

Master's Degree in Automotive Engineering



Path tracking for electric vehicles with multiple chassis actuators based on classical control techniques

Relatori

Prof. Aldo SORNIOTTI

Prof. Alessandro VIGLIANI

Ing. Cecilia FORMENTO

Dott. Luca DIMAURO

Ing. Davide LAZZARINI

Candidati

Giulio SPURI

October 2025

Abstract

Path tracking is a fundamental component of autonomous vehicle control, responsible for ensuring that a vehicle follows a predefined path with high precision. With the rapid advancement of enabling technologies such as sensor systems, real-time data processing, and vehicle-to-everything (V2X) communication, modern autonomous systems can now perceive complex environments and adapt their behavior accordingly. These innovations have significantly improved motion planning, environmental awareness, and overall driving performance. However, accurate and robust path tracking remains a critical challenge, particularly in dynamic or uncertain scenarios.

This work focuses on the design and evaluation of path tracking strategies based on different linear controllers. In particular, the study explores how they can be effectively integrated with different vehicle actuation mechanisms, such as Four-Wheel Steering (4WS) and Torque Vectoring (TV), to enhance lateral and yaw control during path following. These actuation strategies offer additional degrees of freedom that can be leveraged to improve trajectory adherence, stability, and manoeuvrability, especially at higher speeds or under demanding conditions. Through simulation and experimental analysis, the proposed control frameworks have been fine tuned in order to guarantee a fair comparison and understand how, with these innovative technologies it is possible to increase the level of precision of the path tracking strategies.

Table of Contents

List of Figures	III
1 Introduction and State of the Art	1
2 Path Tracking: Definition and Purpose	4
2.1 Cross-Track Error (CTE)	4
2.2 Heading Error	5
2.3 Trajectory Generation	5
2.4 Performance Metrics	6
3 Description of the System	8
3.1 Automation Level	8
3.2 Powertrain and Chassis	9
3.3 Vehicle Parameters	9
3.4 Sensors and Control Platform	10
3.4.1 Simulink Architecture	11
4 Vehicle Dynamics Model Description	13
4.1 Purpose	13
4.2 Degrees of Freedom	13
4.3 Dynamic Equations	14
4.4 Tire Modeling	15
4.5 Steering Kinematics	15
4.6 Experimental Use	17
4.6.1 Control System Interface	18
5 Controllers Architecture	20
5.1 Design Procedure	20
5.2 PD + feedforward Controller	22
5.2.1 Controller Formulation	23
5.2.2 Feedforward Term and Understeer Gradient	23

5.2.3	Lookahead Distance in Feedforward	24
5.2.4	Tuning Procedure	24
5.3	Four-Wheel Steering (4WS) Controller	26
5.3.1	Enhanced 4WS with Rear Feedforward Contribution	27
5.4	Torque Vectoring (TV) Controller	30
5.5	Four-Wheel Steering with Torque Vectoring (4WS + TV)	34
5.6	LQR Controller (Proposal for future application)	36
5.6.1	LQR Problem Formulation	36
5.6.2	Choosing Q and R	37
5.6.3	Control Structure: Feedback and Feedforward	37
5.6.4	2WS Controller	38
5.6.5	4WS Controller	39
5.6.6	2WS + TV Controller	39
5.6.7	4WS + TV Controller	40
6	Experimental results	42
6.1	Gain Scheduling Approach	42
6.2	PD + FF 2WS	44
6.3	4WS pure feedback	48
6.4	4WS with feedforward and feedback	52
6.5	2WS + TV	56
6.6	4WS + TV	63
6.7	Results of the LQRs	74
7	Experimental vs. Simulation Results	78
7.1	1 st Montecarlo Analysis	78
7.2	2 nd Montecarlo Analysis	83
8	Conclusions and Future Work	87
8.1	Conclusions	87
8.2	Future Work	89
	Bibliography	91

List of Figures

1.1	SAE levels of driving automation, reproduced from [1].	2
2.1	Circuits on which the path tracking has been tested.	6
3.1	PIX vehicle employed for the experimental activity.	8
3.2	CHCNAV Inertial Measurement Unit	10
3.3	Kistler Correxit sensor.	10
3.4	dSPACE MicroAutoBox III for data processing.	11
3.5	Block diagram of the developed Simulink architecture.	12
4.1	Schematization of the model's degrees of freedom.	14
4.2	Ackermann steering geometry scheme [2].	16
4.3	Comparison between experimental tests and simulation results for the validation circuit.	17
4.4	Comparison between experimental tests and simulation results for the validation skidpad.	18
5.1	Optimization of controller gains via <code>surrogateopt</code>	21
5.2	Sensitivity analysis of the controller design parameters.	22
5.3	Rear steering command generated by yaw-rate error feedback in simulation.	28
5.4	Rear steering command generated by yaw-rate error feedback in experimental tests.	28
5.5	Mz domain.	33
6.1	Results obtained from the PD + FF 2WS controller (Soft Circuit).	44
6.2	Results obtained from the PD + FF 2WS controller (Sharp Circuit).	46
6.3	Results obtained from the 4WS controller (Soft Circuit).	48
6.4	Results obtained from the 4WS controller (Sharp Circuit).	50
6.5	Results obtained from the 4WS with feedforward (Soft Circuit).	52
6.6	Results obtained from the 4WS with feedforward (Sharp Circuit).	54
6.7	Results obtained from the 2WS + TV controller (Soft Circuit).	56

6.8	Yaw moment command (Soft Circuit).	57
6.9	Torques at wheel level (Soft Circuit).	58
6.10	Results obtained from the 2WS + TV controller (Sharp Circuit). . .	59
6.11	Yaw moment command (Sharp Circuit).	60
6.12	Torques at wheel level (Sharp Circuit).	61
6.13	Results obtained from the 4WS + TV controller (Soft Circuit). . . .	63
6.14	Results obtained from the 4WS + TV controller (Sharp Circuit). . .	65
6.15	Overlap of results for the soft circuit.	70
6.16	Histogram of cross track error, soft circuit.	71
6.17	Overlap of results for the sharp circuit.	72
6.18	Histogram of cross track error, sharp circuit.	73
6.19	Results obtained from the LQR 2WS (Soft Circuit).	74
6.20	Results obtained from the LQR 4WS (Soft Circuit).	75
7.1	Pairwise scatter of the Monte Carlo parameters.	79
7.2	Distribution of the Monte Carlo parameters.	80
7.3	Peaks of the errors in the first Monte Carlo analysis.	81
7.4	Rms of the errors in the first Monte Carlo analysis.	81
7.5	New trajectory.	84
7.6	Curvature of the new trajectory.	84
7.7	Peaks of the errors in the second Monte Carlo analysis.	85
7.8	Rms of the errors in the second Monte Carlo analysis.	85

Chapter 1

Introduction and State of the Art

In recent years, the development of autonomous vehicle technology has advanced rapidly, driven by progress in artificial intelligence, sensor systems, and control theory. Across a wide range of sectors—from agriculture and industry to defence and disaster response—automation has been adopted to improve efficiency, reduce human labour, and ensure safety in environments that are hazardous or difficult to access. While much of the literature and technological focus has traditionally been centred on wheeled autonomous vehicles, tracked vehicles have gained attention in off-road and uneven terrain scenarios due to their superior traction, load-bearing capability, and manoeuvrability. Despite their advantages, a comprehensive review of path tracking control methods specifically tailored to autonomous tracked vehicles has been largely absent from the literature, a gap that some recent research efforts are beginning to address.

In order to classify the degree of automation in vehicles, the Society of Automotive Engineers (SAE) has defined six levels ranging from Level 0 to Level 5. At **Level 0**, no automation is provided, and the driver is fully responsible for all driving tasks, although driver-assistance features such as warnings or emergency braking may be present. **Level 1** systems introduce either steering or acceleration/braking assistance, but never both simultaneously, requiring the driver’s constant supervision. At **Level 2**, the vehicle can combine steering and acceleration/braking assistance (e.g., adaptive cruise control with lane centering), yet the driver must remain engaged and ready to intervene at any time. **Level 3** represents conditional automation, where the system can fully manage driving within specific conditions, but the driver must be available to take control when requested. **Level 4** vehicles are capable of full driving automation within defined operational domains (e.g., urban shuttles), without expecting driver intervention. Finally, **Level 5** denotes

complete automation under all conditions, where the vehicle no longer requires any driver input for its operation.

SAE J3016™ LEVELS OF DRIVING AUTOMATION							
		SAE LEVEL 0	SAE LEVEL 1	SAE LEVEL 2	SAE LEVEL 3	SAE LEVEL 4	SAE LEVEL 5
What does the human in the driver's seat have to do?		You are driving whenever these driver support features are engaged - even if your feet are off the pedals and you are not steering			You are not driving when these automated driving features are engaged - even if you are seated in "the driver's seat"		
		You must constantly supervise these support features; you must steer, brake or accelerate as needed to maintain safety			When the feature requests you must drive	These automated driving features will not require you to take over driving	
		THESE ARE DRIVER SUPPORT FEATURES			THESE ARE AUTOMATED DRIVING FEATURES		
What do these features do?		These features are limited to providing warnings and momentary assistance	These features provide steering OR brake/ acceleration support to the driver	These features provide steering AND brake/ acceleration support to the driver	These features can drive the vehicle under limited conditions and will not operate unless all required conditions are met		This feature can drive the vehicle under all conditions
Example Features		• Automatic emergency breaking • Blind spot warning • Lane departure warning	• Lane centering OR • Adaptive cruise control	• Lane centering AND • Adaptive cruise control at the same time	• Traffic jam chauffeur	• Local driverless taxi • Pedals/steering wheel may or may not be installed	• Same as level 4, but feature can drive everywhere in all conditions.

Figure 1.1: SAE levels of driving automation, reproduced from [1].

The control architecture of autonomous vehicles is typically structured into three layers: *perception*, *planning*, and *execution*. The perception system acts as the vehicle's sensory input, collecting and interpreting data from the environment. Planning algorithms—responsible for determining the vehicle's path and making decisions such as when to overtake or change lanes—constitute the system's "brain." Finally, execution or control modules ensure that the vehicle follows the planned path accurately, coordinating steering, throttle, and braking mechanisms. As highlighted across the literature, path tracking lies at the heart of the control layer and is essential for maintaining accurate trajectory adherence, particularly under varying road and traffic conditions.

A wide variety of control strategies have been explored for path tracking, ranging from classical approaches like Pure Pursuit, Stanley, and PID controllers to more advanced and robust strategies. Model Predictive Control (MPC) has emerged as a powerful framework due to its ability to anticipate future system states and manage constraints, although it remains computationally intensive and its stability properties under uncertain conditions are still active areas of research. Likewise, Linear Quadratic Regulator (LQR) controllers offer optimal solutions in linearized scenarios, while robust strategies such as Sliding Mode Control (SMC), H_∞ control,

and adaptive methods have been introduced to address modeling uncertainties and external disturbances. Several hybrid and enhanced variants of SMC have been proposed in recent studies to reduce chattering and improve convergence time, such as terminal SMC, super-twisting SMC, and adaptive integral approaches.

A particularly complex case of motion planning and path tracking appears in autonomous overtaking manoeuvres. These scenarios require not only lateral control but also decision-making under uncertainty, where the vehicle must consider the intentions and trajectories of nearby traffic participants. Overtaking involves a sequence of sub-manoeuvres, such as lane changes and passing, each influenced by traffic laws, real-time sensor data, and road conditions. To handle this, researchers have employed techniques such as probabilistic models, decision trees, and Markov Decision Processes, often enhanced with Vehicle-to-Everything (V2X) communication to extend environmental awareness beyond the sensor range. However, despite the sophistication of current sensor fusion and communication technologies, most autonomous systems remain limited to low-speed overtaking due to constraints in prediction accuracy and communication reliability.

In conclusion, while substantial progress has been made in both planning and control for autonomous vehicles, the literature continues to evolve, particularly in terms of robust control strategies and their integration with perception and decision-making layers. There is a growing recognition of the need for more flexible and adaptive controllers that can deal with uncertainty, variable terrain, and complex environments, especially in the case of tracked vehicles and high-level maneuvers like overtaking. The realization of the newest technologies such as four-wheel steering or torque vectoring can help in that and, generally speaking, in all the automotive field in which safety and flexibility have always been a must.

Chapter 2

Path Tracking: Definition and Purpose

Path tracking is a fundamental task in autonomous driving and advanced driver-assistance systems (ADAS), where the objective is to guide the vehicle to follow a predefined path as accurately as possible. The controller continuously adjusts steering and/or wheel torques to minimize the deviation between the vehicle's current state and the reference trajectory.

A **path tracking controller** uses feedback from the vehicle's sensors (e.g., position, velocity, orientation) to correct its motion and align it with the desired path. This process involves minimizing the main types of errors:

2.1 Cross-Track Error (CTE)

The **cross-track error**, denoted usually as e_y , is the **lateral distance** between the vehicle's current position and the closest point on the reference path. It measures how far the vehicle is from the trajectory, perpendicular to the path direction.

Mathematically:

$$e_y = (P_{\text{veh}} - P_{\text{ref}}) \cdot n_{\text{ref}} \quad (2.1)$$

Where:

- P_{veh} is the position of the vehicle.
- P_{ref} is the closest point on the reference path.
- n_{ref} is the unit normal vector to the path at P_{ref} .

A positive e_y means the vehicle is to the left of the path (in the path's local frame), while a negative value indicates it is to the right.

2.2 Heading Error

The **heading error**, often denoted e_ψ , is the **difference in orientation** between the vehicle's current heading and the direction of the reference path at the closest point.

Mathematically:

$$e_\psi = \psi - \psi_{\text{ref}} \quad (2.2)$$

Where:

- ψ is the vehicle's current yaw angle.
- ψ_{ref} is the heading (tangent) of the path at the closest point.

The heading error indicates how well the vehicle is aligned with the path direction. Even if the vehicle is on the path (zero cross-track error), a non-zero heading error implies it is not aligned and may deviate in the future.

2.3 Trajectory Generation

The reference trajectory used for path tracking validation consists of a circuit including segments with varying curvature profiles, such as straights, constant-radius curves, and clothoids. This trajectory was generated using a custom MATLAB script designed to allow full customization of the path in a modular way.

The trajectory is defined as a sequence of segments, where each segment can be independently configured by specifying:

- The segment type: straight line, circular arc (constant curvature), or clothoid (linearly varying curvature)
- The segment length
- The curvature (for curves and clothoids)

This flexible approach enables the design of realistic test tracks, including complex combinations of turns and straights, suitable for assessing the performance of path tracking controllers under diverse dynamic conditions.

The resulting trajectory provides at each point:

- Positional references (x, y)
- Orientation reference (ψ_{ref})
- Curvilinear cumulative coordinate (s)

These quantities are essential for computing cross-track and heading errors during controller evaluation. The s coordinate is also required for the feedforward term, which will be discussed later.

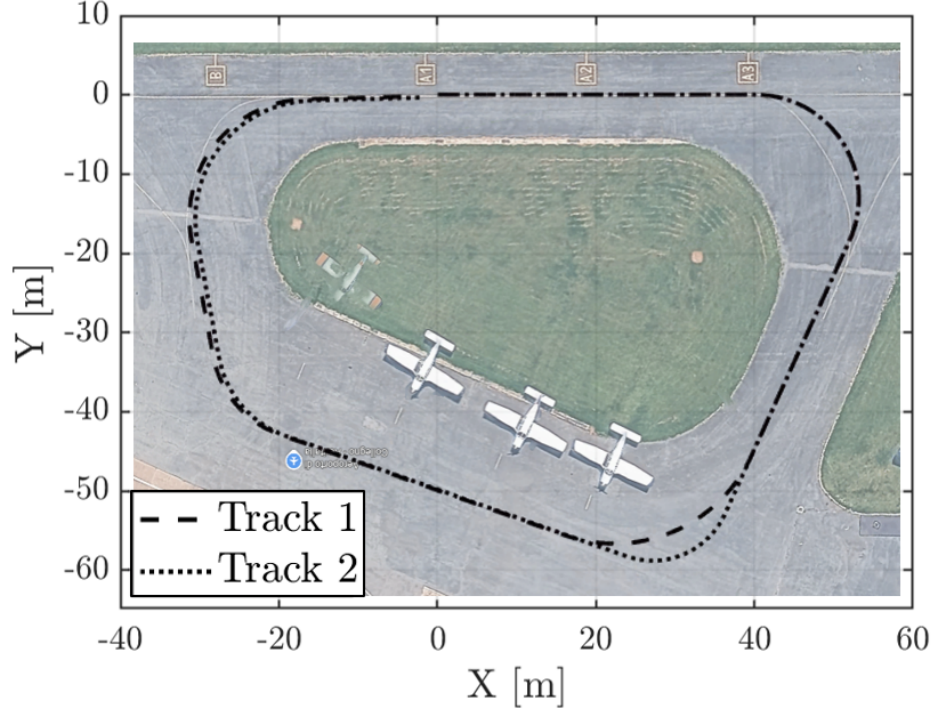


Figure 2.1: Circuits on which the path tracking has been tested.

From this point forward, we will refer to Track 1 as to "Soft circuit" and to Track 2 as to "Sharp circuit". The reason is straightforward, the sharp has a knee curve, harder to follow, which generates higher lateral accelerations.

2.4 Performance Metrics

To quantitatively assess the performance of a path tracking controller, a set of **Key Performance Indicators (KPIs)** is adopted. These metrics provide a numerical evaluation of how accurately the vehicle follows the reference trajectory and how stable its dynamic behavior remains during the maneuver.

The main KPIs considered are:

- **Maximum value** of the error signals, which highlights the worst-case deviation from the reference.

- **Root Mean Square (RMS) value** of the error signals, which gives an overall measure of the tracking accuracy throughout the entire maneuver.

In particular, the monitored quantities are:

- Cross-Track Error (e_y): provides a direct measure of the lateral deviation from the reference path.
- Heading Error (e_ψ): indicates the vehicle's orientation mismatch with respect to the path tangent.
- Sideslip Angle (β): provides information about the lateral stability of the vehicle, complementing the tracking accuracy indicators.

The combined analysis of these metrics allows for an objective evaluation of both **trajectory-following performance** and **vehicle stability**, ensuring that the controller achieves accurate path tracking without compromising dynamic safety.

Chapter 3

Description of the System

The case study vehicle is the **PIXKIT**, a platform equipped with four in-wheel electric motors and featuring four-wheel steering (4WS) capability. The PIXLOOP HOOKE drive-by-wire chassis, developed by PIX Moving, is a fully drive-by-wire platform specifically designed for engineers and researchers working on low-speed Level 4 autonomous driving applications, particularly in controlled environments such as the Aeroclub of Turin, where it has been tested since its arrival in Italy.



Figure 3.1: PIX vehicle employed for the experimental activity.

3.1 Automation Level

Level 4 and Level 5 represent the highest degrees of automation in self-driving vehicles. Level 4, also known as “high automation,” allows the vehicle to drive itself under specific conditions without human intervention, such as within a defined area or during certain weather conditions. Level 5, or “full automation,” means

the vehicle can handle all driving tasks in all conditions, without any human input, effectively removing the human driver from the equation.

For the considered vehicle, an operator was required to remain within the operational range of the remote controller (50 m), both for monitoring and intervening in case of unexpected events, and because otherwise the connection with the vehicle would be lost.

3.2 Powertrain and Chassis

The system is powered by a 72 V main battery pack, complemented by a 12 V maintenance-free lead-acid battery for auxiliary functions. Constructed from high-strength steel, the chassis ensures durability and structural integrity under a variety of operating conditions. Steering and propulsion are fully electronically controlled and can be managed via remote control or integrated into an autonomous driving system.

3.3 Vehicle Parameters

The main vehicle parameters are summarized in Table 3.1.

Table 3.1: Main physical parameters of the PIXKIT platform.

Parameter	Symbol	Value
Vehicle mass	m	510 kg
Yaw moment of inertia	I_z	948 kg · m ²
Roll moment of inertia	I_x	325 kg · m ²
Wheel radius	R	0.305 m
Front axle to CG	a_f	0.975 m
Rear axle to CG	a_r	0.925 m
Front track width	b_f	1.470 m
Rear track width	b_r	1.470 m
CG height	h_{CG}	0.300 m
Roll center height	h_{roll}	0.110 m
Roll stiffness (front/rear)	k_{roll}	9048 N · m/rad
Roll damping (front/rear)	c_{roll}	711 N · m · s/rad
Sprung mass	m_s	257 kg
Wheel inertia	J_w	1.5 kg · m ²
Steering ratio	—	0.053

3.4 Sensors and Control Platform

The vehicle is equipped with a comprehensive set of sensors and control hardware to enable accurate state estimation and advanced control development. The sensor suite includes:

- **CHCNAV Inertial Measurement Unit (IMU)** for measuring accelerations and angular rates.



Figure 3.2: CHCNAV Inertial Measurement Unit .

- **Kistler Correvit sensor** for measuring longitudinal and lateral velocity components with high precision.



Figure 3.3: Kistler Correvit sensor.

- **Additional perception sensors**, including LiDAR, radar, ultrasonic sensors, and cameras, which are available on the vehicle but not employed in the present activity.

For rapid control prototyping, the platform uses a **dSPACE MicroAutoBox III** real-time system, which executes the vehicle dynamics control algorithms and interfaces with the actuators via the drive-by-wire system.



Figure 3.4: dSPACE MicroAutoBox III for data processing.

3.4.1 Simulink Architecture

The Simulink model developed for this work is organized according to the scheme reported in Figure 3.5. The *Inputs* block collects the signals provided by the vehicle sensors. Each of these signals is properly filtered before being processed: in fact, a sensitivity analysis was performed for all the quantities relevant to path tracking in order to select the most suitable filtering strategy, taking into account the trade-off between signal cleanliness and the delay introduced by the filter.

The filtered signals are then fed into the subsequent subsystems. If the input data generates an emergency condition — for instance, the activation of the anti-collision bar — the signal is redirected to the *Safety System*, which forces a safety stop and overrides the controller outputs. Otherwise, the data is sent to the *Control System* block, which represents the core of the path tracking algorithm. Within this subsystem, the control strategy can be manually selected among different alternatives, such as a Linear Quadratic Regulator (LQR), a Proportional-Derivative (PD) controller, or a Neural Network (NN).

Finally, the control actions computed by the selected strategy are transmitted to the *Outputs* subsystem, which interfaces with the vehicle actuators and applies the commanded steering and traction/braking inputs.

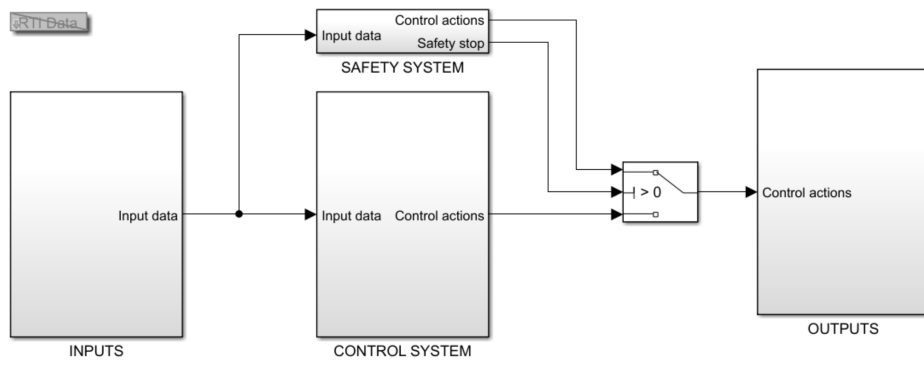


Figure 3.5: Block diagram of the developed Simulink architecture.

Chapter 4

Vehicle Dynamics Model Description

The vehicle model implemented in MATLAB is a nonlinear **9 degrees-of-freedom (DOF)** representation designed for simulation and experimental validation within the context of path tracking control. This model captures the essential dynamic behaviour of a four-wheel vehicle equipped with **four-wheel steering (4WS)** and **individual wheel torque control**, making it suitable for advanced control strategies such as LQR and MPC for autonomous driving.

4.1 Purpose

This model is used as a high-fidelity simulation tool to replicate experimental conditions during the development and tuning of automated path tracking algorithms. Its flexibility allows the evaluation of different control strategies, vehicle configurations (e.g., 4WS, torque vectoring), and actuator commands (steering and torque inputs).

4.2 Degrees of Freedom

The model includes the following **7 DOFs**:

1. Longitudinal velocity V_x
2. Lateral velocity V_y
3. Yaw rate $\dot{\psi}$
4. Wheel angular speeds ω_i for each wheel (FL, FR, RL, RR)

Additionally, derived quantities such as longitudinal and lateral accelerations (a_x, a_y) and the vehicle side-slip angle β are computed for accurate motion estimation and control evaluation.

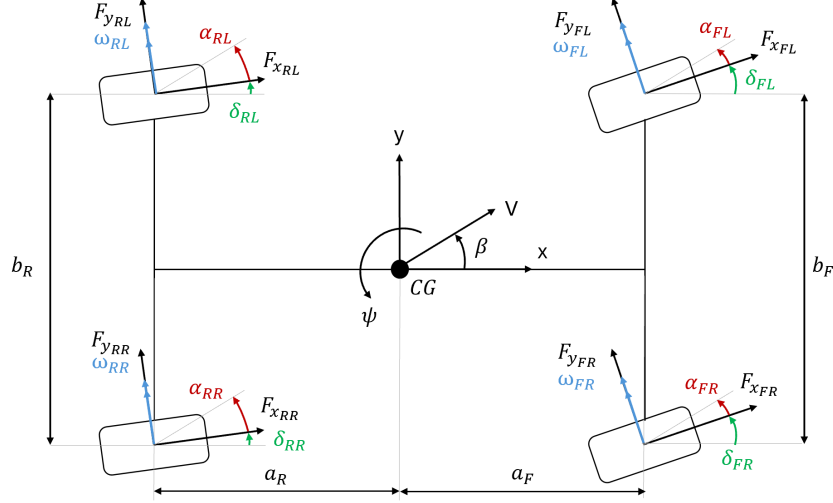


Figure 4.1: Schematization of the model's degrees of freedom.

4.3 Dynamic Equations

The nonlinear dynamics of the system are governed by the following differential equations:

Longitudinal dynamics:

$$\dot{V}_x = \frac{1}{m} \left(\sum_{i=1}^4 F_{x_i} \cos \delta_i - F_{y_i} \sin \delta_i \right) - C_{rr4} V_x^2 + V_y \dot{\psi} \quad (4.1)$$

Lateral dynamics:

$$\dot{V}_y = \frac{1}{m} \left(\sum_{i=1}^4 F_{x_i} \sin \delta_i + F_{y_i} \cos \delta_i \right) - V_x \dot{\psi} \quad (4.2)$$

Yaw dynamics:

$$\ddot{\psi} = \frac{1}{I_z} \left(\sum (a_i F_{x_i} \sin \delta_i + a_i F_{y_i} \cos \delta_i) - \sum (b_i F_{x_i} \cos \delta_i - b_i F_{y_i} \sin \delta_i) \right) \quad (4.3)$$

Wheel rotational dynamics:

$$\dot{\omega}_i = \frac{1}{J_w} (T_i - F_{x_i} R_i + M_{y_i}) \quad (4.4)$$

4.4 Tire Modeling

The tire forces F_{x_i} , F_{y_i} and aligning moments M_{y_i} are calculated using a **reduced Pacejka 1996 tire model**, extended to account for:

- Camber effects
- Load sensitivity
- Combined slip conditions
- Longitudinal and lateral load transfer

Vertical loads are dynamically computed based on vehicle accelerations and roll motion, influencing tire performance.

4.5 Steering Kinematics

An **Ackermann-based steering geometry** is implemented, supporting:

- Conventional front-wheel steering (2WS)
- Four-wheel steering (4WS), with both in-phase and counter-phase configurations

Inner and outer wheel angles are corrected according to Ackermann geometry for both axles.

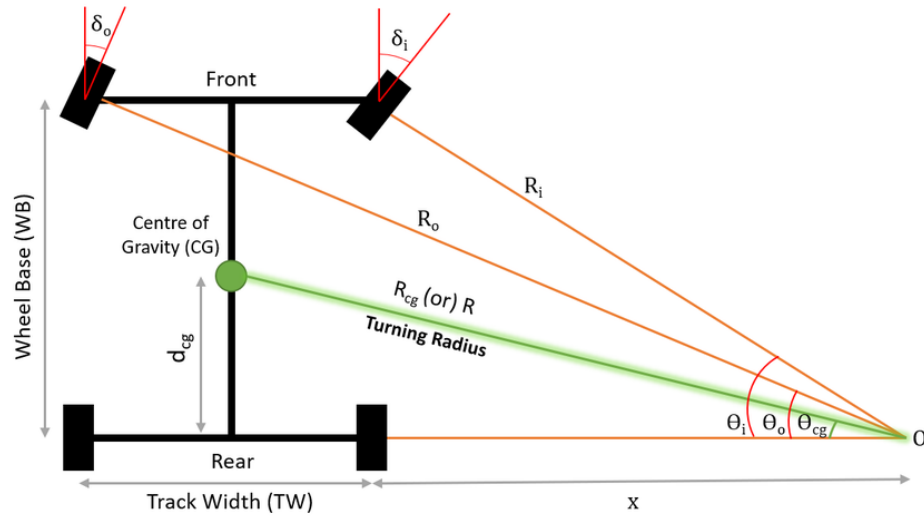


Figure 4.2: Ackermann steering geometry scheme [2].

4.6 Experimental Use

This model is used as a digital twin for:

- Simulating vehicle response for given control inputs
- Validating and tuning the control architecture
- System identification and parameter estimation

Its compatibility with sensor data and actuator commands (steering angles and torques) makes it ideal for Hardware-in-the-Loop (HIL) and Software-in-the-Loop (SIL) testing.

In order to properly tune all the parameters needed in the model an optimizer has been adopted in order to achieve, in simulation, the closest results possible to those obtained in a rounded-like circuit as well as a skidpad manoeuvre with the real vehicle. The good agreement between simulations and experiments makes the model a valuable tool for control system assessment.

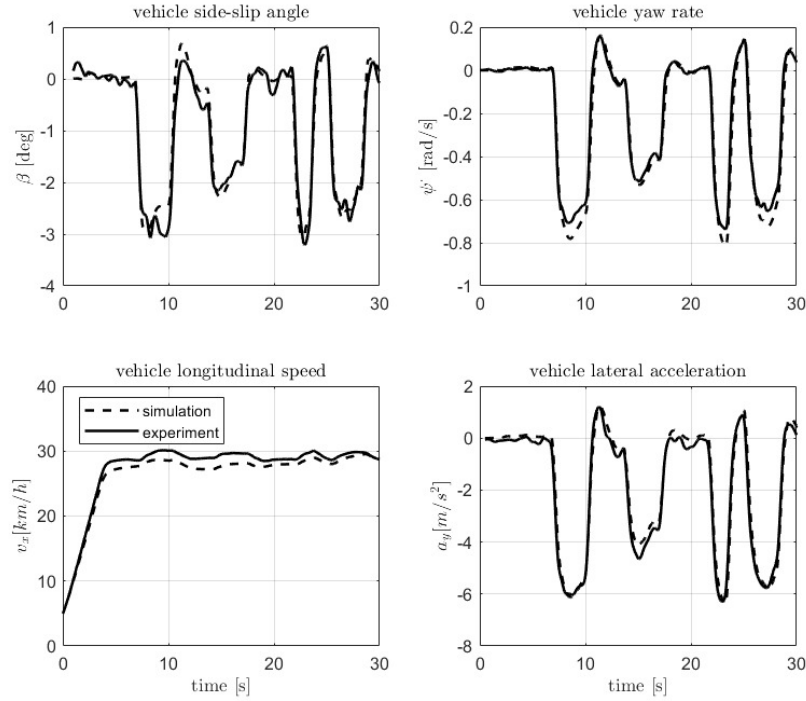


Figure 4.3: Comparison between experimental tests and simulation results for the validation circuit.

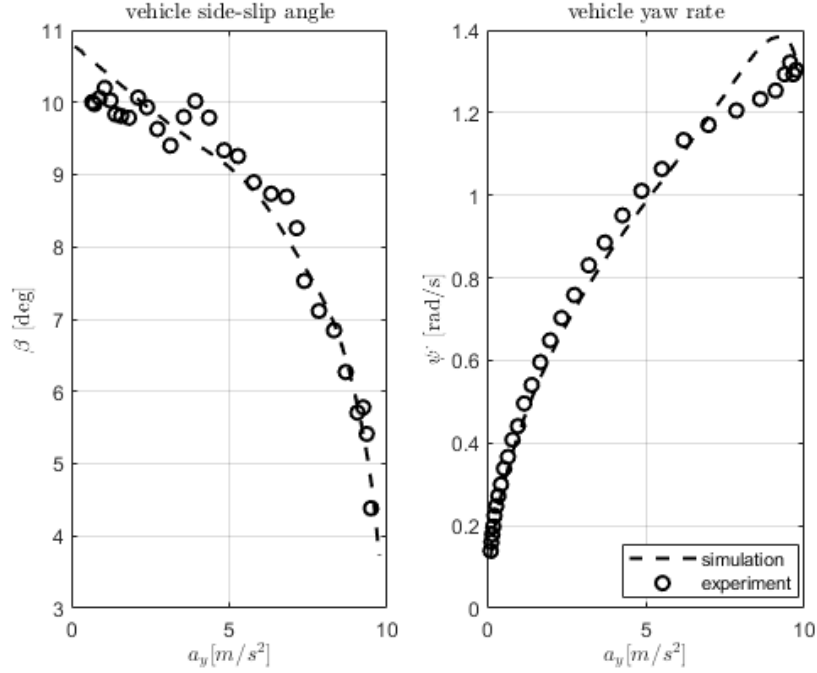


Figure 4.4: Comparison between experimental tests and simulation results for the validation skidpad.

4.6.1 Control System Interface

The interface of the control subsystem follows a structure that is common to all linear controllers and, to a large extent, also to nonlinear ones. The first step concerns the acquisition of the vehicle position through GPS sensors. However, the GPS operates at a very low sampling frequency (5 Hz), which results in a sparse and noisy signal. To improve its effectiveness, a moving average filter is applied to both latitude and longitude signals. The filtered data are then converted into planar coordinates (x, y) using the MATLAB function `latlon2local`, which transforms geographic coordinates into a local Cartesian reference frame by exploiting relative differences with respect to a chosen initial point.

Since the trajectory is defined in the (x, y) plane, it is necessary to correctly initialize the vehicle orientation. The yaw angle ψ is computed at each time step using the `atan2` function, which provides values in the interval $[-\pi, \pi]$. Two issues must be addressed:

- **Initial offset:** the value of ψ obtained from latitude and longitude is not zero at the initial position, but depends on the absolute location. Therefore, the initial offset ψ_{init} must be subtracted from the instantaneous yaw angle.

- **Angle discontinuity:** the periodicity of `atan2` introduces a jump when crossing the $-\pi$ to π boundary. This discontinuity must be handled, since it propagates into the heading error and especially into its derivative, generating spurious spikes.

Once the vehicle position (x, y) is available, the reference trajectory point is selected as the one with the minimum Euclidean distance from the current position. Consequently, the following variables are available at each time step:

- x_{GPS}, y_{GPS} : current vehicle position in the local frame,
- $x_{\text{ref}}, y_{\text{ref}}$: selected reference point,
- ψ : actual vehicle yaw angle,
- ψ_{ref} : reference yaw angle,
- s : curvilinear coordinate associated with the chosen reference point.

These variables allow the computation of the path tracking errors (cross-track error, heading error, etc.), which in turn are exploited by the controller to generate the desired control actions.

Chapter 5

Controllers Architecture

5.1 Design Procedure

The design of all the proposed controllers has been carried out starting from the validated **Simulink model**, on which a surrogate-based optimization procedure was performed using the `surrogateopt` algorithm. A total of **100 iterations** were selected, of which **75 random** and **25 adaptive**, as a good compromise between optimization effectiveness and computational cost. These optimizations already required a considerable amount of time: with more iterations, the computational effort would have increased substantially without providing significant improvements in terms of KPI, since the results already obtained can be considered sufficiently **asymptotic**.

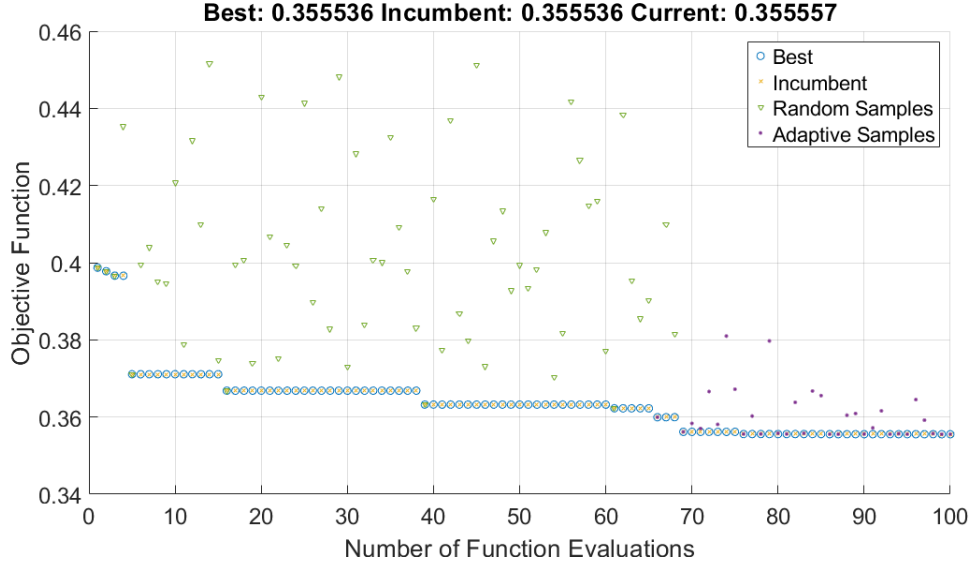


Figure 5.1: Optimization of controller gains via `surrogateopt`.

In any case, the path tracking activity is mainly focused on the **experimental results**. Although the model has been properly configured and validated, this is a matter of centimeter-level precision. Therefore, the results obtained through such optimization represent an excellent starting point, but they should not be considered perfect, since a certain discrepancy between the model and the real experiment will always exist. This is especially true for an application where the objective is to optimize performance to the centimeter.

So far, two different tuning procedures have been described:

1. **Computer-based optimization** via `surrogateopt`.
2. **Manual real-life tuning**, performed directly on the experimental vehicle.

In addition to these two approaches, another powerful tool was exploited in simulation: a **sensitivity analysis** of the main design parameters. Using simple `parfor` loops, each parameter was varied **individually** (no overlapping effects) with respect to the value returned by the optimizer, according to the following scaling factors:

$$0.2 ; 0.4 ; 0.6 ; 0.8 ; 1.0 ; 1.2 ; 1.4 ; 1.6 ; 1.8 .$$

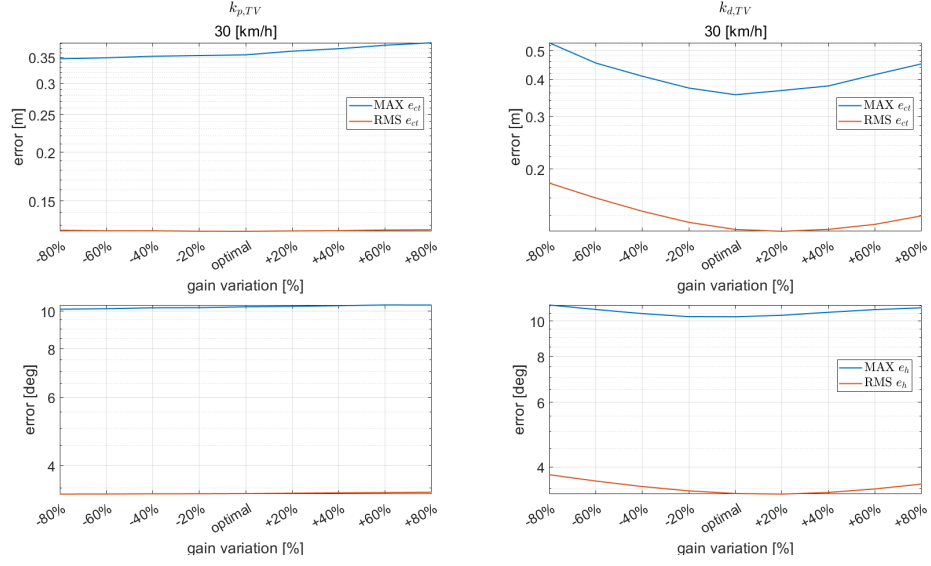


Figure 5.2: Sensitivity analysis of the controller design parameters.

This tool proved to be effective for two main reasons:

1. If the resulting trend does not show a **V-shape** around the optimum (as in Figure 5.2), it means that the optimizer provided a solution that is only optimal within the imposed boundaries, but the true optimum could lie outside the selected domain. In this case, a further optimization could be considered.
2. Beyond the numerical outcome, conducting a sensitivity analysis is a valuable tool to understand the **trend of parameter variations**: whether they lead to instability, whether some changes are more harmful than others, and which parameters have the strongest influence on performance.

5.2 PD + feedforward Controller

The Proportional-Derivative (PD) controller represents the starting point of the path tracking activity carried out in this research group. In fact, the first attempts to achieve reliable automated guidance were based on a PD feedback controller enriched with a feedforward contribution. Despite its conceptual simplicity, this architecture proved to be particularly effective and provided a solid basis for subsequent developments.

5.2.1 Controller Formulation

The general structure of the PD controller is defined by the following law:

$$u(t) = K_p e_{ct}(t) + K_d \dot{e}_{ct}(t) \quad (5.1)$$

where $e_{ct}(t)$ is the cross-track error, i.e. the lateral deviation of the vehicle from the reference path, and $\dot{e}_{ct}(t)$ is its time derivative. The proportional term $K_p e_{ct}(t)$ acts to reduce the magnitude of the error, while the derivative term $K_d \dot{e}_{ct}(t)$ provides damping and anticipates variations, thus improving stability and transient response.

In the context of vehicle path tracking, this simple formulation is enriched with a feedforward component based on the curvature of the reference trajectory. The complete steering command can therefore be expressed as:

$$\delta = K_p e_{ct}(t) + K_d \dot{e}_{ct}(t) + \delta_{ff} \quad (5.2)$$

where δ_{ff} is the feedforward contribution.

5.2.2 Feedforward Term and Understeer Gradient

The design of the feedforward term is crucial in order to eliminate steady-state errors. For the PD controller, it was defined according to the classical formulation:

$$\delta_{ff} = K_l \kappa (1 + K_{us} v^2) \quad (5.3)$$

where κ is the path curvature, v the vehicle longitudinal velocity, K_{us} the understeer gradient, and K_l a scaling constant related to the vehicle wheelbase and steering ratio.

This expression reflects the well-known relation between steering demand and curvature in steady-state cornering, corrected by the understeer gradient K_{us} to account for deviations from neutral steering behavior. In practice:

- If $K_{us} > 0$, the vehicle exhibits an understeering tendency, requiring a steering angle larger than the ideal geometric one.
- If $K_{us} = 0$, the vehicle behaves neutrally and δ_{ff} reduces to the purely geometric term.
- If $K_{us} < 0$, the vehicle shows an oversteering behavior, demanding a smaller steering angle for the same curvature.

The determination of K_{us} is not straightforward, since it requires knowledge of the cornering stiffnesses of both front and rear axles. Ideally, these stiffnesses should be obtained from experimental $\alpha - F_y$ curves (lateral slip angle vs lateral force).

In absence of such direct measurements, an approximate estimation was carried out starting from the Pacejka tire model. Using the Magic Formula parameters, the initial slope of the $F_y(\alpha)$ curve can be extracted, which corresponds to the cornering stiffness C_α . Once C_f and C_r are available, the understeer gradient is computed as:

$$K_{us} = \frac{W_F}{C_F} - \frac{W_R}{C_R} \quad (5.4)$$

where W_F and W_R are the static load distributions on the front and rear axles, respectively. This activity required a dedicated analysis, since the lack of direct experimental tire data forced us to rely exclusively on model-based evaluations.

5.2.3 Lookahead Distance in Feedforward

An additional element introduced in the PD + FF controller is the so-called *lookahead distance*, denoted here as d_{shift} . Unlike the typical use of lookahead in feedback control, in this implementation d_{shift} was applied only to the feedforward term. The rationale is to emulate a more realistic driving behavior, where the steering input anticipates the curve rather than reacting to it only when the vehicle reaches a certain deviation. By shifting the reference point forward along the trajectory, the feedforward contribution δ_{ff} becomes anticipative, reducing the delay in steering action and improving the overall smoothness of the maneuver.

5.2.4 Tuning Procedure

A systematic tuning campaign was conducted for the PD + FF controller across the entire operational velocity range of the vehicle, from 5 km/h up to 30 km/h. This activity went far beyond a simple gain scheduling procedure. While proportional and derivative gains were indeed adjusted depending on velocity, additional tuning was required for the parameters involved in the feedforward term, namely K_{us} and the lookahead distance d_{shift} .

In particular:

- At low speeds, the influence of K_{us} is limited, but the lookahead distance becomes crucial to prevent late steering actions.
- At higher speeds, the quadratic contribution $K_{us}v^2$ significantly affects the steering angle demand, making the correct tuning of K_{us} essential for stability and accuracy.
- The parameter d_{shift} was calibrated so as to balance anticipation and stability: too small values resulted in delayed steering, whereas excessively large values led to oscillatory behavior.

The final result of this tuning activity was a PD + FF controller capable of handling the entire velocity range, providing a good compromise between simplicity, robustness, and performance. This controller therefore played a central role in the initial phase of the research project and remains a reference baseline for evaluating the more advanced strategies introduced later.

Moreover, the effort invested in this tuning activity is not only relevant to the specific path tracking task. The PD + FF controller, thanks to its clear structure and relatively limited computational demand, constitutes a suitable basis for combined activities involving both path tracking and adaptive cruise control (ACC). In such a framework, the steering control must be seamlessly integrated with longitudinal control actions, and the robustness achieved through the tuning process ensures that lateral and longitudinal dynamics can coexist without introducing excessive coupling effects or instability risks.

In addition, the availability of a well-documented and extensively validated controller is of great value for future developments on the PIX platform. Researchers and engineers working on subsequent stages of the project can rely on this reference implementation either as a benchmark for evaluating more advanced controllers, or as a safe fallback solution in case of integration issues. The PD + FF controller thus represents not only a milestone in the present work, but also a cornerstone for the continuity and scalability of research activities on the vehicle.

5.3 Four-Wheel Steering (4WS) Controller

Starting from the baseline PD + FF architecture, an additional control strategy was developed by exploiting the rear steering capability of the vehicle. This controller was designed and validated at a fixed velocity of 30 km/h, which represents the maximum speed attainable by the platform and, consequently, the most demanding operational condition. From this point onwards, all comparison activities between different controllers are therefore restricted to this velocity, as it constitutes the most meaningful benchmark for assessing performance improvements.

The design philosophy behind the 4WS controller did not involve a complete rethinking of the previous scheme. On the contrary, the PD + FF formulation was retained for the front axle, so as to reduce the complexity of the design process and to ensure a direct comparability with the baseline controller. Specifically:

$$\delta_f = K_p e_{ct} + K_d \dot{e}_{ct} + \delta_{ff}, \quad (5.5)$$

where the feedforward contribution δ_{ff} was implemented through the same spatial shift approach described earlier.

On the rear axle, the steering command was instead obtained from a proportional-derivative action on the yaw rate error:

$$\delta_r = K_p^r e_{\dot{\psi}} + K_d^r \dot{e}_{\dot{\psi}}, \quad (5.6)$$

where the reference yaw rate $\dot{\psi}_{ref}$ was computed as the time derivative of the desired yaw angle ψ_{ref} , associated with the point of the reference trajectory selected through the lookahead mechanism, while the actual yaw rate $\dot{\psi}$ was directly measured by the Kistler sensor.

During the tuning process, the introduction of a lookahead distance was also evaluated for the rear control loop. However, the best results were consistently obtained without any anticipation, as the inclusion of a lookahead term tended to deteriorate stability. As a consequence, the rear control action remained inherently reactive, being directly tied to the instantaneous yaw rate error.

This 4WS controller proved to be highly effective in practice, leading to a marked reduction in lateral error across both tested circuits. Nonetheless, the absence of lookahead in the yaw rate feedback introduced a slight delay in the corrective action, which manifested as an increase in the angular error. This suggests that, while the current configuration already provides a significant performance improvement with relatively small rear steering angles, further enhancements could be achieved by complementing the yaw rate regulation with an explicit control action on the heading error.

5.3.1 Enhanced 4WS with Rear Feedforward Contribution

While the baseline 4WS controller already proved effective in reducing lateral error, a more refined version was developed in order to improve the rear steering behaviour within cornering maneuvers. The analysis of both simulation and experimental results revealed that the pure yaw-rate-based control law was able to provide an initial corrective spike in counter-phase at the beginning of the curve, which effectively amplified the rotational dynamics of the vehicle. However, this beneficial contribution was not maintained throughout the manoeuvre: the corrective action tended to vanish after the initial transient and, in some cases, even produced an in-phase steering input at the end of the curve, counteracting the desired effect. To overcome this limitation, an additional feedforward contribution was introduced for the rear steering command:

$$\delta_r = K_p^r e_\psi + K_d^r \dot{e}_\psi + \delta_r^{ff}, \quad (5.7)$$

$$\delta_r^{ff} = K \cdot ffGain. \quad (5.8)$$

The feedforward term δ_r^{ff} was designed to ensure a persistent contribution during cornering, preventing oscillations and providing a more stable corrective behaviour. The calibration of the parameter $ffGain$ followed the same procedure already adopted for the other gains: an initial optimization through `surrogateopt`, followed by a sensitivity analysis and finally a fine-tuning through trial-and-error during experimental tests. Figures 5.3 and 5.4 illustrate the behaviour of the rear steering command generated by the original yaw-rate feedback controller, both in simulation and in real-world tests. It can be observed that while the first spike is beneficial, the subsequent phases of the manoeuvre highlight the necessity of a feedforward term to sustain the corrective action.

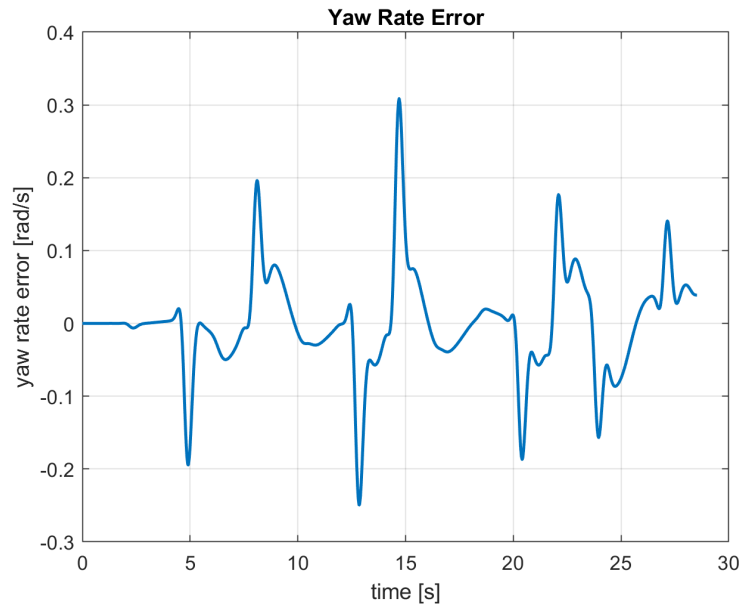


Figure 5.3: Rear steering command generated by yaw-rate error feedback in simulation.

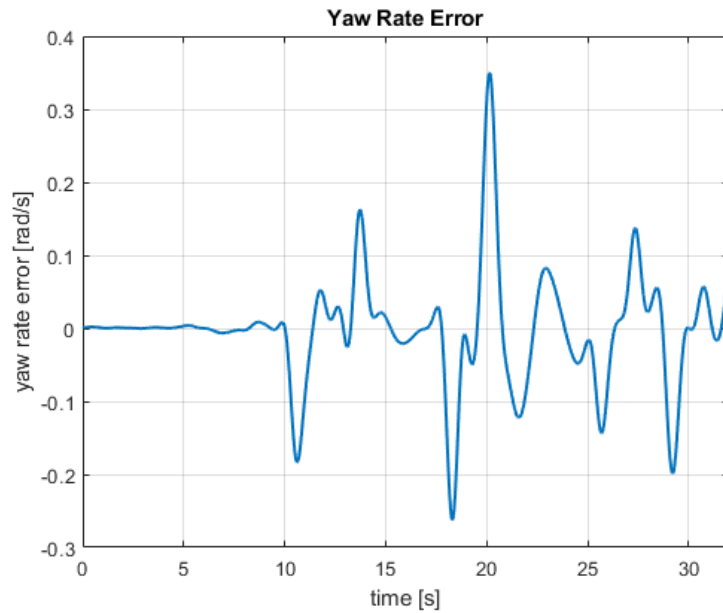


Figure 5.4: Rear steering command generated by yaw-rate error feedback in experimental tests.

Overall, the introduction of the rear feedforward contribution significantly improved the behaviour of the 4WS controller. The rear steering action became more consistent and better aligned with the vehicle dynamics, effectively sustaining the beneficial counter-phase effect throughout the entire curve. This resulted in smoother manoeuvres, reduced oscillations, and enhanced path tracking accuracy, without requiring large steering angles at the rear axle.

5.4 Torque Vectoring (TV) Controller

Torque Vectoring (TV) is a control strategy aimed at generating an additional yaw moment M_z through a differentiated distribution of the wheel torques. By actively shaping the yaw dynamics of the vehicle, torque vectoring significantly increases stability and robustness, especially under demanding manoeuvres. This technique, once considered difficult to implement due to the constraints of conventional powertrains, has become much more feasible with the advent of electric vehicles. In particular, the adoption of four in-wheel motors unlocks an exceptionally wide margin of application, since each wheel torque can be commanded independently and with high bandwidth.

A well-known drawback of in-wheel architectures is the increase in unsprung masses, as the electric machines are directly integrated into the wheel hubs. Higher unsprung mass can be potentially detrimental because it amplifies the transmission of road irregularities to the vehicle body, reducing ride comfort and deteriorating tire-road contact at high frequencies. These effects may negatively impact both vertical dynamics and handling performance. However, several studies (see, for example [3]) have shown that the overall vehicle dynamics of in-wheel and on-board motor architectures are largely comparable. In particular, the work by H. de Carvalho Pinheiro, A. Messana and M. Carello demonstrates through multibody analyses that the impact of increased unsprung mass on lateral and vertical dynamics remains limited, suggesting that the handling benefits of TV can be fully exploited without substantial degradation of comfort or safety.

In the present implementation, the TV controller was designed as a dual proportional action:

$$M_z = K_{he} e_\psi + K_{yr} e_{\dot{\psi}}, \quad (5.9)$$

where e_ψ is the heading error and $e_{\dot{\psi}}$ is the yaw rate error. The resulting yaw moment demand M_z is then mapped into individual wheel torques according to the relation:

$$T_{ij} = M_z \cdot \frac{2r_w}{t} \cdot \begin{bmatrix} 1 & -1 & 1 & -1 \end{bmatrix}, \quad (5.10)$$

where r_w denotes the wheel radius and t the track width. The four elements of the vector correspond respectively to the front-left, front-right, rear-left, and rear-right wheels.

The obtained wheel torque contributions are superimposed onto the baseline longitudinal torques generated by the cruise control system. The latter, in the present work, is implemented as a simple proportional controller on the velocity tracking error. This choice was considered sufficient, as the focus of the study is primarily on the lateral dynamics of the vehicle.

Torque Vectoring represents a control strategy aimed at generating an additional yaw moment through the differential distribution of wheel torques. This technique

is particularly relevant in the context of electric vehicles, where the presence of in-wheel motors or, more generally, independent torque control on each axle enables a fine modulation of the longitudinal forces. In the case of path tracking, however, the role of the TV must be carefully reconsidered. Unlike steering-based control actions, a feedforward contribution is not meaningful in this context. The reason lies in the nature of the yaw moment demand: depending on the specific maneuver, the control objective may either require the amplification of the vehicle's natural yaw response (for instance, to improve agility in tight cornering) or, conversely, the suppression of the yaw moment (to stabilize the vehicle in conditions close to instability). Consequently, the control action cannot be defined a priori with a fixed sign, as the same feedforward law could be beneficial in one scenario and harmful in another. For this reason, the torque vectoring input must necessarily follow the instantaneous yaw error, ensuring that the correction is dynamically adapted to the vehicle's actual behavior.

This consideration highlights an important conceptual difference between torque vectoring and steering-based controllers. While the latter can rely on a predictable geometric relationship between steering angle, curvature and trajectory, torque-based control acts directly on the balance of lateral forces, which is strongly influenced by nonlinear effects such as tire saturation, load transfer and road friction conditions. As a result, the control allocation must remain fully feedback-driven, with the possibility of integrating adaptive or gain-scheduled structures to cope with varying operating conditions.

Moreover, the adoption of TV is not without trade-offs. On one side, it offers clear advantages in terms of stability, maneuverability and the possibility of achieving a decoupled control of yaw and lateral dynamics. On the other side, the actuation of differential torques requires high-bandwidth power electronics and accurate current control of the electric machines, in addition to raising concerns about the additional energy consumption compared to pure steering interventions. Another practical limitation lies in the asymmetric thermal load on the electric machines, which could lead to problems in prolonged maneuvers if not properly managed.

In summary, torque vectoring represents a powerful complement to conventional steering strategies for path tracking, but its effective integration requires a purely feedback-oriented formulation. Only by linking the yaw moment demand directly to the yaw rate error can the controller ensure that the corrective action is properly aligned with the stability and maneuverability requirements of the vehicle.

Considering that our vehicle is four-wheel-drive, equipped with an independent electric motor for each wheel, it is assumed that every motor can provide the same maximum torque in traction and in regeneration:

$$T_{i,\max}^+(\omega_i) = |T_{i,\max}^-(\omega_i)| = T_{i,\max}(\omega_i), \quad i \in \{FL, FR, RL, RR\}.$$

The sum of the maximum deliverable torques is therefore

$$S(\omega) = \sum_{i=1}^4 T_{i,\max}(\omega_i).$$

Given a reference value of total longitudinal torque T^{req} , the residual margin available for yaw moment generation results in

$$S_{\text{residual}}(\omega, T^{req}) = S(\omega) - |T^{req}|,$$

with the condition $S_{\text{residual}} \geq 0$.

The geometric gain of conversion between lateral torque differences and yaw moment is defined as

$$K_{tv} = \frac{t}{2 R_w},$$

where t is the track width and R_w the wheel radius.

Therefore, the yaw moment saturation limits are

$$M_{z,\max}(\omega, T^{req}) = K_{tv} \max(0, S(\omega) - |T^{req}|),$$

$$M_{z,\min}(\omega, T^{req}) = -M_{z,\max}(\omega, T^{req}).$$

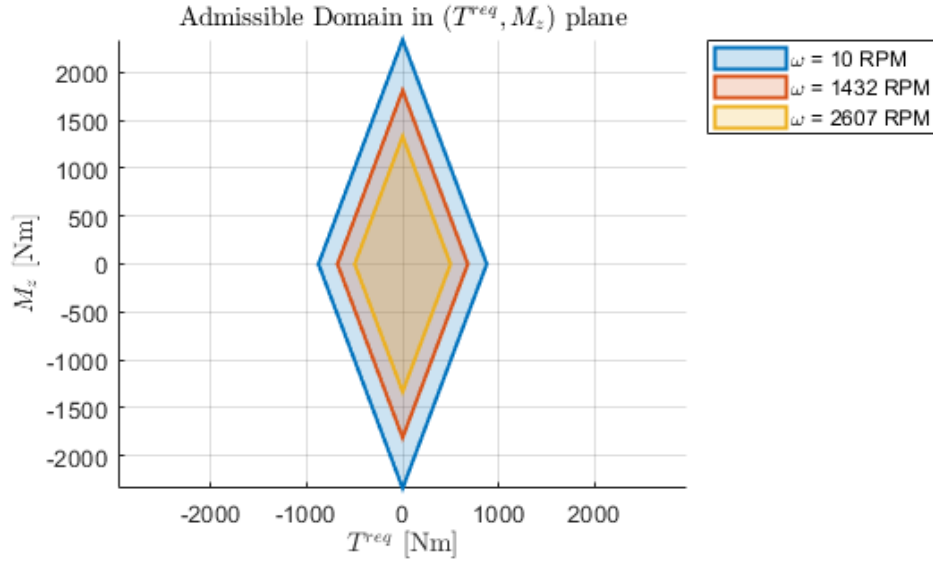


Figure 5.5: M_z domain.

The admissible domain in the (T^{req}, M_z) plane thus takes the form of a diamond-shaped polygon, centered at the origin and symmetric with respect to both axes. Its vertices are determined by the available torque limits and by the vehicle geometry.

5.5 Four-Wheel Steering with Torque Vectoring (4WS + TV)

The final configuration combines four-wheels steering with torque vectoring, thus exploiting the simultaneous action of three independent control actuators. This represents the most complex architecture investigated, since the steering angles of both axles and the differential torque distribution must be coordinated within a single control framework.

The rationale behind this integration lies in the complementary benefits offered by the two approaches. On one side, four-wheel steering (4WS) provides direct geometric control of the vehicle trajectory, enabling improved maneuverability at low speeds and enhanced stability at higher speeds. On the other side, torque vectoring (TV) introduces an additional degree of freedom by directly acting on the yaw dynamics, allowing the controller to stabilize the vehicle or amplify its agility in situations where steering alone may not suffice. The synergistic effect of both actuators promises significant improvements in terms of path tracking performance, especially in demanding maneuvers where steering authority or tire saturation would otherwise limit the achievable accuracy.

It must be emphasized, however, that this configuration cannot be obtained by a simple superposition of the previously designed controllers. The tuning of gains and the definition of the lookahead distance need to be revisited from scratch, considering the strong interaction between the two actuators. In fact, parameters that proved optimal in the standalone 4WS or 2WS+TV architectures may lead to suboptimal or even conflicting behaviors when combined, as the control action must now be shared and coordinated across multiple channels. For this reason, a dedicated optimization process was carried out, including the retuning of the front axle steering law, to identify a configuration capable of exploiting the joint potential of steering and torque distribution.

From a control design perspective, this dual-actuation system also poses additional challenges. The allocation of control effort between steering and yaw moment generation is not unique and must be carefully balanced to avoid redundancy or excessive actuation demand. Furthermore, the interaction between lateral force generation at the tires and the induced yaw moment introduces nonlinear couplings that require a robust and adaptive approach to guarantee consistent performance over a wide range of operating conditions.

Despite the higher complexity, the results obtained with the 4WS + TV controller highlight the promising nature of this technology. The combined use of geometric trajectory correction and direct yaw moment control allows for the minimization of both cross-track and heading errors, while maintaining a good compromise between agility and stability. This architecture therefore represents the most advanced

solution among those investigated, and it clearly demonstrates the potential of coordinated multi-actuator control for future autonomous driving applications.

Table 5.1: Controller gains for different architectures.

Controller	K_p Front	K_d Front	K_p Rear	K_d Rear	KFF Gain	K_p TV	K_d TV
PD + FF 2WS	40	50	0	0	0	0	0
4WS	70	50	-30	-3	0	0	0
4WS + FF	70	50	-30	-3	-71.11	0	0
2WS + TV	40	50	0	0	0	-120	-70
4WS + TV	25	20	-30	-3	-82.22	-160	-80

5.6 LQR Controller (Proposal for future application)

The Linear Quadratic Regulator (LQR) is an optimal control technique designed to minimize a cost function that balances two main objectives:

1. Reducing the energy of the state signal, which improves system performance and reduces oscillations.
2. Reducing the energy of the control input, which helps limit energy consumption and keep the control effort within acceptable bounds.

These objectives are expressed as two integrals over time:

- The first one measures the weighted energy of the states using the matrix Q .
- The second one measures the weighted energy of the control inputs using the matrix R .

The weighting matrices Q and R play a crucial role because they allow us to manage the trade-off between system performance and control effort. Increasing Q emphasizes the importance of keeping the states small (leading to faster convergence and fewer oscillations), while increasing R penalizes large control inputs (reducing energy consumption and command effort).

5.6.1 LQR Problem Formulation

The LQR problem can be formulated as finding the control input $u(t)$ that minimizes the following cost function:

$$J(u, x) = \int_0^\infty \left(x(t)^T Q x(t) + u(t)^T R u(t) \right) dt \quad (5.11)$$

subject to the system dynamics:

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = x_0 \quad (5.12)$$

The solution of this optimization problem, under the assumption that the pair (A, B) is controllable, leads to an optimal control law of the form:

$$u^*(t) = -Kx(t) \quad (5.13)$$

where the gain matrix K is given by:

$$K = R^{-1}B^T P \quad (5.14)$$

and P is the solution of the Algebraic Riccati Equation:

$$A^T P + PA + Q - PBR^{-1}B^T P = 0 \quad (5.15)$$

5.6.2 Choosing Q and R

The initial choice of the elements of Q and R is guided by system requirements:

- If there are specific performance requirements on a state x_i , set $Q_{ii} > 0$; otherwise, $Q_{ii} = 0$.
- If there are constraints on the control input u_i , set $R_{ii} > 0$; otherwise, $R_{ii} = 0$.

The final tuning is typically done through a trial-and-error process:

- Increasing $Q_{ii} \rightarrow$ reduces state energy $\rightarrow$ decreases oscillations and convergence time.
- Increasing $R_{ii} \rightarrow$ reduces control effort $\rightarrow$ decreases command energy and energy consumption.

In MATLAB, the LQR gain can be computed using:

$$K = \text{lqr}(A, B, Q, R) \quad (5.16)$$

5.6.3 Control Structure: Feedback and Feedforward

In general, the control vector U is composed of two parts:

$$U = U_b + U_f \quad (5.17)$$

The feedback component is based on the LQR law:

$$U_b = -KX \quad (5.18)$$

The feedforward component is designed to eliminate steady-state errors:

$$u_f = K_{ff} \cdot \kappa \quad (5.19)$$

where κ is the path curvature and K_{ff} is the feedforward gain.

Design Principles:

- For the **2WS configuration**, the feedforward gain is computed to cancel the steady-state cross-track error.
- For the **4WS configuration**, two feedforward gains are introduced to satisfy two conditions: zero cross-track error and zero heading error.
- For **2WS + torque vectoring**, only the feedforward for the front wheels steering command is used.

- For **4WS + torque vectoring**, same condition for the 4WS only:

$$e_{ct} = 0, \quad e_{\psi} = 0 \quad (5.20)$$

corresponding to zero cross-track error and zero heading error.

The computation of the feedforward gains has been carried out using MATLAB Symbolic Toolbox.

5.6.4 2WS Controller

The state-space matrices are defined as:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2(c_f + c_r)}{mv_x} & \frac{2(c_f + c_r)}{m} & \frac{2(bc_r - ac_f)}{mv_x} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{2(bc_r - ac_f)}{Jv_x} & \frac{2(c_fa - c_rb)}{J} & -\frac{2(c_fa^2 + c_rb^2)}{Jv_x} \end{bmatrix} \quad (5.21)$$

$$B_1 = \begin{bmatrix} 0 \\ \frac{2c_f}{m} \\ 0 \\ \frac{2c_fa}{J} \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 \\ -v_x + \frac{2(bc_r - c_fa)}{mv_x} \\ 0 \\ -\frac{2(c_fa^2 + c_rb^2)}{Jv_x} \end{bmatrix} \quad (5.22)$$

The feedback gain is computed as:

$$K_{LQR} = \text{lqr}(A, B_1, Q, R) \quad (5.23)$$

The steady-state feedforward gain is obtained by solving:

$$X_{ss} = -(A - B_1K)^{-1}(B_1K_{ff} + B_2v_x)K_L \quad (5.24)$$

Solving:

$$\text{solve}([X_{ss}(1) = 0], [K_{ff}]) \quad (5.25)$$

The resulting analytical expression is:

$$K_{ff} = \frac{mv_x^2}{(a+b)} \left(\frac{b}{2c_f} - \frac{a}{2c_r} + \frac{a}{2c_r}k_3 \right) + (a+b-bk_3) \quad (5.26)$$

This gain ensures zero cross-track error at steady state.

5.6.5 4WS Controller

The A matrix remains the same, while the input matrix B_1 becomes:

$$B_1 = \begin{bmatrix} 0 & 0 \\ \frac{2c_f}{m} & \frac{2c_r}{m} \\ 0 & 0 \\ \frac{2c_f a}{J} & -\frac{2c_r b}{J} \end{bmatrix} \quad (5.27)$$

The feedforward gains K_{ff1} and K_{ff2} are computed by imposing:

$$X_{ss} = -(A - B_1 K)^{-1} (B_1 K_{ff} + B_2 v_x) K_L \quad (5.28)$$

Solving:

$$\text{solve}([X_{ss}(1) = 0, X_{ss}(3) = 0], [K_{ff1}, K_{ff2}]) \quad (5.29)$$

Final expressions:

$$K_{ff,f} = \frac{2c_f a^2 + 2bc_f a + bmv_x^2}{2c_f(a+b)} \quad (5.30)$$

$$K_{ff,r} = -\frac{2c_r b^2 + 2ac_r b - amv_x^2}{2c_r(a+b)} \quad (5.31)$$

These gains ensure zero cross-track error as well as zero heading error at steady state.

5.6.6 2WS + TV Controller

The state-space formulation is extended by including the yaw moment M_z produced through torque vectoring. The state matrix A remains the same as in Eq. (5.21), while the input matrix B_1 becomes:

$$B_1 = \begin{bmatrix} 0 & 0 \\ \frac{2c_f}{m} & 0 \\ 0 & 0 \\ \frac{2c_f a}{J} & \frac{1}{J} \end{bmatrix}, \quad (5.32)$$

where the first column corresponds to the steering angle δ and the second column to the external yaw moment M_z .

The relation between M_z and the wheel torques is defined as

$$M_z = \frac{t_f}{2R} (T_{FR} - T_{FL}) + \frac{t_r}{2R} (T_{RR} - T_{RL}), \quad (5.33)$$

with t_f and t_r being the front and rear track widths, and R the wheel radius.

Introducing weighting coefficients w_f and w_r ($w_f + w_r = 1$), the torque differences per axle are

$$\Delta T_f = \frac{2R}{t_f} w_f M_z, \quad (5.34)$$

$$\Delta T_r = \frac{2R}{t_r} w_r M_z. \quad (5.35)$$

Finally, the wheel torques are obtained as

$$T_{FR} = T_{FR}^0 + \frac{1}{2} \Delta T_f, \quad T_{FL} = T_{FL}^0 - \frac{1}{2} \Delta T_f, \quad (5.36)$$

$$T_{RR} = T_{RR}^0 + \frac{1}{2} \Delta T_r, \quad T_{RL} = T_{RL}^0 - \frac{1}{2} \Delta T_r, \quad (5.37)$$

where $T_{(\cdot)}^0$ are the base.

5.6.7 4WS + TV Controller

In this case, the vehicle is equipped with both front and rear steering and torque vectoring. The state matrix A remains unchanged with respect to the previous formulations, while the input matrix B_1 becomes:

$$B_1 = \begin{bmatrix} 0 & 0 & 0 \\ \frac{2c_f}{m} & \frac{2c_r}{m} & 0 \\ 0 & 0 & 0 \\ \frac{2c_f a}{J} & -\frac{2c_r b}{J} & \frac{1}{J} \end{bmatrix}, \quad (5.38)$$

where the first and second columns correspond to the steering angles δ_f and δ_r , while the third column corresponds to the external yaw moment M_z generated by torque vectoring.

The mapping between M_z and the wheel torques is the same as in the 2WS + TV case:

$$M_z = \frac{t_f}{2R} (T_{FR} - T_{FL}) + \frac{t_r}{2R} (T_{RR} - T_{RL}). \quad (5.39)$$

By introducing the front and rear weighting coefficients w_f and w_r ($w_f + w_r = 1$), the torque differences per axle are:

$$\Delta T_f = \frac{2R}{t_f} w_f M_z, \quad (5.40)$$

$$\Delta T_r = \frac{2R}{t_r} w_r M_z. \quad (5.41)$$

The final wheel torques are then obtained as

$$T_{FR} = T_{FR}^0 + \frac{1}{2} \Delta T_f, \quad T_{FL} = T_{FL}^0 - \frac{1}{2} \Delta T_f, \quad (5.42)$$

$$T_{RR} = T_{RR}^0 + \frac{1}{2} \Delta T_r, \quad T_{RL} = T_{RL}^0 - \frac{1}{2} \Delta T_r, \quad (5.43)$$

where $T_{(\cdot)}^0$ represent the baseline traction/braking torques.

In this case, similarly to what happens for the PD controllers, there is the need for a whole retuning. The lqr command anyway helps in this sense since it automatically computes the new state feedback gains taking into account the influence of the newer added terms.

Chapter 6

Experimental results

6.1 Gain Scheduling Approach

One of the main challenges in the design of path tracking controllers for autonomous vehicles is the strong dependence of vehicle dynamics on the longitudinal velocity. In particular, the effectiveness of control actions applied through steering or torque vectoring actuators may vary significantly depending on the operating point of the vehicle. A controller tuned at a fixed speed may therefore result in degraded performance or even instability when the velocity changes. To overcome this limitation, a *gain scheduling* strategy was introduced.

Gain scheduling consists in designing the controller parameters as explicit functions of a measurable scheduling variable, typically the longitudinal velocity of the vehicle. Instead of relying on a single set of proportional and derivative gains, the control law is adapted continuously by interpolating among different values of the gains, which are pre-tuned at selected operating conditions. This approach enables the controller to maintain robust and consistent performance across a wide range of speeds, guaranteeing stability at low velocity and accuracy at higher velocity, where the path tracking task becomes more demanding.

From a practical standpoint, the gain scheduling procedure adopted in this work involved the following steps:

1. Identification of the most representative operating points in terms of vehicle speed (from 5 to 30 km/h, with increments of 5 km/h).
2. Optimization and fine-tuning of the controller gains at each operating point, using a combination of simulation-based sensitivity analysis and performance indices (cross-track error, heading error, and sideslip angle).
3. Definition of a set of look-ahead distances, consistently scheduled with velocity, to ensure that the trajectory preview used in the control law scales

proportionally with the dynamics of the vehicle.

4. Implementation of a continuous interpolation mechanism to guarantee smooth transitions between different operating points, avoiding discontinuities in control effort and ensuring a natural adaptation of the vehicle behaviour.

The resulting gain scheduling maps are reported in Table 6.1, where for each velocity the corresponding gain and look-ahead distance are listed. These values were later embedded in the Simulink control architecture, enabling automatic adaptation of the path tracking controller as a function of velocity.

Table 6.1: Gain scheduling maps for path tracking controller.

Velocity [km/h]	K_p [-]	K_d [-]	Look-ahead [m]	Max CTE [m]	RMS CTE [m]
5	500	500	0.00	0.15	0.05
10	300	300	0.10	0.16	0.07
15	150	150	0.65	0.19	0.08
20	60	60	1.70	0.45	0.20
25	45	45	2.40	0.34	0.18
30	40	40	3.50	0.40	0.15

It is important to underline that, although the gain scheduling framework was developed and validated across a wide range of velocities, the subsequent analyses presented in this work – concerning the comparison of different control strategies and actuator configurations – were all conducted at the maximum attainable speed of the vehicle, namely 30 km/h. This choice was motivated by the fact that operating at the velocity limit represents the most challenging condition for path tracking: the vehicle dynamics are more demanding, actuator authority is more critical, and the sensitivity to modelling uncertainties increases. Therefore, evaluating the controllers at this operating point allows assessing their robustness and performance in the most severe scenario, ensuring that any improvement observed in this condition will also translate into effective performance at lower speeds.

6.2 PD + FF 2WS

Soft circuit:

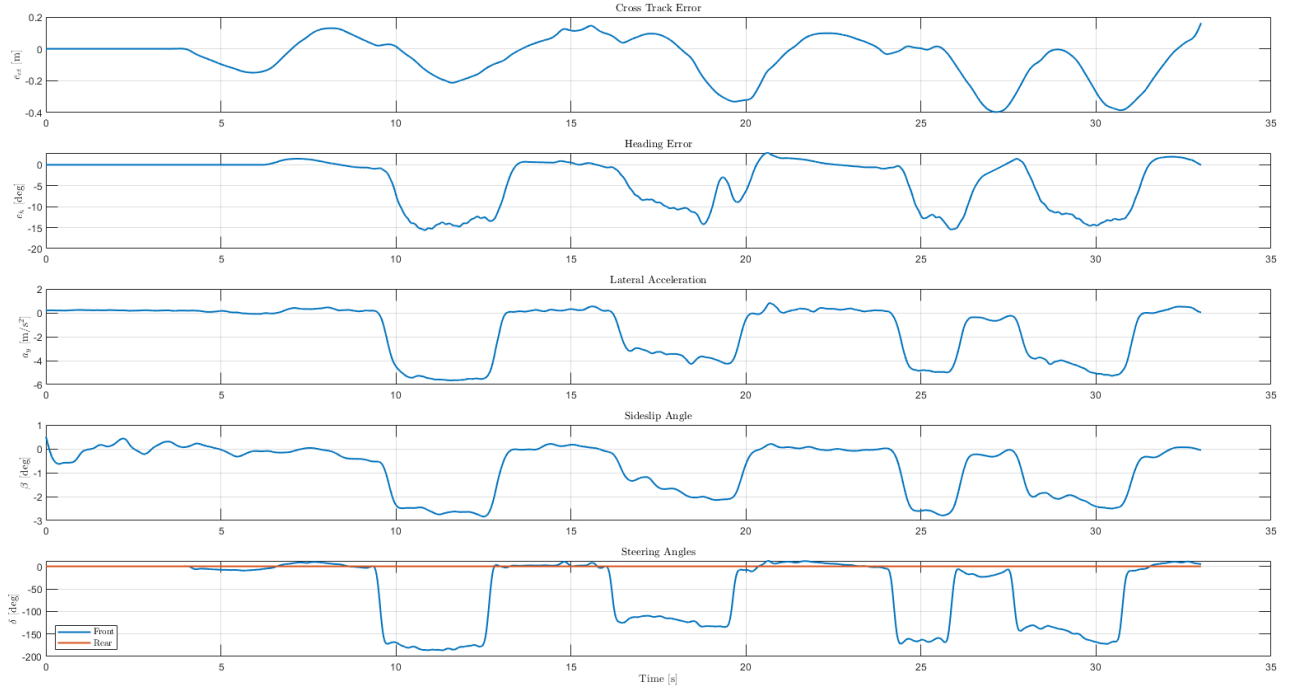


Figure 6.1: Results obtained from the PD + FF 2WS controller (Soft Circuit).

— KPIs —

- Max $|CTE|$ [m] : 0.397
- RMS CTE [m] : 0.148
- Max $|\beta|$ [deg] : 2.836
- RMS β [deg] : 1.315
- Max $|HE|$ [deg] : 15.613
- RMS HE [deg] : 6.953
- Max $|\Delta_{\text{Front}}|$ [deg] : 186.281
- RMS Δ_{Front} [deg] : 88.710

- Max $|\Delta_{\text{Rear}}|$ [deg] : 0.000
- RMS Δ_{Rear} [deg] : 0.000

Sharp circuit:

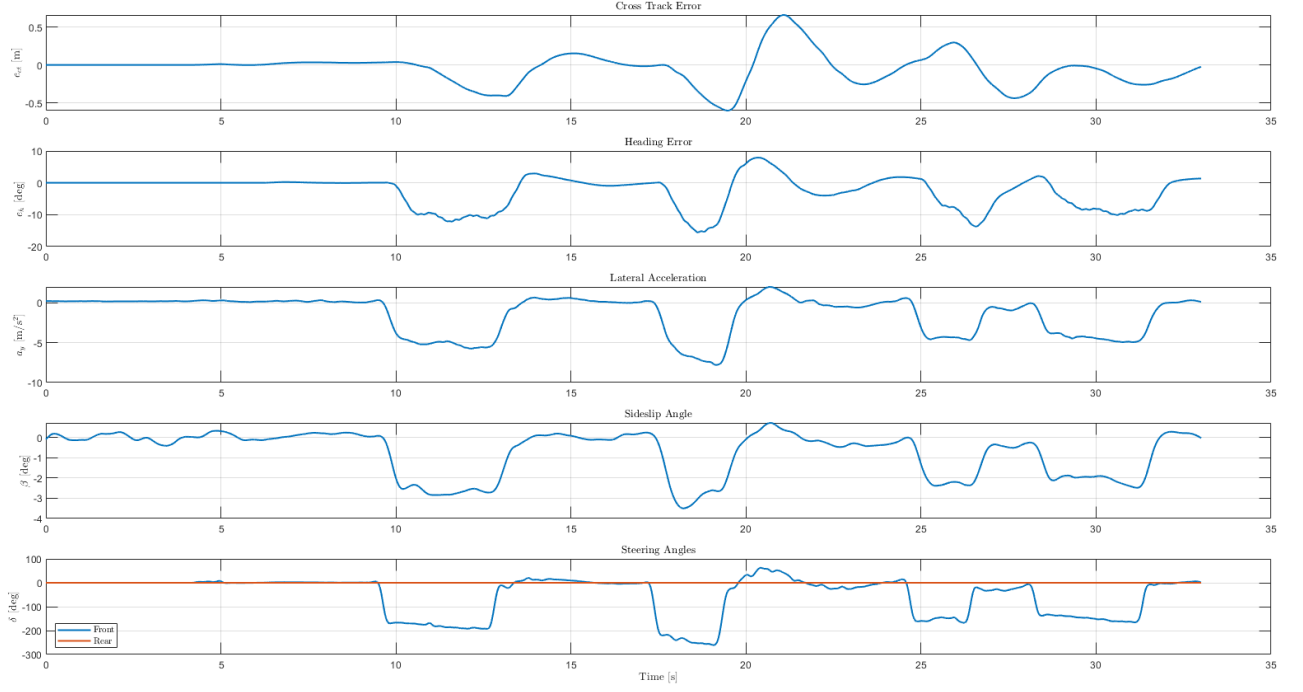


Figure 6.2: Results obtained from the PD + FF 2WS controller (Sharp Circuit).

— KPIs —

- Max $|CTE|$ [m] : 0.661
- RMS CTE [m] : 0.217
- Max $|\beta|$ [deg] : 3.500
- RMS β [deg] : 1.349
- Max $|HE|$ [deg] : 15.546
- RMS HE [deg] : 5.552
- Max $|\Delta_{\text{Front}}|$ [deg] : 260.907
- RMS Δ_{Front} [deg] : 98.685
- Max $|\Delta_{\text{Rear}}|$ [deg] : 0.000

- RMS Δ_{Rear} [deg] : 0.000

From these first two sets of results it is possible to state that the 2WS configuration can be sufficient if the reference trajectory doesn't provide struggles. As far as the so called " Sharp circuit ", we reach almost 70 cm of peak in lateral error which is not at all acceptable. The controller, in its simplicity, can be considered a good benchmark for other strategies but has to be improved.

6.3 4WS pure feedback

Soft circuit:

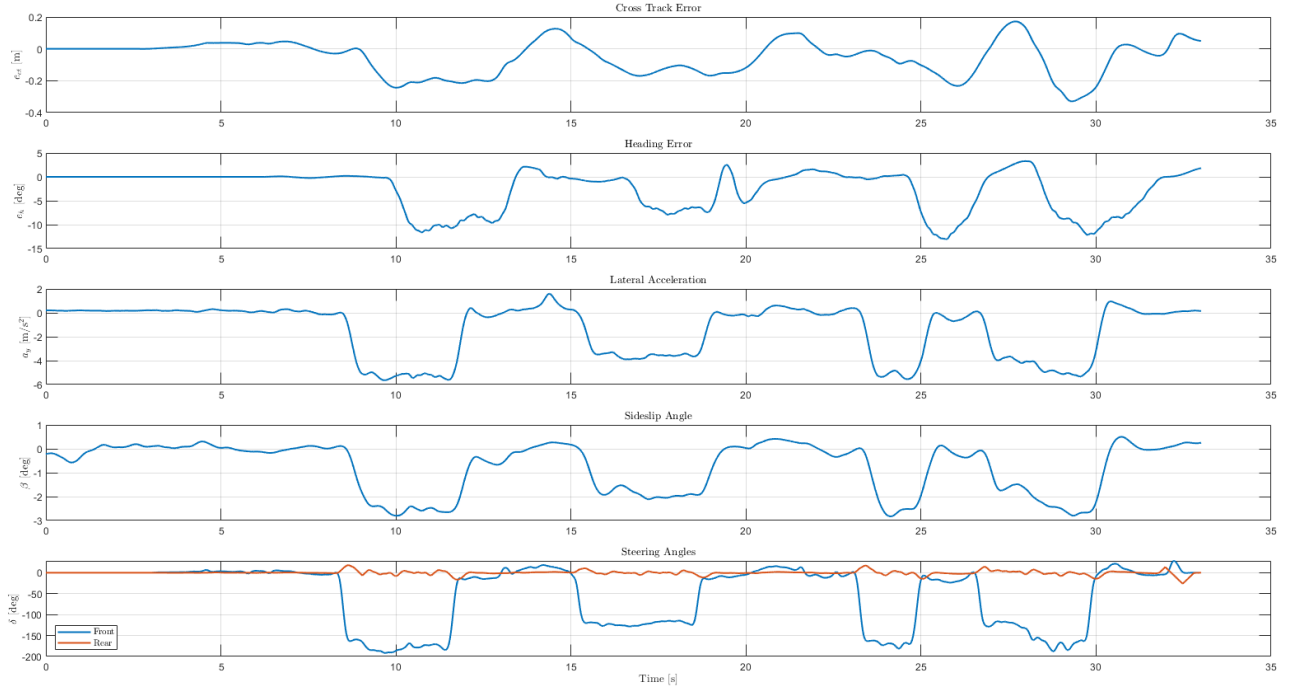


Figure 6.3: Results obtained from the 4WS controller (Soft Circuit).

— KPIs —

- Max $|CTE|$ [m] : 0.330
- RMS CTE [m] : 0.120
- Max $|\beta|$ [deg] : 2.828
- RMS β [deg] : 1.303
- Max $|HE|$ [deg] : 13.063
- RMS HE [deg] : 4.997
- Max $|\Delta_{\text{Front}}|$ [deg] : 191.721
- RMS $|\Delta_{\text{Front}}|$ [deg] : 88.916

- Max $|\Delta_{\text{Rear}}|$ [deg] : 26.310
- RMS $|\Delta_{\text{Rear}}|$ [deg] : 4.686

Sharp circuit:

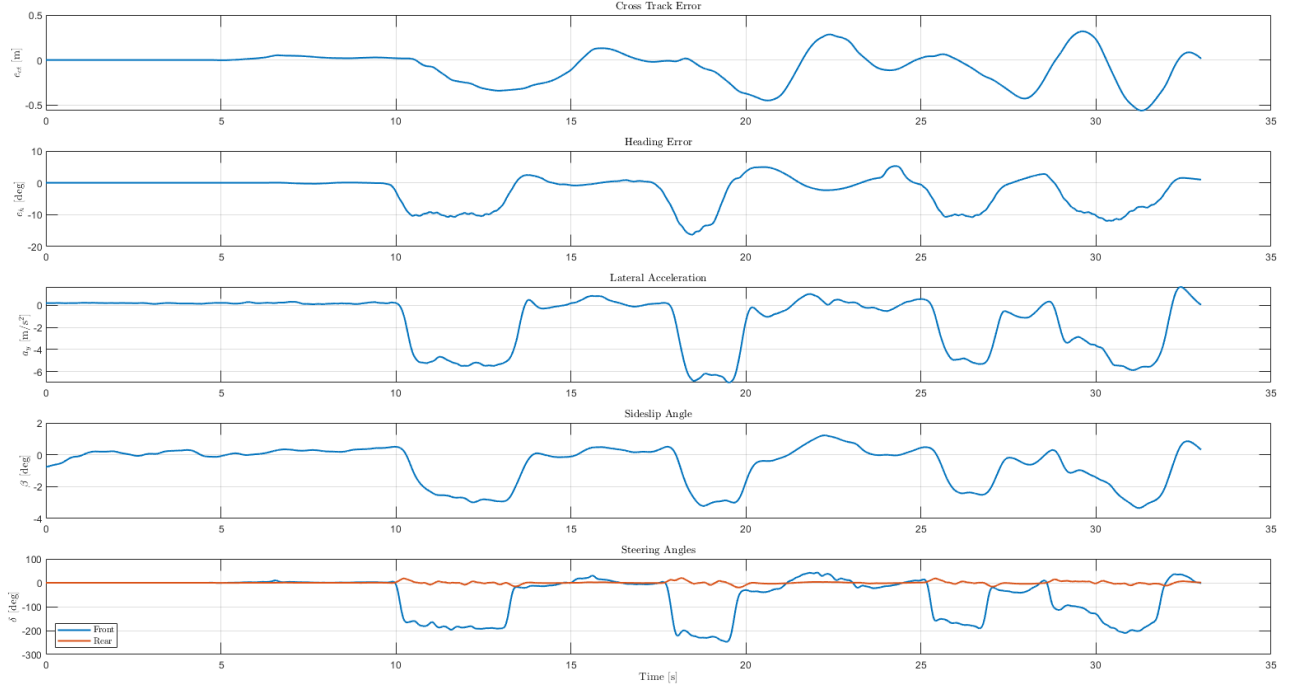


Figure 6.4: Results obtained from the 4WS controller (Sharp Circuit).

— KPIs —

- Max $|CTE|$ [m] : 0.561
- RMS CTE [m] : 0.195
- Max $|\beta|$ [deg] : 3.342
- RMS β [deg] : 1.363
- Max $|HE|$ [deg] : 16.300
- RMS HE [deg] : 5.499
- Max $|\Delta_{\text{Front}}|$ [deg] : 247.200
- RMS $|\Delta_{\text{Front}}|$ [deg] : 98.298
- Max $|\Delta_{\text{Rear}}|$ [deg] : 19.784

- RMS $|\Delta_{\text{Rear}}|$ [deg] : 4.803

For these and the following considerations, the heading error was also evaluated with a lookahead distance, so that the control action could become in some sense predictive. This explains the slightly increased values with respect to the benchmark case.

The 4WS technology proved to be effective in reducing the lateral error, which represents the main goal of the proposed path tracking task. However, this improvement comes at the cost of a limited contribution from the rear steering angle: the imposed control gains were kept small. A detailed sensitivity analysis was carried out, showing that for larger rear steering feedback contributions, the system progressively approached instability.

As already mentioned in the reference section, the rear steering control is based only on a feedback Proportional–Derivative action on the yaw rate error. In order to provide the rear steering with greater influence on the overall dynamics, it was therefore considered appropriate to introduce a feedforward contribution, which is presented in the following.

6.4 4WS with feedforward and feedback

Soft circuit:

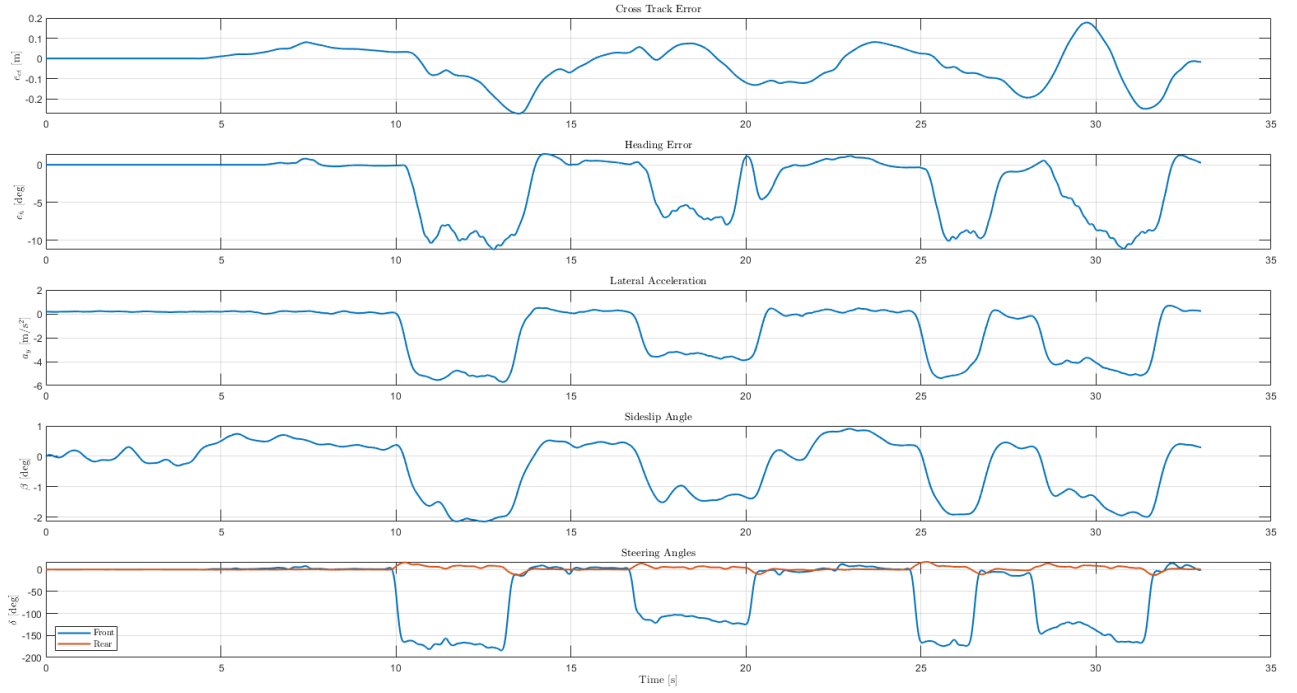


Figure 6.5: Results obtained from the 4WS with feedforward (Soft Circuit).

- Max $|CTE|$ [m] : 0.272
- RMS CTE [m] : 0.094
- Max $|\beta|$ [deg] : 2.137
- RMS β [deg] : 0.989
- Max $|HE|$ [deg] : 11.251
- RMS HE [deg] : 4.717
- Max $|\Delta_{\text{Front}}|$ [deg] : 184.134
- RMS $|\Delta_{\text{Front}}|$ [deg] : 86.079

- Max $|\Delta_{\text{Rear}}|$ [deg] : 17.456
- RMS $|\Delta_{\text{Rear}}|$ [deg] : 5.200

Sharp circuit:

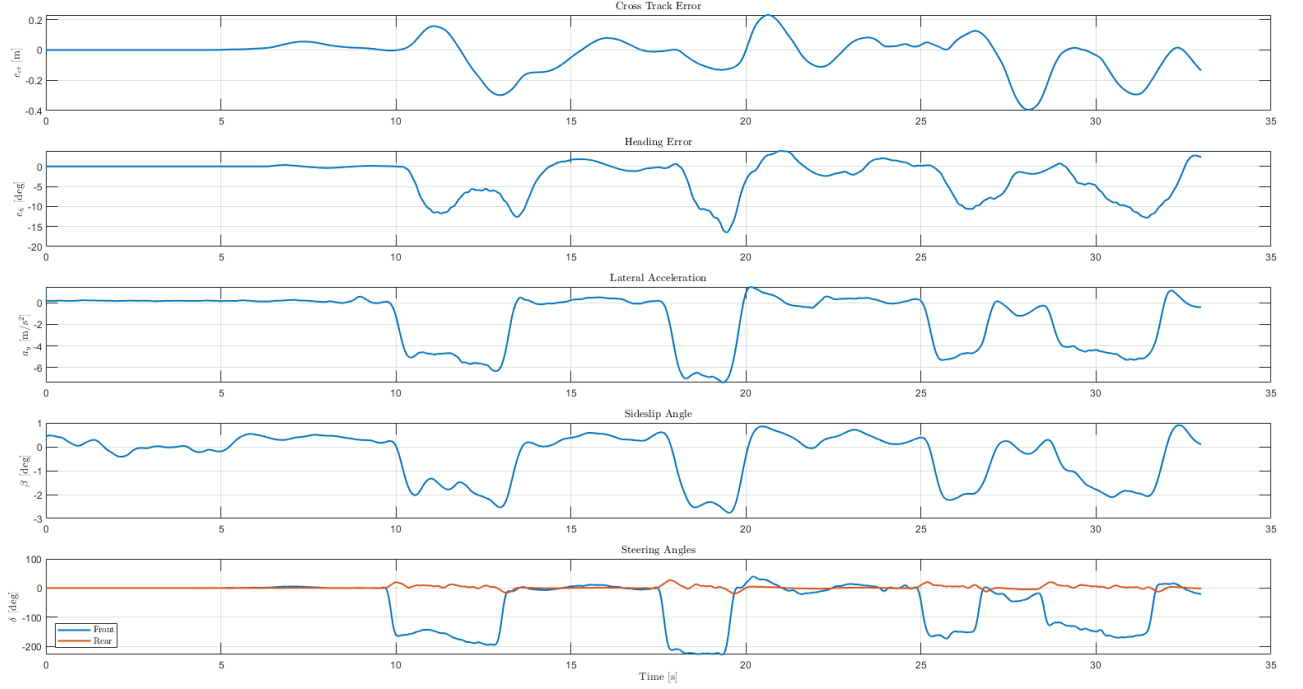


Figure 6.6: Results obtained from the 4WS with feedforward (Sharp Circuit).

— KPIs —

- Max $|CTE|$ [m] : 0.395
- RMS CTE [m] : 0.117
- Max $|\beta|$ [deg] : 2.762
- RMS β [deg] : 1.079
- Max $|HE|$ [deg] : 16.449
- RMS HE [deg] : 5.180
- Max $|\Delta_{Front}|$ [deg] : 228.123
- RMS $|\Delta_{Front}|$ [deg] : 92.881
- Max $|\Delta_{Rear}|$ [deg] : 27.935

- RMS $|\Delta_{\text{Rear}}|$ [deg] : 5.762

From the analysis carried out so far, it clearly emerges that the introduction of a feedforward contribution is not only beneficial, but almost essential for achieving satisfactory path tracking performance. The main advantage lies in the fact that the feedforward term directly accounts for the deterministic component of the trajectory, which is strongly related to the reference curvature. As a consequence, the feedback loop is relieved from the task of compensating steady-state errors, and can instead focus exclusively on the suppression of dynamic disturbances and unpredictable variations. This leads to reduced control effort, smoother actuator commands, and an overall more stable closed-loop behaviour.

Another aspect worth underlining is that the presence of feedforward makes the controller inherently more *predictive*. Rather than reacting to an error that has already developed, the system anticipates the required steering action based on the reference geometry. This results in a noticeable reduction of tracking errors, both in terms of peak and root-mean-square values, as already highlighted by the KPI analysis.

For these reasons, the feedforward contribution is considered a convenient and robust design choice. Its use will therefore be retained also in the following formulations involving torque vectoring. In this context, the combination of an anticipative control action and the additional yaw moment generation capability is expected to provide further improvements in terms of lateral error reduction, yaw stability, and overall robustness of the control architecture.

6.5 2WS + TV

Soft circuit:

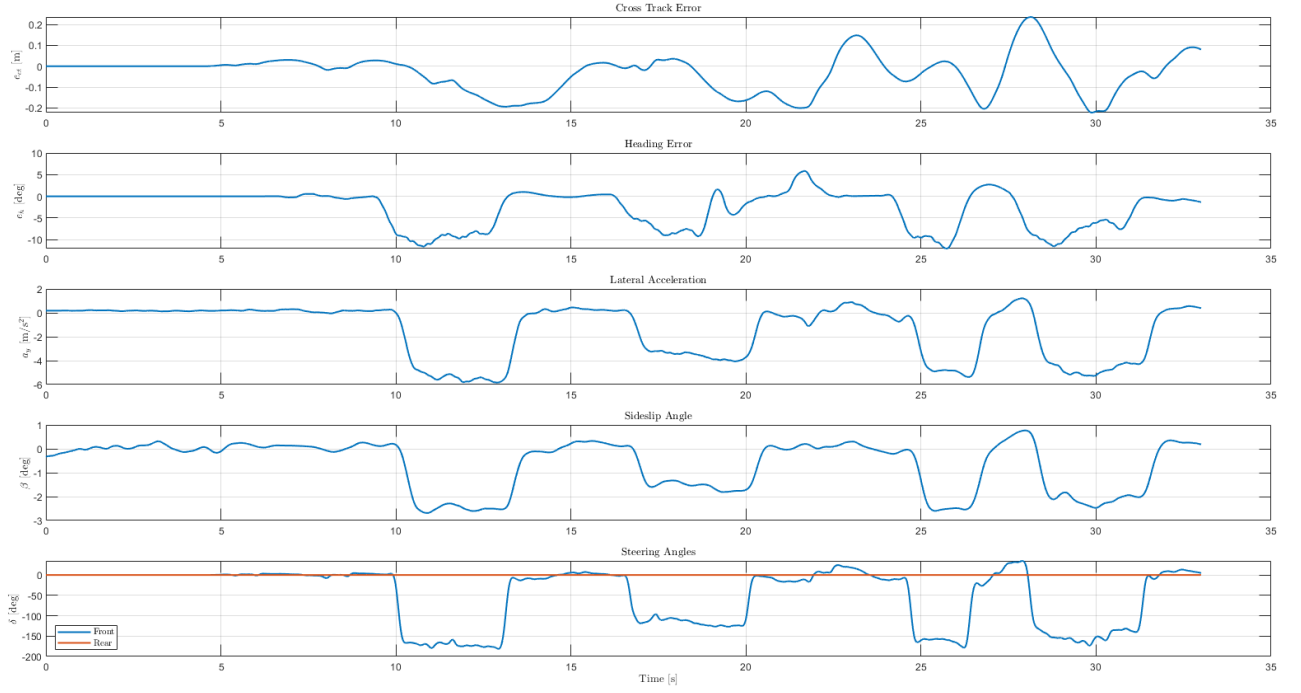


Figure 6.7: Results obtained from the 2WS + TV controller (Soft Circuit).

— KPIs —

- Max $|CTE|$ [m] : 0.236
- RMS CTE [m] : 0.097
- Max $|\beta|$ [deg] : 2.686
- RMS β [deg] : 1.237
- Max $|HE|$ [deg] : 12.059
- RMS HE [deg] : 4.933
- Max $|\Delta_{\text{Front}}|$ [deg] : 181.641
- RMS $|\Delta_{\text{Front}}|$ [deg] : 86.610

- Max $|\Delta_{\text{Rear}}|$ [deg] : 0.000
- RMS $|\Delta_{\text{Rear}}|$ [deg] : 0.000

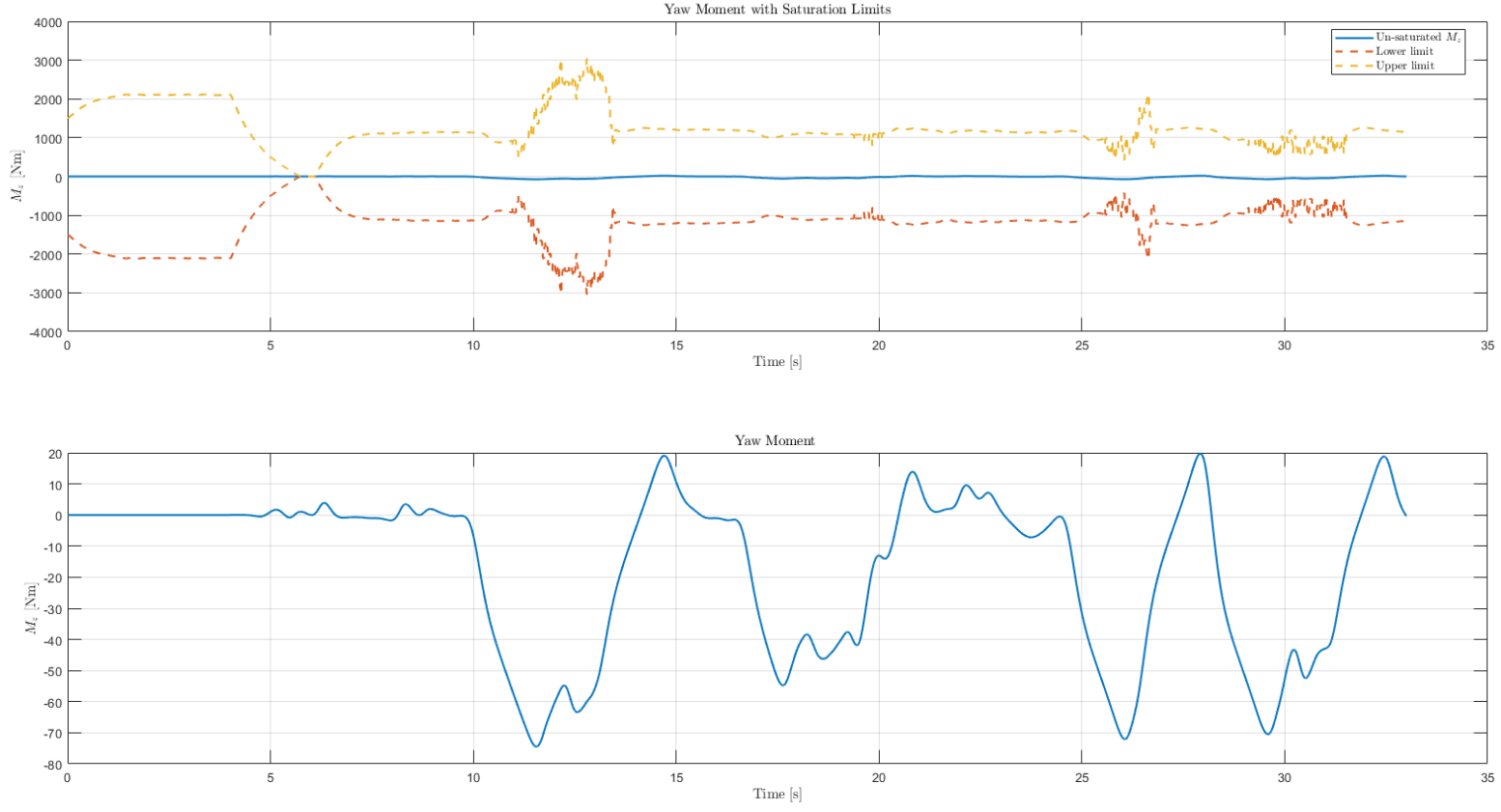


Figure 6.8: Yaw moment command (Soft Circuit).

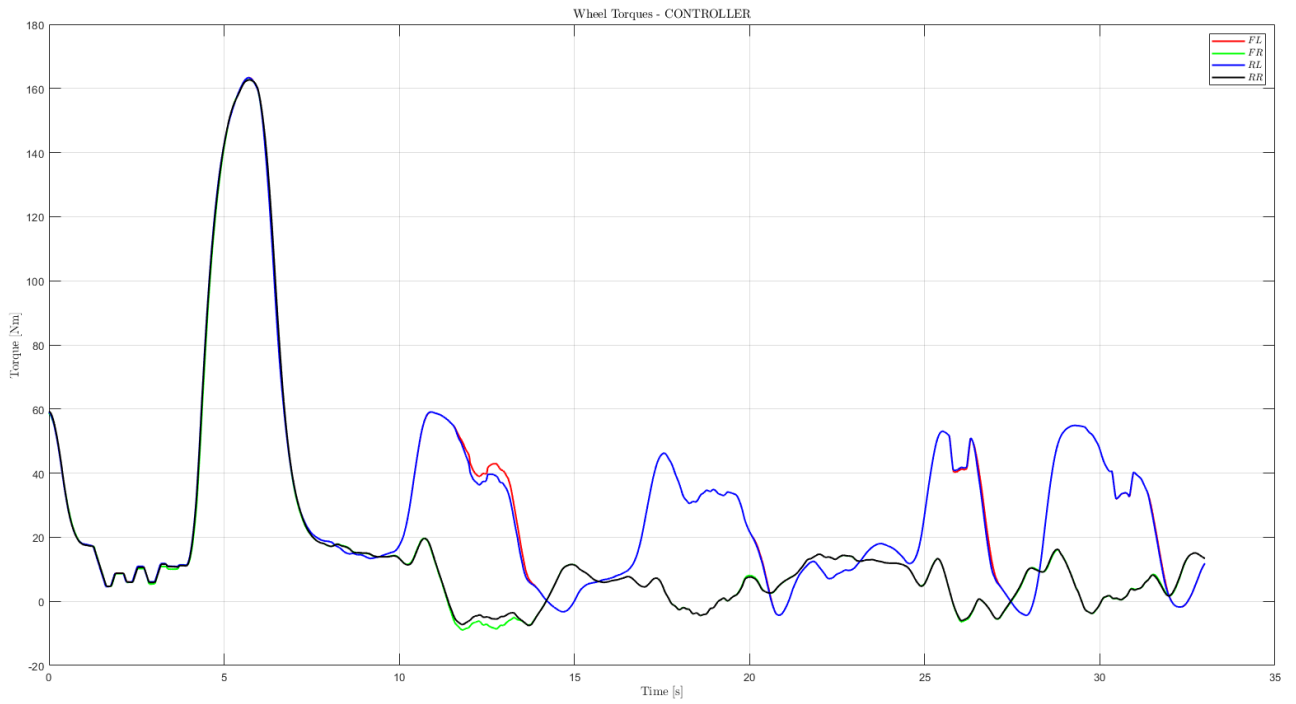


Figure 6.9: Torques at wheel level (Soft Circuit).

Sharp circuit:

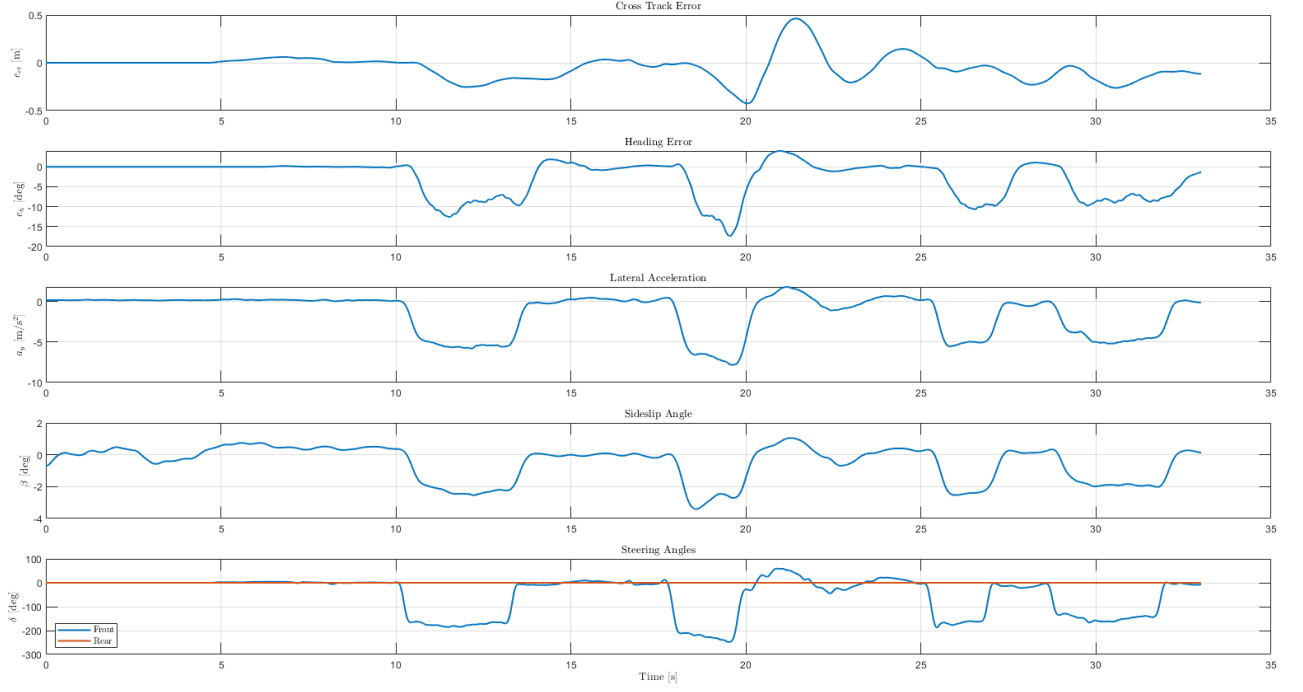


Figure 6.10: Results obtained from the 2WS + TV controller (Sharp Circuit).

— KPIs —

- Max $|CTE|$ [m] : 0.463
- RMS CTE [m] : 0.144
- Max $|\beta|$ [deg] : 3.410
- RMS β [deg] : 1.233
- Max $|HE|$ [deg] : 17.388
- RMS HE [deg] : 5.231
- Max $|\Delta_{Front}|$ [deg] : 248.245
- RMS $|\Delta_{Front}|$ [deg] : 94.877
- Max $|\Delta_{Rear}|$ [deg] : 0.000

- RMS $|\Delta_{\text{Rear}}|$ [deg] : 0.000

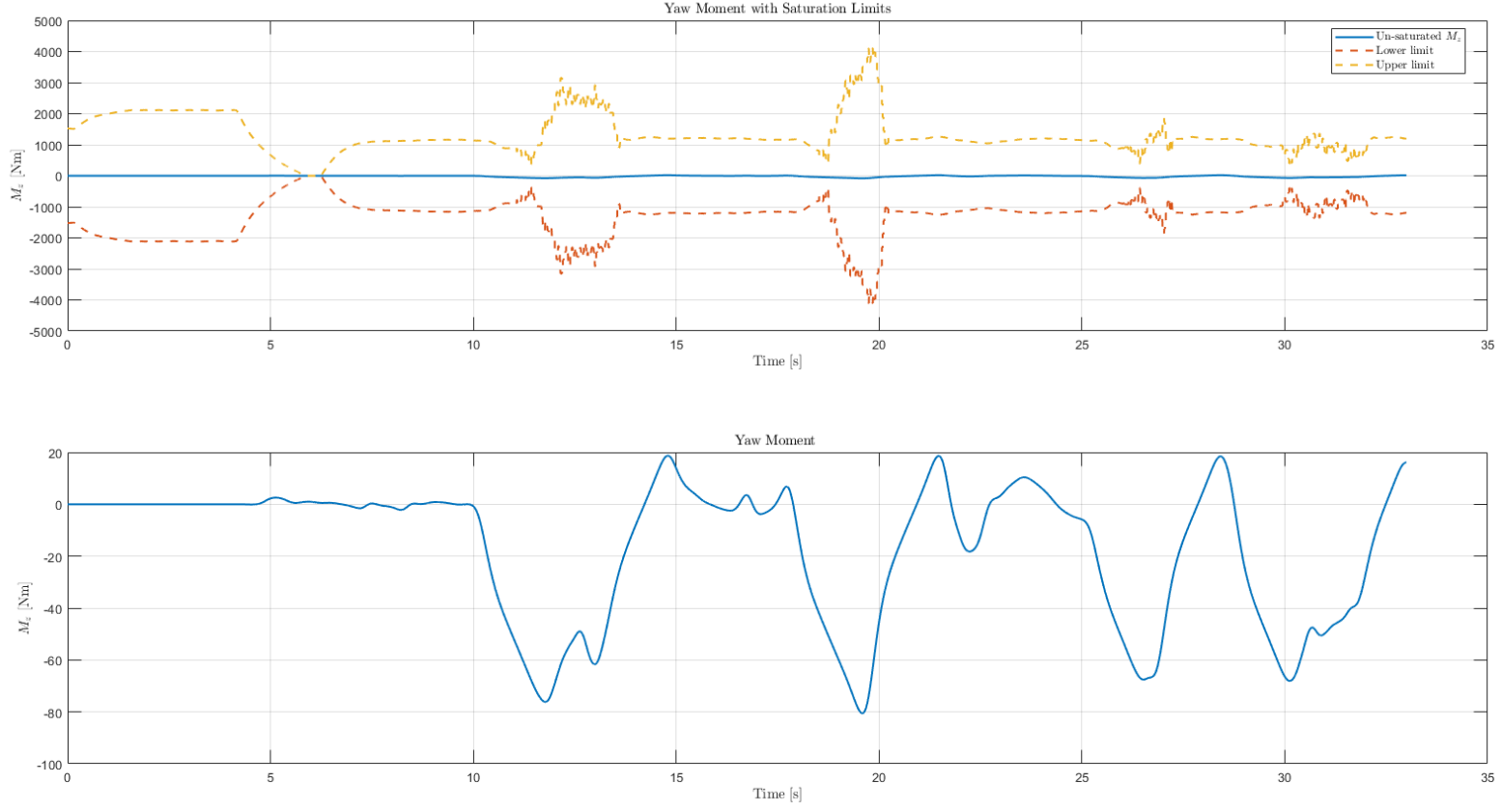


Figure 6.11: Yaw moment command (Sharp Circuit).

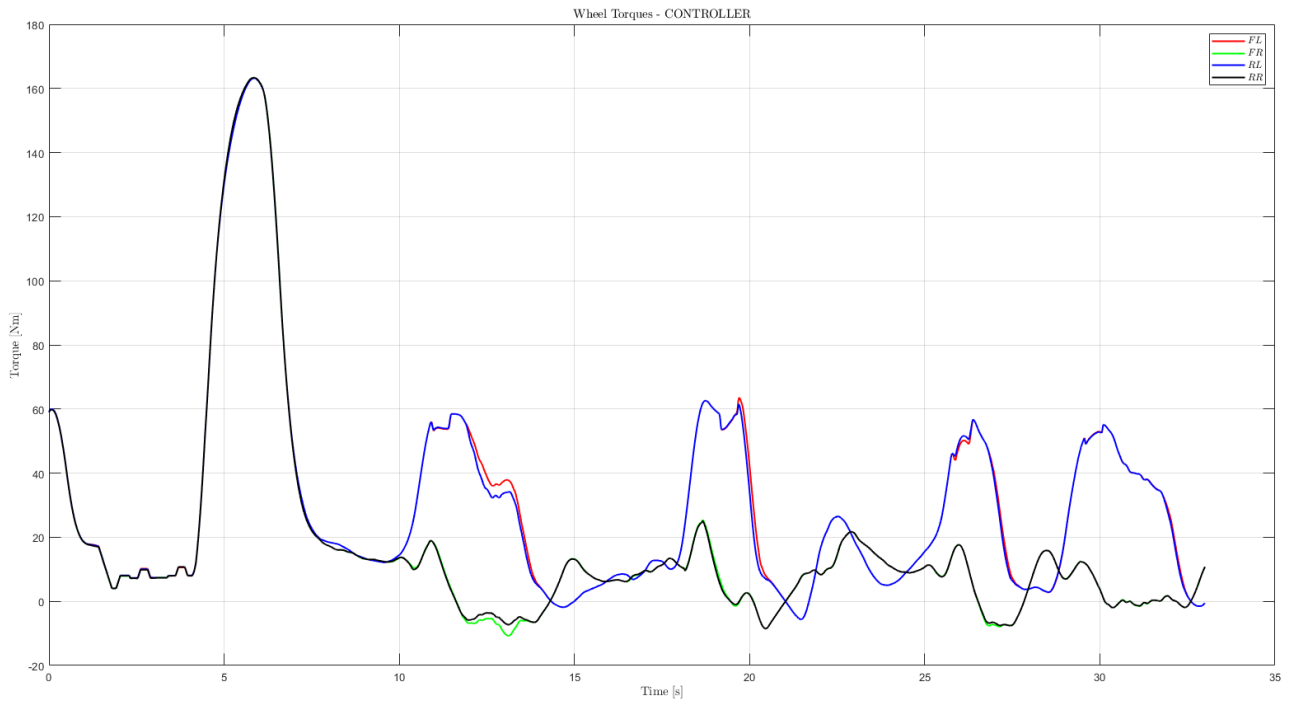


Figure 6.12: Torques at wheel level (Sharp Circuit).

The results obtained on the Soft Circuit with the 2WS + TV controller represent the best performance achieved in the entire project, with a maximum lateral deviation of approximately 23 cm. This clearly demonstrates the effectiveness of torque vectoring as an additional actuator for enhancing path tracking.

A closer inspection of the results shows that the torque vectoring activity can be further increased, since the wheel torque differences never reach values of hundreds of Newton-meters, and the commanded yaw moment remains relatively low. This observation indicates that the system still has unused potential for generating additional corrective yaw moments. Nevertheless, the outcome is fully consistent with the initial design philosophy: starting from a working 2WS controller and progressively adding one actuator at a time, without altering the original control action. As a consequence, the majority of the path tracking performance is still provided by the front steering, while torque vectoring acts as an auxiliary input that improves stability and error reduction, but in a controlled and limited manner.

A sensitivity analysis was also performed, and it confirmed that increasing the TV controller gains beyond the chosen values quickly led the system towards instability. This underlines that the presented tuning already represents the optimal trade-off for the Soft Circuit. It can therefore be considered the best achievable result in this scenario. For the Sharp Circuit, there might still be room for further improvement by exploiting more aggressive torque vectoring actions; however, such an approach would come at the risk of deteriorating the performance on the Soft Circuit, and therefore must be evaluated carefully.

6.6 4WS + TV

Soft circuit:

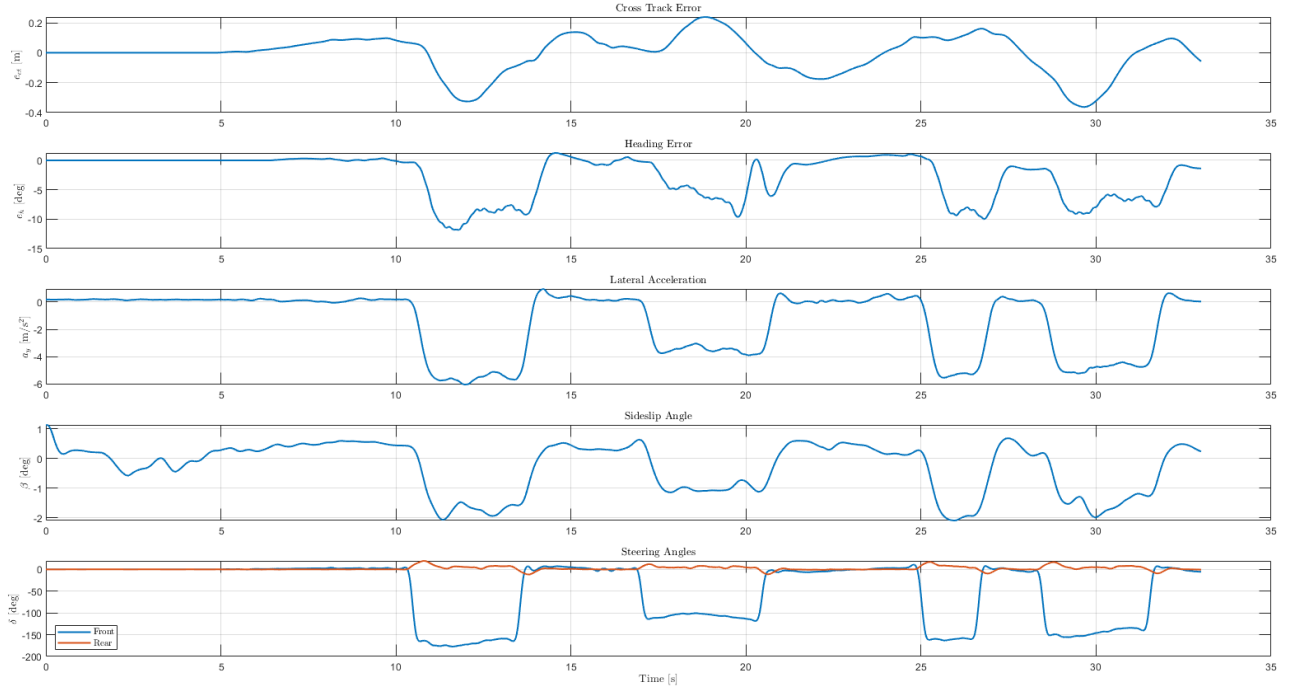


Figure 6.13: Results obtained from the 4WS + TV controller (Soft Circuit).

— KPIs —

- Max $|CTE|$ [m] : 0.363
- RMS CTE [m] : 0.126
- Max $|\beta|$ [deg] : 2.092
- RMS β [deg] : 0.906
- Max $|HE|$ [deg] : 11.848
- RMS HE [deg] : 4.532
- Max $|\Delta_{\text{Front}}|$ [deg] : 177.237
- RMS $|\Delta_{\text{Front}}|$ [deg] : 82.894

- Max $|\Delta_{\text{Rear}}|$ [deg] : 19.490
- RMS $|\Delta_{\text{Rear}}|$ [deg] : 4.901

Sharp circuit:

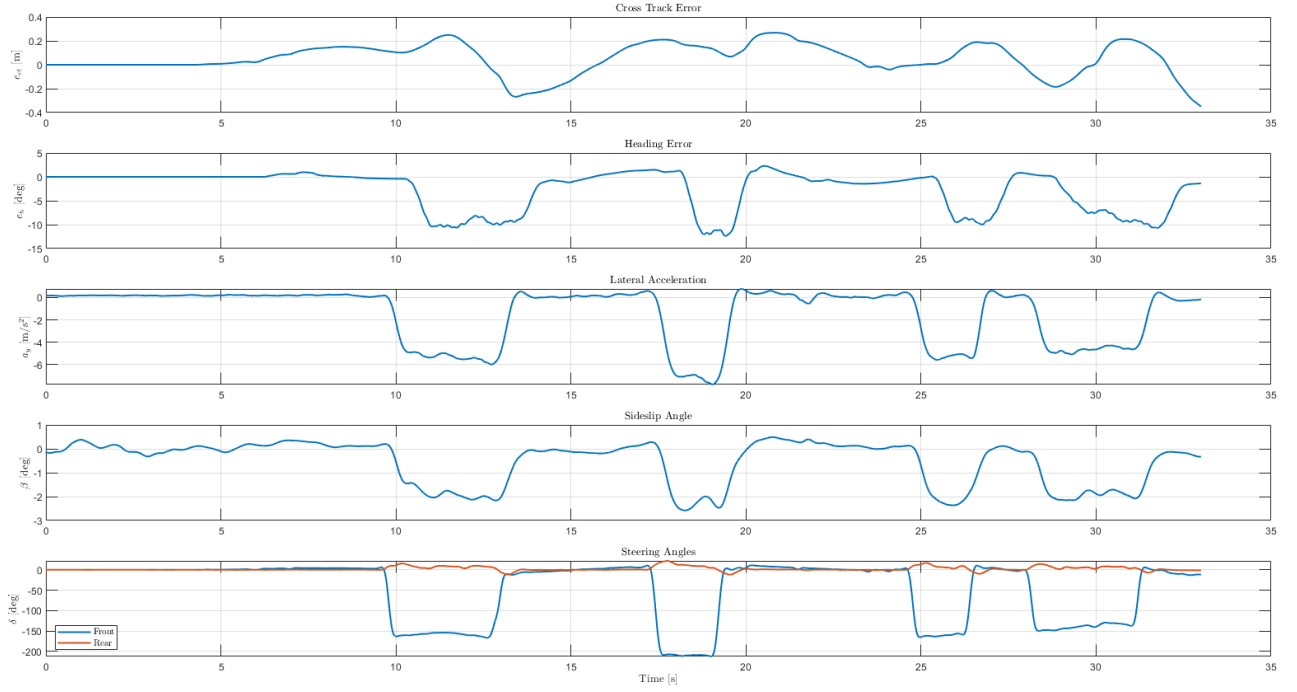


Figure 6.14: Results obtained from the 4WS + TV controller (Sharp Circuit).

— KPIs —

- Max $|CTE|$ [m] : 0.2843
- RMS CTE [m] : 0.1295
- Max $|HE|$ [deg] : 14.8841
- RMS HE [deg] : 5.3458
- Max $|\beta|$ [deg] : 2.9053
- RMS β [deg] : 1.3663
- RMS δ_f [deg] : 93.3938
- RMS δ_r [deg] : 6.5184
- RMS $\dot{\delta}_f$ [deg/s] : 106.8857
- RMS $\dot{\delta}_r$ [deg/s] : 13.5477

Global Comparison of Control Architectures

Once the performance indicators have been collected for each controller architecture, it becomes essential to move from an isolated analysis of the single results to a more comprehensive overview. The rationale behind this approach is that the numerical values reported for each configuration, although useful in absolute terms, acquire their true significance only when placed in direct comparison with one another. In fact, it is only through such a relative perspective that the role and the contribution of each actuator can be properly assessed, allowing us to understand whether a given performance improvement is primarily attributable to front steering, rear steering, or torque vectoring.

It must be stressed that the objective of the present work is not to identify an absolute "winner" among the different control logics, but rather to highlight the relative importance of each actuator within the path tracking problem. This comparative reading is of particular interest because the present thesis is meant as a complementary contribution to a parallel research activity, aimed at developing a neural-network-based controller trained through imitation learning from a Nonlinear Predictive Model Control (NMPC) policy. The NMPC is widely acknowledged as the current state-of-the-art solution in terms of path tracking performance; however, its application is often hindered by the severe computational burden and the consequent challenges in real-time implementation. The imitation learning approach aims precisely at overcoming this limitation, transferring the control policy learned from the NMPC into a neural network that can be deployed efficiently in real-time environments.

In this context, the results obtained with the more classical linear control architectures (PD + FF, 4WS, 4WS + FF, 2WS + TV, 4WS + TV) should not be considered as stand-alone solutions, but as fundamental references for understanding the dynamics of the system and the respective contribution of each actuator. For this reason, it is appropriate to plot, alongside the results presented so far, also the outcomes obtained from the neural network controllers. By placing the neural networks on the same performance maps as the conventional controllers, it becomes possible to carry out a truly meaningful comparative analysis, where the relative weight of each actuator is put into perspective against the behavior of a learning-based strategy inspired by an NMPC. This holistic view not only provides clarity on the incremental benefits offered by each actuator, but also helps frame the classical approaches as a necessary stepping stone towards advanced machine-learning-based control solutions.

Table 6.2: KPIs for each controller in the SOFT CIRCUIT.

controller	MAX e_{ct} [m]	RMS e_{ct} [m]	MAX e_h [deg]	RMS e_h [deg]	MAX β [deg]	RMS β [deg]	RMS δ_f [deg]	RMS δ_r [deg]	RMS $\dot{\delta}_f$ [deg/s]	RMS $\dot{\delta}_r$ [deg/s]
PD+FF 2WS	0.393 (+0%)	0.157 (+0%)	16.81 (+0%)	7.51 (+0%)	2.89 (+0%)	1.37 (+0%)	94.2 (+0%)	-	93 (+0%)	-
NN 2WS	0.269 (-32%)	0.123 (-21%)	15.89 (-6%)	7.33 (-2%)	3.29 (+14%)	1.45 (+6%)	94.6 (+0%)	-	90 (-3%)	-
NN+PD 2WS	0.286 (-27%)	0.120 (-24%)	12.58 (-25%)	5.12 (-32%)	3.49 (+20%)	1.40 (+2%)	92.8 (-1%)	-	87 (-7%)	-
PD 4WS	0.330 (-16%)	0.120 (-23%)	13.10 (-22%)	4.99 (-34%)	2.83 (-2%)	1.30 (-5%)	93.5 (-1%)	4.6	91 (-2%)	12
PD+FF 4WS	0.272 (-31%)	0.094 (-40%)	11.30 (-33%)	4.72 (-37%)	2.14 (-26%)	0.99 (-28%)	90.2 (-4%)	6.1	89 (-5%)	12
NN 4WS	0.263 (-33%)	0.093 (-41%)	11.58 (-31%)	4.73 (-37%)	1.74 (-40%)	0.40 (-71%)	72.6 (-23%)	24.1	85 (-9%)	27
NN+PD 4WS	0.242 (-38%)	0.109 (-31%)	11.86 (-29%)	4.91 (-35%)	1.14 (-61%)	0.38 (-73%)	71.8 (-24%)	24.4	84 (-10%)	26
PD TV 2WS	0.236 (-40%)	0.097 (-38%)	12.10 (-28%)	4.93 (-34%)	2.69 (-7%)	1.24 (-10%)	91.9 (-2%)	-	92 (-2%)	-
NN TV 2WS	0.285 (-27%)	0.118 (-25%)	12.26 (-27%)	5.27 (-30%)	3.95 (+36%)	1.47 (+7%)	92.6 (-2%)	-	97 (+4%)	-
PD TV 4WS	0.363 (-8%)	0.126 (-20%)	11.80 (-30%)	4.53 (-40%)	2.09 (-28%)	0.91 (-34%)	87.4 (-7%)	5.8	90 (-4%)	12

Table 6.3: KPIs for each controller in the SHARP CIRCUIT.

controller	MAX e_{ct} [m]	RMS e_{ct} [m]	MAX e_h [deg]	RMS e_h [deg]	MAX β [deg]	RMS β [deg]	RMS δ_f [deg]	RMS δ_r [deg]	RMS $\dot{\delta}_f$ [deg/s]	RMS $\dot{\delta}_r$ [deg/s]
PD+FF 2WS	0.657 (+0%)	0.229 (+0%)	17.50 (+0%)	6.09 (+0%)	3.73 (+0%)	1.48 (+0%)	103.0 (+0%)	-	111 (+0%)	-
NN 2WS	0.485 (-26%)	0.172 (-25%)	16.42 (-6%)	5.42 (-11%)	3.68 (-1%)	1.60 (+8%)	100.2 (-3%)	-	106 (-5%)	-
NN+PD 2WS	0.487 (-26%)	0.140 (-39%)	18.21 (+4%)	5.55 (-9%)	3.43 (-8%)	1.51 (+2%)	100.4 (-3%)	-	102 (-8%)	-
PD 4WS	0.561 (-15%)	0.195 (-15%)	16.30 (-7%)	5.55 (-9%)	3.50 (-6%)	1.35 (-9%)	102.0 (-1%)	5.4	106 (-4%)	13
PD+FF 4WS	0.395 (-40%)	0.117 (-49%)	16.45 (-6%)	5.50 (-10%)	2.76 (-26%)	1.08 (-27%)	97.0 (-6%)	6.3	107 (-4%)	15
NN 4WS	0.333 (-49%)	0.132 (-42%)	16.87 (-4%)	5.40 (-11%)	1.11 (-70%)	0.38 (-74%)	79.9 (-22%)	25.6	100 (-10%)	30
NN+PD 4WS	0.240 (-63%)	0.087 (-62%)	17.22 (-2%)	5.17 (-15%)	1.41 (-62%)	0.51 (-65%)	79.4 (-23%)	23.3	94 (-15%)	29
PD TV 2WS	0.463 (-29%)	0.144 (-37%)	17.39 (-1%)	5.23 (-14%)	3.41 (-9%)	1.23 (-17%)	99.1 (-4%)	-	108 (-3%)	-
NN 2WS TV	0.362 (-45%)	0.155 (-32%)	18.98 (+8%)	5.67 (-7%)	4.30 (+15%)	1.53 (+3%)	98.9 (-4%)	-	121 (+9%)	-
PD TV 4WS	0.284 (-57%)	0.130 (-44%)	14.88 (-15%)	5.35 (-12%)	2.91 (-22%)	1.37 (-8%)	93.4 (-9%)	6.5	107 (-4%)	14

SHARP CURVE

controller	MAX e_{ct} [m]	RMS e_{ct} [m]	MAX e_h [deg]	RMS e_h [deg]	MAX β [deg]	RMS β [deg]	RMS δ_f [deg]	RMS δ_r [deg]	RMS $\dot{\delta}_f$ [deg/s]	RMS $\dot{\delta}_r$ [deg/s]
PD+FF 2WS	0.657 (+0%)	0.361 (+0%)	17.50 (+0%)	7.62 (+0%)	3.73 (+0%)	1.66 (+0%)	130.3 (+0%)	-	165 (+0%)	-
NN 2WS	0.485 (-26%)	0.278 (-23%)	16.42 (-6%)	6.07 (-20%)	3.68 (-1%)	1.71 (+3%)	115.7 (-11%)	-	138 (-16%)	-
NN+PD 2WS	0.487 (-26%)	0.240 (-33%)	18.21 (+4%)	6.75 (-11%)	3.43 (-8%)	1.53 (-8%)	118.2 (-9%)	-	137 (-17%)	-
PD 4WS	0.445 (-32%)	0.232 (-36%)	16.85 (-4%)	7.31 (-4%)	3.36 (-10%)	1.68 (+1%)	119.1 (-9%)	6.7	114 (-31%)	52
PD+FF 4WS	0.225 (-66%)	0.102 (-72%)	18.75 (+7%)	6.41 (-16%)	2.98 (-20%)	1.44 (-13%)	119.3 (-8%)	8.6	111 (-33%)	15
NN 4WS	0.229 (-65%)	0.136 (-62%)	16.87 (-4%)	6.34 (-17%)	1.11 (-70%)	0.45 (-73%)	99.2 (-24%)	26.9	144 (-12%)	39
NN+PD 4WS	0.189 (-71%)	0.105 (-71%)	17.22 (-2%)	5.96 (-22%)	1.38 (-63%)	0.60 (-64%)	95.4 (-27%)	25.0	125 (-24%)	38
PD TV 2WS	0.455 (-31%)	0.231 (-36%)	19.14 (+9%)	6.74 (-12%)	3.77 (+1%)	1.78 (+7%)	122.5 (-6%)	-	109 (-34%)	-
NN TV 2WS	0.362 (-45%)	0.222 (-38%)	18.98 (+8%)	6.51 (-15%)	3.43 (-8%)	1.56 (-6%)	113.2 (-13%)	-	138 (-16%)	-
PD TV 4WS	0.198 (-70%)	0.098 (-73%)	14.88 (-15%)	5.38 (-29%)	2.91 (-22%)	1.47 (-11%)	110.8 (-15%)	8.6	107 (-35%)	14

The design of the gains for each controller was carried out with the clear intention of identifying the most appropriate trade-off between the complexity of the design process and the accuracy of the resulting performance. This principle represents, in many respects, the very core of the engineering mindset: it is rarely sufficient to pursue accuracy alone, and it is equally unwise to overemphasize simplicity at the expense of performance. Instead, the true art of control system design lies in the balanced compromise between these two aspects. Within the context of the present work, this philosophy has been reflected in the systematic tuning of the controllers, where each architecture was carefully adjusted to achieve a fair balance between computational or structural complexity and the improvement in path tracking accuracy.

The benefits of this approach are particularly evident in the **Sharp Circuit**, where the imposed maneuver can be considered truly demanding and critical. Here, the refined tuning of the more advanced control strategies has led to remarkable improvements in performance. In fact, for the most critical curve of the circuit, the peak lateral error was reduced to an astonishing value of only 20 cm, which corresponds to an improvement of approximately 70% when compared to the baseline configuration of the PD + FF 2WS controller used as a benchmark. Such a drastic reduction not only highlights the effectiveness of the proposed multi-actuator strategies, but also provides a concrete demonstration of the real value of integrating multiple control actions in a synergistic manner.

On the other hand, the **Soft Circuit** presents a less extreme scenario, where the intrinsic smoothness of the trajectory naturally attenuates the difficulties encountered by the controllers. In this case, although improvements are still noticeable when adopting more sophisticated configurations, the margins of gain

are considerably smaller compared to the Sharp Circuit. In particular, it becomes evident that configurations such as 2WS + TV or even pure 4WS can already provide satisfactory results, without requiring the complexity of the full 4WS + TV formulation. This observation reinforces an important consideration for engineering practice: in less demanding operational conditions, it can be more convenient to rely on simpler architectures, which offer adequate performance while minimizing design, calibration, and implementation costs.

In summary, the comparative reading of the two scenarios demonstrates how the choice of the control architecture should not be based on absolute principles, but rather on the specific characteristics of the driving task and the operational context. Advanced architectures reveal their full potential under harsh and challenging maneuvers, while simpler configurations may be sufficient—and sometimes even preferable—in smoother scenarios.

For this reason, a subsequent simulation-based analysis will be carried out, aiming to further confirm, through a complementary environment, the validity of the model and the effectiveness of the different controllers. This twofold validation process, first experimental and then simulated, provides a robust and comprehensive evaluation framework that strengthens the conclusions drawn and supports the reliability of the proposed control solutions.

It is also possible to extend our analysis to a more narrow comparison, only taking into account the proposed linear controllers. In this way, it results easier and tidier to plot also the significant quantities instead of just reporting the tables.

Soft circuit:

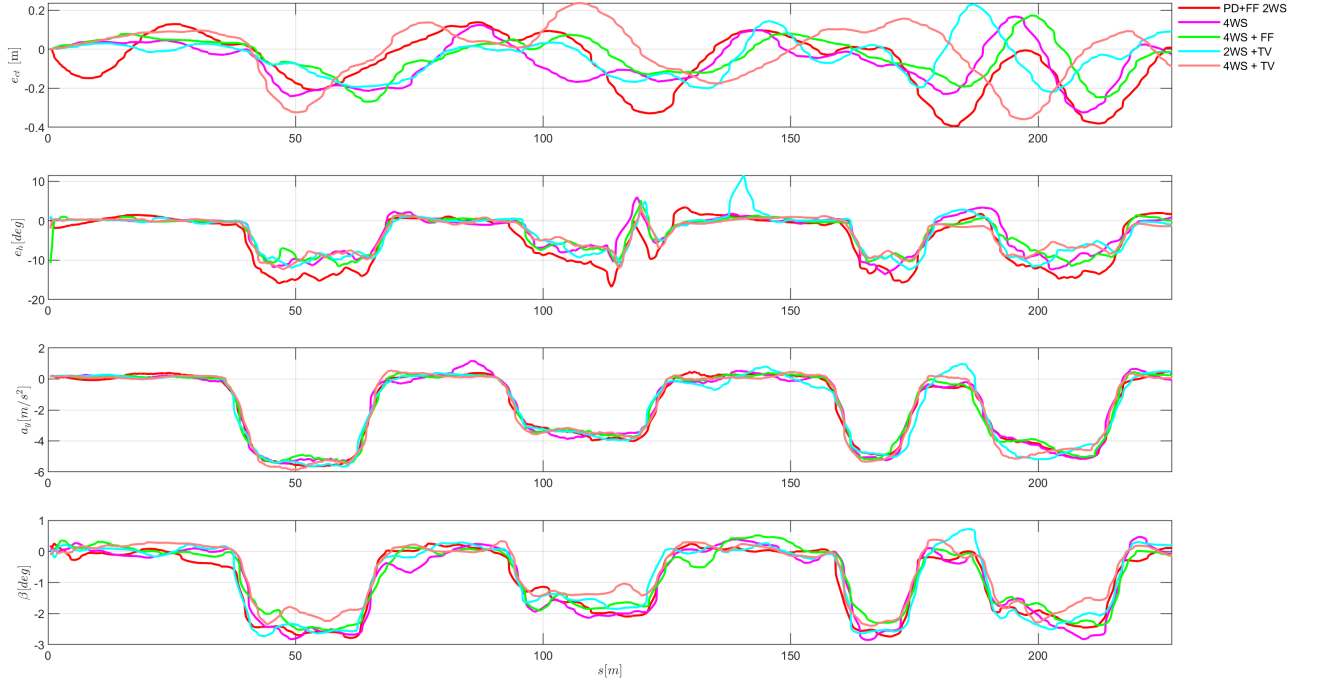


Figure 6.15: Overlap of results for the soft circuit.

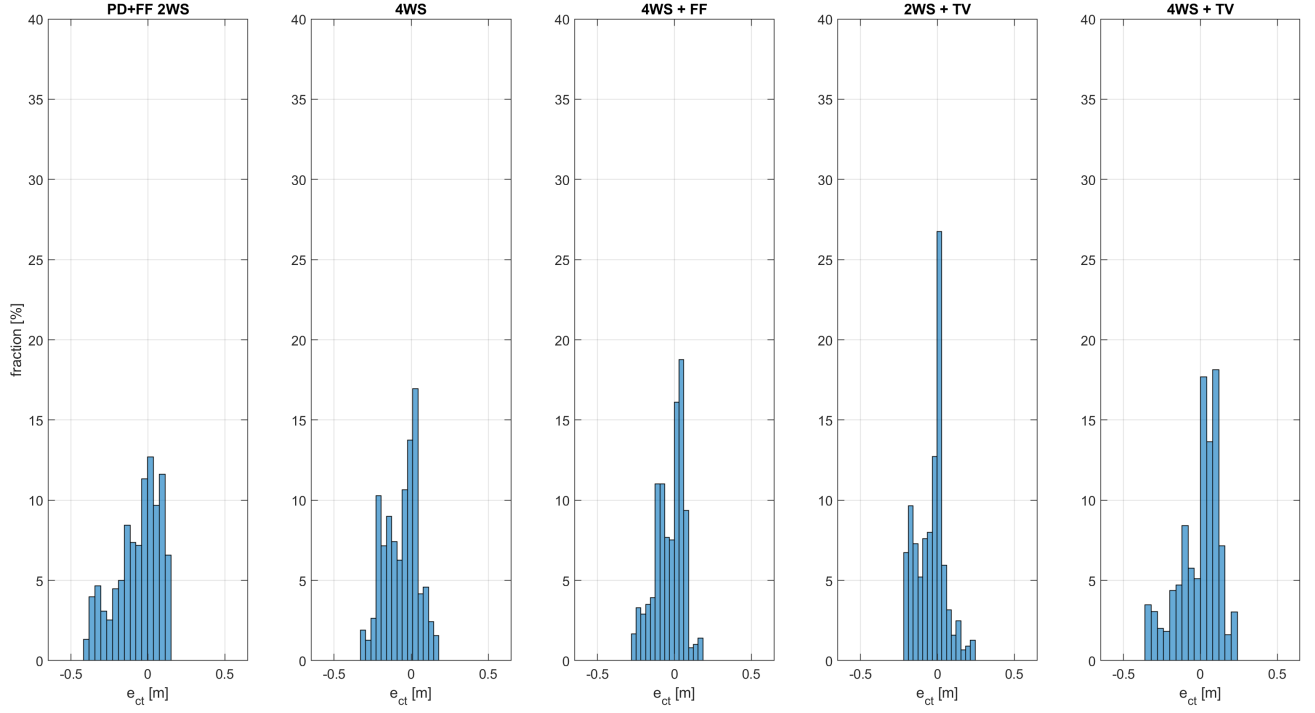


Figure 6.16: Histogram of cross track error, soft circuit.

It is clear to notice that the 2WS + TV is the best controller, providing the lowest peak error as well as the highest amount of times in which the vehicle was close to the zero lateral error condition.

Sharp circuit:

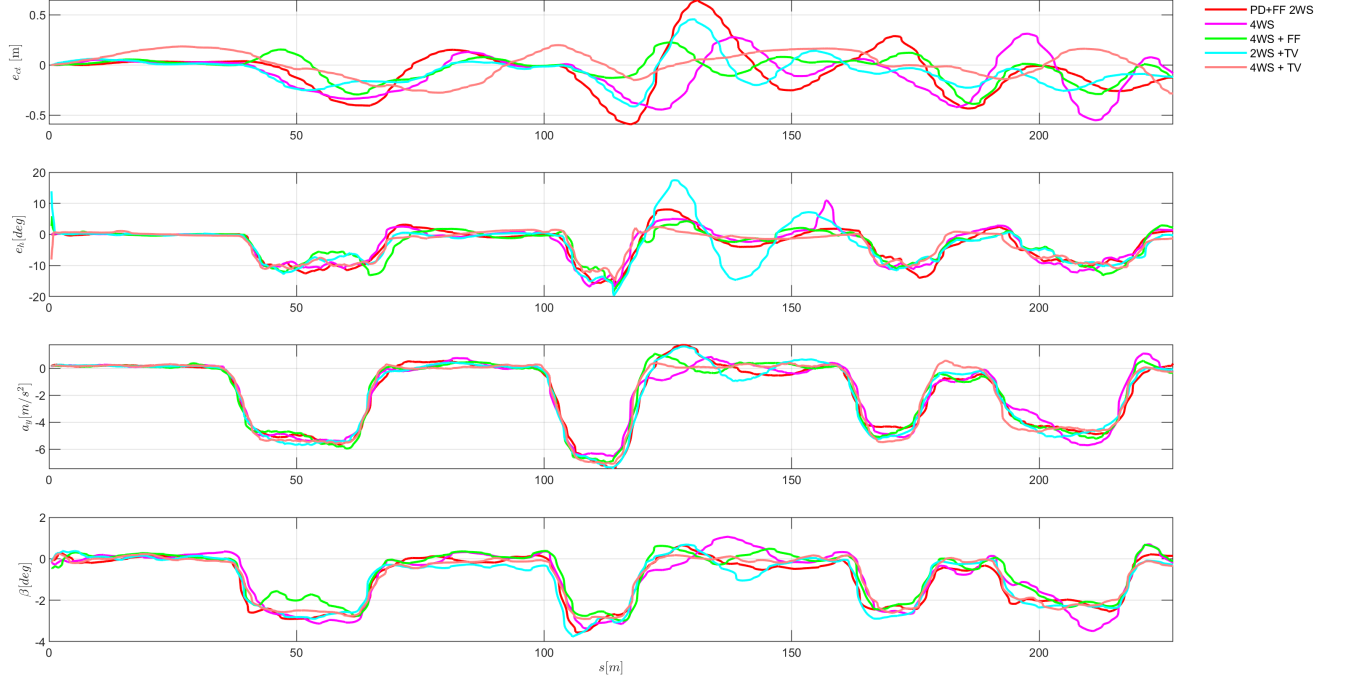


Figure 6.17: Overlap of results for the sharp circuit.

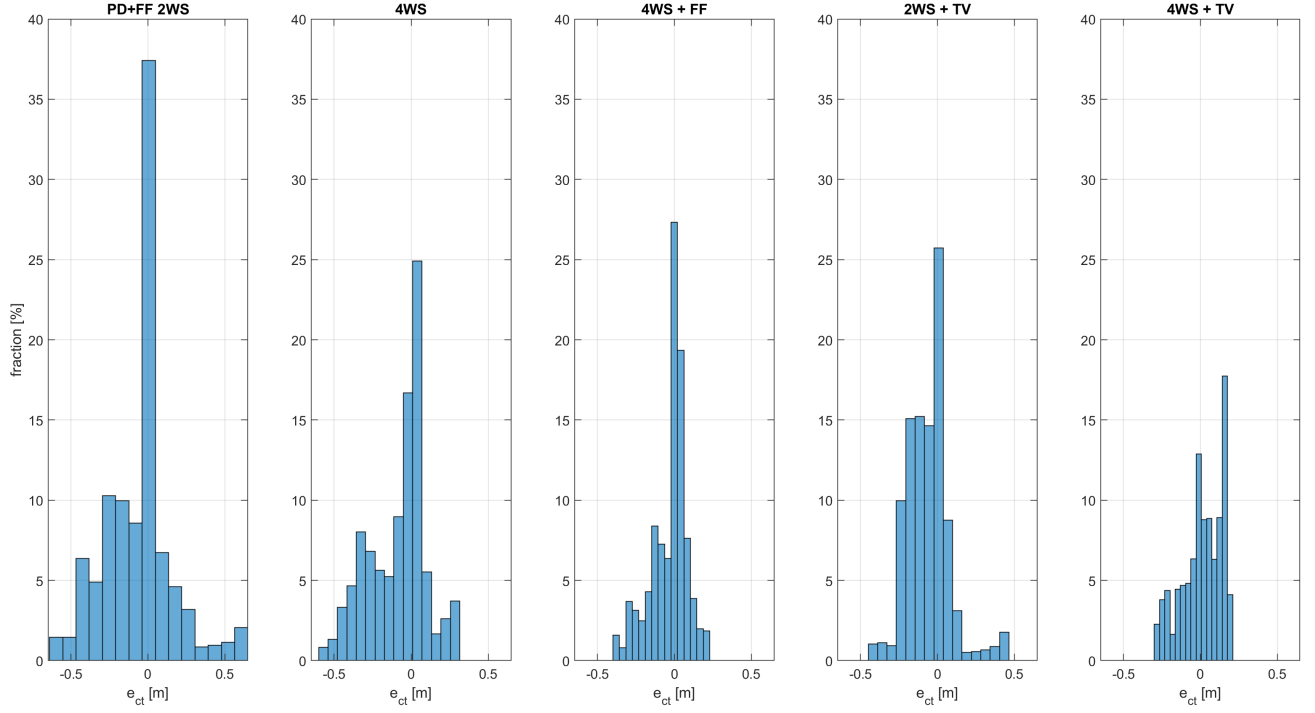


Figure 6.18: Histogram of cross track error, sharp circuit.

As far as the sharp circuit is concerned, the situation has to be properly analyzed. It is true that the highest peak in the histograms is present for the benchmark 2WS but that controller is also the one that brings the widest range of error, meaning that the peak, for that controller is way higher with respect to other strategies such as 4WS + FF or 4WS + TV in which, not only the peak is significantly reduced, but the amount of times in which the vehicle is close to zero error condition is significant too.

6.7 Results of the LQRs

LQR 2WS

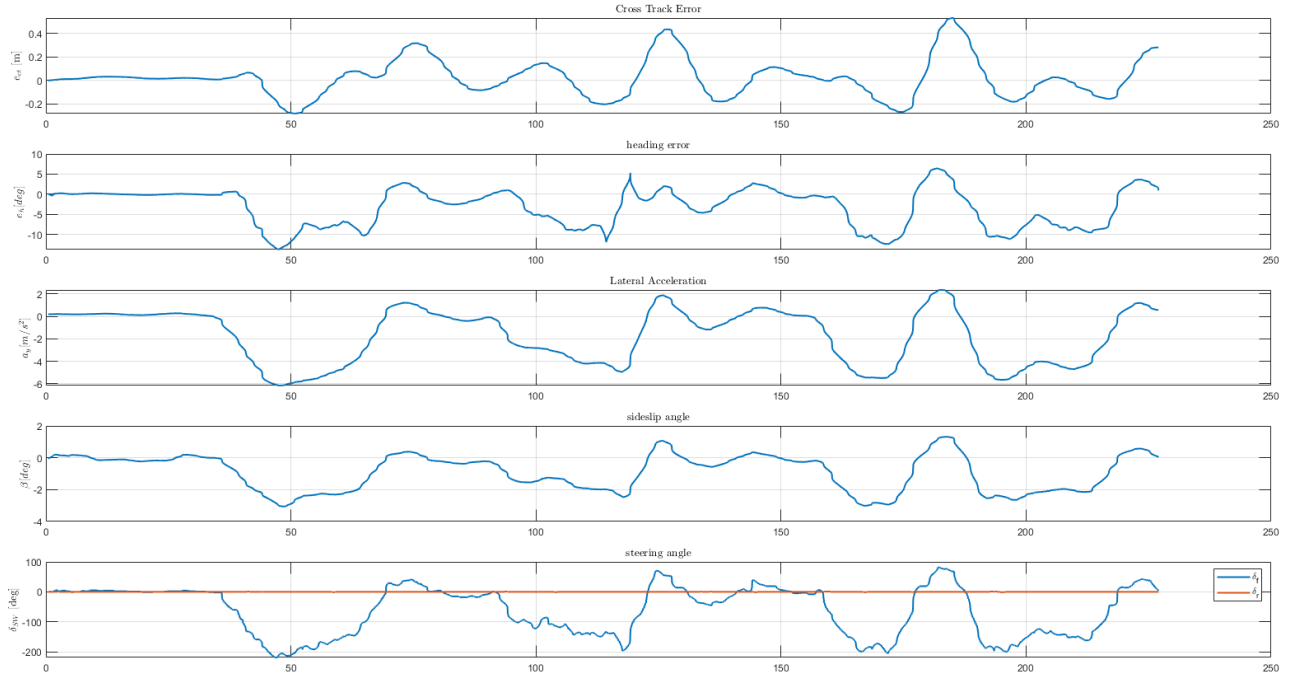


Figure 6.19: Results obtained from the LQR 2WS (Soft Circuit).

— KPIs —

- Max $|CTE|$ [m] : 0.533
- RMS CTE [m] : 0.160
- Max $|HE|$ [deg] : 13.720
- RMS HE [deg] : 5.547
- Max $|\beta|$ [deg] : 3.057
- RMS β [deg] : 1.421
- RMS Δ_f [deg] : 100.530
- RMS Δ_r [deg] : 0.202

- RMS Δ_{pd} [deg] : 0.000
- RMS $\dot{\Delta}_f$ [deg/s] : 107.616
- RMS $\dot{\Delta}_r$ [deg/s] : 0.543

LQR 4WS

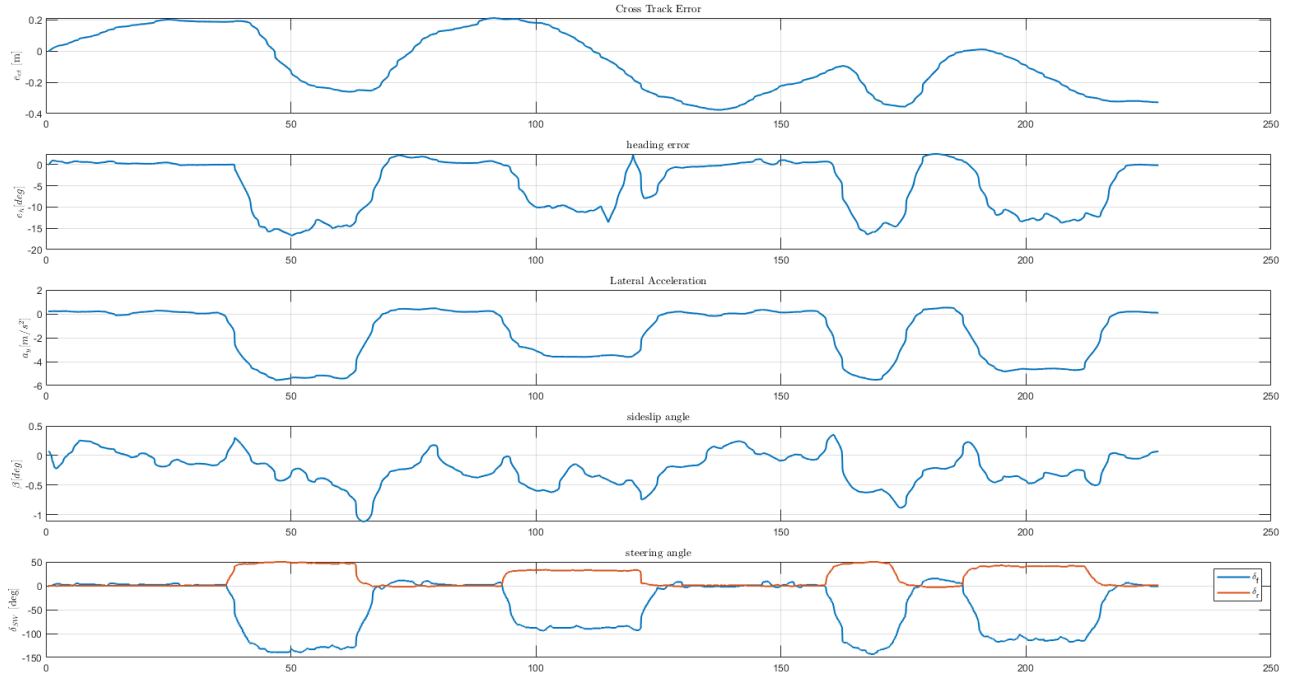


Figure 6.20: Results obtained from the LQR 4WS (Soft Circuit).

— KPIs —

- Max $|CTE|$ [m] : 0.377
- RMS CTE [m] : 0.205
- Max $|HE|$ [deg] : 16.742
- RMS HE [deg] : 7.655
- Max $|\beta|$ [deg] : 1.117
- RMS β [deg] : 0.346

- RMS Δ_f [deg] : 69.913
- RMS Δ_r [deg] : 25.735
- RMS Δ_{pd} [deg] : 0.000
- RMS $\dot{\Delta}_f$ [deg/s] : 66.895
- RMS $\dot{\Delta}_r$ [deg/s] : 26.411

The analysis of the LQR controllers highlights some important considerations. First of all, it must be remarked that the LQR formulation completely failed when tested on the sharp circuit, in both the 2WS and the 4WS configurations. For this reason, the following discussion is restricted to the results obtained on the soft circuit only. Even in this more favourable scenario, a clear deterioration of the performance can be observed when comparing the LQR outcomes with those obtained with the PD controllers. Both in the 2WS and in the 4WS case, the cross-track error increases significantly, confirming that the tuning flexibility of the PD formulation provided more robust and effective results.

Despite this general decline in performance, some useful insights can still be extracted. The addition of the rear steering actuator within the LQR framework did in fact prove beneficial: the maximum lateral deviation decreased from approximately 53 cm in the 2WS case to about 37 cm in the 4WS case. This confirms once more that additional actuation is valuable for improving tracking performance. Nevertheless, when compared with the PD results, where the peak error drops from 39 cm in 2WS to 27 cm in 4WS, the superiority of the PD design becomes evident. The enhanced tuning freedom associated with proportional-derivative structures – particularly in their decoupled and intuitive gain design – clearly emerged as a winning factor, allowing the controller to be successfully deployed also on the more demanding sharp circuit, unlike the LQR.

Another interesting aspect lies in the different actuator usage patterns between the two approaches. With the LQR design, steering angles are distributed in a markedly different way: the front axle requires smaller steering inputs, while the rear axle becomes more actively engaged, thus moving closer to what would be considered an “optimal” distribution of control authority. By contrast, in the PD framework, the philosophy was fundamentally different: the baseline 2WS architecture was retained as a benchmark, and the influence of additional actuators was introduced in a limited and gradual manner. As a result, the effect of 4WS or TV was neither negligible nor dominant, but always complementary to the primary action of the front steering.

Overall, the LQR exercise underlines two key points: on the one hand, it confirms that additional actuators are beneficial even within a suboptimal control architecture; on the other, it reveals the limits of the LQR design in this specific

context, where the lack of tuning flexibility and the intrinsic coupling of the Riccati formulation prevented the controller from achieving the same level of robustness and adaptability that was reached with the PD approach.

Chapter 7

Experimental vs. Simulation Results

7.1 1st Montecarlo Analysis

The experimental campaign provided results of remarkable quality, especially when considering the demanding conditions under which the tests were performed. The selected reference velocity of 30 km/h may appear modest in absolute terms, yet it is rather significant for the considered test track. In fact, the circuit is characterized by a sharp corner with a radius of curvature of only 10 m, which represents a highly challenging manoeuvre for any path-tracking controller. Despite these critical conditions, the controllers exhibited a stable behaviour, managing to keep both the lateral and heading errors within acceptable limits throughout the tests.

In order to validate and strengthen the comparison between experimental and numerical results, the latter were not limited to a single deterministic simulation. Instead, a Monte Carlo analysis was performed entirely within the simulation environment. This methodology is particularly effective in evaluating the robustness of the control strategies, as it allows the introduction of variability in key vehicle and environmental parameters, and consequently the observation of the system response under perturbed conditions.

The following parameters were identified as the most influential for the considered application:

- Road friction coefficient μ (scaled with K_μ);
- Longitudinal stiffness C_x (scaled with K_x);
- Lateral stiffness C_y (scaled with K_y);
- Vehicle mass m (increased starting from the nominal value);

- Actuation delay τ in the steering command.

The parameters were varied according to the following ranges, consistently with the uncertainties expected in the real system:

$$\begin{aligned} K_\mu &= \text{linspace}(0.9, 1.1, 20), \\ K_x &= \text{linspace}(0.9, 1.1, 20), \\ K_y &= \text{linspace}(0.9, 1.1, 20), \\ m &= \text{linspace}(0, 40, 20), \\ \tau &= \text{round}(\text{linspace}(-10, 10, 20)). \end{aligned}$$

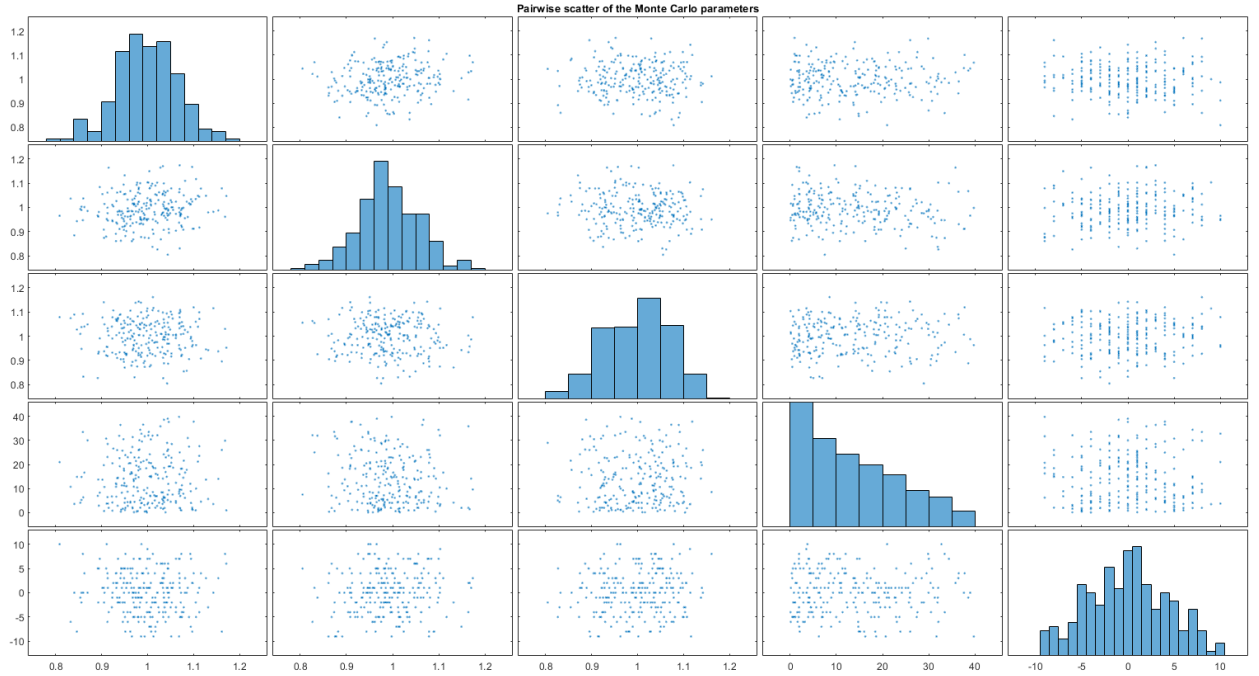


Figure 7.1: Pairwise scatter of the Monte Carlo parameters.

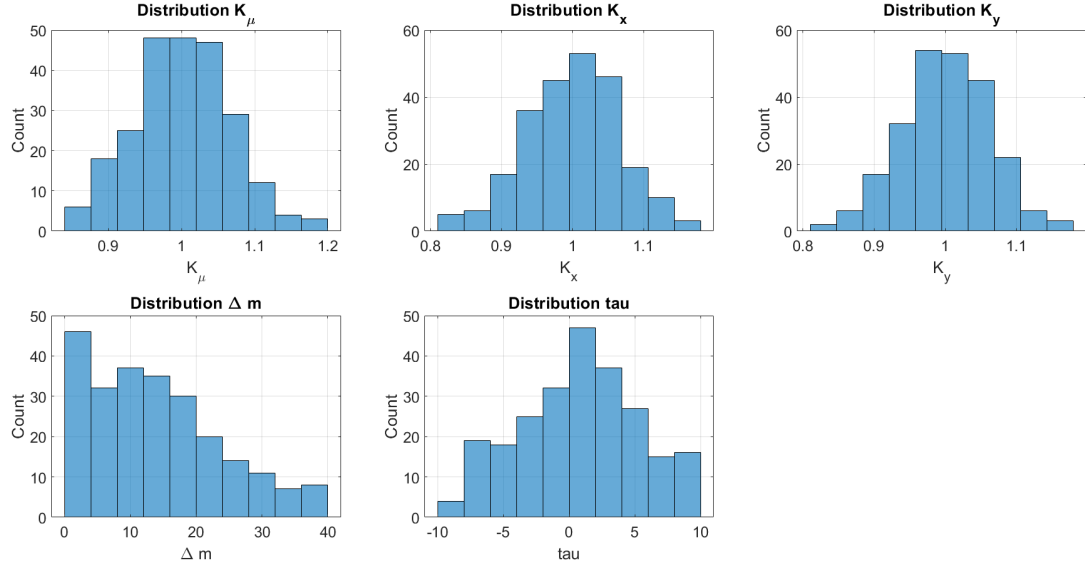


Figure 7.2: Distribution of the Monte Carlo parameters.

For each configuration, the Monte Carlo analysis generated a set of trajectories, from which the peak and RMS values of both the lateral error and the heading error were extracted. This procedure ensures a statistically consistent evaluation, highlighting not only the nominal performance of each controller but also its robustness with respect to modelling uncertainties and environmental disturbances.

The results presented in the following sections refer to the same track employed for the experimental campaign, with the identical set of controllers under investigation. This guarantees a direct comparison between experimental measurements and simulated data, ultimately allowing for a critical assessment of the advantages and limitations of the proposed control architectures.

The results of the Monte Carlo analysis confirm the expectations associated with the adopted methodology. Since the perturbed parameters were varied only within a limited neighbourhood of their nominal values, the resulting point clouds exhibit a remarkable density and uniformity. This outcome is fully consistent with the hypothesis of robustness of the proposed controllers: the scattering of the points remains confined, and only a few outliers can be noticed across the different configurations. In other words, despite the intrinsic linearity of the considered control strategies — which might be regarded as a “classical” approach compared to modern nonlinear or adaptive formulations — the obtained behaviour demonstrates that they are capable of ensuring coherent and reliable performance even under multiple simultaneous parameter variations, both of vehicle dynamics and environmental conditions.

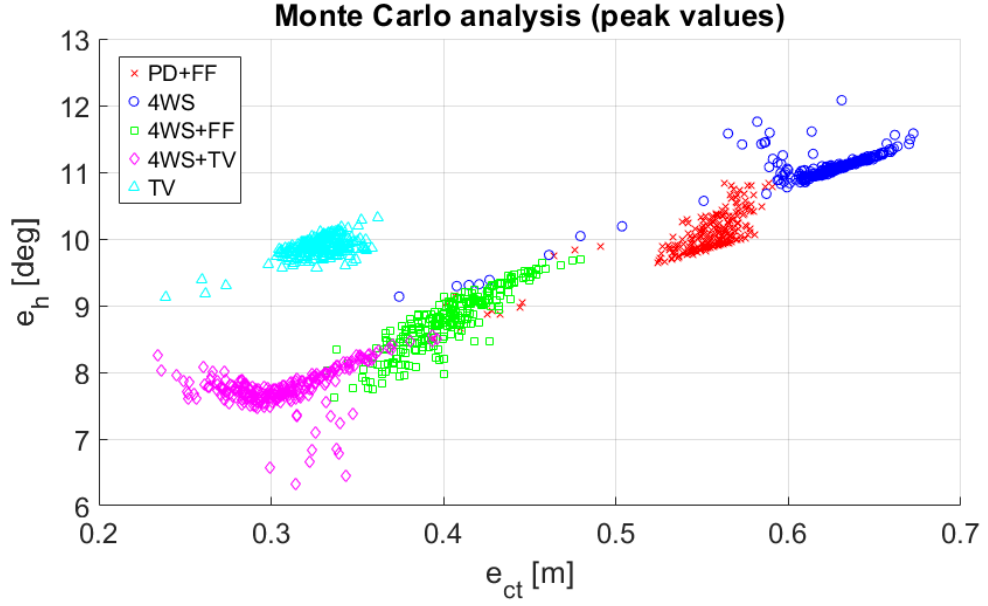


Figure 7.3: Peaks of the errors in the first Monte Carlo analysis.

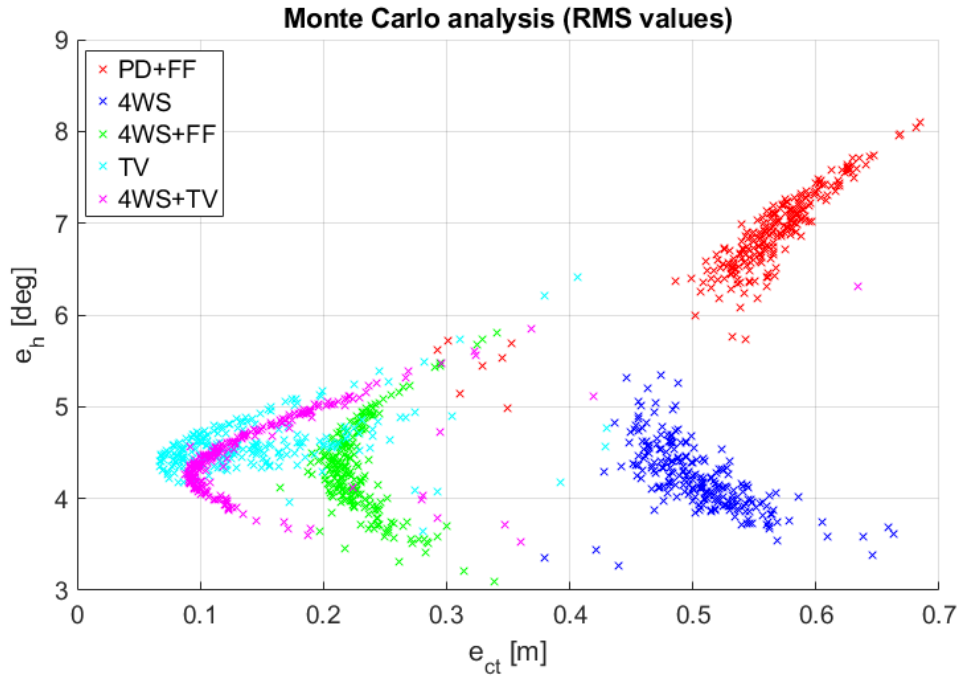


Figure 7.4: Rms of the errors in the first Monte Carlo analysis.

Another important aspect is the preservation of the same qualitative trends that were already appreciated in the experimental campaign. Specifically, the baseline configurations (PD + FF with two-wheel steering, and four-wheel steering without feedforward) prove to be insufficient, as their associated error values remain consistently higher than those of the other strategies. The introduction of a feedforward term on the rear axle already allows for a significant reduction of both lateral and heading errors, confirming the effectiveness of this design improvement.

Furthermore, torque vectoring once again emerges as a powerful actuator, capable of enhancing stability and path-tracking accuracy more than the pure four-wheel steering configuration. Finally, the combined use of both torque vectoring and four-wheel steering yields the best overall performance, with point clouds clustered in the lowest-error region of the plot. This final configuration thus represents the most effective control solution among those tested, providing substantial improvements at the cost of increased design complexity and the need for synergistic tuning of multiple actuation channels.

Overall, the Monte Carlo analysis provides strong evidence that the proposed architectures are not only effective under nominal conditions, but also robust with respect to variations in vehicle parameters and environmental factors. This robustness, combined with the observed performance hierarchy among the different controllers, consolidates the conclusions already drawn from experimental testing and further supports the adoption of advanced actuation strategies — such as the integration of four-wheel steering and torque vectoring — for high-performance path tracking.

7.2 2nd Montecarlo Analysis

A second Monte Carlo campaign was carried out, this time conducted entirely in a purely simulation-based environment. The set of controllers implemented at actuator level remained unchanged, respectively: **PD + FF**, **2WS**, **4WS**, **4WS + FF**, **2WS + TV**, and **4WS + TV**.

The structure of the Monte Carlo parameters was the same as in the previous study: small variations were introduced in the friction coefficient, longitudinal and cornering stiffnesses, total mass, and steering delay.

The reference manoeuvre considered in this second analysis was deliberately more complex and demanding, not particularly suitable for linear path-tracking strategies, since it pushes the vehicle dynamics close to the handling limits. The trajectory consisted of a first straight segment, followed by two consecutive curves with opposite directions (a right turn immediately followed by a left turn), and then another straight segment. The two curves were separated by a very short rectilinear portion of only 5 m and both had a radius of curvature of just 10 m.

Due to time constraints, this activity was restricted to simulation only. The controller gains were therefore not tuned experimentally but rather extracted directly from an optimization process, which provided the most effective results compatible with the considered actuators.

Table 7.1: Gains adopted in the second Monte Carlo analysis.

Controller	K_p Front	K_d Front	K_p Rear	K_d Rear	KFF Gain Rear	K_p TV	K_d TV
2WS PD + FF	6.00	21.00	0.00	0.00	0.00	0.00	0.00
4WS	0.00	45.28	-85.00	0.00	0.00	0.00	0.00
4WS + FF	0.00	43.87	0.00	-84.17	-140.88	0.00	0.00
2WS + TV	33.77	50.00	0.00	0.00	0.00	-191.80	-178.39
4WS + TV	2.22	2.91	-12.57	0.00	-55.00	-120.00	-25.85

The reference manoeuvre considered in this second analysis was deliberately more complex and demanding, not particularly suitable for linear path-tracking strategies, since it pushes the vehicle dynamics close to the handling limits. The trajectory consisted of a first straight segment, followed by two consecutive curves with opposite directions (a right turn immediately followed by a left turn), and then another straight segment. The two curves were separated by a very short straight portion of only 5 m and both had a radius of curvature of just 10 m.

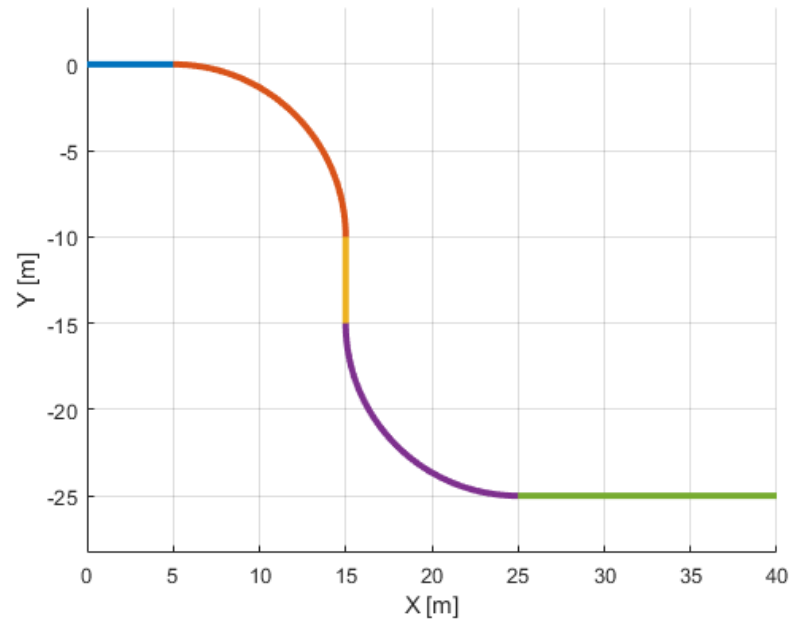


Figure 7.5: New trajectory.

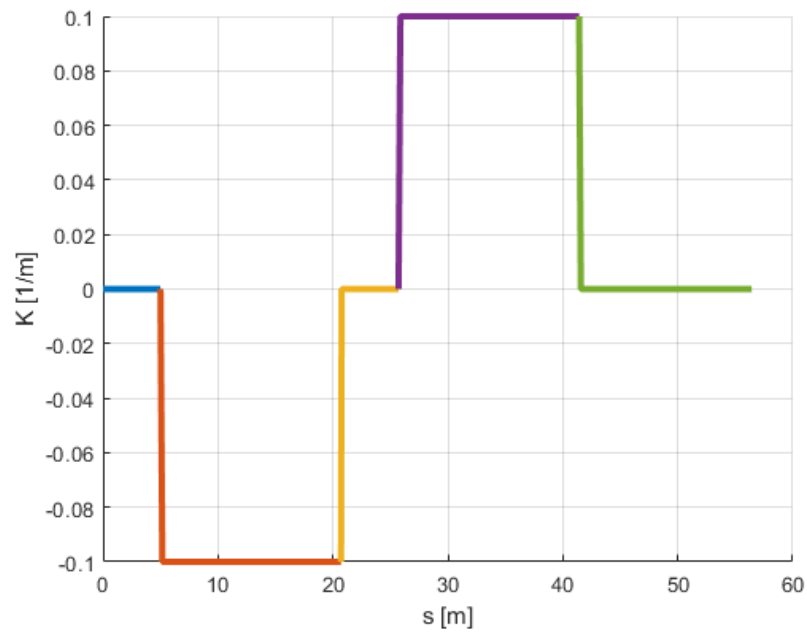


Figure 7.6: Curvature of the new trajectory.

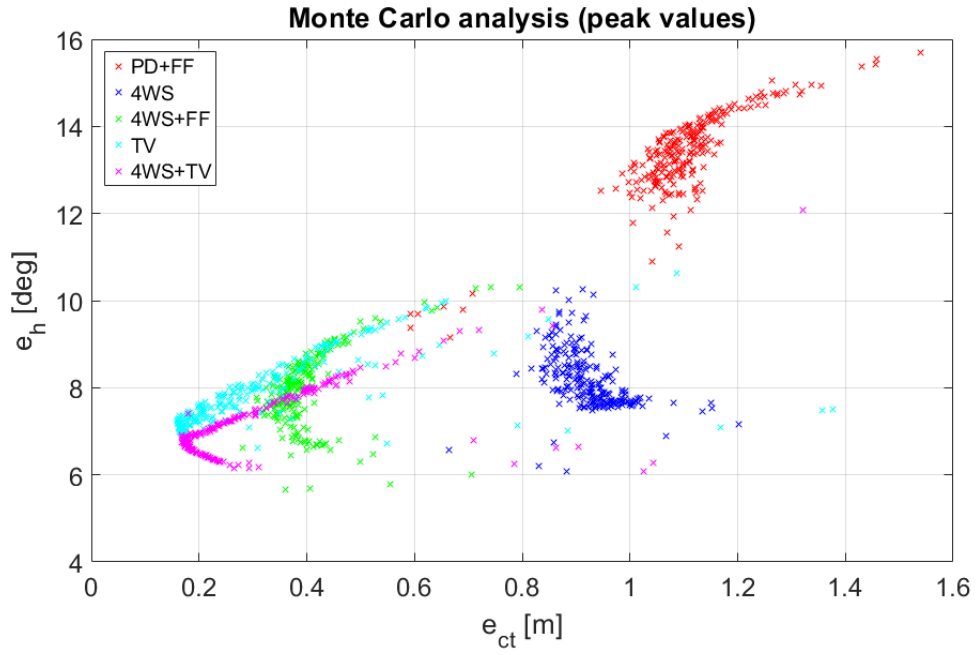


Figure 7.7: Peaks of the errors in the second Monte Carlo analysis.

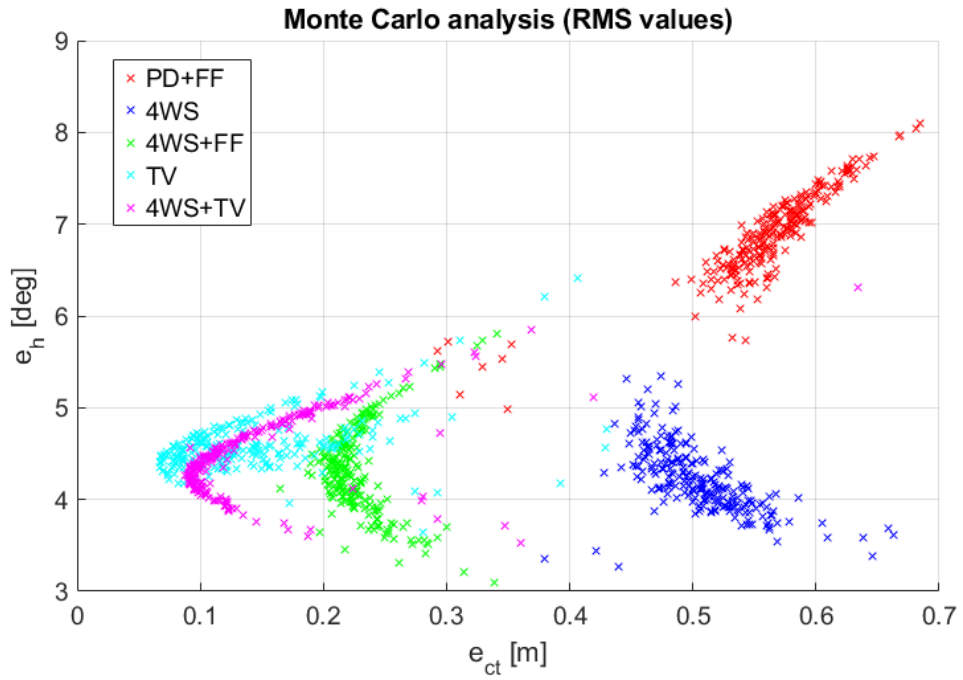


Figure 7.8: Rms of the errors in the second Monte Carlo analysis.

The obtained results are consistent with expectations: the manoeuvre under investigation is extremely demanding. A basic linear controller based on front steering only (2WS) – but also its rear-steered counterpart (4WS) – does not achieve satisfactory tracking performance. The inclusion of a feedforward contribution at the rear axle significantly enhances the results, highlighting the importance of anticipatory and constant actions. Torque Vectoring once again proves to be the most effective actuator in terms of improving stability and accuracy. Finally, the combined strategy **4WS + TV** emerges as the most promising overall solution, offering the highest performance across the evaluated scenarios.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

The experimental campaign, supported by extensive simulation work, has clearly demonstrated that the use of advanced actuators such as four-wheel steering (4WS) and torque vectoring (TV) can provide a measurable improvement in path-tracking performance. What emerges is a picture of high precision: deviations are to be measured in the order of centimeters, which makes the correlation between the simulated model and the real vehicle both evident and meaningful. In fact, the model has proven extremely valuable not only in predicting general performance trends when parameters are modified, but also in detecting potential instability conditions before these could manifest themselves during the physical tests. Nevertheless, it is essential to stress that this correlation must be interpreted as primarily qualitative rather than quantitative. It is unrealistic to expect that the exact same peak and RMS errors in terms of cross-track and heading deviations can be obtained repeatedly in different experimental runs, let alone when comparing simulated and real-world conditions. Repeatability, therefore, should be considered in terms of consistency of general behaviour, observable tendencies, and robustness of the control architecture, rather than strict numerical equivalence. This methodological approach is aligned with the broader literature on vehicle path tracking, where the reliability of results is generally understood in this same qualitative framework. From this perspective, the choice of testing the controllers on two circuits – here referred to as the “Soft” and the “Sharp” track – has proven extremely functional. On one side, it ensures the required scientific repeatability of results, validating the ability of the control schemes to reproduce comparable behaviours under different yet controlled conditions. On the other, it provides a natural benchmark to assess

the limits of applicability of classical linear controllers when faced with increasingly demanding manoeuvres. In particular, the soft track allows to appreciate performance under relatively benign conditions, whereas the sharp track pushes the architectures closer to their operational boundaries, revealing strengths and weaknesses that would otherwise remain hidden. The results collected in this work highlight that classical linear controllers remain a powerful and highly relevant tool for vehicle path tracking. As shown by the comparative tables, even when tested against neural networks trained via imitation learning from nonlinear predictive model control (NMPC) – which currently represents the state-of-the-art albeit with significant issues in real-time implementation – the gap in terms of peak lateral and angular errors is surprisingly narrow. This strongly suggests that the proposed architectures, despite their conceptual simplicity, are still competitive solutions with excellent robustness and relatively low implementation costs. Furthermore, the experiments confirm that the use of multiple actuators in combination becomes particularly significant in the most challenging scenarios. For instance, the 4WS+TV controller achieved an impressive 70% reduction in peak cross-track error on the sharp circuit compared to the PD+FF 2WS benchmark. Conversely, in the soft circuit the improvements remained much smaller, with reductions of only a few percentage points. This dichotomy illustrates that in simpler scenarios a limited control action is sufficient, and in those cases, relying on a more elementary and less complex architecture is not only acceptable but even preferable. In other words, complexity should be introduced only where strictly necessary to guarantee performance. Finally, it must be recalled that the effectiveness of any controller is highly dependent on its tuning. Gains must always be designed specifically for the trajectory, the velocity regime, and the geometric and environmental conditions of the vehicle. Although the Monte Carlo analysis carried out in this work showed that, for a fixed trajectory, variations in parameters such as mass, tire stiffness, or steering delay did not drastically alter the general trends of performance, this does not hold true for more complex trajectories. In those cases, redesigning the gains would be required, and similar adaptations would also likely be necessary under different operational conditions. Therefore, the results presented here should be regarded as both a confirmation of the robustness of classical controllers under certain variability, and a reminder of the need for careful re-tuning when moving towards more challenging environments.

8.2 Future Work

While the results presented in this thesis already demonstrate the strong potential of classical controllers enriched by advanced actuation mechanisms, a number of promising directions remain open for future research and development. These possibilities extend from refinements of the existing approaches to the exploration of entirely new control paradigms, all aimed at further enhancing the effectiveness, robustness, and adaptability of vehicle path tracking architectures.

A first natural extension concerns the exploitation of the gain scheduling framework, which had already been outlined in the early stages of this project. Such an approach could provide a valuable starting point for developing combined control strategies capable of simultaneously managing both longitudinal and lateral dynamics. In this context, a more advanced tuning process would likely be necessary in order to maximize performance across a wide range of operating conditions. A gain-scheduled controller could, for instance, adapt the actuation strategy as a function of vehicle speed, progressively shifting the emphasis between steering and yaw moment generation. This perspective opens the door to an even more ambitious line of research: the integration of path tracking objectives with lap-time optimization. In other words, the actuators investigated in this work could be exploited not only to ensure trajectory fidelity, but also to minimize lap times while preserving stability, thereby bridging the gap between classical control theory and performance-oriented driving tasks.

A second line of research, still focused on pure path tracking, involves the introduction of an integral term in the controller design. Integral action is well known to be effective in eliminating steady-state errors, as it accounts for the history of the deviation rather than its instantaneous value. This could provide a decisive improvement in trajectory fidelity, ensuring an almost perfect tracking of the reference path. In the present activity, integral action was deliberately omitted for two main reasons. First, the combination of proportional-derivative action with feedforward terms (in the steering-based controllers) already yielded satisfactory results, reducing the perceived need for additional complexity. Second, the use of an integral term would necessarily require the design and tuning of an anti-windup strategy, indispensable to avoid the uncontrolled accumulation of error when the actuator inputs reach saturation. Several techniques are available for this purpose, such as progressively reducing the integral contribution as saturation is approached, or entirely disabling it by multiplying the integral gain by zero whenever saturation occurs. The design and optimization of such anti-windup mechanisms, together with the tuning of both the integral and discharge gains, could significantly improve performance, especially in highly constrained scenarios.

A third and highly relevant proposal concerns the systematic implementation of all the controllers developed in this thesis within an LQR framework. A preliminary

formulation of an LQR controller has already been discussed in the dedicated section, and its extension to the complete set of control architectures represents a promising avenue for future work. However, two major challenges need to be addressed. The first relates to the fidelity of the vehicle model: the standard linear tire formulation embedded in the MATLAB `lqr` command is a useful starting point, but it requires extensive refinement to accurately represent the cornering stiffness behaviour under varying lateral accelerations. It is well established that C_f and C_r decrease with increasing a_y , and this variation can be modelled either linearly or quadratically. To ensure realistic performance, however, a direct dependency on a_y should be incorporated, which would require a dedicated experimental campaign to characterize the cornering stiffness as a function of lateral acceleration. The second challenge lies in the limited flexibility of the LQR tuning process. Unlike direct state-feedback designs, in which individual gains can be adjusted independently, LQR tuning operates through the weighting matrices Q and R . Altering one of these parameters inevitably modifies the entire set of Riccati equations, thus affecting all feedback gains simultaneously. For example, increasing the weight q_1 associated with the cross-track error does not only amplify the corresponding feedback gain K_1 , but also alters the complete set of gains in a non-trivial and non-bijective way. This makes intuitive manual tuning particularly difficult. Overcoming this limitation would require a highly reliable MATLAB/Simulink environment, capable of accurately reproducing real-world issues such as the low-frequency sampling rate of GPS sensors and other noise sources. Once such a model is available, numerical optimizers such as `fmincon` or `surrogateopt` could be employed to determine the most effective configuration in simulation, and then transfer it to experimental tests. Preliminary attempts in this direction have already proven promising, but issues with derivative-based correlations – especially given the step-like nature of GPS signals sampled at only 5 Hz – have made validation challenging. A refined LQR approach, combined with a carefully characterized stiffness model, could therefore emerge as a powerful tool for future vehicle path tracking controllers.

In summary, the future development of this research may proceed along three main lines: (i) the integration of gain scheduling to harmonize longitudinal and lateral control and pursue lap-time optimization; (ii) the design and implementation of integral action with robust anti-windup strategies to further improve tracking fidelity; and (iii) the systematic application and refinement of LQR-based controllers, supported by accurate modelling of tire behaviour and simulation-driven optimization. Together, these directions could significantly extend the scope and robustness of the present work, providing valuable insights not only for academic research but also for the practical deployment of advanced control strategies in real-world autonomous and high-performance vehicles.

Bibliography

- [1] SAE International. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. Revised April 2021, originally issued January 2014. Apr. 2021 (cit. on p. 2).
- [2] Suraj Prashad and Udit Narayana Gedala. *Simplified Numerical Approach for Steering System Design Parameter Selection for Off-Road Racing*. July 2022 (cit. on p. 16).
- [3] Henrique de Carvalho Pinheiro, Alessandro Messana, and Massimiliana Carello. «In-wheel and on-board motors in BEV: lateral and vertical performance comparison». In: *2022 IEEE International Conference on Environment and Electrical Engineering and 2022 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe)*. 2022, pp. 1–6. DOI: 10.1109/EEEIC/ICPSEurope54979.2022.9854774 (cit. on p. 30).
- [4] Raj Rajamani. «Vehicle Dynamics and Control». In: *Springer* (2012). DOI: 10.1007/978-1-4614-1433-9.
- [5] J. Ackermann and W. Sienel. «Robust yaw damping of cars with front and rear wheel steering». In: *Proceedings of the 34th IEEE Conference on Decision and Control*. 1995, pp. 2241–2246. DOI: 10.1109/CDC.1995.478982.
- [6] A. Sorniotti, P. Gruber, T. Goggia, and S. S. Pennycott. «Direct yaw moment control for four-wheel-drive electric vehicles with independently actuated drivetrains». In: *IEEE Transactions on Vehicular Technology* 64.5 (2015), pp. 1888–1900. DOI: 10.1109/TVT.2015.2389615.
- [7] Paolo Falcone, Francesco Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat. «Predictive active steering control for autonomous vehicle systems». In: *IEEE Transactions on Control Systems Technology* 15.3 (2007), pp. 566–580. DOI: 10.1109/TCST.2007.894653.
- [8] J. Y. Wong. *Theory of Ground Vehicles*. 4th. Wiley, 2008. ISBN: 978-0470170380.
- [9] Carmine Caponio et al. «Nonlinear Model Predictive Yaw Moment Control Through Electric Axle and Friction Brake Torque Distribution». In: *IEEE Access* 13 (2025), pp. 106540–106560. DOI: 10.1109/ACCESS.2025.3568635.

- [10] Pietro Stano, Davide Tavernini, Umberto Montanaro, Manuela Tufo, Giovanni Fiengo, Luigi Novella, and Aldo Sorniotti. «Enhanced Active Safety Through Integrated Autonomous Drifting and Direct Yaw Moment Control via Nonlinear Model Predictive Control». In: *IEEE Transactions on Intelligent Vehicles* 9.2 (2024), pp. 4172–4190. DOI: 10.1109/TIV.2023.3340992.
- [11] Shiyue Zhao, Junzhi Zhang, Chengkun He, Yuan Ji, Heye Huang, and Xiaohui Hou. «Autonomous vehicle extreme control for emergency collision avoidance via Reachability-Guided reinforcement learning». In: *Advanced Engineering Informatics* 62 (2024), p. 102801. ISSN: 1474-0346. DOI: <https://doi.org/10.1016/j.aei.2024.102801>. URL: <https://www.sciencedirect.com/science/article/pii/S147403462400449X>.
- [12] N. R. Kapania and J. C. Gerdes. «Design of a feedback-feedforward steering controller for accurate path tracking and stability at the limits of handling». In: *Vehicle System Dynamics* 53.12 (2015), pp. 1687–1704. DOI: 10.1080/00423114.2015.1055279. URL: <https://doi.org/10.1080/00423114.2015.1055279>.
- [13] Qiang Hua, Baoshan Peng, Xiaolin Mou, Ouwen Zhang, Tao He, Li Xia, and Heyan Li. «Model Prediction Control Path Tracking Algorithm Based on Adaptive Stanley». In: *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*. 2022, pp. 1–5. DOI: 10.1109/VTC2022-Fall157202.2022.10013074.
- [14] U.S. Federal Highway Administration, Office of Operations. *Chapter 2: Overview of Shared Mobility Services*. Accessed: 2025-10-01. 2020. URL: <https://ops.fhwa.dot.gov/publications/fhwahop20013/ch2.htm>.
- [15] Paolo Tims. «Path tracking control solutions via Enhanced Model Reference Adaptive Control algorithms augmented with Neural Networks and their experimental validation in scaled fully autonomous vehicles». Supervisors: Prof. Alessandro Vigliani, Dr. Umberto Montanaro, Prof. Aldo Sorniotti. Master’s Thesis. Turin, Italy: Politecnico di Torino, 2023.
- [16] AUMOVIO Engineering Solutions. *ARS 404-21 “entry” long range radar (far field radar sensor) 77 GHz*. Accessed: 2025-10-01. 2025. URL: <https://engineering-solutions.aumovio.com/components/ars-404/>.
- [17] Laurent Lessard. *Lecture 13: Linear Quadratic Regulator (LQR)*. Lecture notes – ME 7247, University of Wisconsin–Madison. Scribe: Milad Alipour Shahraki. Accessed: 2025-10-01. 2022. URL: <https://laurentlessard.com/teaching/me7247/lectures/lecture%2013%20-%20linear%20quadratic%20regulator.pdf>.