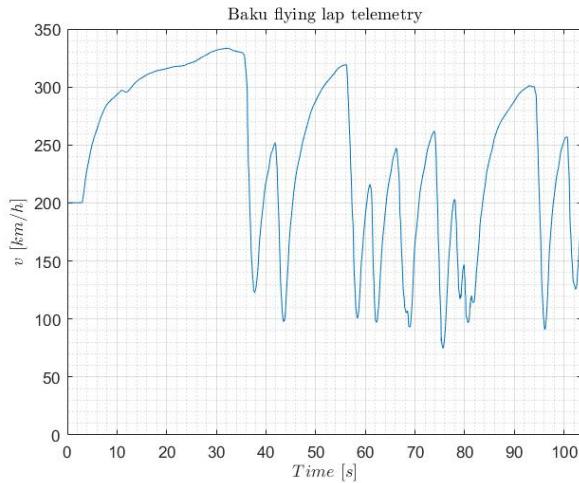


Figure 3.41: **Baku City Circuit** flying lap telemetry

This track is particularly relevant as its extremely long main straight triggered, in some single-seaters, the most aggressive porpoising behaviour of all the 2022 F1 season. The repeated vertical oscillations

were so aggressive that some drivers experienced moderate health impact which forced the FIA to pass a technical directive that imposed teams to limit porpoising even if it came at the cost of performance.

The figures reported below highlight the performance of the control by comparing the behaviour of the Open-Loop system with the Closed-Loop system.

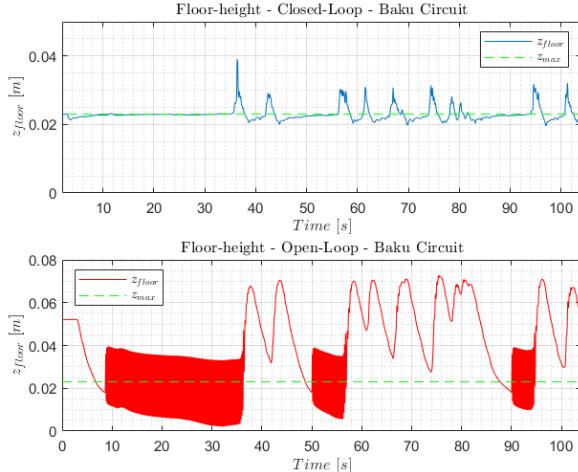


Figure 3.42: Closed-Loop vs Open-Loop comparison of the car ride-height during a flying lap in **Baku City Circuit**

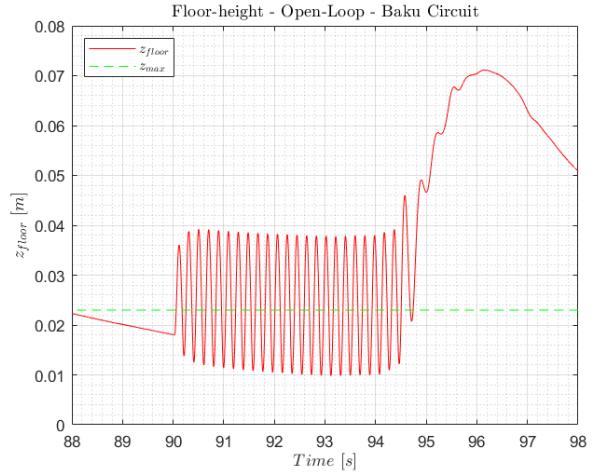


Figure 3.43: Zoomed in view of the Open-Loop time evolution of the ride-height of the vehicle while porpoising

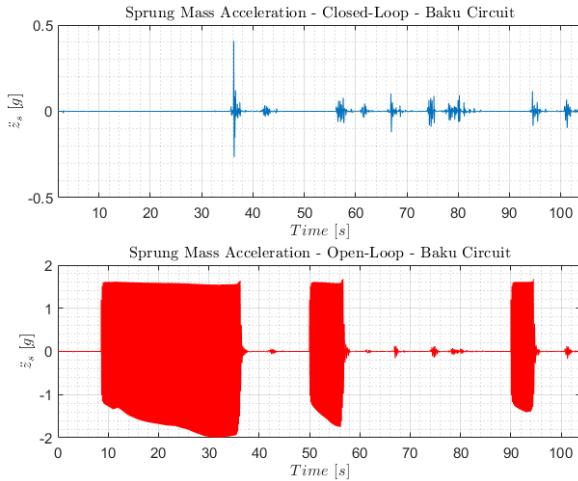


Figure 3.44: Closed-Loop vs Open-Loop comparison of the driver vertical acceleration during a flying lap in **Baku City Circuit**



Figure 3.45: Zoomed in view of the Open-Loop time evolution of the vertical acceleration of the driver while porpoising

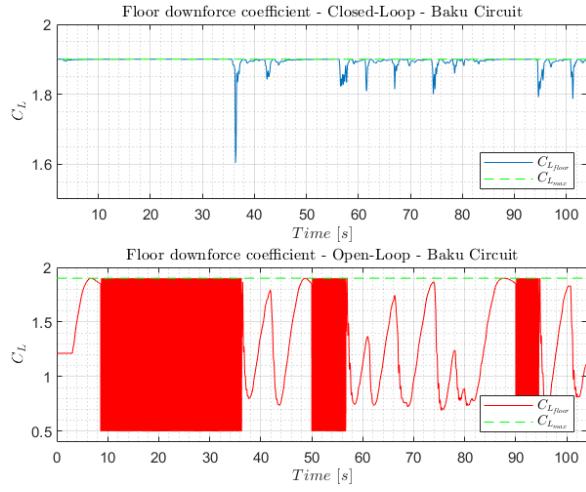


Figure 3.46: Closed-Loop vs Open-Loop comparison of the floor downforce coefficient during a flying lap in **Baku City Circuit**



Figure 3.47: Zoomed in view of the Open-Loop time evolution of the floor downforce coefficient while porpoising

Scenario 2: Flying lap at Barcelona Circuit

In scenario 2, the simulation was performed exposing the system to a time-varying disturbance obtained from telemetry data, depicted in fig. 3.48, recorded at **Circuit de Barcelona-Catalunya**.

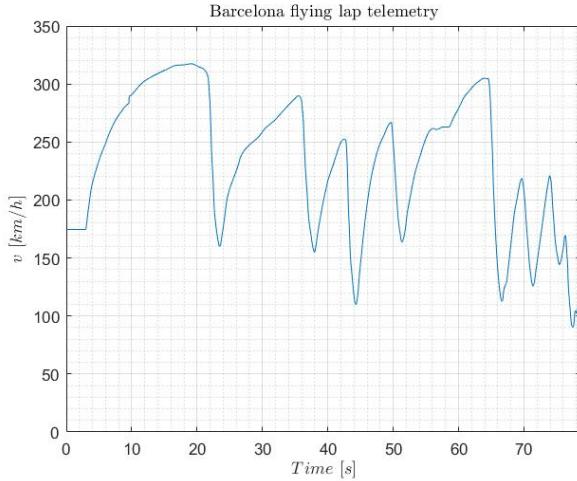


Figure 3.48: **Circuit de Barcelona-Catalunya** flying lap telemetry

This track was chosen as it was used during pre-season testing and therefore a lot of official data was released from F1 regarding porpoising at this specific venue.

The figures reported below highlight the performance of the control by comparing the behaviour of the Open-Loop system with the Closed-Loop system.

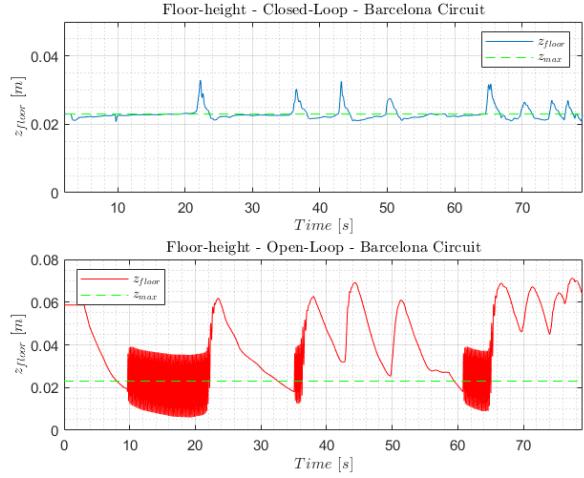


Figure 3.49: Closed-Loop vs Open-Loop comparison of the car ride-height during a flying lap in **Circuit de Barcelona-Catalunya**

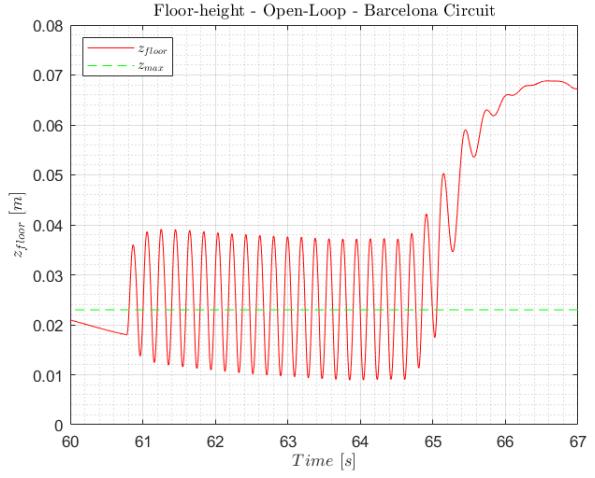


Figure 3.50: Zoomed in view of the Open-Loop time evolution of the ride-height of the vehicle while porpoising

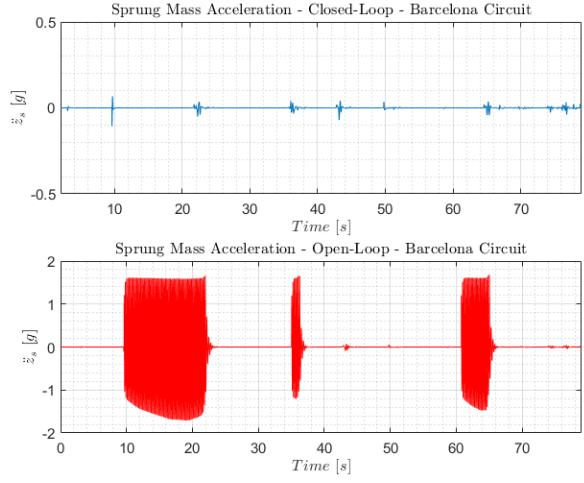


Figure 3.51: Closed-Loop vs Open-Loop comparison of the driver vertical acceleration during a flying lap in **Circuit de Barcelona-Catalunya**

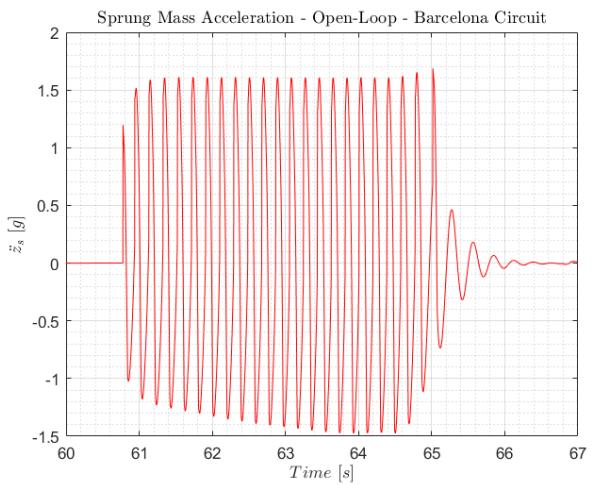


Figure 3.52: Zoomed in view of the Open-Loop time evolution of the vertical acceleration of the driver while porpoising

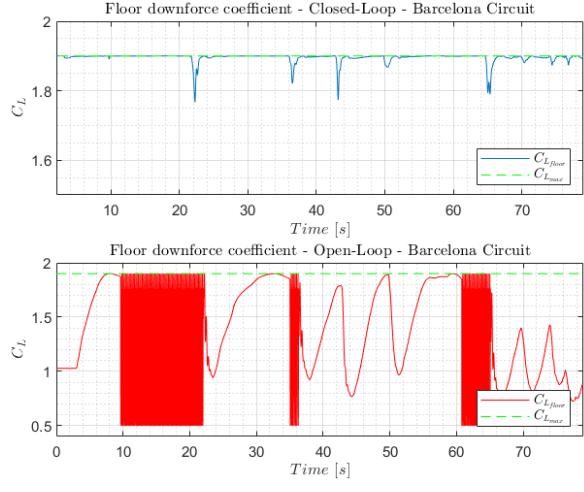


Figure 3.53: Closed-Loop vs Open-Loop comparison of the floor downforce coefficient during a flying lap in **Circuit de Barcelona-Catalunya**

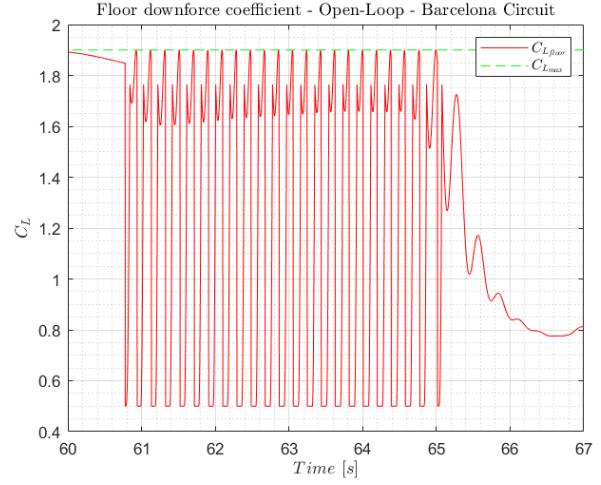


Figure 3.54: Zoomed in view of the Open-Loop time evolution of the floor downforce coefficient while porpoising

Scenario 3: Flying lap at Jeddah Circuit

In scenario 3, the simulation was performed exposing the system to a time-varying disturbance obtained from telemetry data, pictured in fig. 3.55, recorded at **Jeddah Corniche Circuit**.

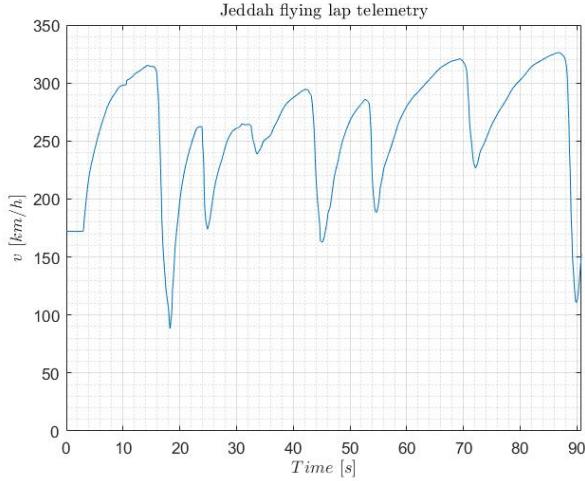


Figure 3.55: **Jeddah Corniche Circuit** flying lap telemetry

This track was chosen as the city circuit with the highest average speed during a flying lap, like Baku, resulted to be one the most challenging circuits today.

The figures reported below highlight the performance of the control by comparing the behaviour of the Open-Loop system with the Closed-Loop system.

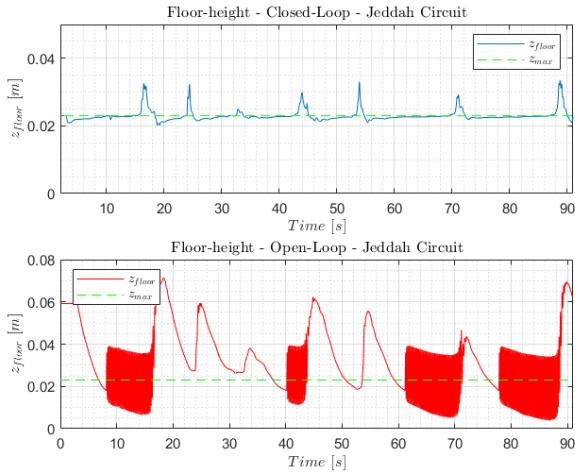


Figure 3.56: Closed-Loop vs Open-Loop comparison of the car ride-height during a flying lap in **Jeddah Corniche Circuit**

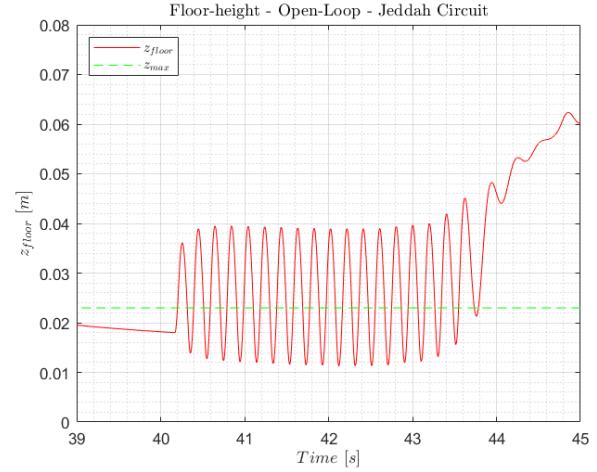


Figure 3.57: Zoomed in view of the Open-Loop time evolution of the ride-height of the vehicle while porpoising

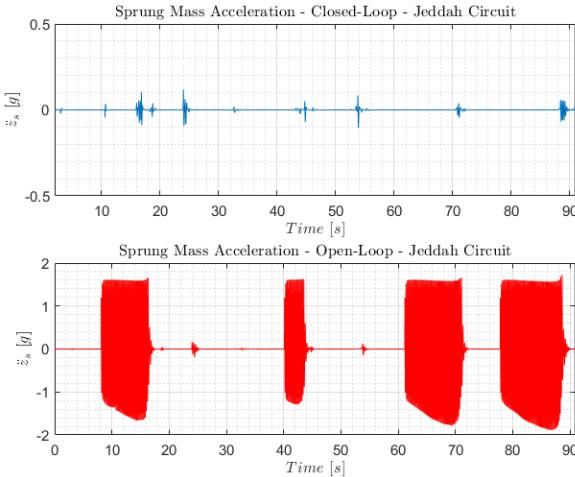


Figure 3.58: Closed-Loop vs Open-Loop comparison of the driver vertical acceleration during a flying lap in **Jeddah Corniche Circuit**

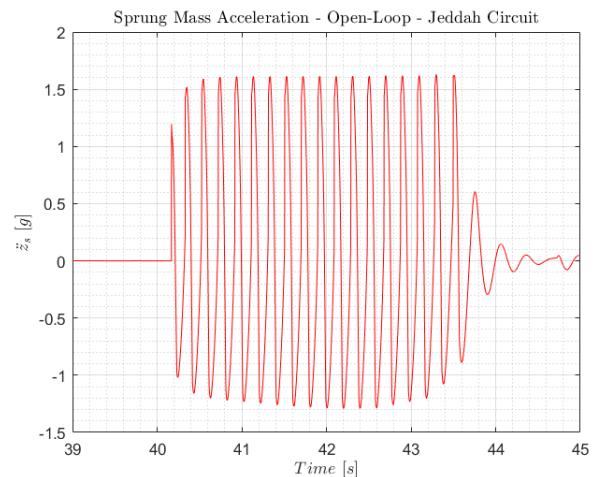


Figure 3.59: Zoomed in view of the Open-Loop time evolution of the vertical acceleration of the driver while porpoising

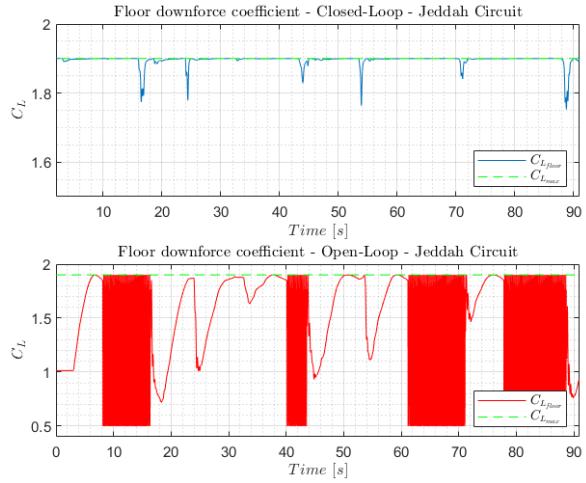


Figure 3.60: Closed-Loop vs Open-Loop comparison of the floor downforce coefficient during a flying lap in **Jeddah Corniche Circuit**

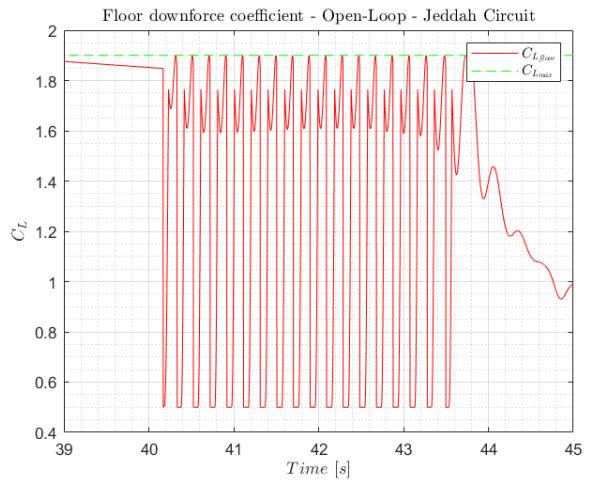


Figure 3.61: Zoomed in view of the Open-Loop time evolution of the floor downforce coefficient while porpoising

Scenario 4: Flying lap at Melbourne Circuit

In scenario 4, the simulation was performed exposing the system to a time-varying disturbance obtained from telemetry data, pictured in fig. 3.62, recorded at **Albert Park Circuit** in Melbourne.

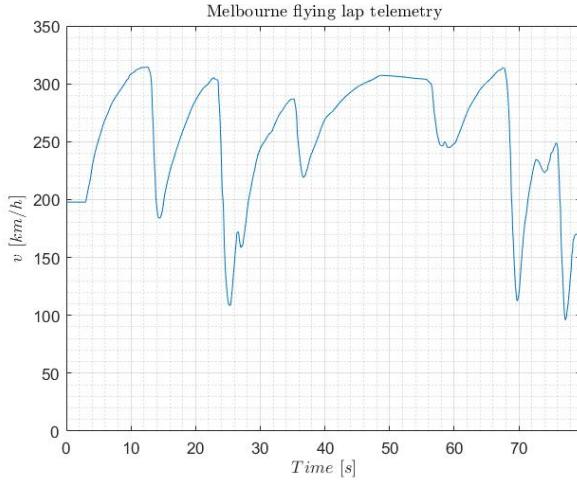


Figure 3.62: **Albert Park Circuit** flying lap telemetry

This track was chosen as its revised 2022 layout transformed it from a low-speed city circuit into a 4 DRS zones high-speed one and therefore a challenging circuit for porpoising.

The figures reported below highlight the performance of the control by comparing the behaviour of the Open-Loop system with the Closed-Loop system.

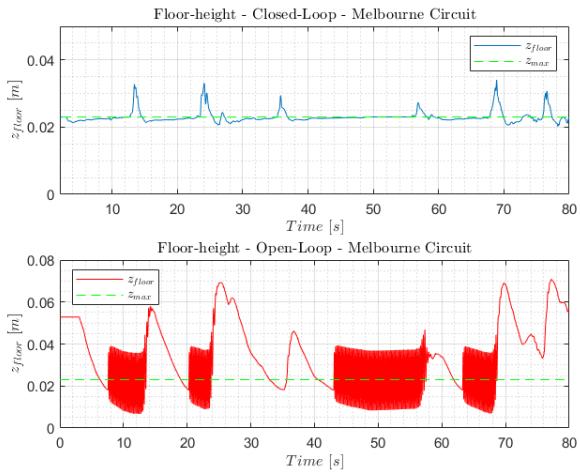


Figure 3.63: Closed-Loop vs Open-Loop comparison of the car ride-height during a flying lap in **Albert Park Circuit**

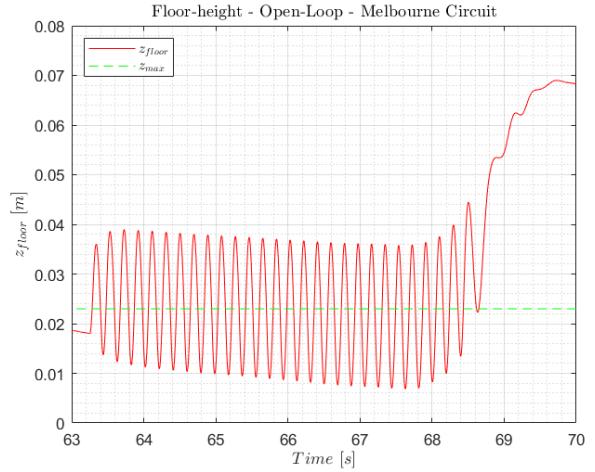


Figure 3.64: Zoomed in view of the Open-Loop time evolution of the ride-height of the vehicle while porpoising

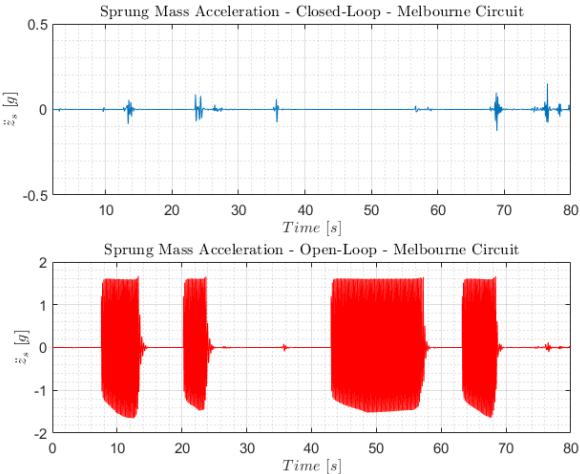


Figure 3.65: Closed-Loop vs Open-Loop comparison of the driver vertical acceleration during a flying lap in **Albert Park Circuit**

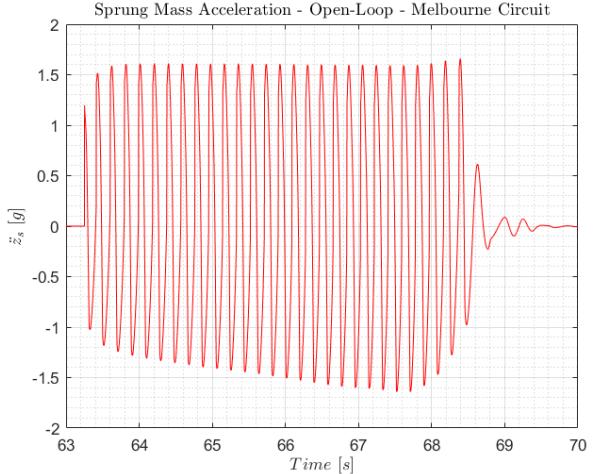


Figure 3.66: Zoomed in view of the Open-Loop time evolution of the vertical acceleration of the driver while porpoising

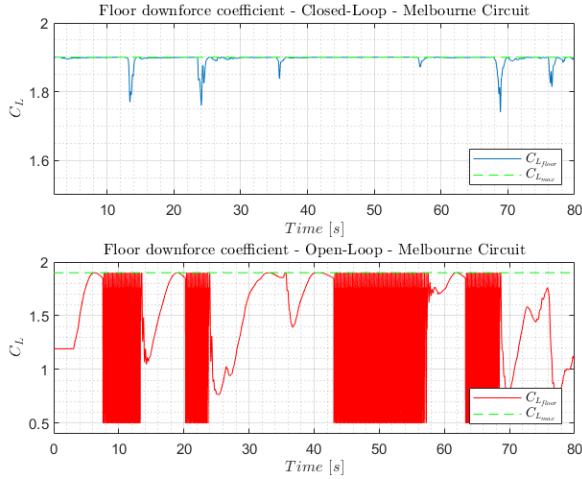


Figure 3.67: Closed-Loop vs Open-Loop comparison of the floor downforce coefficient during a flying lap in **Albert Park Circuit**

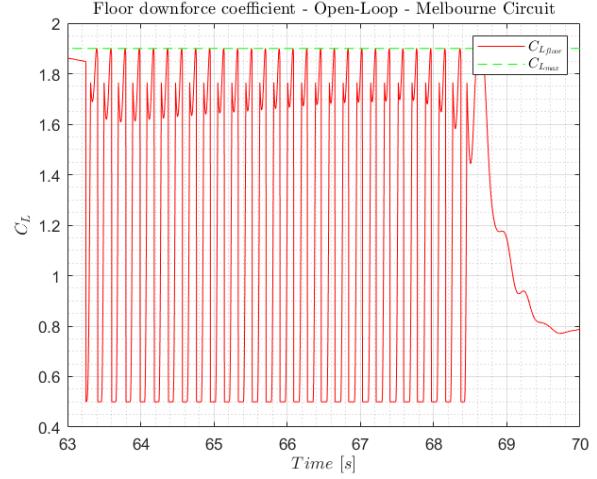


Figure 3.68: Zoomed in view of the Open-Loop time evolution of the floor downforce coefficient while porpoising

The performance of the optimal control approach has been demonstrated through different simulations which compared Open-Loop against Closed-Loop regarding the floor-height, vertical acceleration, and aerodynamic response during four current F1 circuits laps representing a wide range of scenarios, in which it is clear that there is a common behaviour in the response to disturbance coming from the Open-Loop versus the Closed-Loop plant.

As it can be seen in all the tracks, at high-speed, in response to the aerodynamic load increasing, the suspension system is compressed and therefore the vehicle starts to oscillate vertically. This also results in an oscillating trend of the downforce production, inducing an unreliable behaviour for the driver, that cannot trust the amount of downforce generated at one time. In another part, during porpoising the sprung mass experiences accelerations with an amplitude of around $1.5g$ in average, which would pose a severe health concern for the driver. The ride-height is also time-varying and depends on the vehicle longitudinal speed, this at low speeds leads to a sub-optimal downforce production from the floor. Analysing the Open-Loop, when the driver is braking after porpoising at high-speed, the oscillation stops pretty fast consistently with what was seen on F1 cars in the 2022 season.

On the other hand, observing the Closed-Loop, one notices that the response is not significantly affected by the applied disturbance, hence it can be observed that the oscillating behaviour that characterises the Open-Loop is not present in the Closed-Loop plant. It is evident from the results that the controller is effective in stopping harsh vibrations and manages to attenuate the vast majority of these oscillations experienced in the Open-Loop. The floor-height, being constant at the optimal value, also improve the downforce production at all speeds optimising the vehicle performance. During heavy breaking manoeuvres the high negative magnitude of the longitudinal acceleration interfere the ability of our control strategy to keep the ride-height constant; nevertheless, the downforce reduction experienced in these sections of the tracks is much less significant than the ones observed in the Open-Loop during the same manoeuvres.

These numerical simulations support that the proposed strategy can be efficient and real time implementable. In fact, the experiment results in different ideas to improve our level of accuracy for a possible iteration of this project.

Chapter 4

Conclusions and Further Investigations

4.1 Conclusions

In conclusion, an active suspension control system capable to avoid the porpoising behaviour of ground effect F1 cars and to optimise the Venturi duct performance, by keeping the ride-height constant with speed, has been developed. Such control system demands the implementation of an hydraulic actuator installed in each wheel to provide the requisite force to keep the ride-height constant at any speed.

The system is capable to steer large errors to zero when switching on the vehicle at 0-speed without providing excessive accelerations of the chassis, even if one could think of having the driver not enter the cockpit until the ride-height is fully set. The system, even at this large starting error condition, is not strong enough to be able to lift the tire from the ground, which would be unacceptable.

The system proved effective in a wide range of different tracks including those which caused the severies porpoising points, hence it would be able to cope with all the tracks scheduled in the calendar. The robustness of the system has been proven in a scenario much more challenging than those possibly faced in real life operating conditions ensuring reliability in the limited range of operating conditions of an F1 car.

The capability of the racecar to have the same operating ride-height at all possible speeds, if a system like the one described above was allowed by the regulation [3], would also open the possibility to F1 teams to design cars with a much narrower operating window. This last specs have a particular interest as usually this kind of cars are faster than those with larger operating windows if one is able to keep them in that specific operating window [6].

All of this would surely make the cars faster allowing the teams to run their cars very close to the ground while being able to avoid floor stall leading to an optimisation of the underfloor downforce generation [9].

This could potentially make them further reduce the requirement of downforce generation from the wings and in turn further reduce the amount of “dirty air” sent to the following car improving the cars’ raceability as the new regulation aimed.

4.2 Further Investigations

As the dataset available was limited and the mathematical model much simplified many improvements could be made if using: a more realistic half-car model similar to the one implemented by Gadola et al. in “Analyzing Porpoising on High Downforce Race Cars: Causes and Possible Setup Adjustments to Avoid It” and a dataset coming from an actual F1 team that experienced porpoising during the 2022 F1 season.

Some of the possible improvements implementing these changes include:

- The possibility to also control the pitching behaviour of the vehicle.

- The implementation of a more realistic non-linear suspension model.
- A more accurate knowledge of all the available sensors in an F1 car.
- The possibility to build the aerodynamic load model on dedicated experimental data.
- The possibility to split the downforce production between front and rear axle appropriately.

Other interesting future investigations could explore the behaviour of the system when riding on kerbs and the implementation of a gain scheduling control policy to have a softer control at low speeds to diminish vertical acceleration at start-up and stronger control at high-speed to avoid porpoising. Finally, as the aerodynamic forces are due to the relative airspeed of the car and not due to the longitudinal speed of the vehicle, an analysis of the effect of wind on the system could be of relevant interest. The effect of altitude on the phenomena and a high altitude specialised control policy could also be of interest to optimise the working of the control system for high altitude tracks such as the **Autódromo Hermanos Rodríguez** in Mexico City and the **Autódromo José Carlos Pace** in São Paolo.

Chapter 5

Appendix

5.1 Matlab Launch code

```
%> LAUNCH CODE FOR THE CONTROL SYSTEM SIMULATION
clc
close all
clear

%> SIMULATION SETUP
scenario=1; % 0 other disturbance, 1 Baku sector, 2 Baku lap,
%3 Barcelona sector, 4 Barcelona lap, 5 Jeddah sector, 6 Jeddah lap,
%7 Melbourne sector, 8 Melbourne lap, 9 Max acceleration
telemetry=load('telemetry.mat');
switch scenario
    case 0 %other disturbance
        TimeSpan=10;
    case 1 %Baku sector
        TimeSpan=ceil(telemetry.Baku_sector.time(end)); %simulation time ...
        [s]
        telemetry=timeseries(telemetry.Baku_sector.speed, ...
            telemetry.Baku_sector.time);
    case 2 %Baku lap
        TimeSpan=ceil(telemetry.Baku_lap.time(end)); %simulation time [s]
        telemetry=timeseries(telemetry.Baku_lap.speed, ...
            telemetry.Baku_lap.time);
    case 3 %Barcelona sector
        TimeSpan=ceil(telemetry.Barcelona_sector.time(end)); %simulation ...
        %time [s]
        telemetry=timeseries(telemetry.Barcelona_sector.speed, ...
            telemetry.Barcelona_sector.time);
    case 4 %Barcelona lap
        TimeSpan=ceil(telemetry.Barcelona_lap.time(end)); %simulation ...
        time [s]
        telemetry=timeseries(telemetry.Barcelona_lap.speed, ...
            telemetry.Barcelona_lap.time);
    case 5 %Jeddah sector
        TimeSpan=ceil(telemetry.Jeddah_sector.time(end)); %simulation ...
        time [s]
        telemetry=timeseries(telemetry.Jeddah_sector.speed, ...
            telemetry.Jeddah_sector.time);
    case 6 %Jeddah lap
        TimeSpan=ceil(telemetry.Jeddah_lap.time(end)); %simulation time [s]
```

```

telemetry=timeseries(telemetry.Jeddah_lap.speed, ...
    telemetry.Jeddah_lap.time);
case 7 %Melbourne sector
TimeSpan=ceil(telemetry.Melbourne_sector.time(end)); %simulation
%time [s]
telemetry=timeseries(telemetry.Melbourne_sector.speed, ...
    telemetry.Melbourne_sector.time);
case 8 %Melbourne lap
TimeSpan=ceil(telemetry.Melbourne_lap.time(end)); %simulation ...
    time [s]
telemetry=timeseries(telemetry.Melbourne_lap.speed, ...
    telemetry.Melbourne_lap.time);
case 9 %Max acceleration
TimeSpan=telemetry.acceleration.t(end); %simulation time [s]
telemetry=timeseries(telemetry.acceleration.speed, ...
    telemetry.acceleration.t);
end

DT=1e-4; %sample time fixed-step integration [s]
PLANT=1; % 0=linear, 1=nonlinear

%% DECLARATION OF VECTOR DIMENSIONS
n=4; %state dimension, x is: [z_s-z_u,v_s-v_u,z_u,v_u] '
p=1; %control dimension, u is: [F_u]
q=2; %output dimension, y is [a_s,suspension length] '
m=1; %regulated output dimension, e is: [height of center of mass
% respective to ideal]
nd=1; %disturbance dimension, d is: [longitudinal speed]
r=nd+q+m; %exogenous dimension, w is: [d accelerometer noise
%potentiometer noise height reference]'

%% INITIAL CONDITIONS FOR THE NONLINEAR PLANT
%Initial conditions
switch scenario
    case 0 %other simulations (starting from 0 longitudinal speed)
        zs_0=0.2; %initial sprung mass height [m]
        zu_0=0.1; %initial unsprung mass height [m]
    case 1 %Baku sector
        zs_0=0.19542; %initial sprung mass height [m]
        zu_0=0.09826; %initial unsprung mass height [m]
    case 2 %Baku lap
        zs_0=0.17715; %initial sprung mass height [m]
        zu_0=0.09133; %initial unsprung mass height [m]
    case 3 %Barcelona sector
        zs_0=0.18591; %initial sprung mass height [m]
        zu_0=0.09466; %initial unsprung mass height [m]
    case 4 %Barcelona lap
        zs_0=0.18367; %initial sprung mass height [m]
        zu_0=0.09381; %initial unsprung mass height [m]
    case 5 %Jeddah sector
        zs_0=0.18383; %initial sprung mass height [m]
        zu_0=0.09387; %initial unsprung mass height [m]
    case 6 %Jeddah lap
        zs_0=0.1842; %initial sprung mass height [m]
        zu_0=0.09401; %initial unsprung mass height [m]
    case 7 %Melbourne sector
        zs_0=0.19426; %initial sprung mass height [m]
        zu_0=0.09782; %initial unsprung mass height [m]

```

```

    case 8 %Melbourne lap
    zs_0=0.17785; %initial sprung mass height [m]
    zu_0=0.0916; %initial unsprung mass height [m]
    case 9 %max acceleration
    zs_0=0.2; %initial sprung mass height [m]
    zu_0=0.1; %initial unsprung mass height [m]
end
vs_0=0; %initial sprung mass velocity [m]
vu_0=0; %initial unsprung mass velocity [m]

%State initial condition
x_0=[zs_0-zu_0
      vs_0-vu_0
      zu_0
      vu_0];

%% NOMINAL PLANT PARAMETERS USED TO DESIGN THE CONTROL SYSTEM
%Quarter car model parameters
g=9.81; %gravity acceleration [m/s^2]
ks=165000; %suspension spring stiffness [N/m]
kt=2.7e5; %tire spring stiffness [N/m]
M=200; %quarter vehicle mass [kg]
mu=20; %unsprung mass [kg]
ms=M-mu; %sprung mass [kg]
zs_unloaded=0.2; %ms unloaded heighth [m]
zu_unloaded=0.1; %mu unloaded heighth [m]
ls=ms*g/ks+zs_unloaded-zu_unloaded; %suspension unloaded length [m]
lt=M*g/kt+zu_unloaded; %tire unloaded length [m]
cs=3500; %damping coefficient [Ns/m]
mr=(mu+ms)/(mu*ms); %reduced mass of two body sistem [1/kg]

%Aero force parameters
z_max=0.023; %optimal ground height [m]
CL_max=1.9; %max downforce coefficient
S=0.4; %floor area [m^2]
rho_air=1.225; %air density [kg/m^3]
Sw_CLw=0.75; %surface area times downforce coefficient of rear
%wing [m^2]
h=0.125; %vertical distance between floor edge and COM [m]
z_stall=0.018; %floor height when stall happens [m]
z_recovery=z_max+0.01; %floor height when stall is recovered [m]

%% LINEARISATION CONDITIONS
%Equilibrium triplet x*,u*,w*
d_star=70; %disturbance at equilibrium [m/s]
x_star=[z_max+h+(M*g+0.5*rho_air*d_star^2*(Sw_CLw+S*CL_max))/kt-lt;
        0
        -(M*g+0.5*rho_air*d_star^2*(Sw_CLw+S*CL_max))/kt+lt;
        0]; %state at equilibrium
u_star=-mu*g+ks*(x_star(1)-ls)-kt*(x_star(3)-lt); %control action at
%equilibrium [N]
nu_star=[0;0]; %sensor noise at equilibrium
r_star=z_max+h; %reference at equilibrium
w_star=[d_star; nu_star; r_star]; %exogenous at equilibrium

y_star=[-g-ks*(x_star(1)-ls)/ms+u_star/ms-rho_air*d_star^2/(2*ms)*...
         (Sw_CLw+S*CL_max)+g;
         x_star(1)-ls]; %output at equilibrium (acceleration [m/s^2] and

```

```

%position [m])
e_star=x_star(1)+x_star(3)-r_star; %regulated output at equilibrium

%% LINEARISED PLANT
%Linearization of A,B1,B2,C,D1,D2,Ce,De1,De2 around equilibrium point
%State
a21=-mr*ks;
a22=-mr*cs;
a23=kt/mu;
a41=ks/mu;
a42=cs/mu;
a43=-kt/mu;
A=[0 1 0 0;
   a21 a22 a23 0;
   0 0 0 1;
   a41 a42 a43 0];

b12=mr;
b14=-1/mu;
b21=-rho_air*d_star/ms*(Sw_CLw+S*CL_max);
B1=[0; b12; 0; b14];
B2=[0 0 0 0;
   b21 0 0 0;
   0 0 0 0;
   0 0 0 0];

%Output
d2_11=-rho_air*d_star/ms*(Sw_CLw+S*CL_max);
C=[-ks/ms -cs/ms 0 0;
   1 0 0 0];
D1=[1/ms; 0];
D2=[d2_11 1 0 0
   0 0 1 0];

%Regulated output
Ce=[1 0 1 0];
De1=0;
De2=[0 0 0 -1];

%Initial conditions
tilde_x_0=x_0-x_star;

%% STATE FEEDBACK CONTROL+INTEGRAL ACTION
%Extended state definition:Ae,B1e, matrixes definition
A_e=[A zeros(n,m)
      Ce zeros(m,m)];
B1_e=[B1;
      De1];

%Definition of C epsilon, D epsilon for the algebraic Riccati equation
Ceps=[1 0 1 0 0
      -ks/ms -cs/ms 0 0 0
      ks/mu cs/mu -kt/mu 0 0];
Depsi=[0
      1/ms
      -1/mu];

%Build of different R,Q matrixes for different scenarios to

```

```

%have different reponses from our control system
eps1max=1.3e-4; %sprung mass height [m]
eps3max=0.008*g; %sprung mass acceleration [m/s^2]
eps4max=0.015*g; %unsprung mass acceleration [m/s^2]

%Definition of Q,R
Q=inv(3*diag([eps1max^2,eps3max^2,eps4max^2]));
umax=5000; %active Force max value [N]
R=inv(umax^2);
R_bar=R+Deps.*Q*Deps; %definition of Rbar to solve ARE

alpha=6; %alpha value to trigger more "instability" in the plant
%if needed to increase robustness
%Definition of auxiliaries matrixes required by MATLAB to solve
%algebraic Riccati equation
Am=A_e+alpha*eye(n+m);
eig_mod=eig(Am);
Em=eye(n+m);
Bm=B1_e;
Gm=0;
Qm=Ceps.*Q*Ceps;
Sm=Ceps.*Q*Deps;
Rm=R_bar;
%Solutions of ARE
[~,Km,lambdam,INFO1] = icare(Am,Bm,Qm,Rm,Sm,Em,Gm);
%Solution check
if INFO1.Report==0
    disp('Accurate solution found for KS,KI')
end
K=-Km;
KS=K(:,1:n); %State feedback gain
KI=K(:,n+1:n+m); %Integral Action gain

%% OBSERVER
%Calculation of Ko using duality of properties of the plant
alpha_d=0; %dual value of alpha for the observer
%Trasposition of main matrixes of the plant
Ad=A';
Bd=C';
Cd=B2';
Dd=D2';

w1max=70; %max possible disturbance
std_pot=1e-5; %potentiometer standard deviation AVIO RACE DIA13 [m]
std_acc=5e-4*g; %accelerometer standard deviation: TE Connectivity
%model 4203 Accelerometer [m/s^2]
%Definition of Rd,Qd,Rdbar matrixes for dual problem
Qd=diag([w1max^2,0,0,0]);
Rd=diag([std_acc^2,std_pot^2]);
Rd_bar=Rd+Dd.*Qd*Dd;

%Definition of auxiliaries matrixes required by MATLAB to solve
%algebraic Riccati equation
Am=Ad+alpha_d*eye(n);
Em=eye(n);
Bm=Bd;
Gm=0;
Qm=Cd.*Qd*Cd;

```

```

Sm= Cd.*Qd*Dd;
Rm=Rd_bar;
[-,Km,-,INFO2] = icare(Am,Bm,Qm,Rm,Sm,Em,Gm);
if INFO2.Report==0
    disp('Accurate solution found for K0')
end

K0=Km.'; %Observer gain
%Calculation of A0, B0, C0, D0
A0=A-K0*C;
B0=B1-K0*D1;
C0=eye(n);
D0=zeros(n,q);

%Initial condition
x0_init=tilde_x_0;

```

5.2 Matlab Non-Linear Plant function

```

function [x_dot,y,e]=F_trial(x,u,w,attiva)
%Real NON LINEAR plant equations
%% VARAIBLES
%STATES
x1=x(1); %zs-zu
x2=x(2); %vs-vu
x3=x(3); %zu
x4=x(4); %vu

%DISTURBANCE
d=w(1); %relative airspeed

%NOISE
nu=w(2:3); %[accelerometer noise, potentiometer noise]

%REFERENCE
r=w(4); %COM height to achieve peak downforce

%% ACTUAL PLANT PARAMETERS
g=9.81; %gravity acceleration [m/s^2]
ks=165000; %suspension spring stiffness [N/m]
kt=2.7e5; %tire spring stiffness [N/m]
M=200; %quarter vehicle mass [kg]
mu=20; %unsprung mass [kg]
ms=M-mu; %sprung mass [kg]
zu_0=0.1; %0 speed value of unsprung mass height [m]
zs_0=zu_0+0.1; %0 speed value of sprung mass height [m]
ls=ms*g/ks+zs_0-zu_0; %suspension unloaded length [m]
lt=M*g/kt+zu_0; %tire unloaded length [m]
cs=3500; %damping coefficient [Ns/m]
mr=(mu+ms)/(ms*mu); %reduced mass of two body sistem [1/kg]

%% STATE DYNAMICS
x_dot=[x2
        mr*(-ks*(x1-ls)-cs*x2+u)+kt*(x3-lt)/mu-f_aero(x1+x3,d,attiva, ...
        x2+x4)/ms
        ];

```

```

x4
-g+ks*(x1-ls)/mu+cs/mu*x2-u/mu-kt*(x3-lt)/mu]; %[vs-vu;as-au;vu;au]

%% MEASUREMENTS
y=[-g-ks/ms*(x1-ls)-cs/ms*x2+u/ms+g-f_aero(x1+x3,d,attiva,x2+x4)/ms;
x1-ls] + nu; %[as;suspension length]+noise

%% CONTROLLED VARIABLE
e=x1+x3-r; %distance from ideal floor height [m]
end

%% SUPPORT FUNCTIONS
function Fz=f_aero(zs,v,attiva,zs_dot)
CL_max=1.9; %max downforce coefficient
z_max=0.023; %optimal ground height [m]
h=0.125; %vertical distance between floor edge and COM [m]
rho_air=1.225; %air density [kg/m^3]
Sw_CLw=0.75; %surface area times downforce coefficient of rear
%wing [m^2]
S=0.4; %floor area [m^2]
z_stall=0.018; %floor height when stall happens [m]
z_recovery=0.01+z_max; %floor height when stall is recovered [m]
CL_stall=0.5; %CL of the floor when stalled
%Stall recovery parameters definition
a=(0.5*rho_air*v^2*S*((z_recovery)/z_max*CL_max.*exp((z_max-...
(z_recovery))/z_max)-CL_stall))/((z_recovery)^2+z_stall^2 ...
-2*z_stall*(z_recovery));
c=0.5*rho_air*v^2*((z_recovery)/z_max*CL_max.*exp((z_max-...
(z_recovery))/z_max)*S+Sw_CLw)+a*(-(z_recovery)^2+2*...
z_stall*(z_recovery));
b=-2*a*z_stall;
%Force definition
if zs-h<z_recovery && zs-h>z_stall && zs_dot>0 && attiva==1 %stall
    %recovery condition
    Fz=a*(zs-h).^2+b*(zs-h)+c;
elseif zs-h≤z_stall %stall condition
    Fz=0.5*rho_air*v^2*(Sw_CLw+S*CL_stall);
else %attached boundry layer condition
    CL=((zs-h)/z_max)*CL_max.*exp((z_max-zs+h)/z_max);
    Fz=0.5*rho_air*v^2*(S*CL+Sw_CLw);
end
end

```

5.3 Matlab Noise generation function

```

function nu=noise_trial(t)
%Function to generate plausible noise on our sensors
g=9.81; %gravity acceleration [m/s^2]
std_pot=1e-5; %potentiometer standard deviation AVIO RACE DIA13-75 [m]
std_acc=5e-4*g; %accelerometer standard deviation: TE Connectivity
%model 4203 Accelerometer [m/s^2]
%Noise vector definition
nu = [std_acc*randn(1);
       std_pot*randn(1)];
end

```

Bibliography

- [1] Andrew Alleyne and J. K. Hedrick. Nonlinear control of a quarter car active suspension. In *1992 American Control Conference*, pages 21–25, 1992.
- [2] Rosheila Darus and Nur Idawati Enzai. Modeling and control active suspension system for a quarter car model. In *2010 International Conference on Science and Social Research (CSSR 2010)*, pages 1203–1206, 2010.
- [3] Fédération Internationale de l'Automobile. *2022 FORMULA 1 TECHNICAL REGULATIONS*.
- [4] Marco Gadola, Daniel Chindamo, Paolo Magri, and Giulia Sandrini. Analyzing porpoising on high downforce race cars: Causes and possible setup adjustments to avoid it. *Energies*, 15(18):6677, 2022.
- [5] Alex Guerrero and Robert Castilla. Aerodynamic study of the wake effects on a formula 1 car. *Energies*, 13(19):5183, 2020.
- [6] Joseph Katz. Aerodynamics of race cars. *Annu. Rev. Fluid Mech.*, 38:27–63, 2006.
- [7] Nicola Mimmo. *Analysis and Design of Control Laws for Advanced Driver-Assistance Systems: Theory and Applications*. Department of Electrical, Electronic and Information Engineering, University of Bologna, 2022.
- [8] Jonathan Noble. Russell suggests active suspension as solution to f1 porpoising issue. *Motosport.com*, feb 2022.
- [9] DJ Purdy and DN Bulman. An experimental and theoretical investigation into the design of an active suspension system for a racing car. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 211(3):161–173, 1997.
- [10] Xin Zhang, Willem Toet, and Jonathan Zerihan. Ground effect aerodynamics of race cars. *Appl. Mech. Rev.*, 2006.