POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Implementation of a Non-optimization based Preview Control in case of significant actuator delays

Tesi di Laurea Magistrale in
Automation and Control Engineering - Ingegneria dell'Automazione

Author: **Alessandro Peverali**

Student ID: 995714
Advisor: Prof. Matteo Corno
Co-advisors: Prof. Olivier Sename
Academic Year: 2024-25

Abstract

The work of this thesis arises from the need to implement a lateral control for a small radio-controlled vehicle (Microcar) of 30 cm. The vehicle works inside a platform equipped with a system of 12 cameras capable of capturing the movement and recognizing with high accuracy the position of the vehicle in real time. The control system is implemented on a remote Desktop PC that integrates the position information and sends the control command to the Microcar via Wi-Fi. This whole communication architecture increases steering actuator delays, that significantly limits the fast dynamics of the Microcar and makes more difficult to follow an already winding and challenging path. For this reason, it was implemented a control that was able to combine high performance with reduced computational costs, thus avoiding worsening the system delay. In this perspective, a non-optimized architecture was used to exploit the curvature preview information of the track. In literature it has been shown that an algorithm of this type, if sufficiently supported by an accurate model, is able to guarantee performances comparable to techniques such as Model Predictive Control, which however presents major computational costs. A Smith Predictor has been developed and integrated into this control scheme, thus allowing to speed up the tracking of the yaw rate and to know explicitly how to set the preview information to best compensate for the actuation delays. A robustness analysis also provided a quantitative knowledge about the limits of the Smith Predictor's operation even in the face of possible uncertainties. Last but not least, the implementation phase was taken care of as much as possible, showing the critical issues of the practical side and the solutions adopted in implementing this architecture.

Keywords: Autonomous vehicles, path-tracking controller, non-optimization based preview control, curvature, actuator delays, Smith-Predictor, H-inf controller, Robustness analysis

Abstract in lingua italiana

Il lavoro di questa tesi nasce dall'esigenza di implementare un controllo laterale per un veicolo radiocomandato (Microcar) di dimensioni ridotte, circa 30 cm. Il veicolo funziona all'interno di una piattaforma provvista di un sistema di 12 telecamere in grado di catturare il movimento e di riconoscere con un'alta accuratezza la posizione del veicolo in tempo reale. Il sistema di controllo è implementato su un Desktop PC in remoto che integra le informazioni di posizione e invia tramite Wi-Fi il comando di controllo alla Microcar. Questa architettura di comunicazione rende il ritardo di attuazione del servosterzo consistente, limitando notevolmente la dinamica veloce di cui la Microcar dispone per poter percorrere un tracciato tortuoso e impegnativo. Per migliorare questa limitazione, è stato implementato un controllo che fosse in grado di unire alte prestazioni con ridotti costi computazionali, ed evitare così di gravare ulteriormente sul ritardo del sistema. In quest'ottica si è utilizzata un'architettura non ottimizzata che sfrutta le conoscenze in anteprima della curvatura del tracciato. In letteratura si è visto che un algoritmo di questo tipo, se supportato sufficientemente da un modello accurato, è in grado di garantire prestazioni paragonabili a tecniche come Model Predictive Control, che presenta però grossi limiti in termini di costi computazionali. Si è sviluppato e integrato a questo schema di controllo un Predittore di Smith che ha reso possibile velocizzare l'inseguimento dello yaw rate e conoscere in modo esplicito come impostare l'informazione di anteprima per compensare al meglio i ritardi di attuazione. Un'analisi di robustezza ha permesso inoltre di conoscere quantitativamente i limiti di funzionamento del Predittore di Smith anche a fronte di possibili incertezze. Infine, e non per importanza, si descrive il lato implementativo, mostrando le criticità del lato pratico e le soluzioni adottate nell'applicare questa architettura.

Parole chiave: Veicoli autonomi, controllo di inseguimento del percorso, controllo con anteprima non ottimo, curvatura, ritardi dell'attuatore, Predittore di Smith, controllo H-inf, analisi di robustezza

Acknowledgements

I would like to sincerely thank everyone who supported and assisted in the work and completion of this thesis.

I would like to express my deepest gratitude to Professor Sename for the opportunity he offered me and for the warm welcome he gave me from the moment I first joined the lab. I also thank him, along with Professor John-Jairo Martinez-Molina, for always stimulating and inspiring the research work within the team.

I am deeply grateful to my advisor, Professor Corno, for assisting me once I returned to Italy. His approach and working method were a fortunate and invaluable learning experience for me on a professional level. I would like to thank everyone on Professor Sename's automotive team for their collaboration and support in this work. I thank Professor Jonathan Dumon and all those who offered their technical assistance on the Microcar platform.

I thank both the human resources at Grenoble and of Polytechnic for assisting me with the various bureaucratic difficulties I encountered in making this internship possible.

A huge thank you to all the people and friends at Gipsa-lab who accompanied me on this wonderful French experience.

To all my friends, I am infinitely grateful for having been by my side over the years, for all the support, experiences, time, shared laughter, and smiles you have given me. Heartfelt thanks to everyone.

My deepest and greatest thanks go to my mom, my dad, my brother Giacomo and his girlfriend Chiara, and the rest of my wonderful family. Thank you for all the love and support you have always shown me. Thank you for the strength you have given me in achieving this goal.

This work has been partially supported by ROBOTEX 2.0, the French Infrastructure in Robotics under the grants ROBOTEX (EQUIPEX ANR-10-EQPX-44-01) and TIRREX (EQUIPEX+ grant ANR-21-ESRE-0015).

Contents

Introduction

Autonomous and intelligent vehicles are at the heart of the societal concerns of tomorrow's transport. Various and broad issues, that go beyond engineering itself, will be part of this field and they will directly impact on choices regarding the control architecture.
It is important to keep people, who are intimately connected to transportation, at the center of this design process. Autonomous driving field was born with the aim of reducing traffic, lowering pollution and above all improving safety. However, it is equally important to consider also how the vehicle is perceived and experienced by passengers, and by the driver in case of Advanced Driver Assistance Systems (ADAS system). ADAS assists the driver in particular needs and situations facilitating the driving and improving safety and comfort. The vast area of autonomous driving, and the particular and different needs require a specific and very challenging design.

The SAE J3016 standard distinguishes vehicles into 6 levels according to their self-driving ability. In particular, level 0 includes vehicles whose tasks are performed by human drivers, in level 1 are considered vehicles with basic driving assistance, steering and speed can be changed, but driver still has the control of the vehicle (adaptive cruise-control, ABS, ESC...). Level 2 regards partial autonomous vehicles where some situations are automated by advanced assistance systems but the driver can intervene in case of necessary corrections. Finally, from level 3 to level 5 vehicles are fully autonomous and gradually human intervention is reduced to a full-autonomous vehicle able to drive in all possible scenarios.[16][20]
In recent history self-driving cars have been tested in several situations [16]. Briefly, in 2004, 23 self-driving cars participated to the Darpa Grand-Challenge where autonomous vehicles drove in off-road conditions but none of them managed to reach the final line. In the following year, 5 of them succeeded in this goal [24][10]. In 2007, Darpa Urban Challenge was the event to test vehicles in a urban environment, where traffic and variable unpredictable events make driving more demanding. Six teams finished it proving that the challenge is possible even in a such different and complex situation. During the 2010s, several companies have invested a lot of resources in the autonomous-driving field.

Google with Google Driverless Car was one of the firsts with a team that included some members of the Darpa Grand Challenge. In 2016 Google Driverless Car became Waymo, and now they are active with a robotaxi service in some cities of the United States [27]. Tesla is one of the first commercial companies that invests the most in this technology. Currently Tesla sells vehicle of level 2 SAE, but according to recent statements a vehicle of Level 5 is still a difficult and far-away goal to achieve. The challenge is much more complicated than expected for many reasons [5][7], and it seems that automotive companies are pushing much more on ADAS systems, considering ADAS as a real final-destination for research and not just a passing point towards a full-autonomous vehicle [15]. This means that their control design has to take into account also how drivers feel the driving assistance, in a way that effectively driving aid is helpful and not counterproductive.

0.1. Optimal Preview control

The article by Brian Paden et al [16] presents an overview of autonomous car research, showing the main layers, generally with a hierarchical structure, for an autonomous car. The study of self-driving cars starts from the perception of the environment, where all the car sensors (e.g. Lidar, GPS, accelerometers...) collect and integrate data to decision-making and path-planning problems, to satisfy the upcoming needs such as choice of the road, obstacle avoidance, optimize consumptions, integrate emergency situations. The result is to generate a path, and eventually a speed profile, that the path-tracking layer has to follow. Each of these topics represents a wide scenario that require deep investigations. The present thesis will focus on an algorithm of the path-tracking layer, where it is assumed to have a generated path and a constant speed profile that the car has to follow. Aiming to develop a lateral controller for a radio controlled vehicle (Microcar). The small size of the vehicle and of the track requires a fast controller that is able to smoothly follow the reference path, with a very low computational cost, and face a significant delay due to the remotely located controller.

Over the years a lot of different control architectures, for path-tracking, have been developed, and a short example among the most well-known regards Sliding Mode, Model Predictive Control, Nonlinear Control, Potential Field Control, Reinforcement Learning and Optimal Preview Control. The best controller does not exist, each of these has its own qualities and weaknesses, and it is possible to find comparative studies in the literature [1]. The choice of the best controller is strongly influenced by the skills one would

like to achieve.

The theory agrees that using knowledge of the future path reference or disturbances, significantly improves performances. Optimal Preview control was born in the '60s , but it became active only in the '90s [2][25]. It started integrating the preview concept to the classic theory of control such as output feedback, LQ/H∞ and LMIs. Over the years the field has expanded more and more including Artificial Neural Network, fuzzy logic, sliding mode...
Prior knowledge of the reference-signal evolution allows the controller to better anticipate and promptly track its variations. For example, in [19] there is a clear example that if there is enough preview information about the future track, the control variable (steering angle) can anticipate its intervention, and the vehicle dynamic that is slower can track better the reference path minimizing the lateral error. Future knowledge of the disturbance/reference potentially allows a two degree of freedom controller (feedforward+feedback) to achieve almost all the possible improvements [2].
Model Predictive Control (MPC) is a control technique that is becoming increasingly used in recent research because it combines very good performances with a flexible architecture. Its drawback is the high computational costs that have prevented its practical use in the past. Nowadays this limit is changing thanks to more powerful computers. [28] shows that if preview information is well exploited by an optimal control problem (an LQR augmented with curvature preview of the track) performances are almost the same of MPC. It has similar tracking accuracy and smoothness with the MPC but more computationally efficient. Its computing load is equal to that of LQR control. The disadvantage is less flexibility in problem formulation.

0.2. Non-optimized Curvature Preview Control and Research Contributions

This thesis focuses on a non-optimized preview control approach. In comparison with the optimized-one previously described its performances are not formally optimized, but still remain comparable. The method presents a more flexible structure — though not as general as Model Predictive Control (MPC) — and allows for more intuitive tuning by leveraging classical control theory.

The architecture is inspired by the one proposed in [6], which also emphasizes its capability

to be easily integrated with a path planner. In that context, the path planner uses the limitations of traditional path-tracking regulators to define the vehicle's dynamic boundaries. The architecture was designed with the goal to have a smooth control scheme for high performances and robust trajectory tracking, so the vehicle can track a path generated for evasive maneuvers like obstacle avoidance, at a given speed and no time to intervene on the brakes.

The control scheme, summarized in fig. 1, is a cascaded architecture based on classical control techniques, where a faster inner loop tracks a yaw-rate reference, keeping heading error as small as possible. While an outer loop, slower, minimizes the lateral error changing the yaw-rate reference directly. This low-frequency implementation is used to prevent vehicle from drifting off the trajectory. The yaw-rate reference is computed by knowing the trajectory of the car: by the path of the trajectory, the curvature is obtained and its preview information is provided to get the yaw-rate preview reference. The preview information of the curvature works as a feed-forward disturbance helping the outer loop regulator to keep small as possible the lateral error. In [6], it is formally shown that for a sufficiently fast bandwidth of the yaw-rate controller, the closed loop transfer function of the inner-loop is approximate to a pure delay. This delay can be estimated and compensated by an anticipative-delay filter set to the yaw-rate reference (feedforward term). This can be seen retracing the closed-loop transfer function from the curvature reference to the lateral error (neglecting the lateral-error feedback):

$$G_{ke} = e^{-Ts} \cdot \frac{v_x^2}{s^2} \cdot A \approx \frac{v_x^2}{s^2}$$

where A is the anticipative filter of the preview element, and k is the curvature input. If $A \approx e^{T \cdot s}$, the lateral error is theoretically null:

$$e_{CoG} = G_{ke} \cdot k - \frac{V_x^2}{s^2} \cdot k \approx 0$$

The delay in the loop, generally introduced by the actuator, imposes limit on the regulator bandwidth. If we assume, just for a moment, to have in the open-loop only a delay, i.e. $G = e^{-T_{act}s}$, its phase would be $\phi = -T_{act}\,\omega$. The system remains stable as long as phase margin $\phi_m > 0$, in fact at least a value of $\phi_m > 45° = \frac{\pi}{4}$ is recommended to avoid oscillations. If the regulator has a pole in zero (which introduces a $-\frac{\pi}{2}$ phase shift), as is usually done to guarantee zero steady-state error, the phase margin becomes:

$$\phi_m = \phi + \pi - \frac{\pi}{2} > \frac{\pi}{4}$$

and the maximum bandwidth would be:

$$\omega_b < \frac{\pi}{4T_{act}} \tag{1}$$

where in case of a significant delay is a strict limit. For example, in the case of the Microcar, where $T_{act} \approx 0.18$ means a limit of $\omega_b < 4.32\,[rad/s]$, and it will be shown that for the trajectories used in the Platform ω_b should be around $14\,[rad/s]$. For this reason, a Smith-Predictor is implemented. By theory it is known that in case of accurate model and accurate parameter estimation, the rational regulator of the Smith-Predictor can be tuned on the rational part of the model as if there were no limits (eq. (1)) due to the delay. Furthermore, the Smith-Predictor allows the time delay to be separated from the rational part of the complementary sensitivity transfer function $F = F_{rational} \cdot e^{-T_{act}s}$, making it possible to accurately identify the delay that needs to be compensated by the preview-based strategy.
In addition to being experimentally applicable, its study provides awareness of delay management. For this reason, and since Smith-Predictor is sensitive to model and parameter uncertainties, a robust analysis is conducted to explore the limits of this approach, in the hope of inspiring future improvements and developments.

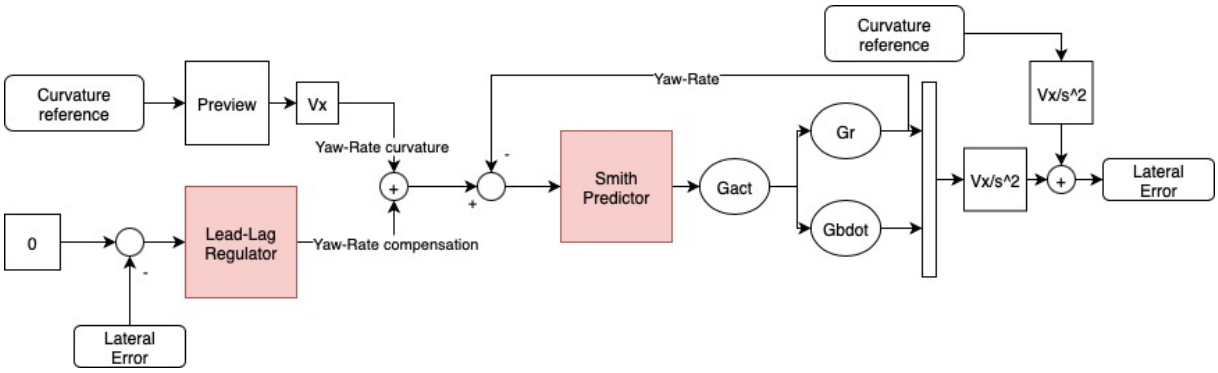


Figure 1: Standard architecture of Non-optimized Preview Control

The work of the present thesis is organized as follow: Chapter 1 analyzes the dynamics of an autonomous vehicle, focusing on the features of a radio controlled scaled vehicle (for more details on the Microcar platform, see 1.2.1). An accurate non-linear model is described and used to perform simulations prior to real-world testing on the Microcar, and to obtain a linearized version of the system. The linear model is visible in the block diagram of fig. 1 with its transfer functions $G_{act}, G_{\dot{\beta}}, G_r$ representing the actuator model, the relationship between the steering angle δ and the yaw-rate r, and the relationship between δ and the vehicle's side-slip rate $\dot{\beta}$, respectively.

This linear representation is used to design and tune the control architecture in Chapter 2, starting with the inner-loop regulator (the Smith-Predictor) followed by the outer-loop controller. For tuning purposes only, a simplified transfer function of the form V_x/s^2 is employed to reconstruct the lateral error from the sum of yaw-rate and side-slip rate. Note that $r + \dot{\beta}$ represents the angular velocity of the vehicle' speed vector considering an inertial reference frame.

Chapter 3 is divided into two parts: the first one describes the generation of the trajectories used in simulations and in Microcar tests, including how the curvatures are obtained and used to get the preview of the yaw-rate reference. It also details how the lateral error is effectively calculated, followed by the implementation of the non-linear model. The second part concerns the presentation and discussion of the results from both simulations and the actual Microcar running tests.

Chapter 4 presents a robustness analysis conducted on the system in the face of uncertainties in both time delay and tire-stiffness coefficients. The chapter opens with a theoretical summary that illustrates the steps conducted experimentally. The analysis initially focuses on the Smith predictor and the actuator delay uncertainty, modeling it as an unstructured uncertainty. It then extends to the whole system and considers all the uncertainties together through a structured representation. Finally, chapter 5 summarizes the main conclusions and briefly discusses future perspective.

1 | Vehicle Model

The following chapter focuses first on the non-linear vehicle model used for simulations, subsequently it presents an overview of the real vehicle "Microcar" used in the test Platform, which was previously developed by the Gipsa-Lab. The chapter includes a description of the hardware and the parameter identification process also developed by the Gipsa-Lab. Finally, a study of a linearized model, crucial for the control system design, is described.
A note to clarify the reading: the yaw rate, generally identified as $\dot{\psi}$, will then be called r in the linearized model. This is done for convenience, but they are exactly the same variable.

1.1. Non-linear model

Describing vehicle dynamics is challenging because it requires a very accurate model to capture all the couplings interactions among various subsystems (e.g. suspensions, tire, chassis...). In the Automotive Control Systems book [12] a very accurate model, well known in literature, is described and analyzed. Its reduced nonlinear two track-model, considered by the book essential for vehicle dynamic control, is revisited in this thesis to help the reader better understanding the SAV (Scaled Automated Vehicle) non-linear model 1.4 and its underlying assumptions.
The SAV non-linear model has been realized in Gipsa-Lab and used to perform simulations. It's carefully described and validated in Ariel Medero Borrell PhD Thesis [4].

In figure 1.1, the yaw ψ is the angle between the the x-axis of an inertial reference frame and the longitudinal axes of the vehicle, while the side-slip angle β represents the angle between the vehicle's heading and its velocity vector v_{CoG}. Both are positive in the counterclockwise direction.

$$\begin{bmatrix} \dot{x}_{in} \\ \dot{y}_{in} \end{bmatrix} = v_{CoG} \begin{bmatrix} \cos(\psi + \beta) \\ \sin(\psi + \beta) \end{bmatrix} \tag{1.1}$$

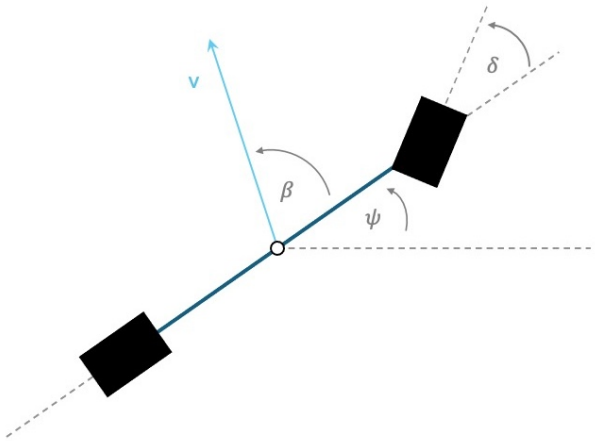


Figure 1.1: Illustration of the main vehicle angles

Switching from the inertial reference frame to the vehicle-fixed one, and assuming that the vehicle mass is concentrated in its center of gravity, we obtain the following equations:

$$\begin{bmatrix} \ddot{x}_{CoG} \\ \ddot{y}_{CoG} \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \cdot \begin{bmatrix} \ddot{x}_{In} \\ \ddot{y}_{In} \end{bmatrix} = v_{CoG} \cdot (\dot{\beta} + \dot{\psi}) \begin{bmatrix} -\sin\beta \\ \cos\beta \end{bmatrix} + \dot{v}_{CoG} \begin{bmatrix} \cos\beta \\ \sin\beta \end{bmatrix} \tag{1.2}$$

This kinematic relationship is inserted into a force balance to derive the vehicle model:

$$\begin{cases} m\ddot{x}_{CoG} = \sum F_x & \text{(1.3a)} \\ m\ddot{y}_{CoG} = \sum F_y & \text{(1.3b)} \end{cases}$$

1.1.1. SAV NL Model

Before proceeding, some assumptions are required: speed and accelerations are not sufficiently high to make gravitational and aerodynamic forces relevant. Additionally, side-slip rate is assumed to be approximately zero, i.e. $\dot{\beta} \approx 0$. According to [14], it means assuming a scenario with steady curves where there are no rapid changes in the side-slip angle β.

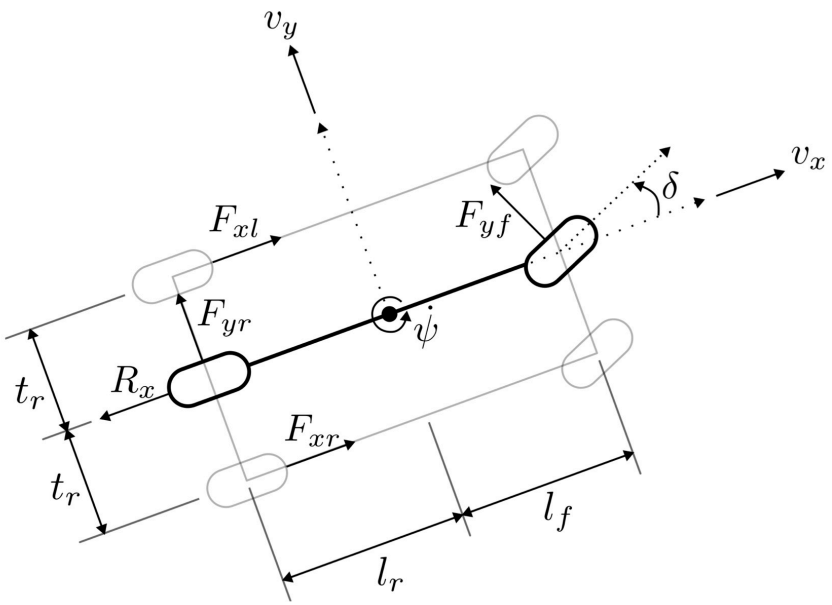


Figure 1.2: Vehicle force diagram with respect to vehicle reference frame, picture from [4]

The vehicle architecture is depicted in the figure 1.2. F_{xl} and F_{xr} denote the longitudinal forces on the left and right rear wheels, respectively. These forces take into account both the traction and braking generated by two independent motors. For this reason they can differ $F_{xl} \neq F_{xr}$, and this generates an induced moment of force considered in the equation 1.4c.
While the two longitudinal forces are considered at their actual point of application, the left and right lateral forces are identical and, for this reason, they are combined in F_{yf} and F_{yr}, and applied on the longitudinal axis of the vehicle, taking up the bicycle model [18]. Their orientation is transverse to the wheel axis, and therefore F_{yf} is rotated by an angle δ, which is the steering angle of the front wheels, around the z-axis. Finally, friction is identified and considered in R_x.
It can now be observed that this model is a hybrid between the classic bicycle model and a two-track model.

The SAV non-linear model is the following:

$$\begin{cases} m(\dot{v}_x - \dot{\psi} v_y) = F_{xl} + F_{xr} - F_{yf} \sin\delta - R_x, & \text{(1.4a)} \\ m(\dot{v}_y + \dot{\psi} v_x) = F_{yf} \cos\delta + F_{yr}, & \text{(1.4b)} \\ I_z \ddot{\psi} = l_f F_{yf} \cos\delta - l_r F_{yr} + t_r F_{xr} - t_r F_{xl}, & \text{(1.4c)} \end{cases}$$

where $v_x = v_{CoG} \cos\beta$, $\dot{v}_x = \dot{v}_{CoG} \sin\beta$,$v_y = v_{CoG} \sin\beta$ and $\dot{v}_y = \dot{v}_{CoG} \sin\beta$.

1.1.2. Tire forces

It is a non-linear vehicle model that differs from the bicycle one and the reduced non-linear two-track one not only in terms of architecture, but also in the way forces are modeled. Lateral and longitudinal forces are linear functions of tire stiffness coefficients:

$$F_{xl,r} = C_{\sigma l,r}(v_x)\sigma_{l,r} \tag{1.5}$$

$$F_{yf,r} = C_{\alpha f,r}(v_x)\alpha_{f,r} \tag{1.6}$$

with longitudinal slip-ratios:

$$\sigma_{l,r} = \frac{r\omega_{l,r} - v_{x_{l,r}}}{v_{x_{l,r}}} \quad \textit{during braking} \tag{1.7a}$$

$$\sigma_{l,r} = \frac{r\omega_{l,r} - v_{x_{l,r}}}{r\omega_{l,r}} \quad \textit{during acceleration} \tag{1.7b}$$

with tire side-slip angles, described in the single-track model of [12]:

$$\begin{aligned} \alpha_f &= \delta - \arctan\left(\frac{v_y + l_f\dot{\psi}}{v_x}\right) \\ \alpha_r &= -\arctan\left(\frac{v_y - l_r\dot{\psi}}{v_x}\right) \end{aligned} \tag{1.8}$$

with longitudinal tire stiffness coefficients:

$$\begin{aligned} C_{\sigma l}(v_x) &= C_{\sigma l2}v_x^2 + C_{\sigma l1}v_x + C_{\sigma l0} \\ C_{\sigma r}(v_x) &= C_{\sigma r2}v_x^2 + C_{\sigma r1}v_x + C_{\sigma r0} \end{aligned} \tag{1.9}$$

and cornering tire stiffness coefficients:

$$\begin{aligned} C_{\alpha f}(v_x) &= C_{\alpha f2}v_x^2 + C_{\alpha f1}v_x + C_{\alpha f0} \\ C_{\alpha r}(v_x) &= C_{\alpha r2}v_x^2 + C_{\alpha r1}v_x + C_{\alpha r0} \end{aligned} \tag{1.10}$$

Here, in eqs. (1.9) and (1.10), the non-linearity of the longitudinal (1.5) and lateral (1.6) forces with respect to the longitudinal velocity v_x is evident. This improves the accuracy of the dynamic model, especially under high-acceleration conditions where the linear relationship between force and tire slip no longer exists.

Finally, the friction force, according to [4], is represented as:

$$R_x = mg\left(\mu_0 + \mu_1 v_x^4\right) \tag{1.11}$$

1.1.3. Inputs and states

Now that vehicle dynamics have been described, attention is directed to the input $[\omega_l, \omega_r, \delta]$ and state variables $[v_x, v_y, \dot{\psi}]$.
The selection of input variables is not totally arbitrary, but it is strictly connected to the actuators and their dynamics. In this situation, the Microcar has two Maxon controllers that allow the DC motors to be controlled in speed. They use the wheel position measurements provided by the encoders to obtain the actual wheel speeds and follow the reference values with sufficiently high bandwidth. For these reasons, their dynamic is neglected.
Differently, the steering angle is not known, no measurements are available, and therefore there is no guarantee on a sufficiently fast bandwidth and especially on the precise pursuit of the reference. This leads to the inclusion of a steering actuator model to extend the vehicle model, which significantly affects both the system behavior and the control design.

1.1.4. Steering actuator model

As described in [4], the choice of the model was influenced by the one used in [11]. It is a second-order model with a pure time delay, which can be represented by the following transfer function:

$$S(s) = \frac{\delta(s)}{\delta^*(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} e^{-\tau s} \tag{1.12}$$

where the *input* δ^* is the steering angle command sent from the remote Desktop PC to the Microcar.

1.1.5. Differential model

In Chapter 2 it will be seen that the control architecture of this thesis actually uses as control variables only the steering angle δ_{ref} and the angular velocity ω_{ref} of a rear wheel, which is assumed to be on the rear part of the vehicle longitudinal axis (point C in figure 1.3). If the steering angle is zero $\delta = 0$, ω_{ref} corresponds to the real speed of the wheels, otherwise a differential model will take care of differentiating the velocities to ensure a correct kinematic relationship. It is assumed that v_{xf} and v_{yf} are the longitudinal and lateral velocities of the vehicle at the center of the front wheel, respectively, fixed with respect to the vehicle reference frame. The yaw-rate of the vehicle around the vertical

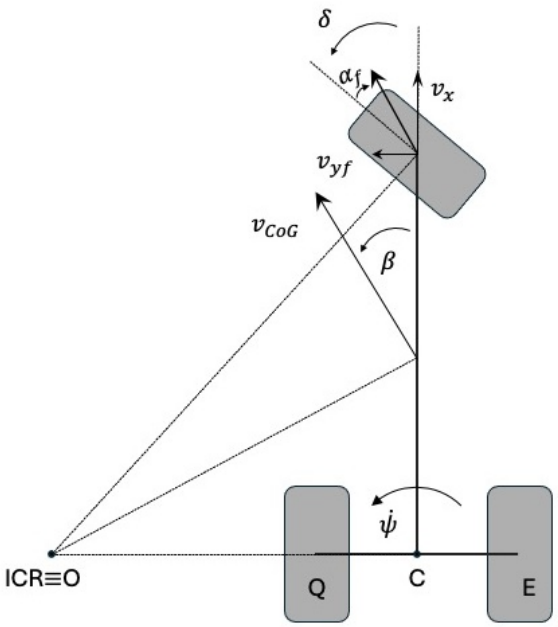


Figure 1.3: kinematic differential model of the vehicle

axes of point C is:

$$\dot{\psi} = \frac{v_{yf}}{L} \tag{1.13}$$

where L is the length of the vehicle. Knowing that v_x is the same along all the longitudinal axes of the vehicle $v_{xf} = v_x$, it is possible to rewrite the eq. (1.13) as function of the inputs $[\delta_{ref}, \omega_{ref}]$:

$$\dot{\psi} = \frac{v_{yf}}{L} = \frac{v_x \tan\delta_{ref}}{L} = \frac{R\omega_{ref}\tan\delta_{ref}}{L} \tag{1.14}$$

Point O is fixed at the instantaneous center of rotation (ICR), and D is the distance between O and point C (rear point of the longitudinal axes of the vehicle). Speed in C is computed to get the differential speed between right rear wheel (point E) and left rear wheel (point Q), using the Rivals' theorem for velocities.

$$\begin{aligned} \omega_{ref} &= \frac{v_x}{R} = \frac{v_C}{R} = \frac{\dot{\psi}D}{R} \\ v_E &= \dot{\psi}\left(D + \frac{W_i}{2}\right) \\ v_Q &= \dot{\psi}\left(D - \frac{W_i}{2}\right) \\ \Delta\omega = \omega_E - \omega_Q &= \frac{\dot{\psi}}{R}\left(D + \frac{W_i}{2} - D + \frac{W_i}{2}\right) = \omega_{ref}\tan\delta_{ref}\frac{W_i}{L} \end{aligned} \tag{1.15}$$

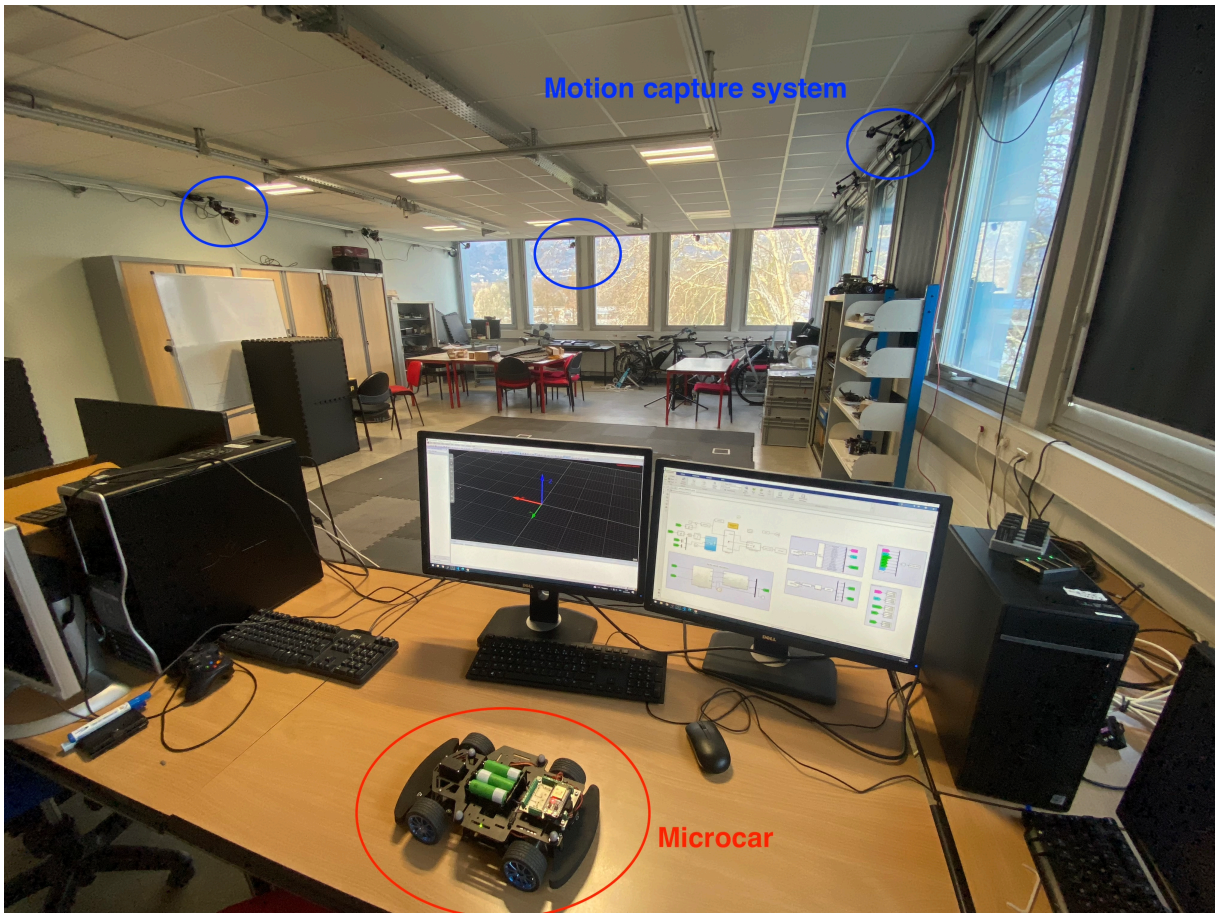


Figure 1.4: Picture of the Microcar Test Platform, by Gipsa-Lab [9]

Then ω_l and ω_r are

$$\begin{aligned} \omega_l &= \omega_C - \frac{\Delta\omega}{2} = \omega_{ref} - \frac{\Delta\omega}{2} \\ \omega_r &= \omega_C + \frac{\Delta\omega}{2} = \omega_{ref} + \frac{\Delta\omega}{2} \end{aligned} \tag{1.16}$$

1.2. Microcar

1.2.1. Microcar Test Platform

The Microcar Test Platform, in fig. 1.4, is a motion room at GIPSA-LAB [9] where a Microcar, a 1:12 scaled autonomous vehicle, runs to test designed control architectures and planning algorithms. The main components of the platform are the motion capture system, a remote desktop PC and the Microcar itself. The motion capture system is constituted by 12 Qualisys cameras with different angles and locations that sample accurately the position and orientation of the Microcar at a frequency of 100Hz. The Microcar is a redesigned 30 cm radio controlled (RC) vehicle, equipped with two Maxon Brush DC

motors for longitudinal traction and braking, one for each rear wheel. Each motor is independently controlled in either torque or speed mode via its own Maxon motor controller. Two Maxon encoders are used to measure the angular position and derive the speed of the rear wheels, and a servomotor is employed as a steering actuator for the front wheels. Finally, a remote PC acquires data from the Qualisys cameras, processes them within a Simulink control scheme and communicates with the vehicle ESP32 board via Wi-Fi. More detailed information on all hardware components is provided in the following table:

Microcar hardware
1x custom board : ESP32, ADC (LTC2309 with 8-Channel, 12-Bit SAR ADC with I2C Interface), DAC (AD7398 with 4-channel, 12-Bit with SPI Interface), 11 GPIOs, SDcard reader, 6.5-36V power supply, WIFI/Bluetooth connectivity
2x DC motor Maxon DCX22L 48V (torque constant: 0.0452Nm/A)
2x DC motor controller Maxon ESCON 36/2 DC (current limit: 1A)
2x encoder Maxon ENX 16 EASY with 1024 impulsions/rev
1x servo motor for steering
1x battery holder for 3x Li-ion cells in series (12.6V)
2x temperature sensors TI LM35DZ

Table 1.1: Microcar hardware components.

1.2.2. Parameter Identification

In this section, a brief description of how parameters have been identified by Gipsa-Lab is provided (for more details see [4]). The identification method solves the following non-linear optimization problem:

$$\min_{\Omega} \sum_{i=1}^{N_y} Q_i \cdot \text{rms}\left(y_i - \hat{y}_i(\hat{x}, u)\right), \quad Q_i \in \mathbb{R}^+ \tag{1.17}$$

where $\Omega \in \mathbb{R}^{16}$ is the set of parameters that have been identified, while Q_i is the weighting matrix that weights with different values the output errors. In particular, this parameter

identification process prioritizes a more accurate estimation of v_y over v_x and ψ, with the output vector $y = [v_x, v_y, \psi]^T$.

Parameters	Value
$C_{\sigma 2}$	1.5993
$C_{\sigma 1}$	1.4247
$C_{\sigma 0}$	0.6515
$C_{\alpha f 2}$	-0.4363
$C_{\alpha f 1}$	6.2295
$C_{\alpha f 0}$	-1.9787
$C_{\alpha r 2}$	3.0642
$C_{\alpha r 1}$	8.5829
$C_{\alpha r 0}$	-2.9295
ζ	1.7206
ω_n	48.8878
τ	0.1818
I_z	0.0060
R	0.0324
μ_0	$1.2643 \cdot 10^-5$
μ_1	0.0040

Table 1.2: List of identified parameters used in the model, by Gipsa-Lab [9].

1.3. Linearized model

As anticipated in the introduction chapter the non-optimized curvature preview control, the control architecture which is the focus of this thesis, is based on a linear model. Moreover the linearized model is only used to build and set the control scheme, and not to perform simulations nor validate its architecture. As shown also in [6][12], it is assumed that:

1. side-slip angle is small: $\sin\beta \approx 0$, $\cos\beta \approx 1$;
2. vehicle speed is constant ($\dot{V} = 0$);
3. linear tire characteristics;
4. a path radius larger than the wheelbase;

5. steering angle is small: $\cos\delta \approx 1$.

From the first assumption, it can be deduced that $v_x \approx V$ and $v_y \approx 0$. So the system 1.2 is reduced to:

$$\begin{cases} m\left[-v_y\left(\dot{\beta}+\dot{\psi}\right)+\dot{v}_x\right] = \sum F_x \quad \Rightarrow \quad \dot{V} = 0 = \sum F_x, & \text{(1.18a)} \\ m\left[v_x\left(\dot{\beta}+\dot{\psi}\right)+\dot{v}_y\right] = \sum F_y \quad \Rightarrow \quad mV\left(\dot{\beta}+r\right) = \sum F_y, & \text{(1.18b)} \end{cases}$$

where $r = \dot{\psi}$ is the yaw-rate.

1.3.1. Lateral model

As a result, the eqs. (1.4b) and (1.4c) of the non-linear model now become:

$$\begin{cases} mV\left(\dot{\beta}+r\right) = F_f + F_r, & \text{(1.19a)} \\ I_z\dot{r} = l_f F_f - l_r F_r, & \text{(1.19b)} \end{cases}$$

The forces are linearized around the equilibrium value of $v_x = V$, according to the assumption 2:

$$\begin{aligned} F_f &= C_f\alpha_f \\ C_f &= C_{\alpha f2}V^2 + C_{\alpha f1}V + C_{\alpha f0} \\ \alpha_f &= \delta - \beta - \frac{rl_f}{V} \end{aligned} \tag{1.20}$$

$$\begin{aligned} F_r &= C_r\alpha_r \\ C_r &= C_{\alpha r2}V^2 + C_{\alpha r1}V + C_{\alpha r0} \\ \alpha_r &= -\beta + \frac{rl_r}{V} \end{aligned} \tag{1.21}$$

where the tire side-slip angles have always been obtained in a kinematic way, as described in the single-track model of [12].
Regarding the impact of longitudinal forces on eqs. (1.4b) and (1.4c), since the speed is constant and the steering angle is assumed to be small, the moment induced by the speed difference of the rear wheels can be neglected:

$$t_r F_{xr} - t_r F_{xl} = F_{xr} - F_{xl} \approx 0 \tag{1.22}$$

Finally, the linearized model is the following:

$$\begin{cases} \dot{\beta} = -\dfrac{C_f + C_r}{MV}\,\beta + \left(\dfrac{C_r l_r - C_f l_f}{MV^2} - 1\right) r + \dfrac{C_f}{MV}\,\delta & \text{(1.23a)} \\ \dot{r} = \dfrac{C_r l_r - C_f l_f}{J_z}\,\beta - \dfrac{C_f l_f^2 + C_r l_r^2}{J_z V}\,r + \dfrac{C_f l_f}{J_z}\,\delta & \text{(1.23b)} \end{cases}$$

The lateral dynamic of the vehicle is described by the states: $[\beta, r]$ the side-slip angle and the yaw rate, respectively. The steering angle δ, output of the servomotor actuator model, instead is the input. The output, not made explicit by the previous equations, is the lateral error of the vehicle. In order to calculate the lateral error, it is necessary to identify the actual rotation speed of the vehicle. This term is not the yaw rate, which identifies the rotation of the vehicle body, but the sum of the two states: $r + \dot{\beta}$. Whenever a vehicle turns, it tends to slip laterally and, in this case, the actual speed vector of the vehicle will no longer be aligned to the vehicle chassis by a β angle. Therefore, it is possible to reconstruct the actual trajectory of the vehicle starting from the rotation of the real velocity vector.

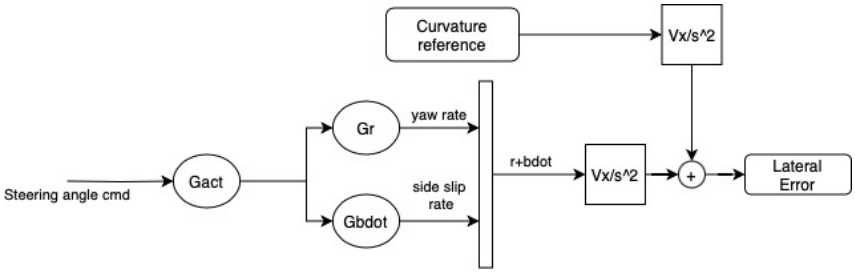


Figure 1.5: Scheme of the linearized model

Proposition 1.1. *According to [17, 26], the curvature k measures how fast the unit tangent vector to the curve at a point P rotates, when point P moves along the curve (at unit speed):*

$$k = \frac{d\theta}{ds}, \tag{1.24}$$

where $d\theta$ represents the infinitesimal angular displacement of the tangent vector, and ds is the infinitesimal path length traveled by point P along the curvilinear abscissa.

Then, integrating $\left(r + \dot{\beta}\right) \cdot V$ you get the angle $\theta = \beta + \psi$ of the real speed vector. As a matter of fact:

$$k \cdot V = \frac{d\theta}{ds} \cdot \frac{ds}{dt} = \frac{d\theta}{dt} \quad \Rightarrow \quad \int_0^t k(\tau) V d\tau = \theta(t) \tag{1.25}$$

Now, it is possible to compute the lateral error integrating the difference between the

speed-vector angle of the car and the one obtained from the reference trajectory:

$$\int_0^t (\theta(\tau) - \theta_{ref}(\tau))\, d\tau = e_{CoG}(t) \tag{1.26}$$

This idea, used in [6], works in theory. The vehicle must actually travel at a constant speed of V, and every variation causes a loss of synchronization with respect to the reference curvature. Therefore, the lateral error estimation would not be reliable if performed in simulation. For this reason, this idea is used only to tune the control scheme properly.

Equations 1.4 show that vehicle dynamics strongly depends on its speed value, specifically on its components v_x and v_y. This speed dependence is also present in equations 1.23, where it is clear that the linearized model can change according to the choice of the equilibrium value V. The numerical analysis of the model, and therefore of its transfer functions, is performed around a speed value of $V = 1.2$ $[m/s]$, which is considered a fairly high value for the Microcar. Subsequently, fig. 1.7 and tables 1.6 and 1.7 provide a brief illustration of how the model would change if a different choice of V were made.

The steering actuator transfer function is the transfer function from the commanded steering angle δ_{cmd}, output of the yaw-rate controller on the PC, to the actual one δ of the vehicle:

$$G_{\text{act}}(s) = \frac{2390}{(s + 15.67)(s + 152.6)} \cdot e^{-0.1818s} \tag{1.27}$$

Poles	Zeros	Static gain
-15.67, -152.6	none	1

Table 1.3: Poles, zeros and gain of G_{act}.

The current paragraph details an analysis of this transfer function. First of all, it is possible to notice that the system is of the second order with a pure time delay, it is asymptotically stable, and it exhibits no oscillations. The presence of two distinct poles suggests that the actuator comprises two sub-dynamics: a very fast one (associated with the pole at -150), which supposedly identifies the electric dynamics of the motor, and a slower one (pole at -15), probably due to the mechanical response of the servo.

The identified time delay of 0.1818 $[s]$ is significant and must be considered during the control design. It is likely influenced by the time required for the steering command to be processed by the remote Simulink/ROS PC, transmitted via Wi-Fi to the Microcar and finally sent to the steering servomotor.

The yaw-rate transfer function from the steering angle δ to the yaw rate r of the vehicle:

$$G_r = 56.068 \cdot \frac{s + 20.72}{s^2 + 32.86s + 318.4} \tag{1.28}$$

Poles	Zeros	Static gain
$-16.43 \pm 6.95\,\mathrm{i}$	-20.72	3.65

Table 1.4: Poles, zeros and gain of G_r.

The side-slip rate transfer function from the steering angle δ to the side-slip rate $\dot{\beta}$:

$$G_{\dot{\beta}} = 3.3996 \cdot \frac{s \cdot (s + 13.38)}{s^2 + 32.86s + 318.4} \tag{1.29}$$

Poles	Zeros	Static gain
$-16.43 \pm 6.95\,\mathrm{i}$	$0, -13.38$	0

Table 1.5: Poles, zeros and gain of $G_{\dot{\beta}}$.

The bode diagram in fig. 1.6 shows a direct comparison among the three transfer functions $G_{act}, G_r, G_{\dot{\beta}}$. It is noted that the bandwidth of the steering actuator with its slowest pole limits the vehicle dynamic. This issue further complicates the control design.

As mentioned, the final discussion aims to show how this linear model would change if the equilibrium velocity were different. Figure 1.7a shows how, as the speed increases, the yaw-rate response to steering commands increases in magnitude and becomes more reactive. Table 1.6 reports the numerical values for different speeds, highlighting the increase in gain and the shift of G_r poles to more negative values, which moves the G_r phase to the right in the Bode diagram. While in figure 1.7b it is illustrated a different behavior for the $G_{\dot{\beta}}$ response, at low frequencies the gain decreases if the Microcar becomes faster. This suggests that for slow steering corrections, such as gradual turns, the faster the vehicle the more filtered the response becomes, meaning the velocity vector angle tends to remain constant and less sensitive to steering changes. For fast steering changes, such as sudden turns, the opposite behavior occurs, with the response magnitude increasing. The phase of $G_{\dot{\beta}}$ starts from 90° at low frequencies, increases at medium frequencies, and goes to zero at high frequencies. Table 1.7 reports gain and pole values for different speeds.

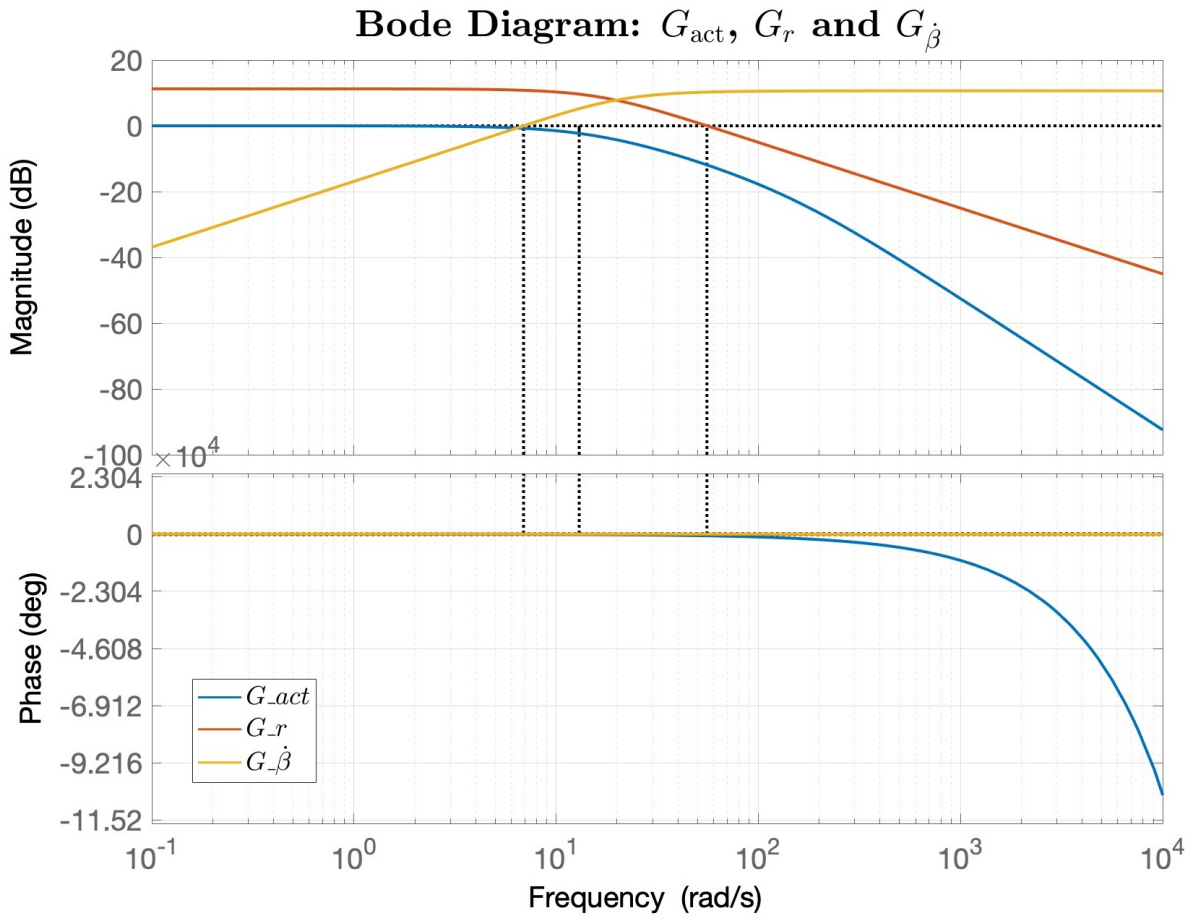
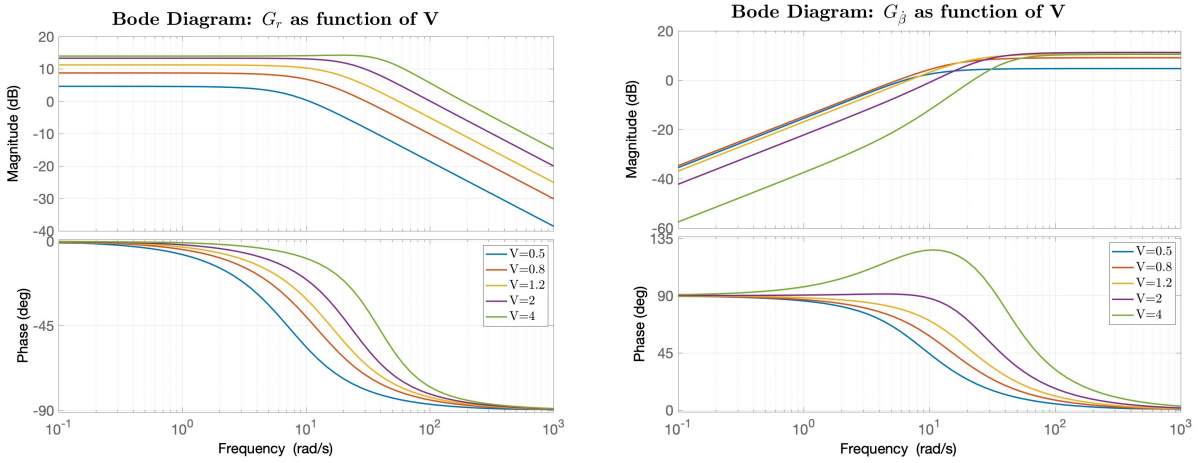


Figure 1.6: Bode diagram of the model transfer functions



(a) $G_r(V)$ (b) $G_{\dot{\beta}}(V)$

Figure 1.7: Bode diagram of G_r and $G_{\dot{\beta}}$ if V changes

Poles	Zeros	Static gain	V
$-7.36 \pm 2.85\,\mathrm{i}$	-8.98	1.70	0.5
$-12.63 \pm 4.28\,\mathrm{i}$	-15.56	2.74	0.8
$-16.43 \pm 6.95\,\mathrm{i}$	-20.72	3.65	1.2
$-21.26 \pm 12.45\,\mathrm{i}$	-27.95	4.63	2
$-30.12 \pm 25.58\,\mathrm{i}$	-42.43	4.99	4

Table 1.6: Poles, zeros and gain of G_r according to different V.

Poles	Zeros	Static gain	V
$-7.36 \pm 2.85\,\mathrm{i}$	0, -6.08	0	0.5
$-12.63 \pm 4.28\,\mathrm{i}$	0, -11.43	0	0.8
$-16.43 \pm 6.95\,\mathrm{i}$	0, -13.38	0	1.2
$-21.26 \pm 12.45\,\mathrm{i}$	0, -12.81	0	2
$-30.12 \pm 25.58\,\mathrm{i}$	0, -6.19	0	4

Table 1.7: Poles, zeros and gain of $G_{\dot{\beta}}$ according to different V.

1.3.2. Longitudinal model

The longitudinal model takes up the equation 1.18a and describes the longitudinal dynamic of the vehicle around the equilibrium value of $v_x = V$. In the control architecture, this model allows to properly tune a model-based regulator ensuring that the vehicle speed actually remains constant.

Considering the eqs. (1.4a) and (1.18a):

$$m\left[\dot{v}_x - \left(\dot{\psi} + \dot{\beta}\right) v_y\right] = F_{xl} + F_{xr} - F_{yf}\sin\delta - R_x \tag{1.30}$$

and recalling the assumptions considered for the linearized model, it can be deduced that:

1. $R_x \approx 0$, due to the assumption 2, $R_x = mg\mu_0$ is negligible due to the very small value of μ_0;
2. $\delta \approx 0$, due to the assumptions 5 and 4;
3. $v_y \approx 0$, as previously analyzed.

According to the previous considerations and if $F_{xl} = F_{xr} = C_\sigma \cdot \frac{R\omega - v_x}{v_x}$, the model can be

reduced to the following:

$$\begin{aligned} m\dot{v}_x = F_{xl} + F_{xr} \quad &\Rightarrow \quad m\dot{V} = 2C_\sigma \cdot \left(\frac{R\omega}{V} - 1\right) \\ \Rightarrow \quad \dot{x} &= -\frac{2C_\sigma}{mV}x + \frac{2C_\sigma R}{mV}u, \end{aligned} \tag{1.31}$$

where the state $x = V$, and the input $u = \omega$

2 | Control Architecture

In the present chapter the control scheme architecture is resumed to highlight the role of the two controllers and their tuning, using both a classic approach and an $\mathcal{H}_\infty$ strategy. Then, it follows with a description of the $\mathcal{H}_\infty$ regulator for the longitudinal control. The chapter ends with a robustness analysis focused on the Smith predictor modeling both a delay uncertainty and an uncertainty on the tire stiffness parameters.

2.1. Smith predictor with anti-windup

The yaw-rate controller has a main role in the control goal, it is located in the innermost loop and its design should be fast enough to track the reference yaw rate. The reference yaw rate is the sum of:

1. yaw rate of the track: $r_{\text{curvature}} = k_{\text{preview}} \cdot v_x$;
2. yaw-rate compensation: it is the output of the outer-loop regulator.

As described in the thesis introduction, it is important that, in the considered bandwidth, the closed-loop transfer function behaves approximately as a pure delay. More precisely, a unity gain over the entire bandwidth ensures an accurate yaw-rate tracking and, additionally, the more the phase of the complementary sensitivity transfer function resembles the one of a pure delay, the more this delay can be compensated by an anticipatory term.

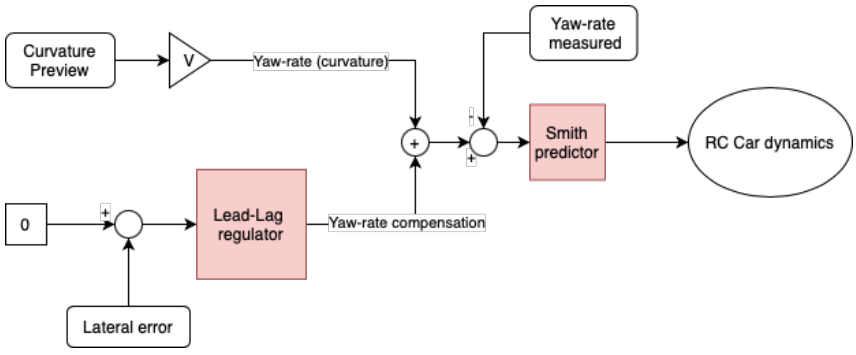


Figure 2.1: Controller architecture diagram

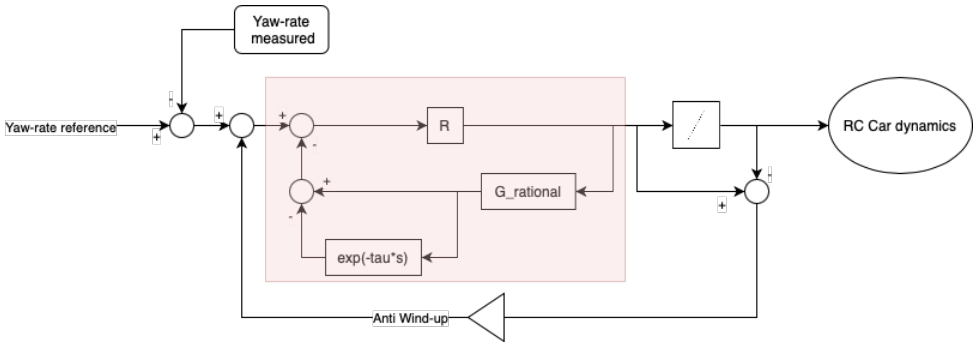


Figure 2.2: The figure shows the structure of the Smith predictor with anti-windup

As already pointed out, this architecture is based on a linear model. The inner loop transfer function is defined as $G = G_{\text{actuator}} \cdot G_{\text{yaw rate}}$:

$$G(s) = G_{\text{act}}(s) \cdot G_r(s) = \frac{1.34e05\,(s + 20.72)}{(s + 15.67)\,(s + 152.6)\,(s^2 + 32.86s + 318.4)} \cdot e^{-0.1818s} \tag{2.1}$$

The use of a Smith predictor is motivated by:

- the non-negligible time delay, $\tau = 0.1818\,[\text{s}]$;
- the tracks employed that require a fast controller. In figure 3.2b it is shown a frequency analysis of the track curvature, its power spectrum contains important information just under the frequency of $2\,[\text{Hz}] \approx 12.5\,[\text{rad/s}]$. A bandwidth of $\omega_b \approx 14\,[\text{Hz}]$ has been chosen.

The scheme of the Smith predictor used is shown in fig. 2.2, where R is the rational controller set on the rational transfer function of the model G_{rational}, that is, G without the delay term. The model in feedback loop constituted by $P(s) = G_{\text{rational}} \cdot (1 - e^{-\tau s})$ identifies the part with the predictor role. This is described in the book [3], where it is shown that the complementary sensitivity transfer function becomes: $F = \frac{RG}{1+RG_{rational}} = F_{rational} \cdot e^{-\tau s}$. Furthermore, an anti-windup block is added to prevent the windup phenomenon caused by the integral action of R in the case of steering angle saturation.

Two different strategies have been adopted to design the rational regulator R: the first one follows a more classical approach, while the second one is based on an $\mathcal{H}_\infty$ strategy.

2.1.1. A more classical approach

The first aspect considered is the approximation of the G_{rational} transfer function with a simpler second-order one, making the tuning of the regulator easier. In the Bode diagram of fig. 2.3, it can be observed that this approximation is sufficiently accurate within the

frequency bandwidth of interest.

$$G_{\text{reduced}} = 3.6163 \cdot \frac{\frac{-1}{229}s + 1}{\frac{1}{226.2}s^2 + \frac{26.63}{226.2}s + 1} \tag{2.2}$$

Poles	Zeros	Static gain
$-13.31 \pm 7\,\mathrm{i}$	229	3.61

Table 2.1: Poles, zeros and gain of $G_{reduced}$.

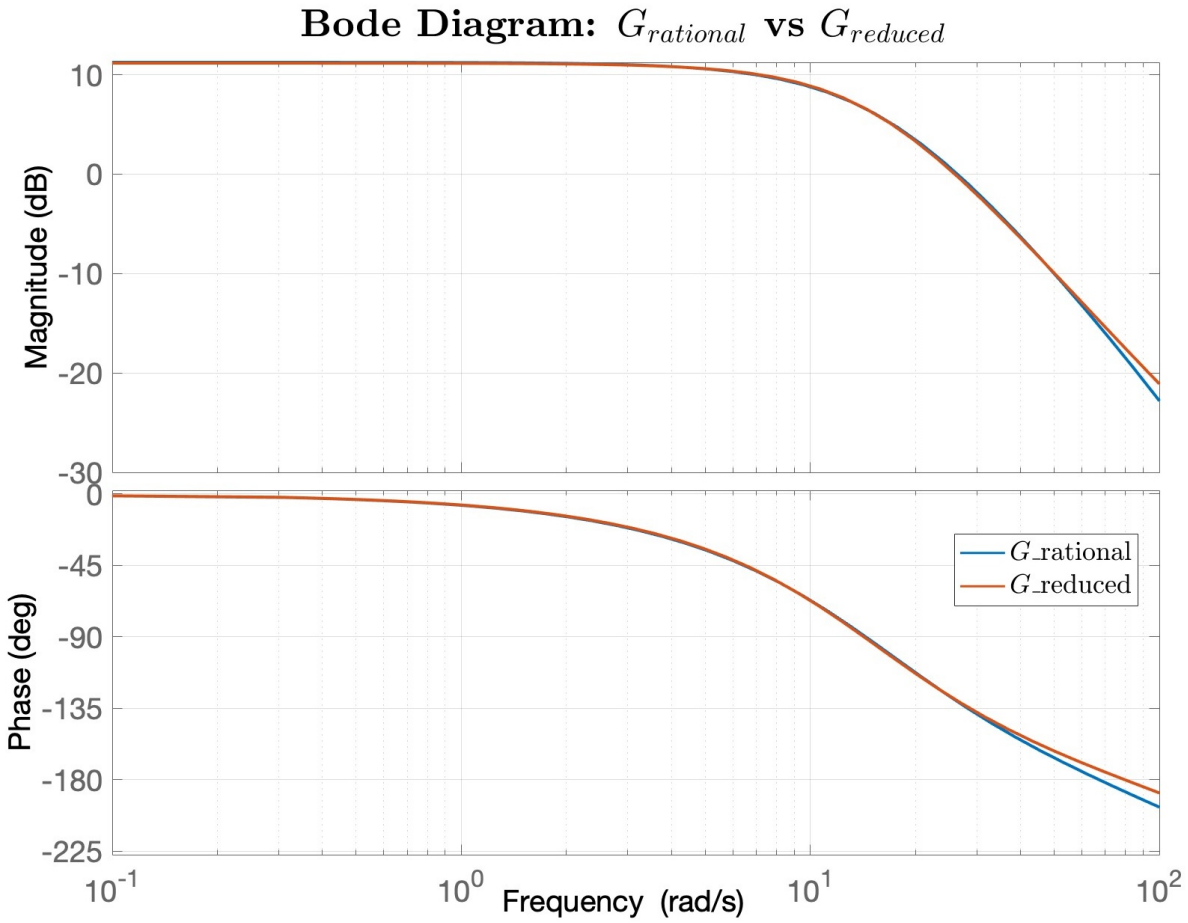


Figure 2.3: Bode diagram of the rational yaw-rate transfer function vs the reduced one G_{reduced}

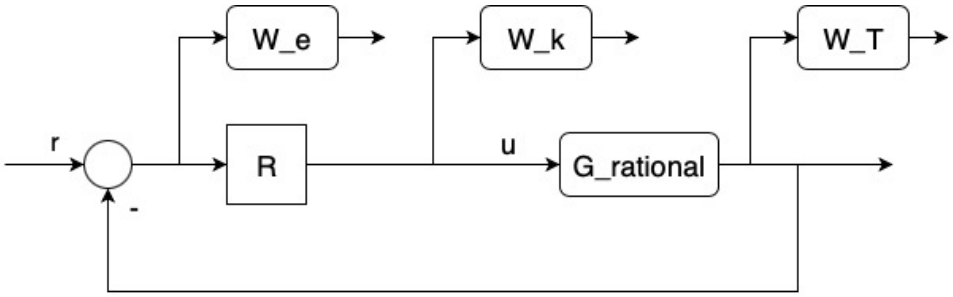


Figure 2.4: Block diagram defining the weighting functions to tune the inner-loop rational regulator $R_{\mathcal{H}_\infty}$

R is tuned on G_{reduced}:

$$R(s) = \frac{14}{3.6163} \cdot \frac{\frac{1}{226.2}s^2 + \frac{26.63}{226.2}s + 1}{s\left(\left(\frac{1}{70}s + 1\right)^2\right)} \tag{2.3}$$

Two zeros are set in $\omega = -13.31 \pm 7i$ to cancel the poles of G_{reduced}, an integrator is used to have zero steady-state error and the gain is selected to have the cutting frequency at 0 dB in $\omega_c = 14$. Finally, two poles are chosen at a frequency slightly higher in such a way to not impact on the performance and to filter out high frequency noise. Otherwise, the small discontinuities and disturbances in the yaw-rate reference or in the yaw-rate feedback would cause high oscillations in the steering angle.

2.1.2. $\mathcal{H}_\infty$ synthesis strategy

This method allows a controller design that satisfies predefined performance constraints. It is an optimized approach by enforcing bounds on the sensitivity transfer function, on the complementary sensitivity transfer function and on the control effort.

Figure 2.4 illustrates where the weighting functions are located within the block diagram. W_e weights the transfer function between the reference r and the input of the rational regulator. W_k is defined on the control variable, i.e., on the input of G_{rational}. Whereas W_T is located on the system output.

The error weighting function W_e is defined as follow:

$$W_e(s) = \frac{\frac{1}{M_e}s + 2\pi f_e}{s + 2\pi f_e A_e} \tag{2.4}$$

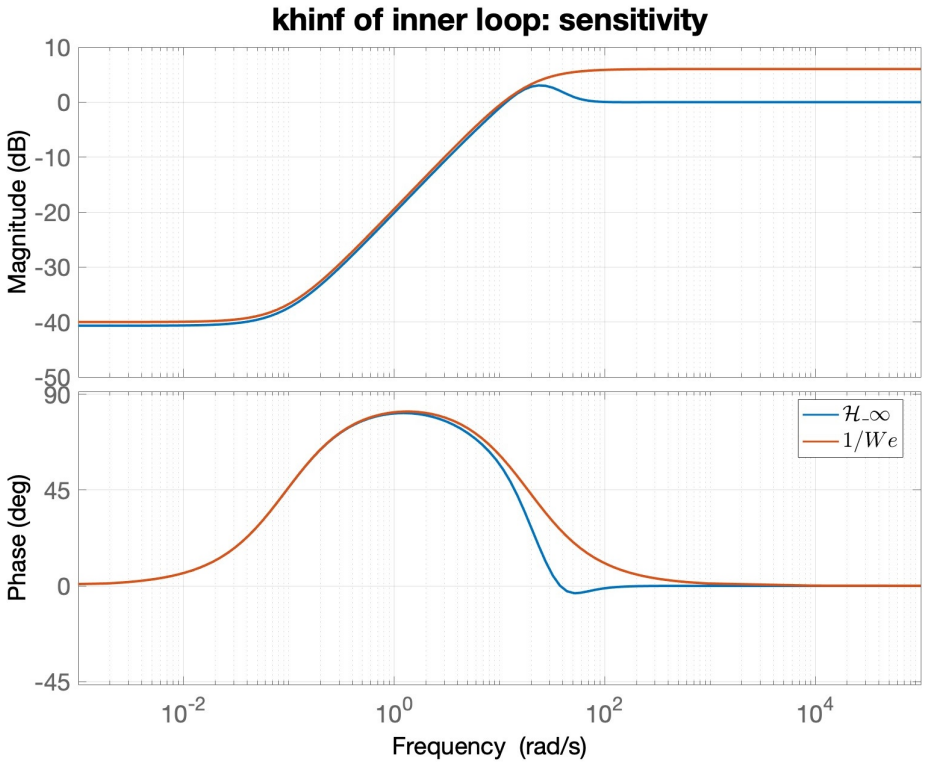


Figure 2.5: bode diagram of sensitivity transfer function vs $1/W_e$

Parameters	Value
f_e	1.5 [Hz]
A_e	0.01
M_e	2

Table 2.2: Parameter values of the error-weighting function used in the $\mathcal{H}_\infty$ yaw-rate controller

W_e weights the sensitivity transfer function $S_r(s) = \frac{1}{1+R_{\mathcal{H}_\infty} G_{\text{rational}}}$ implying $S_r(j\omega) < -40\,\text{dB}$ at low frequencies, and $\omega_c > 10.9\,[\text{rad/s}]$.

The control effort is weighted by W_k:

$$W_k(s) = \frac{s + \frac{2\pi f_k}{M_k}}{A_k s + 2\pi f_k} \tag{2.5}$$

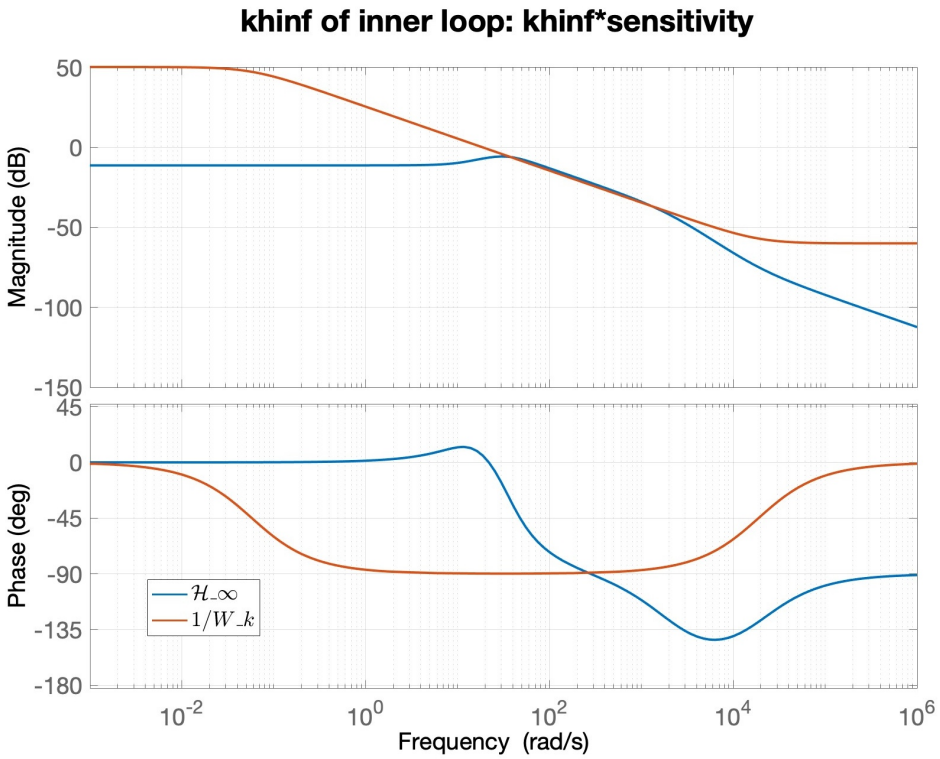


Figure 2.6: bode diagram of $R_{\mathcal{H}_\infty} \cdot S(s)$ vs $1/W_k$

Parameters	Value
f_k	3 [Hz]
A_k	10^{-3}
M_k	326

Table 2.3: Parameter values of the control-effort weighting function used in the $\mathcal{H}_\infty$ yaw-rate controller

W_k penalizes the control action at high frequency, limiting $K(s) = R_{\text{Smith}} \cdot S(s)$.

In the end, W_T is referred to the complementary sensitivity transfer function $T = \frac{R_{\mathcal{H}_\infty} G_{rational}}{1 + R_{\mathcal{H}_\infty} G_{rational}}$ and it is chosen as follows:

$$W_T(s) = \frac{s + \frac{2\pi f_T}{M_T}}{A_T s + 2\pi f_T} \tag{2.6}$$

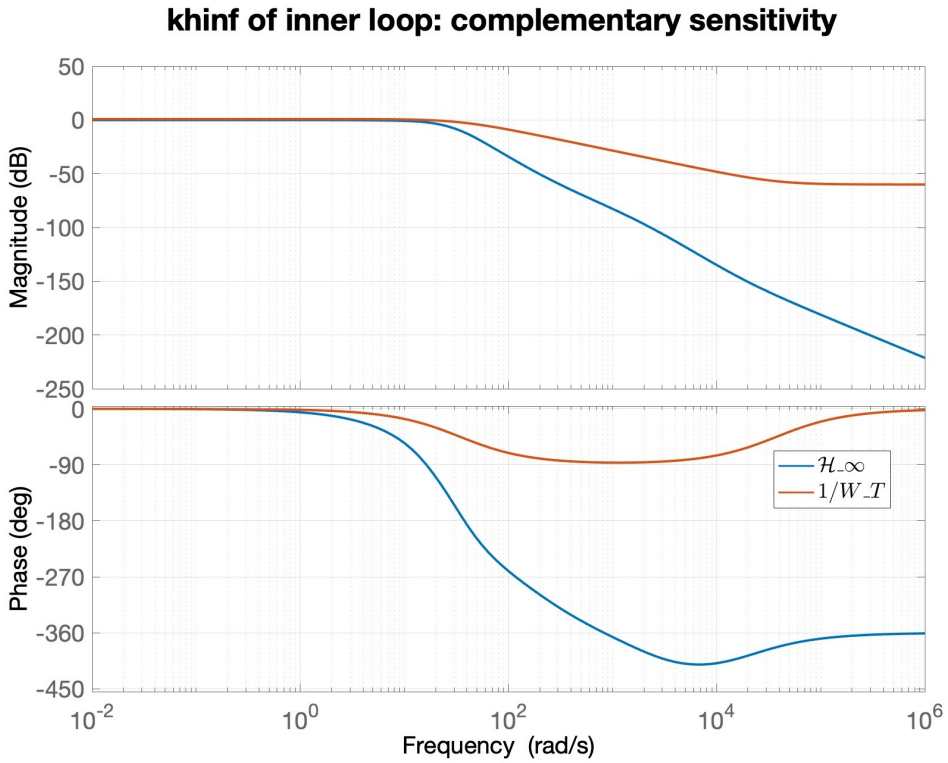


Figure 2.7: bode diagram of complementary sensitivity transfer function vs $1/W_T$

Parameters	Value
f_T	6 [Hz]
A_T	10^{-3}
M_T	1.1

Table 2.4: Parameter values of the complementary sensitivity weighting function used in the $\mathcal{H}_\infty$ yaw-rate controller

W_T limits the bandwidth to an upper value of $\omega_c < 15.7\,[\text{rad/s}]$.

The $\mathcal{H}_\infty$ control strategy minimizes $\|W_e(s)S(s)\|_\infty$, $\|W_k(s)R_{\mathcal{H}_\infty}(s)S(s)\|_\infty$, and $\|W_T(s)T(s)\|_\infty$, aiming to keep them smaller than 1, according to the $\mathcal{H}_\infty$ control theory, see book [13]. This behavior can be observed by the three Bode diagrams shown from figs. 2.5 to 2.7 where the three closed-loop transfer functions are always smaller than their corresponding weighting function, except for a small point in the control effort one, fig. 2.6, that can be considered absolutely negligible and non-impactful.

The Bode diagram in fig. 2.8 shows the results of the two approaches adopted. The regu-

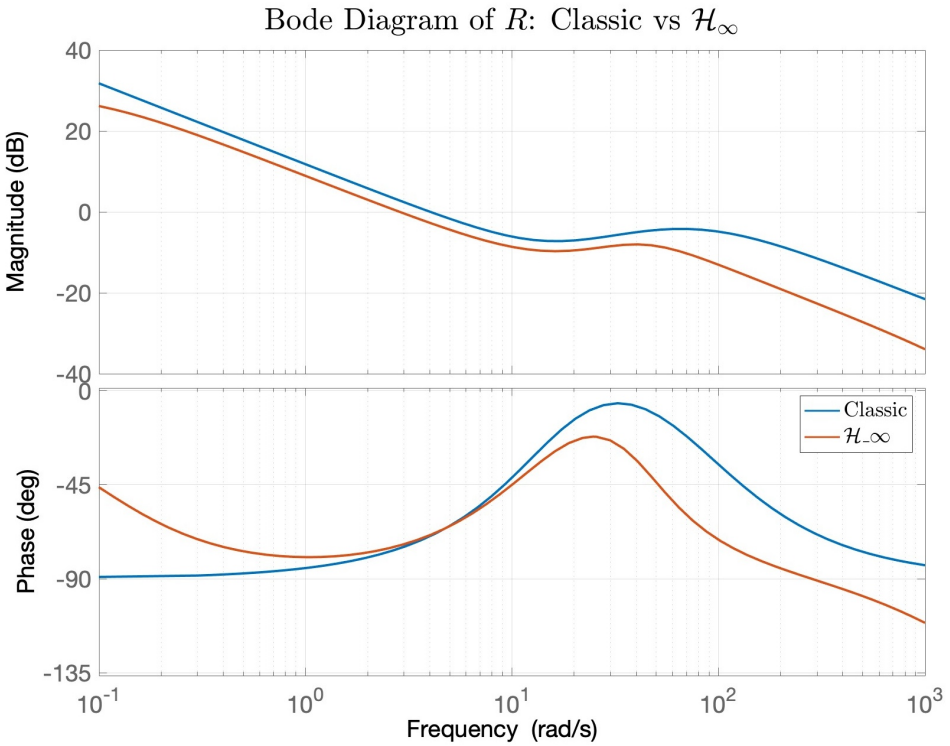


Figure 2.8: Bode diagram comparison for the yaw-rate regulator: more classical approach vs $\mathcal{H}_\infty$ one

lators are practically identical, indicating that despite the classical approach is designed according to an approximate model, its performances are comparable to that of an optimal controller tuned on the original reference model. This combined study gave awareness of both the limitations in the tuning phase of the yaw-rate controller and of its closed-loop performances. In this regard, it can be observed that the two zeros ($\omega = -13.31 \pm 7i$) used to widen the bandwidth of ω_c, also widen the bandwidth of the regulator which, as already described, can lead to oscillation phenomena on the control variable.

In fig. 2.9 is shown that effectively Smith predictor makes the yaw-rate transfer function similar to a delay of $\tau = 0.1818$. It illustrates a comparison between the complementary sensitivity transfer functions of the two strategies, with $F_r = \frac{R_{\text{smith}}G}{1+R_{\text{smith}}G}$. It can be observed that there is a small difference in phase with a pure delay, this is motivated by the fact that $F_{rational}$, visible in fig. 2.7, has a phase variation before the cutting frequency ω_c. Therefore, $F_{rational}$ cannot be perfectly approximated by 1 throughout the bandwidth, and this falls within the limits of the vehicle model and the steering actuator.

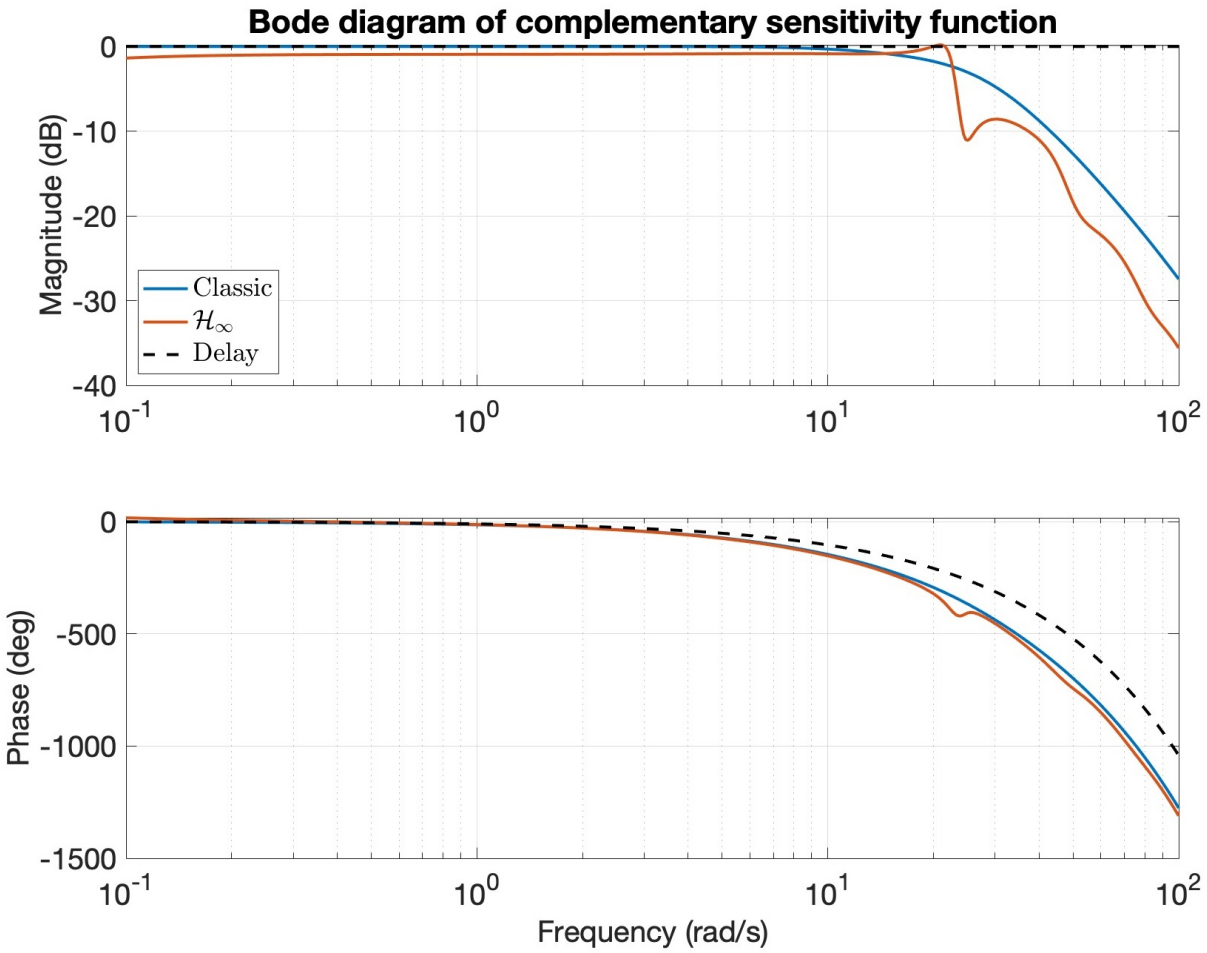


Figure 2.9: Bode diagram of yaw-rate complementary sensitivity transfer function using Smith predictor and the model G with delay

2.2. Lead-lag regulator

The lateral error regulator is tuned according to the transfer function from the steering angle to the lateral error:

$$G_{re} = \left(\frac{R_{\text{smith}}\, G}{1 + R_{\text{smith}}\, G_{rational}} + R_{\text{smith}}\, G_{act}\, G_{\dot{\beta}} \right) \cdot \frac{V}{s^2} \tag{2.7}$$

Its objective is to minimize the lateral error by intervening in the reference yaw-rate and compensating the yaw-rate obtained from the curvature. The trajectory of a vehicle controlled only by yaw-rate has a tendency to drift from the reference trajectory, and this slower regulator keeps the vehicle in its reference position. As it can be observed in the eq. (2.7), there is a double integrator which causes instability. For this reason, a lead-lag compensator is implemented in order to improve phase margin and guarantee stability of the loop. Also in this case two approaches have been used:

2.2.1. Classical approach

Theoretically, a lead-lag compensator is a transfer function:

$$R_e = k \cdot \frac{(s + z_1)(s + z_2)}{(s + p_1)(s + p_2)} \tag{2.8}$$

where $|p_2| < |z_2| < |z_1| < |p_1|$. It consists of a lead term where the zero intervenes before the pole ($|z_1| < |p_1|$), increasing the phase margin, and of a lag term ($|p_2| < |z_2|$) that increases the gain at lower frequencies and reduces the steady-state error. The regulator used is the following one:

$$R_e(s) = 0.75 \cdot \frac{(10s + 1)(30s + 1)}{\left(\frac{1}{15}s + 1\right)\left(\frac{1}{20}s + 1\right)(100s + 1)} \tag{2.9}$$

where tuning is performed empirically, in order that L_e (with $L_e = R_e G_{re}$):

1. $\omega_c \approx 3\,[\text{rad/s}]$;
2. a further pole at $\omega = 20\,[\text{rad/s}]$, such that R_e has a filtering action at higher frequency;
3. increases its phase around ω_c, ($\phi_m > 60°$);
4. reduces steady-state error as much as possible.

Changing a little the gain of the regulator, it changes its behavior: the current gain makes

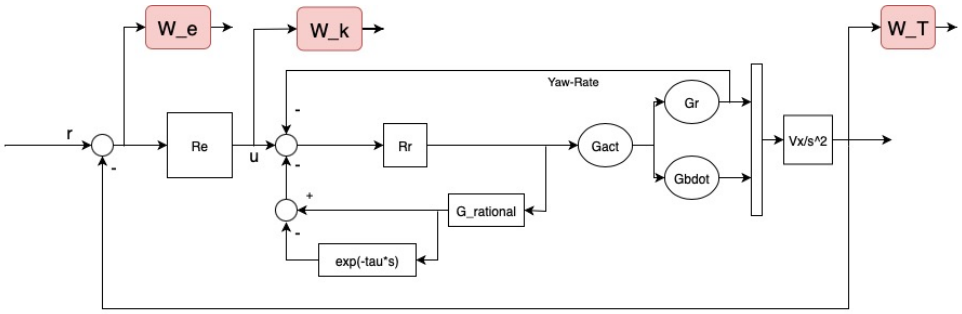


Figure 2.10: Block diagram defining the weighting functions to tune the Lead-Lag regulator with $\mathcal{H}_\infty$ synthesis strategy

the response faster, compensating a lot the lateral error. Alternatively, it is sufficient to reduce a little $\omega_c \approx 2\,[\text{rad/s}]$ with a gain of $k = 0.5$ to have a less nervous regulator and remove any possible oscillation ($\phi_m \approx 70°$) accepting a minimum lateral error.

2.2.2. $\mathcal{H}_\infty$ synthesis strategy

The strategy adopted here is the same as the one used for the yaw-rate regulator, and the parameters (summarized in table 2.5) are consistent with the objectives of the lead-lag compensator. In figure 2.10 it is illustrated the structure used to define the weighting functions.

In the Bode diagrams of figs. 2.11 and 2.12 are highlighted the comparison between the two regulators and the one of their complementary sensitivity transfer functions.

2.3. Longitudinal controller

The longitudinal controller is an $\mathcal{H}_\infty$ regulator tuned on the linearized longitudinal model with the aim of providing the angular velocity input to the differential, so that the vehicle speed remains constant.

Name	Function	Parameters and Notes
$W_e(s)$	$\frac{\frac{1}{M_e}s + 2\pi f_e}{s + 2\pi f_e A_e}$	• **Parameters**: $f_e = 0.3\,[\text{Hz}]$, $A_e = 0.01$, $M_e = 4$; • $S(s)$: high disturbance rejection on the output and low steady-state error; • Slower with respect to the inner-loop regulator; • Implies $S(j\omega) < -40\,\text{dB}$ at low frequencies, $\omega_c > 1.95\,[\text{rad/s}]$.
$W_k(s)$	$\frac{s + \frac{2\pi f_k}{M_k}}{A_k s + 2\pi f_k}$	• **Parameters**: $f_k = 150\,[\text{Hz}]$, $A_k = 10^{-3}$, $M_k = 30$.
$W_T(s)$	$\frac{s + \frac{2\pi f_T}{M_T}}{A_T s + 2\pi f_T}$	• **Parameters**: $f_T = 1.5\,[\text{Hz}]$, $A_T = 10^{-3}$, $M_T = 1.1$; • $T(s)$: limits the bandwidth to $\omega_c < 3.93\,[\text{rad/s}]$.

Table 2.5: Tuning and description of the weighting functions for the $\mathcal{H}_\infty$ Lateral-error regulator

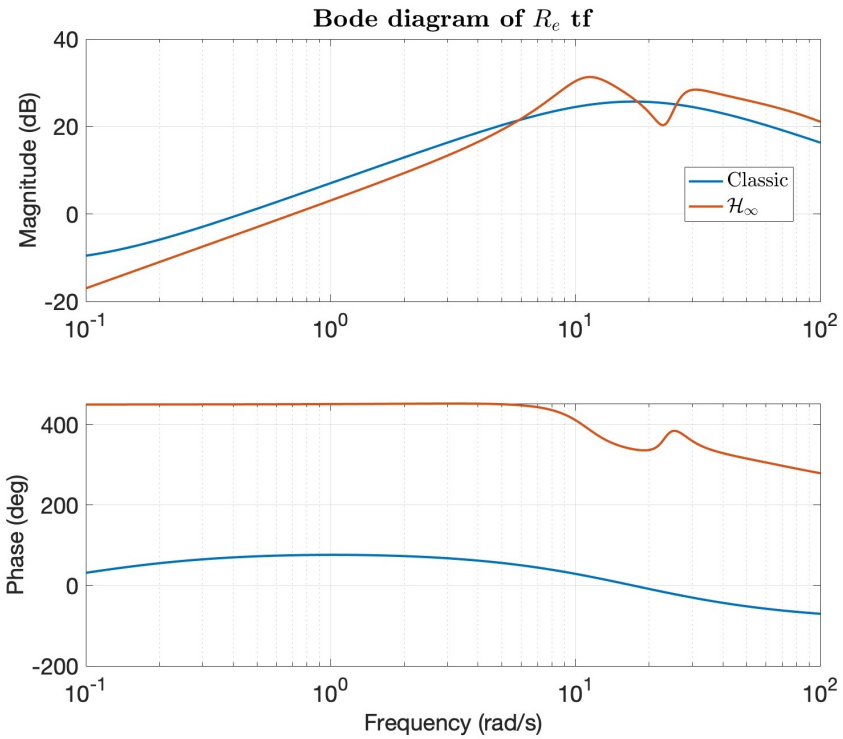


Figure 2.11: Bode diagram comparison for the lateral error regulator: more classical approach vs $\mathcal{H}_\infty$ one

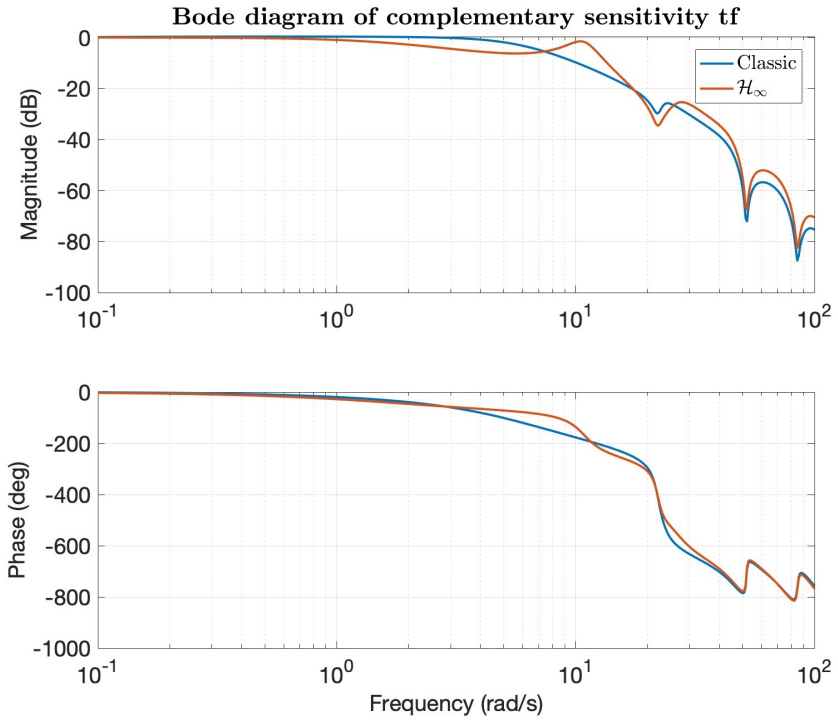


Figure 2.12: Bode diagram of lateral error complementary sensitivity transfer function using Lead-Lag compensator (classical approach vs $\mathcal{H}_\infty$ one) and the model G_{re}

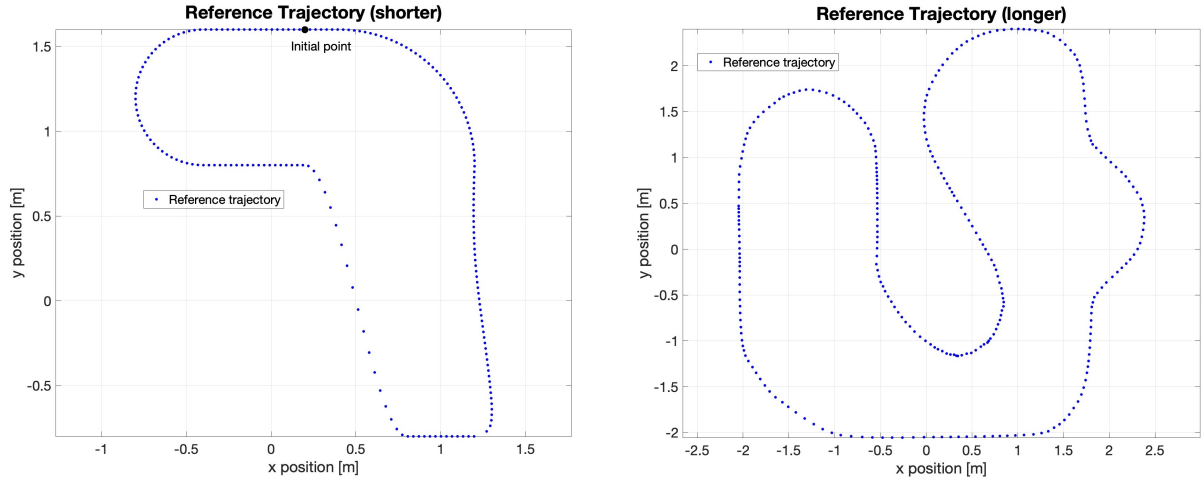
Name	Function	Parameters and Notes
$W_e(s)$	$\frac{\frac{1}{M_e}s + 2\pi f_e}{s + 2\pi f_e A_e}$	• **Parameters**: $f_e = 1\,[\text{Hz}]$, $A_e = 0.001$, $M_e = 2$; • $S(s)$: high disturbance rejection on the output and low steady-state error; • Implies $S(j\omega) < -60\,\text{dB}$ at low frequencies, $\omega_c > 7.26\,[\text{rad/s}]$.
$W_k(s)$	$\frac{s + \frac{2\pi f_k}{M_k}}{A_k s + 2\pi f_k}$	• **Parameters**: $f_k = 110\,[\text{Hz}]$, $A_k = 10^{-3}$, $M_k = 90$.
$W_T(s)$	$\frac{s + \frac{2\pi f_T}{M_T}}{A_T s + 2\pi f_T}$	• **Parameters**: $f_T = 10\,[\text{Hz}]$, $A_T = 10^{-3}$, $M_T = 1.1$; • $T(s)$: limits the bandwidth to $\omega_c < 26.2\,[\text{rad/s}]$.

Table 2.6: Tuning and description of the weighting functions for the $\mathcal{H}_\infty$ longitudinal regulator

3 | Implementation and Simulation

3.1. Trajectory

In this thesis project, two different paths have been studied. Since the non-optimization based curvature preview control is a path-tracking algorithm, it uses the advance knowledge of the path curvature to operate. This can be computed online by a path-planning algorithm or, as in this case, by an offline algorithm that requires full knowledge of the path. These paths were obtained by using measurements from two different trajectories of the Microcar, made available by Gipsa-Lab.



(a) Shorter path (b) Longer path

Figure 3.1: Picture of the two different original paths used in this thesis. They are resampled by a factor of 5 and 10, respectively, to make the image visible.

The first path is smaller, with accentuated curves difficult to travel where the steering angle saturates and inevitably the vehicle leaves the reference trajectory. The second one instead is longer, more winding but the curves are not so tight as to saturate the steering. It is much closer to a common path. Since the paths are the result of various measurements, they display the following issues:

- they are not uniformly sampled. Vehicle speed was not perfectly constant, and consequently the distance between two subsequent points varies;
- there is some noise, especially in the longer path (3.1b);
- there is no perfect continuity between the starting point and the ending point.

Before computing the track curvature, due to the problems described, an algorithm was developed to interpolate and to filter the reference trajectories.

1. First of all, the trajectory is created by assuming that the vehicle moves at a constant speed of V ($V = 1.2\,[\mathrm{m/s}]$) and then, computing the distance between two nearby points, each sample is associated with its instant of time, i.e. $dt = d/V$.
2. A spline function is generated to interpolate the samples and get a continuous function. The `csape(time,path,'periodic')` Matlab function is used with the 'periodic' command in order to guarantee continuity until the second derivative at the extremes. This allows for a closed path, making it possible to run more laps and facilitating the trajectory filtering and derivative computations, which are useful for obtaining the curvature.
3. A uniform time grid is built with `t_uniform=linspace(min(time),max(time),(max(time)-min(time))*fs)`using `ts=median(diff(time))` as a reference sampling time. This sampling time choice (for example instead of using `min(diff(time))`) is motivated by excluding those outliers that would lead to oversampling.
4. The spline function is evaluated on `t_uniform` using the command `ppval(...)`, generating the new trajectory `traj_spline=[t_uniform,x_spline,y_spline]`.

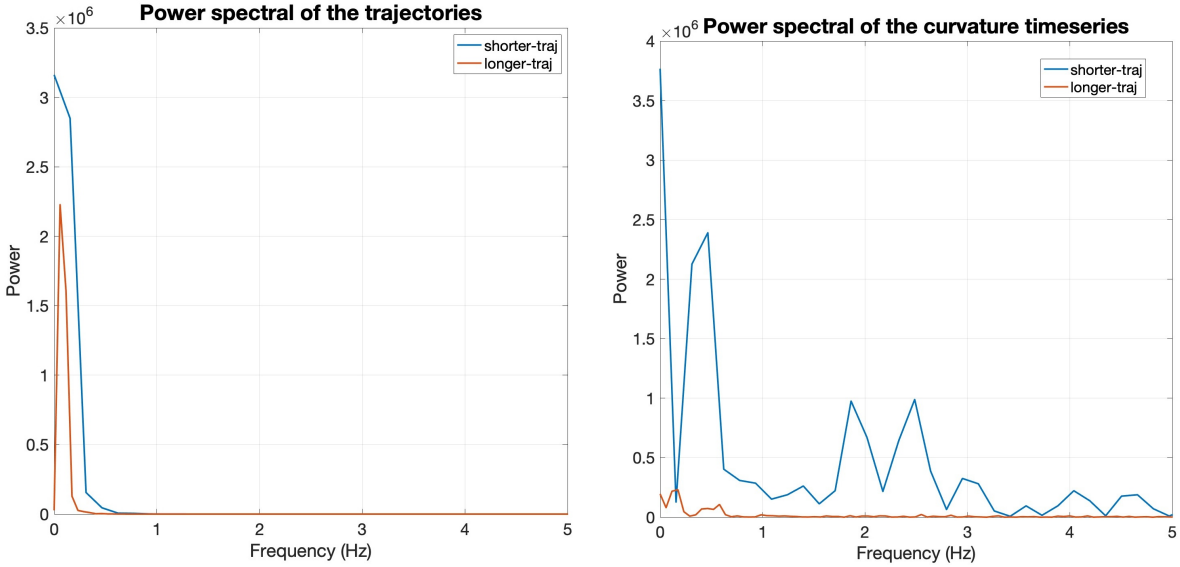
The trajectory is now equally sampled and can be filtered using a function designed to filter both curvature and trajectory. It uses a butterworth filter of the 4^{th} order applied forward and backward (using the function `filtfilt()`) to keep zero phase. The main points of the algorithm, called `filtfiltbutter()`, are the following:

1. The input dataset is extended three times `data_extended=[data;data;data]`, properly managing the associated time instants.
2. A frequency analysis is carried out to choose properly a cutting frequency for the butterworth filter, $f_c = 5$ for the trajectory and $f_c = 1.5$ for the curvature. In the following fig. 3.2 the power spectral densities of the trajectory and curvature timeseries are depicted. Power Spectral Desnity is computed as the square of the Fourier coefficients for each frequency without normalizing with respect to period length.

3. a `filtfilt()` function is used, it filters two times (forward and backward) in order to avoid any phase shift. Each phase variation rotates the trajectory with respect to its initial orientation.

4. only the middle filtered-data (second lap of three) is provided as output of the `filtfiltbutter()` function. This guarantees continuity and no distortion at the extremes.

The dataset is first extended and at the end is cut because at the extremities the signal can be distorted. It is provided as input to the filter, which acts as a transfer function, taking some time to reach a steady state. The dynamic of the transient could, not only filter, but also alter and distort the signal.



(a) PSD of the trajectories.

(b) PSD of the curvature timeseries.

Figure 3.2: Power Spectral Densities of trajectory and curvature

In figure 3.3 are shown the final trajectories, and in fig. 3.4 is depicted a zoom to show noise and filtering action.

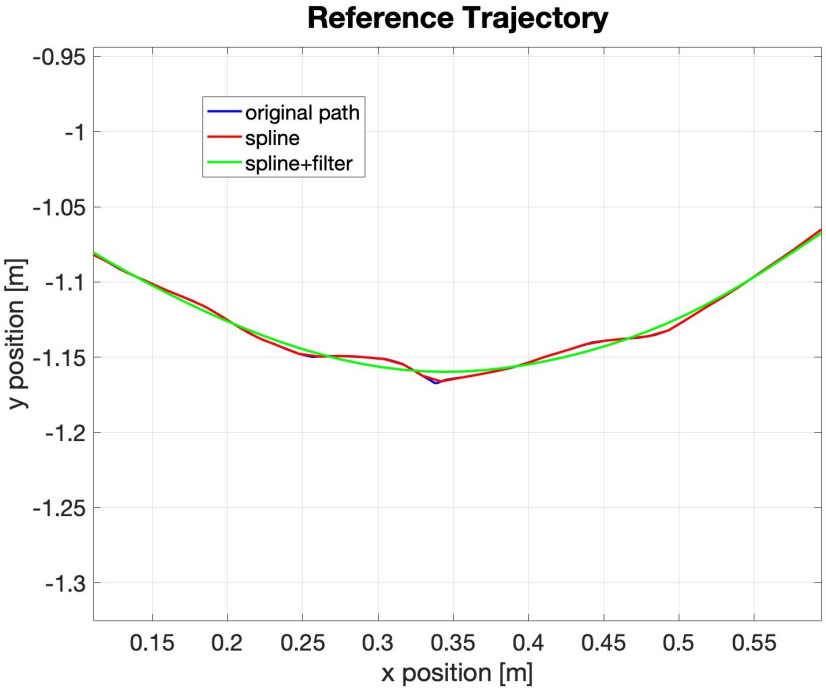
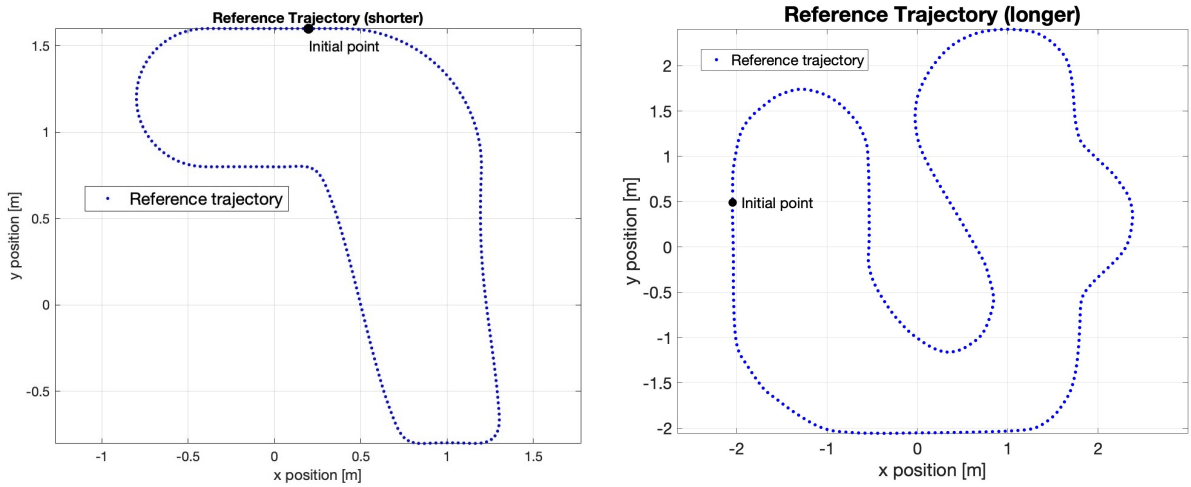


Figure 3.4: Picture from the second trajectory (longer one), zoomed to emphasize the different trajectory steps.



(a) Shorter trajectory.

(b) Longer trajectory.

Figure 3.3: Final trajectories after interpolation and filtering action, resampled to make the image more understandable

3.2. Curvature

Curvature is a measure that represents how tight a curve is. It has several definitions, the most common one is known as the inverse of the radius of an osculating circle at a given point. The definition used in this thesis is the one previously mentioned in proposition 1.1. A curvature computed at a point i on the trajectory is:

$$k(i) = \frac{\alpha(i)}{\Delta s}, \tag{3.1}$$

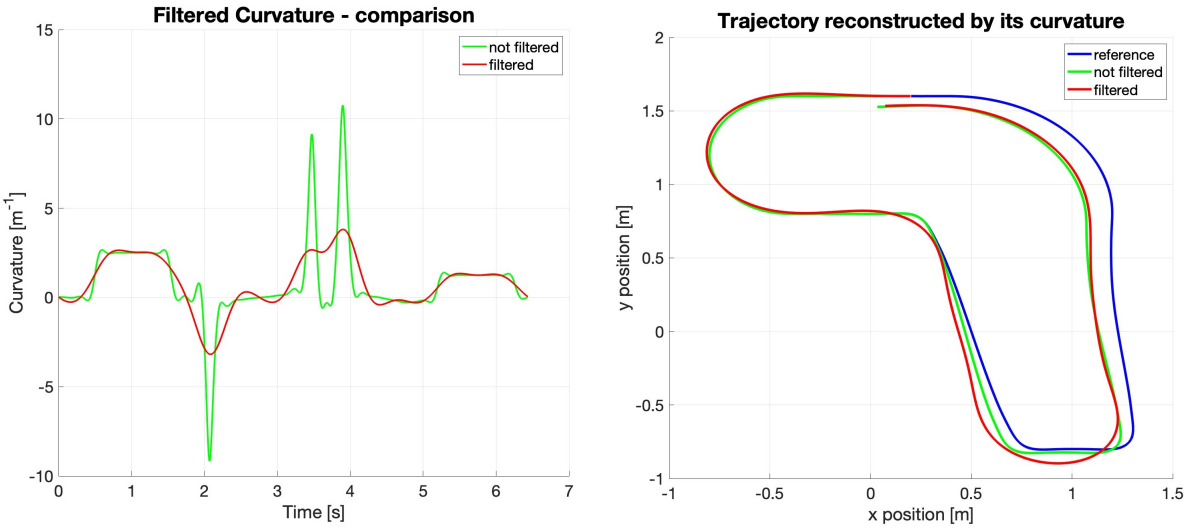
where α can be computed with different strategies. The one used in this thesis is the following:

1. $\alpha(i)$ represents the angle between the tangent vectors at point i and at point $i+1$;
2. the tangent vector is defined as `Der = [gradient(x,s),gradient(y,s)]` where s is the curvilinear abscissa;
3. by the definition of the scalar product between two vectors of the plane in Euclidean space, it is possible to get the

$$\cos(\alpha(i)) = \frac{\mathrm{Der}(i,:) \cdot \mathrm{Der}(i+1,:)}{\|\mathrm{Der}(i,:)\| \cdot \|\mathrm{Der}(i+1,:)\|} \tag{3.2}$$

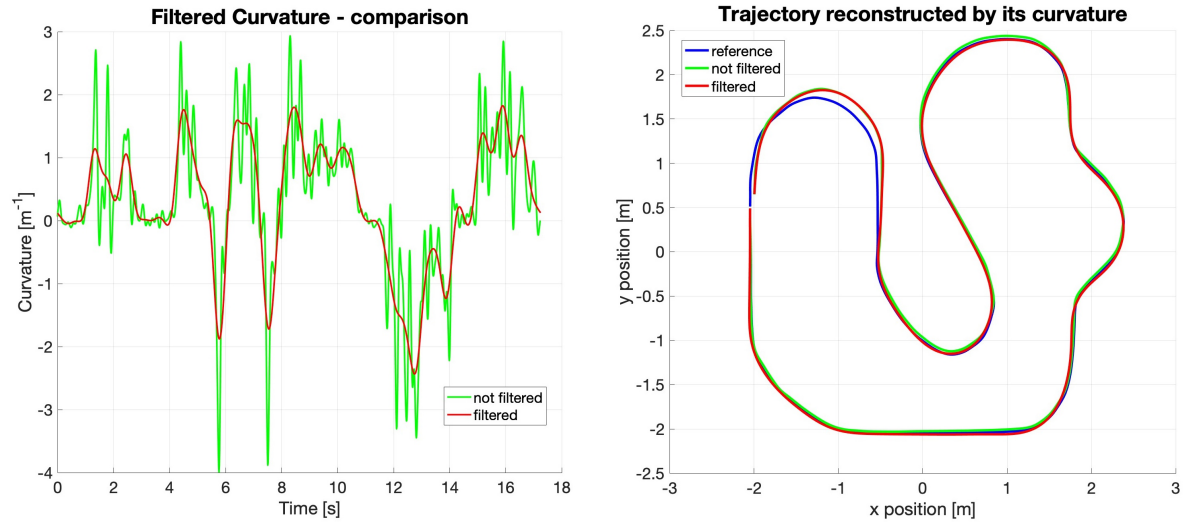
4. finally, the norm of the angle is obtained as $\|\alpha(i)\| = \arccos(\cdot)$, and its sign is the sign of $\det(\mathrm{Der}(i : i+1,:))$.

Due to the derivative factor, the curvature obtained is noisy, it has components at high frequency that cannot be used as a reference for the vehicle control. The curvature is, then, filtered, and it is possible to see its final result in figs. 3.5a and 3.6a. In figs. 3.5b and 3.6b, it is shown the trajectories reconstructed by their corresponding curvature, to understand how much is the loss due to this cascaded filtering actions.



(a) Curvature comparison. (b) trajectory reconstructed.

Figure 3.5: Comparison between the curvature before and after filtering, and their corresponding trajectory. Shorter trajectory.



(a) Curvature comparison. (b) trajectory reconstructed.

Figure 3.6: Comparison between the curvature before and after filtering, and their corresponding trajectory. Longer trajectory.

3.3. Lateral error and curvature synchronization

The topic under discussion in this paragraph concerns the algorithms estimating in real-time the lateral error of the vehicle with respect to the reference trajectory and the algo-

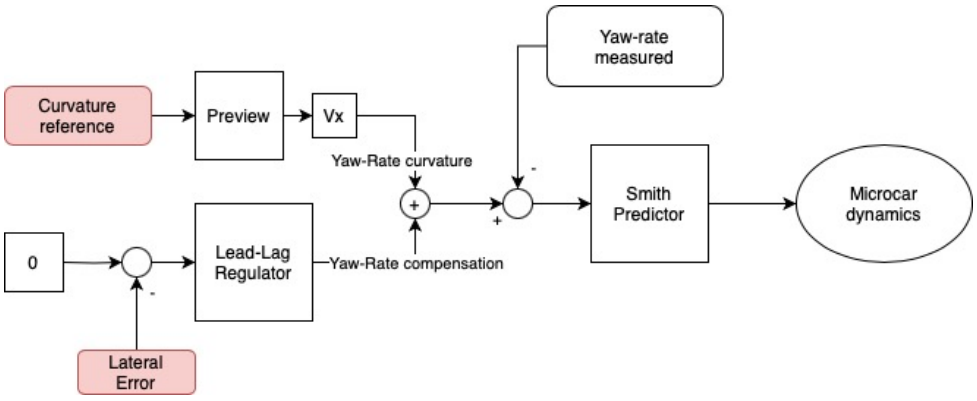


Figure 3.7: Control scheme diagram: lateral error and curvature synchronization

rithm that synchronizes the reference curvature with the current position of the vehicle. Both are highlighted in the control scheme of fig. 3.7.

3.3.1. Lateral-error algorithm

The developed algorithm works in the following way:

1. It finds the two closest points $P1$ and $P2$ on the trajectory using `knnsearch()` command. The research is done within a smaller moving window in order to reduce computational costs. The window size is taken very small at the beginning, and is increased until $P1$ and $P2$ are always inside the window, and never on the boundaries.
2. It computes the line passing through $P1$ and $P2$.
3. It uses the point-line distance formula to get the **absolute value of lateral error**.

   Definition 3.3.1 (Point-line distance). *Given a line $r : ax + by + c = 0$ with $a = y_1 - y_2$, $b = x_2 - x_1$ and $c = y_2x_1 - x_2y_1$, the distance between a point P and the line r is equal to:*

   $$\|distance(P, r)\| = \frac{\|ax_p + by_p + c\|}{\sqrt{a^2 + b^2}} \tag{3.3}$$

   with $P = (x_p, y_p)$
4. The **sign of the lateral error** is the sign of the angle between
   - the **position vector**, from the closest point on the reference trajectory to the current position of the vehicle

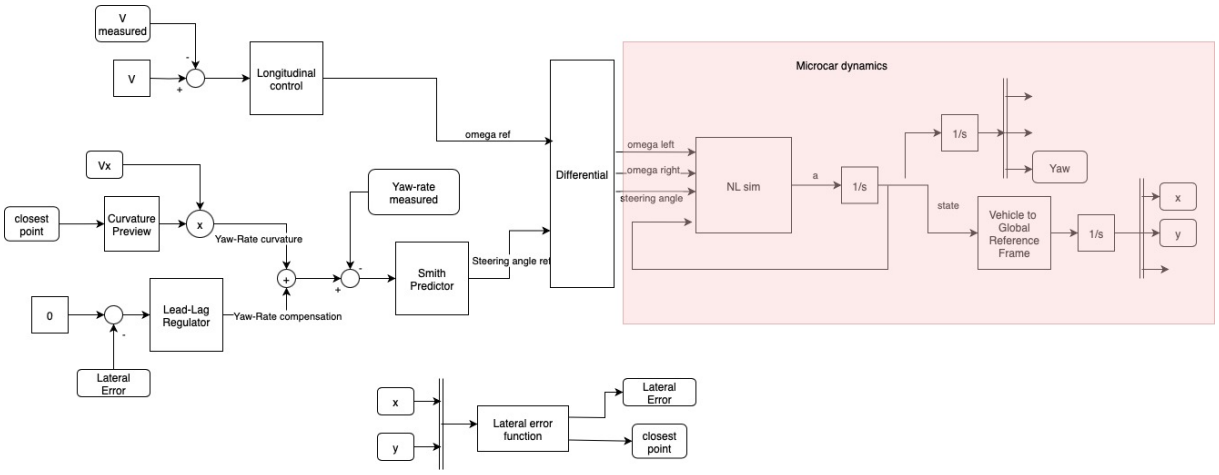


Figure 3.8: Diagram of the whole control architecture

- and the **normal vector** to the trajectory on the closest point

By interpolating the points with a line, it is obtained a lateral error function that is more continuous and does not introduce discontinuities.

3.3.2. Curvature synchronization

In order to make the control architecture more robust to speed variations with respect to its constant value V, the curvature of the reference trajectory has been synchronized with the current position of the vehicle. The algorithm utilizes the closest point on the reference trajectory $P1$ computed real-time by the lateral error function to provide its corresponding curvature. Actually, it provides the preview curvature, that is, the corresponding curvature anticipated by the value τ.

3.4. Implementation of the non-linear model

The software used to perform simulations and control in real-time the Microcar is Simulink [1]. The whole architecture, represented in the diagram of fig. 3.8, is designed to work in discrete time with the exception of the non-linear model simulation that is in continuous time.

The function that takes care of simulating the dynamics of the model 1.4 is `simNL()`. It is summarized in the following steps:

- At each time step this function uses the current values of the input references

[1]Simulink, MathWorks. `https://www.mathworks.com/products/simulink.html`

$[\omega_l, \omega_r, \delta]$ and of the states to compute car accelerations. More precisely, tire stiffnesses $C_{\sigma l,r}$, $C_{\alpha f,r}$, slip ratios $\sigma_{l,r}$ and side-slip angles $\alpha_{f,r}$ are calculated and used to obtain the current forces. At this point the force balance of the nonlinear model defines the corresponding acceleration values $[a_x, a_y, \ddot{\psi}]$.

- Subsequently, starting from the initial speed V_0, the model integrates to compute the state vector $[v_x, v_y, \dot{\psi}]$ which is passed to the `simNL()` function for the next iteration.
- By integrating the yaw rate of the car from its initial orientation, the yaw of the car is obtained.
- Lastly, the vehicle position is computed moving from the reference system of the car to the global one estimating the side-slip angle as $\beta = \text{atan2}(v_y, v_x)$ and speed magnitude as $V = \text{norm}(v_x, v_y)$. Velocities with respect to the global reference frame are computed as $\dot{x} = V\cos(\psi + \beta)$ and $\dot{y} = V\sin(\psi + \beta)$. Finally by integrating $\dot{x}, \dot{y}$ from the initial point the positions $x(t), y(t)$ are obtained.

3.5. Results of simulations and Microcar tests

The current section describes different tests conducted to introduce and present the performance of the control architecture described so far. It is organized into sub-sections, starting with preliminary tests where only the yaw-rate controller is used; continuing with full-architecture simulations where both regulators are used but no saturation limits are imposed to the steering angle (the control variable); ending with some comparative tests between full-architecture simulations with saturation and the actual results of the Microcar running.

3.5.1. Yaw-rate test

The first test carried out uses an artificial curvature (fig. 3.9) to analyze the behavior of the Smith predictor along a single turn.

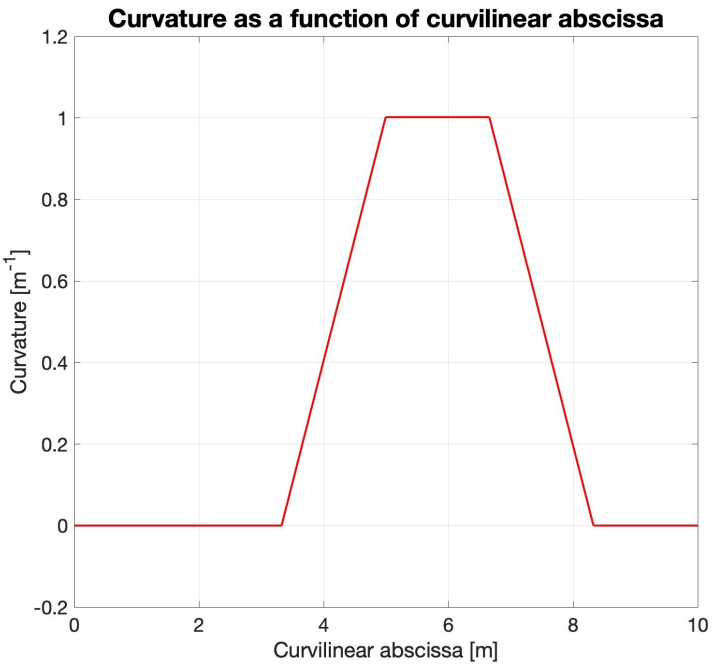


Figure 3.9: Reference of the artificial curvature

The curve is chosen with a minimum radius of $1\,[m]$ and a total length of $10\,[m]$. The simulation is analyzed through a series of pictures, where in fig. 3.10 the reference trajectory, directly reconstructed by the artificial curvature, is compared with the simulated trajectory.

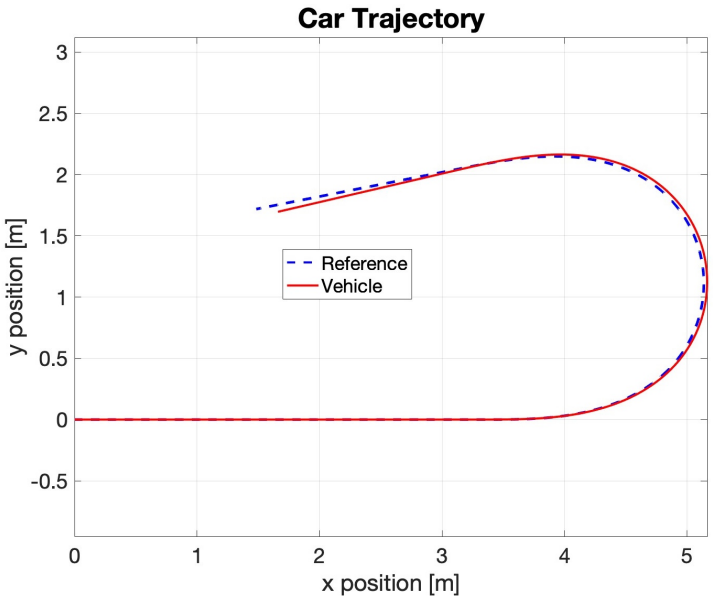


Figure 3.10: Simulation of a single curve trajectory using only the yaw-rate controller

Even within a single curve, a small but noticeable deviation from the reference trajectory can be noticed. It is marked out by a small slip towards the outside of the curve, which causes a small change in direction on the final straight. The small deviation is corroborated by the lateral error plot in fig. 3.11, where the progressive drift of the trajectory is evident, and the sign change highlights the slight difference in direction.

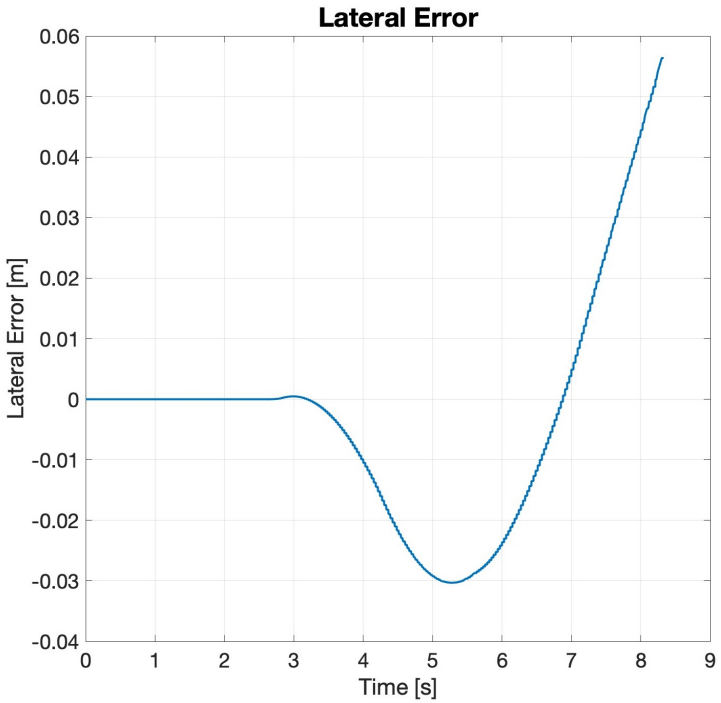


Figure 3.11: Lateral error in the yaw-rate test

This deviation becomes more evident within the entire track simulations. It is a phenomenon already reported and discussed in the literature [6], that prompted the use of an external controller. In fig. 3.12 the progressive trajectory misalignment grows curve after curve, while in fig. 3.13a this phenomenon occurs all together during a very challenging curve. In the latter case, the dynamics required to perform the curve are probably too arduous to be followed perfectly. This is particularly evident in fig. 3.13b, where the yaw-rate reference tracking is shown. The tracking error is clear in the third curve (and slightly at the end of the second curve), significantly affecting the overall trajectory tracking outcome.

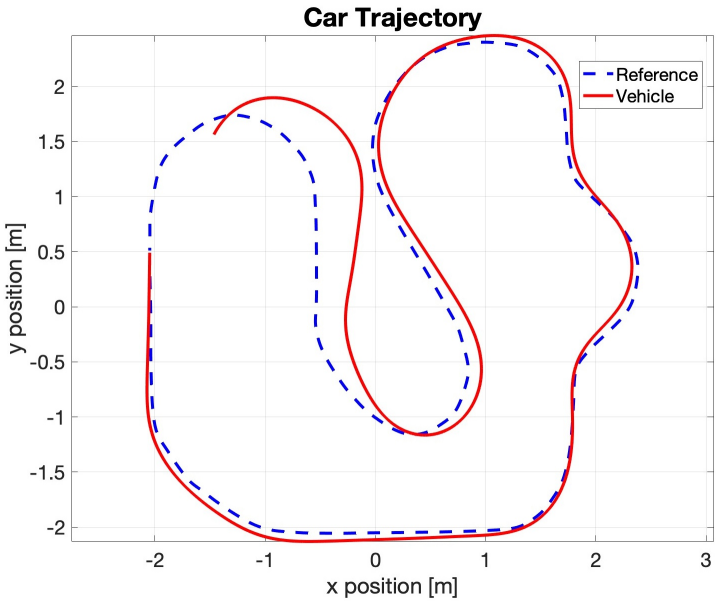
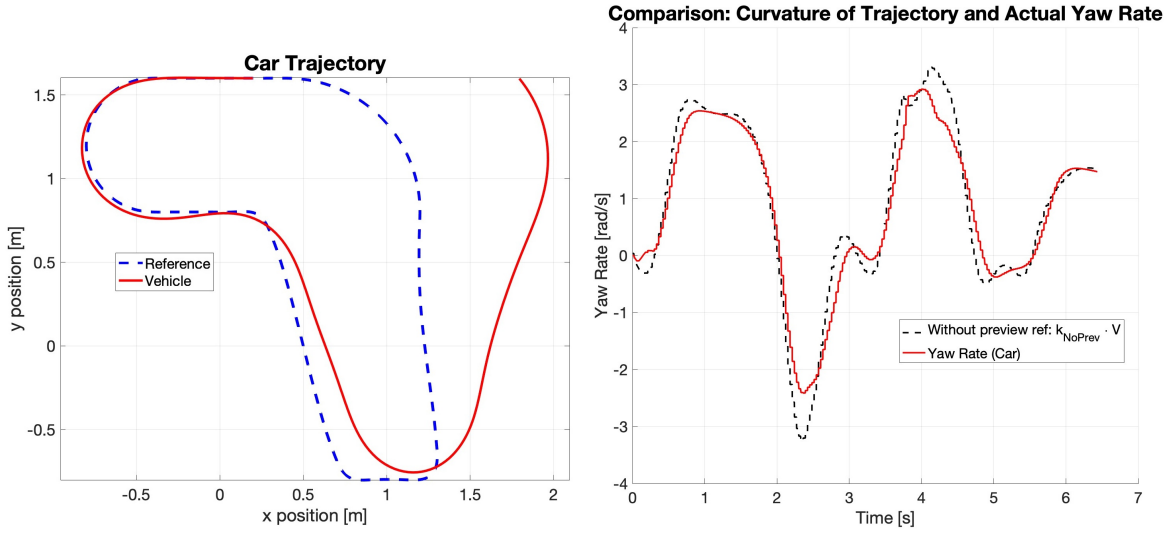


Figure 3.12: Trajectory misalignment (longer trajectory) using only the YR controller



(a) trajectory plot (b) YR plot

Figure 3.13: Trajectory misalignment (shorter trajectory) using only the YR controller

Going back to fig. 3.10, it can also be observed that the simulated trajectory is very smooth. This is due to the fact that the steering angle in fig. 3.14 is soft, and features only very small fluctuations at the points of discontinuity. This shows that the Smith predictor design is effectively robust against small discontinuities in the input, with particular attention given to calibrating the trade-off between fast performance and good filtering at high frequencies.

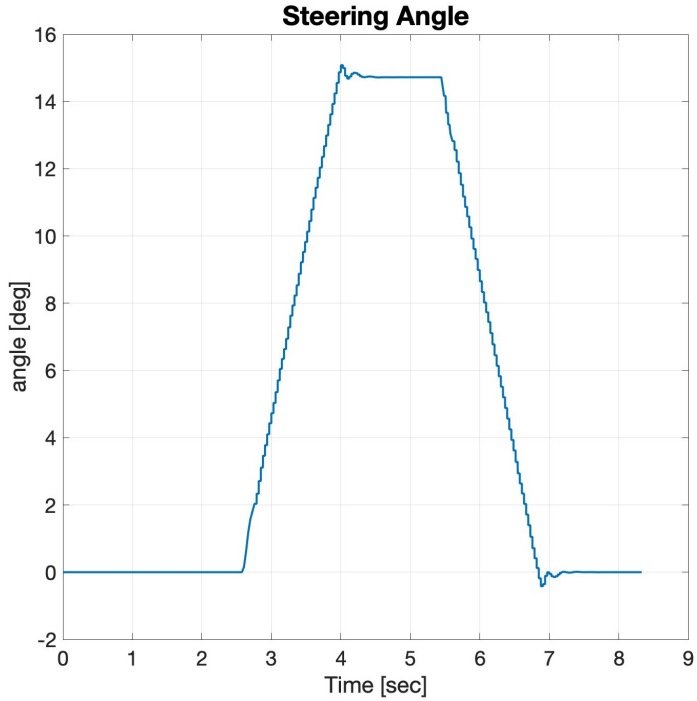
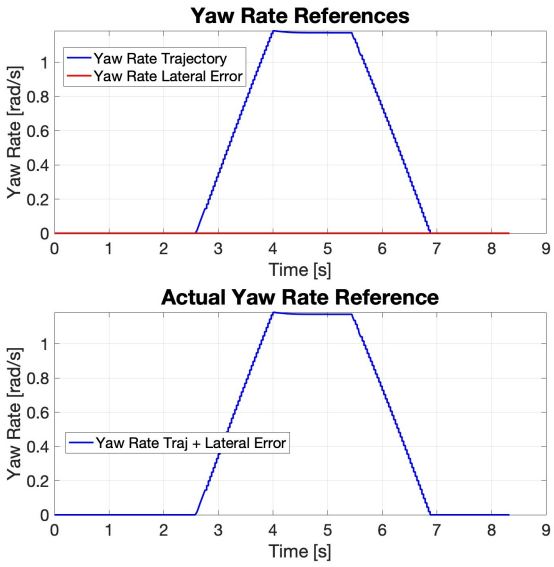
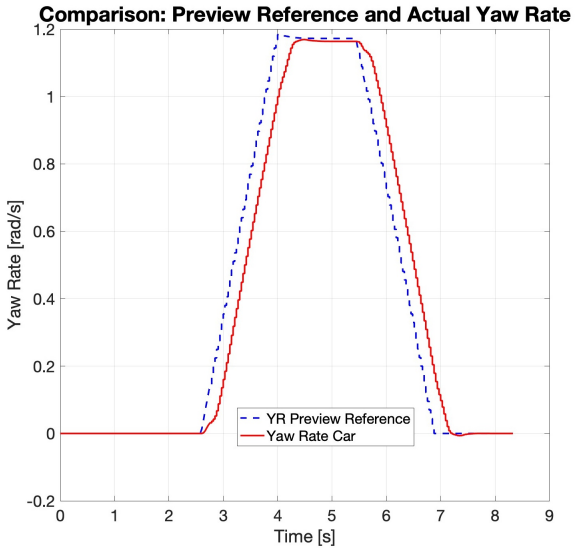


Figure 3.14: Steering angle in the yaw-rate test

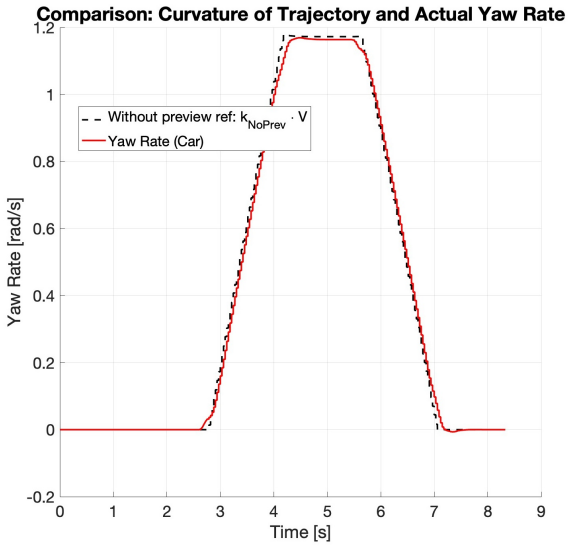
The figs. 3.15a to 3.15c focus on the yaw-rate tracking, the first one differentiates the yaw-rate component of the reference due to the trajectory (the one that acts as a "disturbance") from the component calculated by the external compensator, which in this case, being turned off, is zero. The second one illustrates the actual simulated yaw-rate with respect to the preview reference. It is interesting to see that its tracking creates a time delay. In the third image the track's yaw rate, obtained by multiplying the curvature by the vehicle's speed, is compared to the simulated vehicle's yaw rate showing a precise tracking. Of note, the time delay is fully compensated by the preview component. It means that the preview term is effectively compensating the delay term and, moreover, the Smith predictor is sufficiently fast to follow the yaw-rate dynamic while its integral action is sufficient to avoid steady-state errors.
The figure 3.16 shows the speed values simulated. It shows respectively the longitudinal and lateral speed, while the third plot resumes the actual yaw-rate. It can be noticed that as soon as the car approaches the curve, the longitudinal speed decreases in favor of the lateral speed.



(a) Yaw-rate traj VS yaw-rate of the lateral error



(b) Actual yaw rate VS preview yaw-rate reference



(c) Actual yaw rate VS yaw-rate reference without preview

Figure 3.15: Yaw-rate in the yaw-rate test

3.5.2. Entire track test

The following tests aim to emphasize the role of the external regulator. The architecture used is constituted by the same Smith predictor with $\mathcal{H}_\infty$ tuning of the yaw-rate test, while the lead-lag regulator with $\mathcal{H}_\infty$ approach closes the lateral-error loop. The simulations are performed without saturation constraints on the steering angle, in order to observe

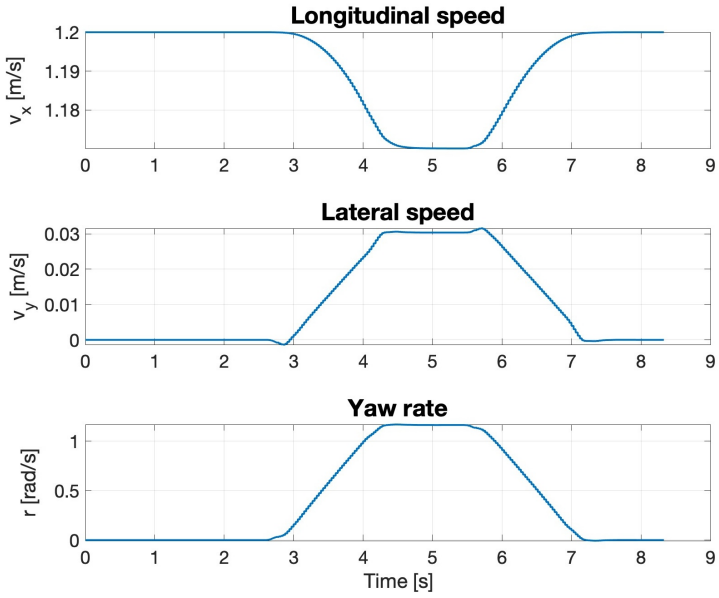
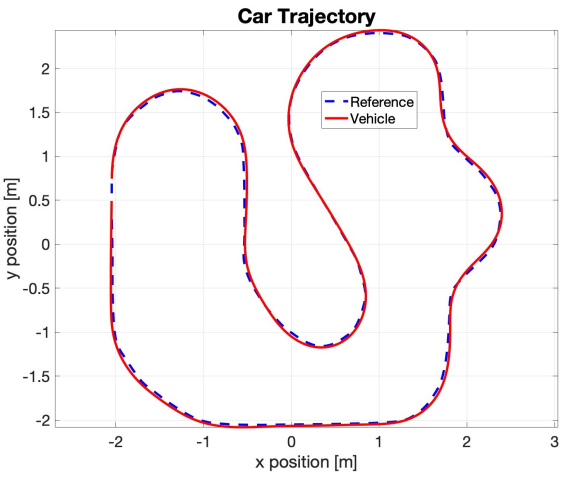


Figure 3.16: Speed profile in the yaw-rate test

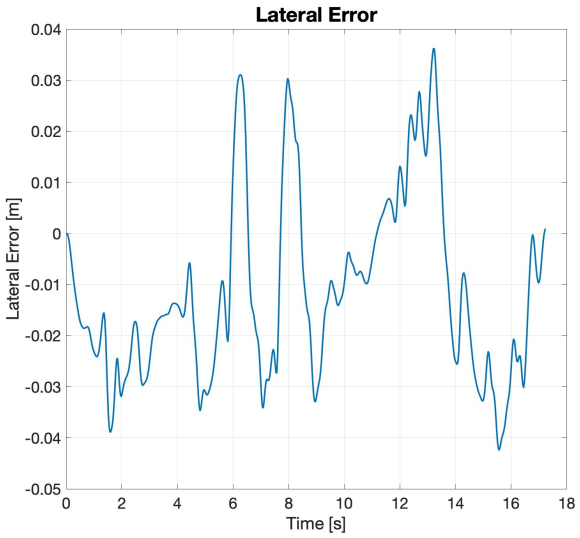
and analyze the limits of vehicle dynamics.

The results of the longer-path simulation are summarized in fig. 3.17. They are characterized by a path-tracking with a lateral error smaller than $4\,[cm]$. The outer-loop regulator intervenes minimally by varying the yaw-rate reference, as can be seen in fig. 3.17d, keeping the vehicle on its reference path. In fig. 3.17c the steering plot appears a little nervous and seems to follow the shape of the yaw-rate reference without too many corrections, allowing an accurate yaw-rate control, see figs. 3.17e and 3.17f, with the exception of few peaks.

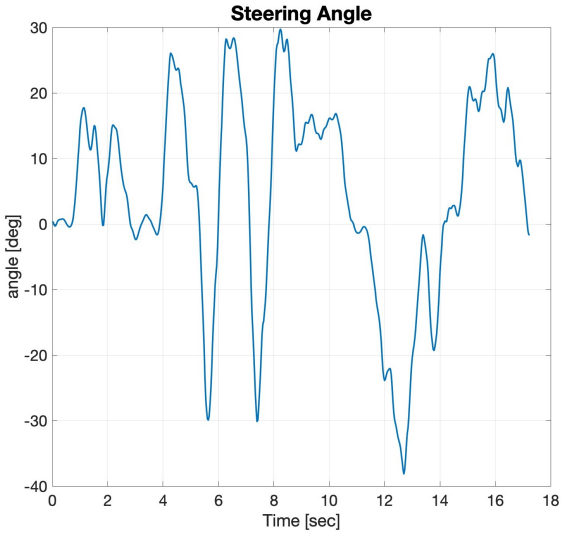
The shorter-path simulation points out the dynamic limits of the vehicle, even though the steering angle reaches very high values up to 60°, where the yaw-rate regulator is not fast enough to have a perfect tracking as already shown in 3.5.1. As already highlighted, this track has much tighter curves than normal, comparable to parking maneuvers, thus explaining why the lateral error increases. In fig. 3.18d it can be noted that the external regulator works to bring the vehicle back on the desired path, although it was not specifically designed for these situations. Its purpose is to correct low-frequency deviations and not sudden changes.



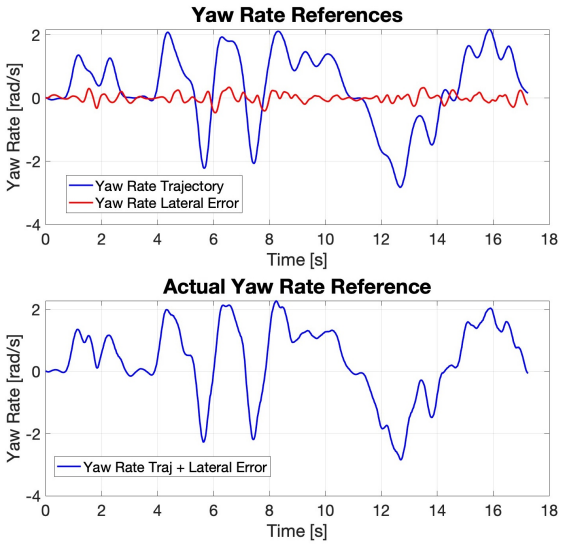
(a) Trajectory



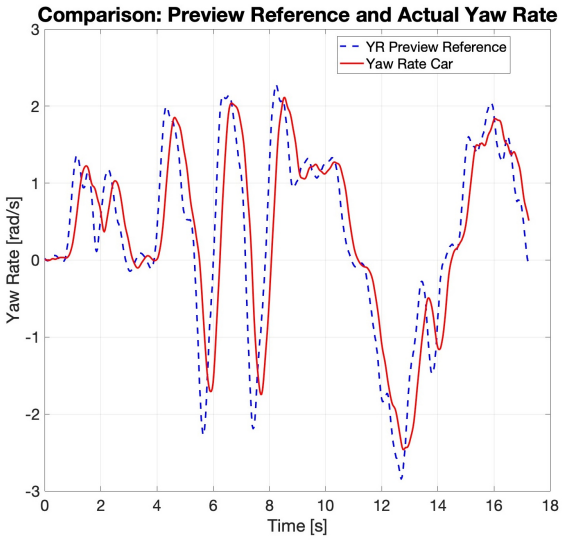
(b) Lateral error



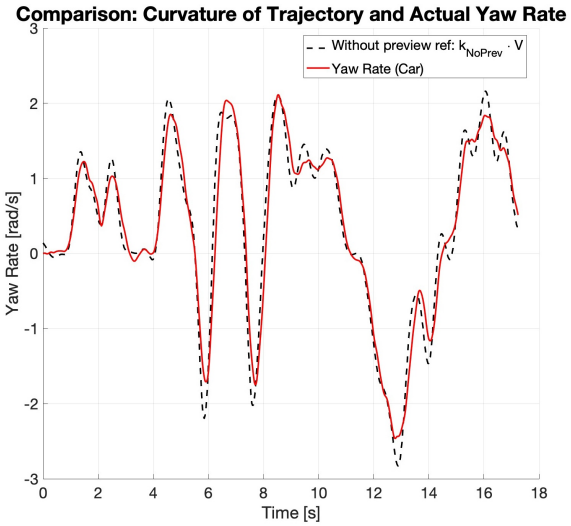
(c) Steering angle



(d) YR traj VS YR e_cog

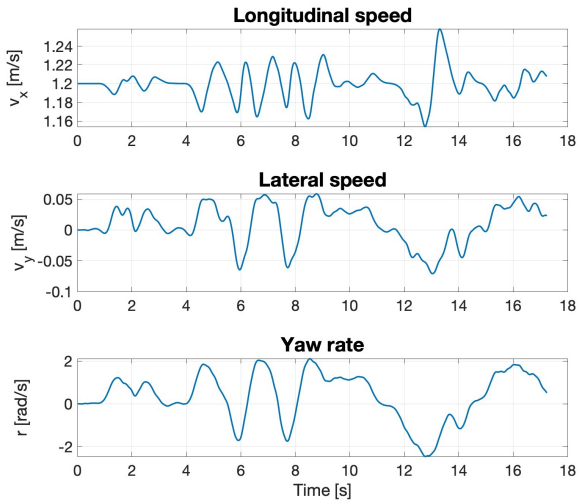


(e) Actual YR VS Preview YR ref



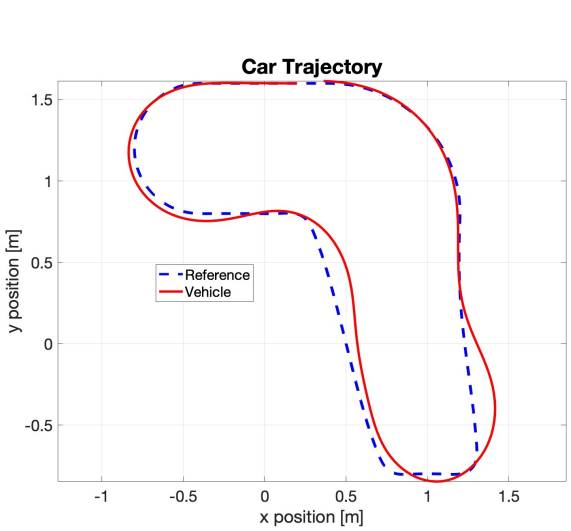
(f) Actual YR VS w/o preview YR ref

Figure 3.17: Longer path - Simulation results (1/2)

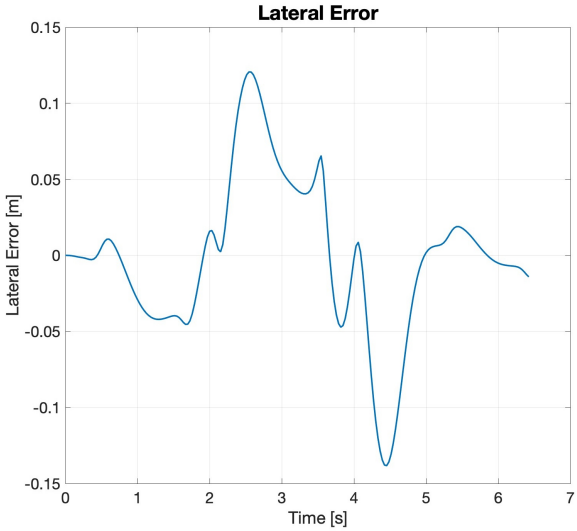


(g) Speed profile

Figure 3.17: Longer path - Simulation results (2/2)

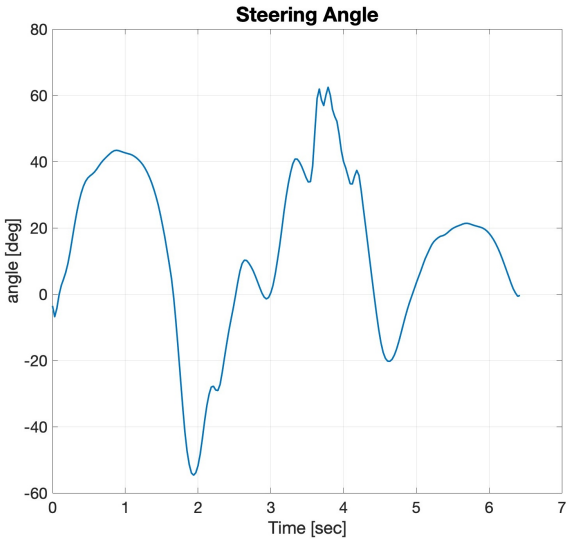


(a) Trajectory

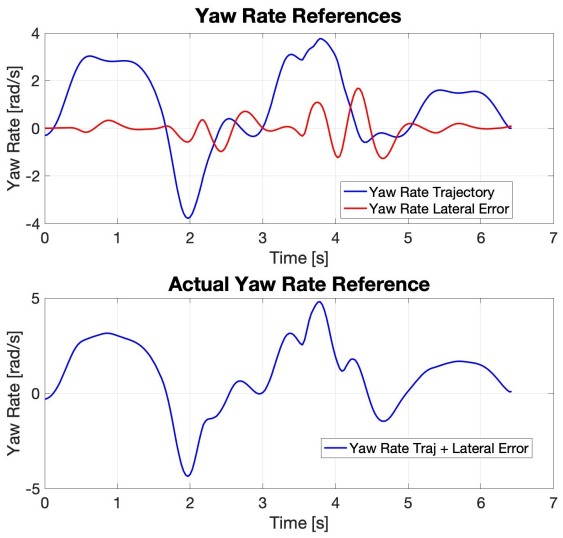


(b) Lateral error

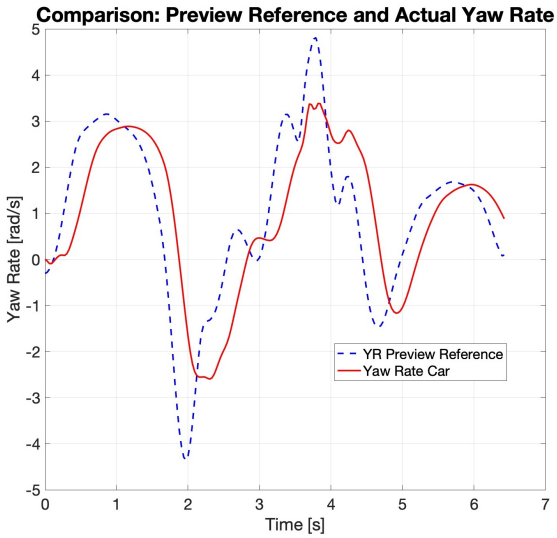
Figure 3.18: Shorter path - Simulation results (1/2)



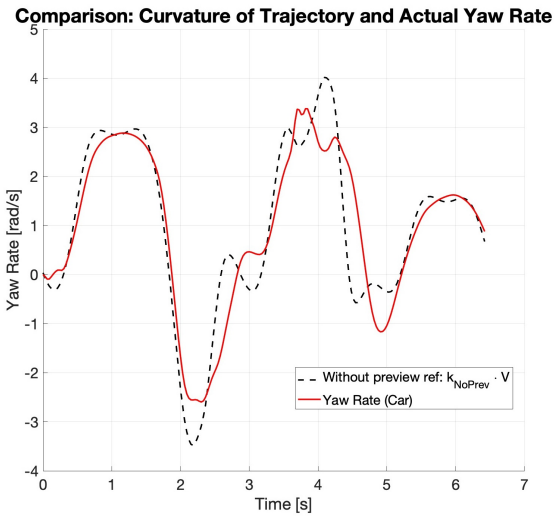
(c) Steering angle



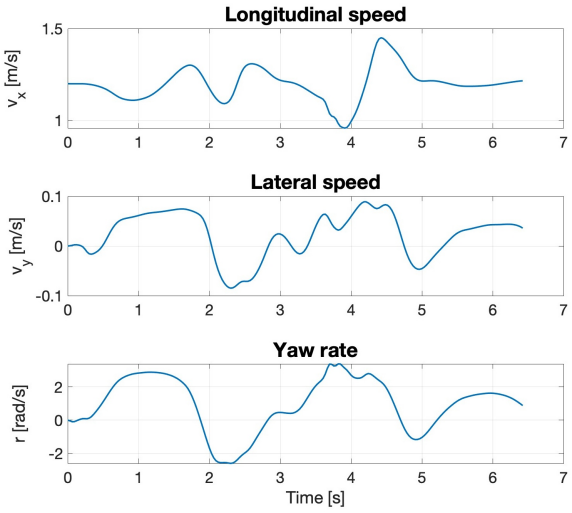
(d) YR traj VS YR e_cog



(e) Actual YR VS Preview YR ref



(f) Actual YR VS w/o preview YR ref



(g) Speed profile

Figure 3.18: Shorter path - Simulation results (2/2)

3.5.3. Simulation vs Microcar test

This last section of the chapter discusses the actual results of the Microcar running in the platform along the two already known paths.

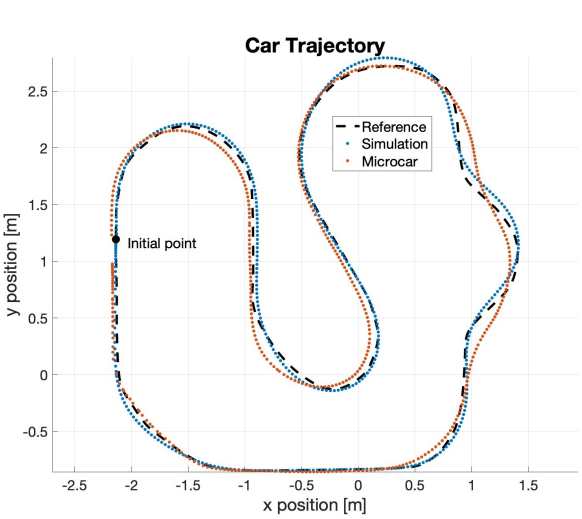
Before starting, it is necessary to make some considerations. The Microcar has undergone some recent changes, although it preserves the same architecture, the parameters identified and used in the platform still belong to the old version. It is likely that some parameters may have slightly varied in value, partially compromising the reliability of the results. Thanks to the upcoming developments and work that the Gipsa laboratory will carry out, it will be possible to access new values in the next future.
In addition, during the testing phase, some issues were encountered using a Kalman filter. It has been developed in the GIPSA laboratory to estimate vehicle velocity based on car position and orientation. Attempts were made to correct the Kalman filter a posteriori to display more reliable values for actual velocities (the issue primarily concerns the vehicle's lateral velocity). During the testing phase, these values were used to close the longitudinal control loop and derive the yaw-rate reference by multiplying the curvature by the longitudinal velocity, which may have degraded final performance.

In order to give greater meaning to the results, the figures present a comparison with the simulations made using the same architecture and under the same conditions. The longer path test is enclosed in the set of images 3.19, where the Microcar starts near the initial point from a standing position (with a zero initial speed). The steering angle is limited to the range of $[-45°, 45°]$ and the angular speed to $[-100\,\text{rad/s}, 100\,\text{rad/s}]$. In the first figure (fig. 3.19a) the Microcar running and the simulation show similar results, where the largest discrepancy is observed between 30% and 50% of the route, exactly in the right part where the path is made up of a series of linked curves (the first curve is to the right, then immediately to the left and finally to the right again to continue in the same direction as before). In the next image, fig. 3.19b, it is noticeable that precisely at this point the lateral error of the Microcar is greater compared to the simulation one, and, indeed, its trajectory tends to cut the curves, as if the car was taking them more smoothly. In fig. 3.19f, always at the same position, the yaw rate of the Microcar cuts the peaks and does not completely follow the trajectory yaw rate. This may be partly due to the external regulator intervention, which corrects the small lateral error of the previous curve by changing the yaw-rate reference and therefore distinguishing the yaw-rate reference from the trajectory one (figs. 3.19d and 3.19e). Furthermore, in fig. 3.19d can also be seen that the lead-lag compensator always intervenes along the entire path to correct and always keep the vehicle on the desired trajectory.

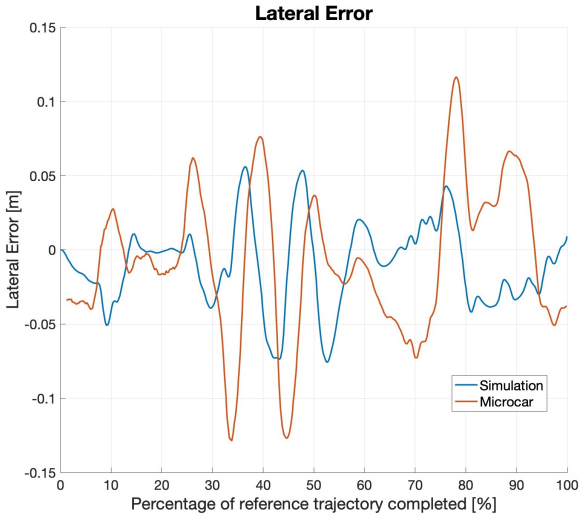
The fig. 3.19c shows the evolution of the control variable. The command of the Microcar steering angle tends to be more nervous and with some more oscillations and corrections than the simulation one. Unfortunately, as already mentioned, there are no sensors capable of measuring the actual steering angle. Considering that the model parameters are slightly different, it's possible that the steering will respond differently than expected. This, along with the other considerations already mentioned, could also explain the observed corrections. However, overall, it can be said that the steering correlates well with the simulation, the vehicle is able to track the path precisely with a lateral error always less than $15[cm]$ and exceeding $10\,[cm]$ only in three points, demonstrating the robustness of the entire control architecture in the face of model uncertainties and measurements which appear to be not entirely accurate (especially regarding lateral speed). With regard to this last consideration, it can be observed in fig. 3.19g a comparison plot between the longitudinal and lateral speeds and the yaw rates. The biggest difference is in the lateral velocity, and this does not necessarily mean that the vehicle's lateral velocity is actually higher, but it is possible that its estimate is simply less accurate.

The shorter path test instead was carried out with an initial speed equal to the equilibrium value around which the model was linearized ($V_0 = V = 1.2\,[\text{rad/s}]$), always starting from a point close to the initial position. To achieve this, the vehicle performed a preparatory lap so that it would arrive in proximity of the starting point with the desired speed. Also in this case, the results were compared with a simulation test performed under the same conditions, and therefore the same controllers and architecture. They have been summarized in the series of figures 3.20, a first analysis immediately reveals that this track is indeed more challenging and takes a particular driving condition to the extreme. It is noticeable that for most of the track, namely in turns 1, 2, and 3, the steering is overloaded, creating great difficulties for this control architecture. The lateral error of the Microcar (fig. 3.20b) is bigger than the previous case (3.19b) at the position located around 70 % of the completed reference trajectory, reaching a value between 20 and 25 [cm]. For the rest of the route the values remain similar implying similar performances. Furthermore, a difference in the evolution of the lateral error around 35 % of the completed reference trajectory, and a significant difference between the trajectory yaw rate and Microcar yaw rate at $4.5\,[s]$ highlights two interesting situations which should be studied deeper to improve control performance. It is possible that the model uncertainties already discussed previously modify the performance of the external compensator. In this case, the Microcar, after the saturation phenomenon has moved it away from its desired position, might take longer to get back on the desired path than the time actually required in simulation where parameters are perfectly accurate. This could cause a shift in the trajectory and a

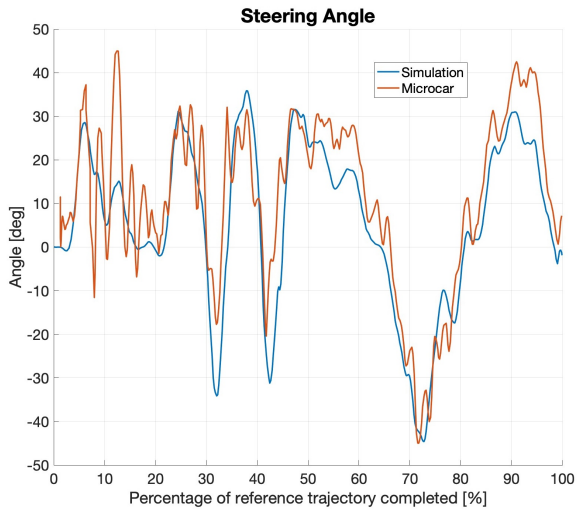
different and out of phase evolution of the lateral error.
Overall, the results from this study show similar behavior to the longer-path test, and although certain situations are more accentuated due to these particular conditions, the same conclusions can be drawn regarding steering angle behavior and overall vehicle control. This demonstrates repeatability in performance and robustness, as we have just seen that even in extreme situations, which were not considered and included in the study of this control architecture, the vehicle demonstrated stability and similar performance.



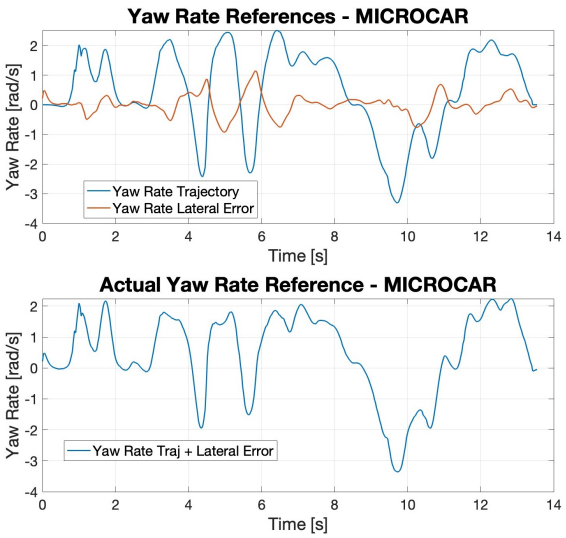
(a) Trajectory



(b) Lateral error

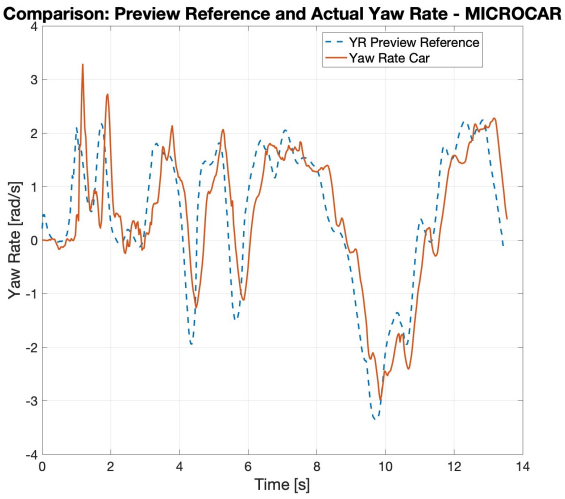


(c) Steering angle

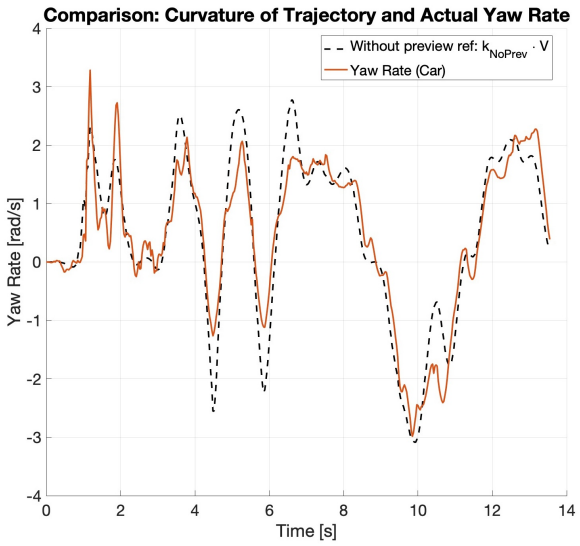


(d) YR traj VS YR e_cog

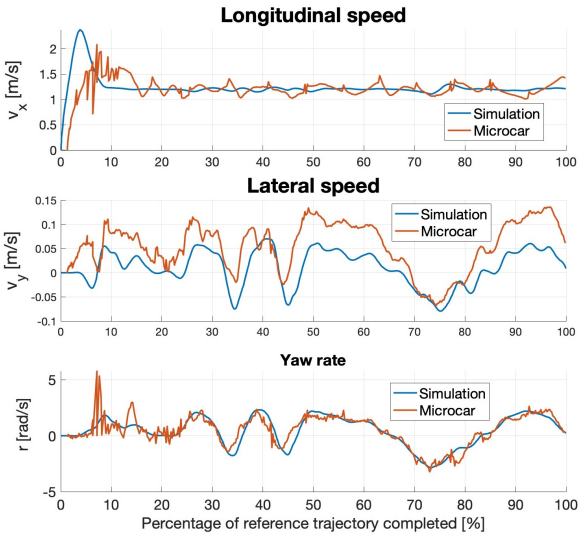
Figure 3.19: Longer path - Simulation and Microcar Test results (1/2)



(e) Actual YR VS Preview YR ref

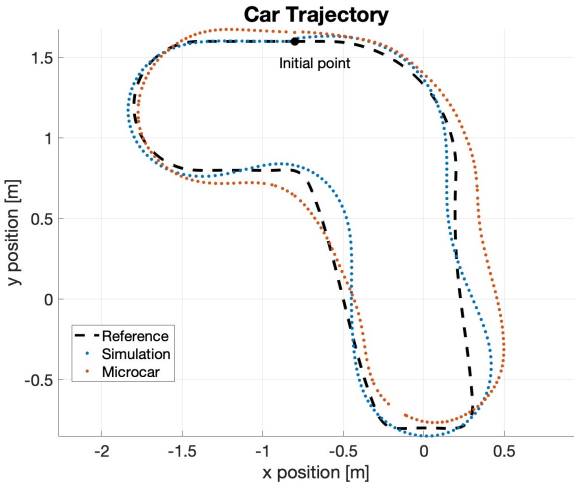


(f) Actual YR VS w/o preview YR ref

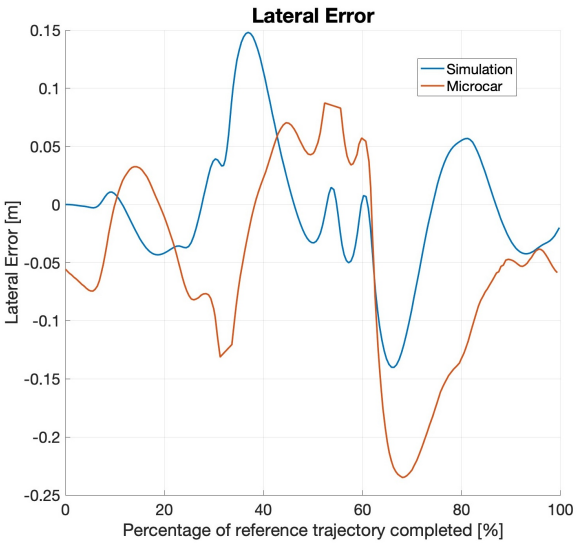


(g) Speed profile

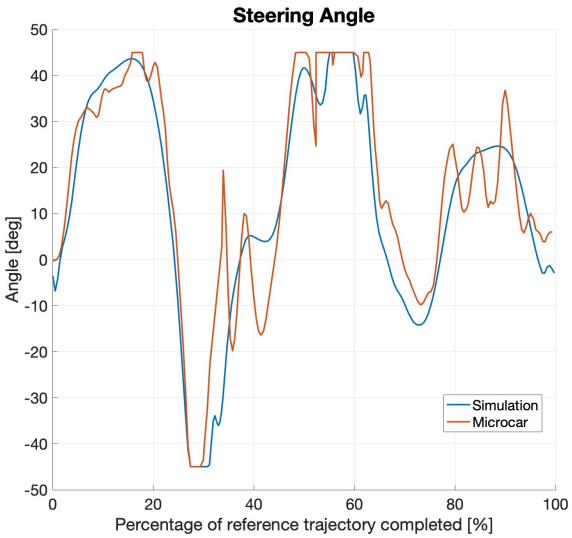
Figure 3.19: Longer path - Simulation and Microcar Test results (2/2)



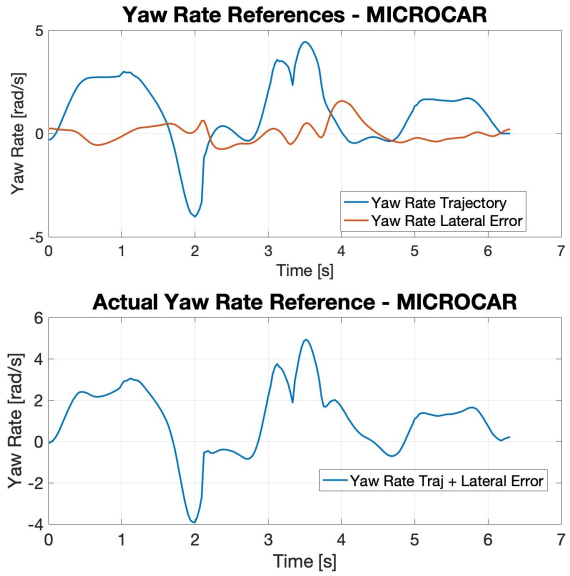
(a) Trajectory



(b) Lateral error

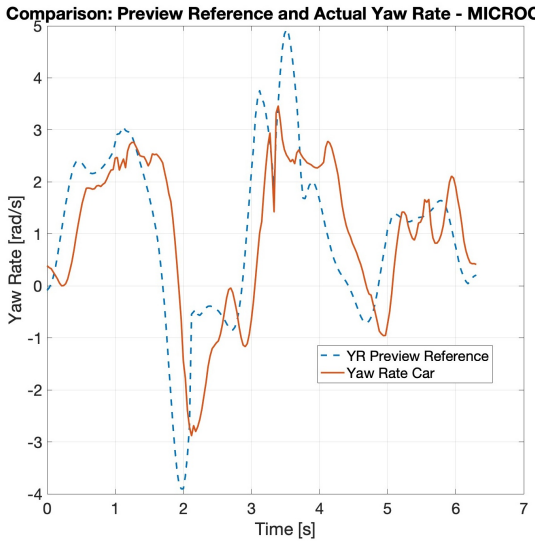


(c) Steering angle

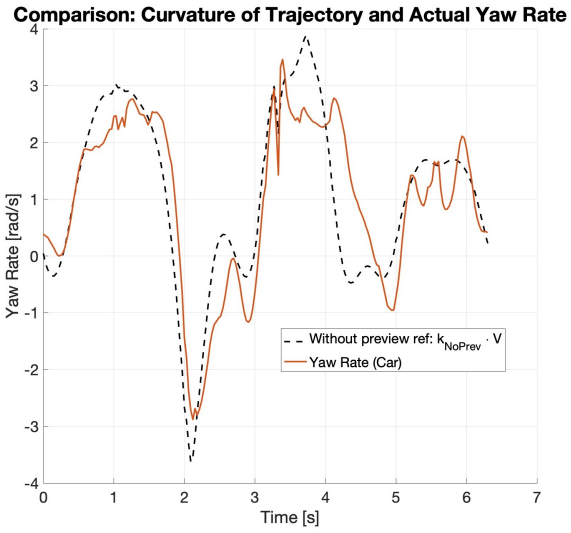


(d) YR traj VS YR e_cog

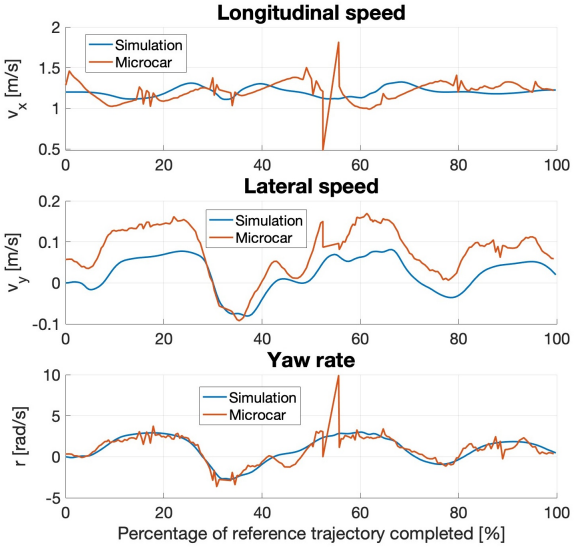
Figure 3.20: Shorter path - Simulation and Microcar Test results (1/2)



(e) Actual YR VS Preview YR ref



(f) Actual YR VS w/o preview YR ref



(g) Speed profile

Figure 3.20: Shorter path - Simulation and Microcar Test results (2/2)

4 | Robustness analysis

In this final chapter, we aim to quantify, through a robustness analysis, the ability of non-optimized preview control to preserve the stability in the face of uncertainty. As described in [21], all mathematical models, no matter how accurate they are, differ from real systems. It is therefore necessary to verify that the control techniques designed and tuned on specific mathematical models actually remain stable once they are used in reality. In chapter 3, we showed that the control architecture, tuned on a linear model, makes the system stable whether its dynamics are simulated by a non-linear model, or whether it is verified in reality through tests conducted on the platform. Now, we intend to quantify how large the uncertainty set can be for the system to remain stable.
The chapter begins by introducing the problem from a theoretical perspective, taking advantage of information collected from [8, 21–23]. It then focuses on quantifying the maximum acceptable time-delay variation for the inner loop to remain stable. It uses an unstructured uncertainty model and focuses on the Smith predictor by showing a strategy to represent it without running into algebraic problems. Finally, the robustness analysis extends to the entire system. Uncertainties in the tire stiffness coefficients are included, along with the time-delay one, and they are collected in the uncertainty matrix through a structured representation.

4.1. Theoretical overview

In order to begin a robustness analysis, it is necessary to distinguish, as just mentioned, the actual real system of the vehicle that has to be investegated and is not perfectly known, hereafter named $H(j\omega)$, from the approximated mathematical model used to tune the regulators, named $G(j\omega)$.
Quantifying the maximum error, for any value of omega:

$$|H(j\omega) - G(j\omega)| < r_{max} \quad \forall\omega$$

it is possible to find a stable set Δ which takes this uncertainty into account.
Through a mathematical model, called an uncertainty model, it is possible to use the

uncertainty Δ to describe $H(s)$ starting from the nominal model $G(s)$. Among the various existing models, the multiplicative one is used, and is described according to the definition in [21, 23]

$$H(j\omega) \in \mathcal{H}(\omega) \subset G(j\omega)\,(I + W(j\omega)\mathbf{\Delta}_c) \quad \forall\omega \in \mathbb{R} \cup \{\infty\} \tag{4.1}$$

where

- G(s) is a real rational proper transfer function
- $\mathbf{\Delta}_c$ is the open unit disk around 0: $\mathbf{\Delta}_c \coloneqq \{\Delta_c \in \mathbb{C} \mid |\Delta_c| < 1\}$
- $W(s)$ is a real rational weighting function.

In this definition, the idea is to say that $H(j\omega)$ is not known but it certainly belongs to the set $\mathcal{H}(\omega)$ modeled through the uncertainty $W(s)\mathbf{\Delta}_c$, in order to subsequently verify that the control architecture used is robust, according to the definition of robustness stability. Referring to [23], it means ensuring that the system remains stable for all perturbations in the uncertainty set.

$\Delta(s) = W(s)\mathbf{\Delta}_c$ is defined as a disk-shaped region around the nominal model. This is the only conservatism that is introduced by the uncertainty model, indeed the original uncertainty region created by varying the uncertain parameters along their uncertainty range can have complicated shapes and complex mathematical descriptions. For this reason, at each frequency, it is approximated by a larger disc-shaped region with maximum radius of

$$l_I(\omega) = \max_{H(j\omega)\in\mathcal{H}} \left| \frac{H(j\omega) - G(j\omega)}{G(j\omega)} \right| \tag{4.2}$$

and $W(s)$ is chosen in order to be:

$$W(j\omega) \geq l_I(\omega) \quad \forall\omega \tag{4.3}$$

Before continuing, it is interesting to note that the definition 4.1 considers $|\Delta_c| < 1$ but in other books, as in [23], a closed set $|\Delta_c| \leq 1$ is used to perform the analysis.

Stability of the system in figure 4.1 can be proved in different ways, in [23] are distinguished three different cases, all of these are based on the Nyquist stability condition. All referenced books focus on the $M\Delta$-structure, a general analysis that uses an alternative structure of the system. It is based on the block scheme consisting of Δ, that differs from the previous one and can be identified as Δ_c, and on the block M. M is the transfer

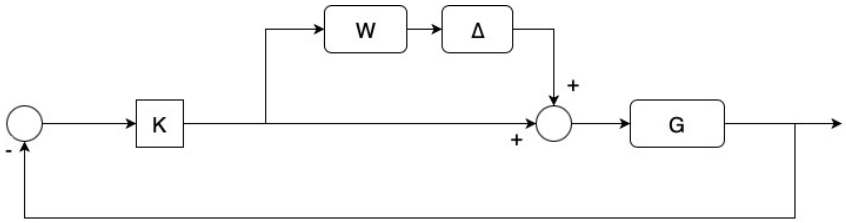


Figure 4.1: Closed-loop system with multiplicative uncertainty

function seen by the uncertainty:

$$M = WK(1 + GK)^{-1}G = WT \tag{4.4}$$

where K is the regulator that guarantees the stability of the nominal closed-loop system, and T is the complementary sensitivity transfer function. Note that the weighting function, which defines the width of the disk-shaped uncertainty region, is included in M while Δ is defined to be < 1.

Before showing the stability condition it is also necessary to study the $PK\Delta$-structure:

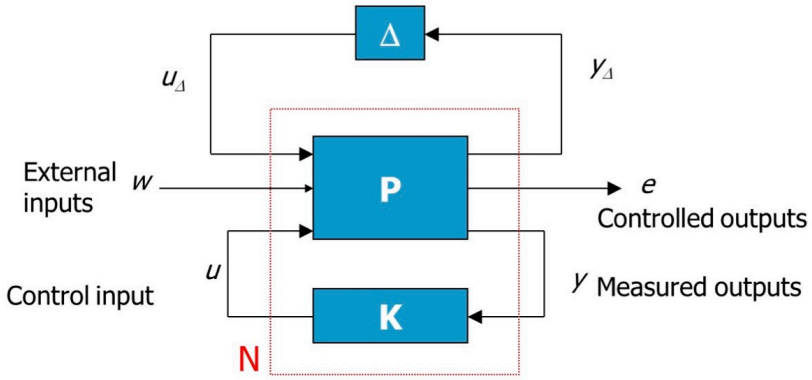


Figure 4.2: $PK\Delta -$ structure, by [22]

where P is the general plant that takes as input:

- u_Δ the output of the uncertainty Δ, with $|\Delta| < 1$
- w where are collected all the external inputs of the system
- and u that is the control variable, id est the output of the controller K.

P gives an output of:

- y_Δ, that is the input to the uncertainty matrix Δ

- the controlled output of the system e
- and the output variables of the system y that are measured and provided in feedback to the controller.

If N is defined as the interconnection of the plant with the controller, the system can be reformulated as:

$$\begin{bmatrix} y_\Delta \\ e \end{bmatrix} = \begin{bmatrix} M & N_{12} \\ N_{21} & N_{22} \end{bmatrix} \begin{bmatrix} u_\Delta \\ w \end{bmatrix} \tag{4.5}$$

where $M = N_{11}$, thus obtaining the $M\Delta$-structure described before.

The following theorem, described and analyzed in [21], is fundamental to robust stability analysis:

Theorem 4.1. *If K stabilizes P, and if*

$$\det\left(I - M(j\omega)\Delta_c\right) \neq 0 \qquad \forall \Delta_c \in \mathbf{\Delta}_c,\ \omega \in \mathbb{R} \cup \{\infty\} \tag{4.6}$$

then K robustly stabilizes $S(\Delta, P)$ against Δ.

Unstructured case

Theorem 4.1 is a sufficient condition, and in case of a general uncertainty matrix without a particular structure (but bounded <1) the condition 4.6 can be more easily verified by using the small gain theorem where:

$$\det\left(I - M(j\omega)\Delta_c\right) \neq 0 \tag{4.7}$$

is (sufficiently) equivalent to the inequality:

$$\|M(j\omega)\|\|\Delta_c\| < 1 \tag{4.8}$$

thus, it is sufficient to check if ([21]):

$$\|WT\| < 1 \tag{4.9}$$

where W is the weighting function and T the complementary sensitivity transfer function.

Structured case

If there are multiple uncertainties, it is possible to limit the conservatism discussed above by giving the Δ matrix a structure. Instead of limiting the entire matrix, it is possible to impose a limit on each block. The Δ matrix has the following structure:

$$\Delta_c = \begin{bmatrix} p_1 I & & & & & & & & 0 \\ & \ddots & & & & & & & \\ & & p_{n_r} I & & & & & & \\ & & & \delta_1 I & & & & & \\ & & & & \ddots & & & & \\ & & & & & \delta_{n_c} I & & & \\ & & & & & & \Delta_1 & & \\ & & & & & & & \ddots & \\ 0 & & & & & & & & \Delta_{n_f} \end{bmatrix} \in \mathbb{C}^{p \times q} \tag{4.10}$$

- $p_j \in \mathbb{R} \quad \text{with} \quad |p_j| < 1, \quad j = 1, \dots, n_r$
- $\delta_j \in \mathbb{C} \quad \text{with} \quad |\delta_j| < 1, \quad j = 1, \dots, n_c$
- $\Delta_j \in \mathbb{C}^{p_j \times q_j} \quad \text{with} \quad \|\Delta_j\| < 1, \quad j = 1, \dots, n_f$

where $p_j I,\ j = 1, \dots, n_r$ represents real parametric blocks, $\delta_j I,\ j = 1, \dots, n_c$ are complex parametric blocks, and $\Delta_j,\ j = 1, \dots, n_f$ are full complex uncertainty blocks.

Now, it is considered the $M\Delta$-structure with structured uncertainties $\Delta \in \Delta_c$. The theory uses the structured singular value to get necessary and sufficient conditions for robust stability. In particular, [22, 23] states that the structured singular value is the smallest structured Δ which makes $I - M\Delta$ singular.

Definition 4.1.1. *The structural singular value (SSV) of the matrix M with respect to the set Δ_c, with $M(s)$ a complex matrix and Δ_c, as defined before, a diagonal matrix with $\bar{\sigma}(\Delta) < 1$, $\forall \Delta \in \Delta_c$, is defined as:*

$$\mu_{\Delta_c}(M) \coloneqq \frac{1}{\min\{\bar{\sigma}(\Delta) \,|\, \Delta \in \Delta_c,\ \det(I - M\Delta) = 0\}}, \qquad \text{for } M \in \mathbb{C}^{q \times p}$$

The SSV is used in the following theorem to prove robust stability:

Theorem 4.2. *Assume that $M(s)$ is a MIMO LTI stable system and $\Delta(s)$ is an LTI uncertain stable matrix ($\Delta \in \mathcal{RH}_\infty$).*

Then the $M\Delta$-structure is stable $\forall \Delta \in \Delta_c$ with $\bar{\sigma}(\Delta) < 1$ if and only if:

$$\mu_{\Delta_c}(M(j\omega)) \leq 1, \qquad \forall \omega \in \mathbb{R}$$

The computation of an exact SSV is a really complicated problem. The computational method for the μ-analysis provides a lower (γ_2) and an upper bound (γ_1) for the SSV. They can be interpreted according to the following theorem ([22]):

Theorem 4.3. *Assume that $M(s)$ and $\Delta(s)$ are stable. Then, $\forall \Delta \in \Delta_c \mid \|\Delta\|_\infty < 1/\gamma$ the $M\Delta$-structure is stable if and only if:*

$$\mu_{\Delta_c}(M(j\omega)) \leq \gamma, \qquad \forall \omega \in \mathbb{R} \tag{4.11}$$

If γ^* is the smallest value of γ for eq. (4.11) to hold, it is clear that γ^* is the upper bound of μ_{Δ_c}. Thus, $\gamma_1 = \gamma^*$ gives a guaranteed but conservative limit to Δ_c. It indirectly tells how large the structured uncertainties can grow to still guarantee robustness stability ([21]). The lower bound is instead the value corresponding to the first time the calculator finds the system unstable.
In robustness analysis, as in the command `robstab()` of the Robust Control Toolbox in Matlab, the concept of robust stability margin is often used. It is defined in [22] as:

Theorem 4.4. *The robust stability margin k_r is defined as the inverse of the largest value of γ according to Theorem 4.3. Then:*

$$k_r = \frac{1}{\sup_\omega \mu_{\Delta_c}(M(j\omega))}$$

4.2. Delay uncertainty in the unstructured case

The following analysis will study the stability of the Smith predictor in the face of time-delay uncertainties. It is well known that the Smith predictor is sensitive to model uncertainties. For this reason, it is useful to quantify what is the maximum uncertainty width that preserves the stability of the entire system. This initial analysis focuses specifically on the inner loop, because for the whole architecture stability it is fundamental that the yaw-rate loop is stable.

The vehicle model G_0 is obtained by connecting in series the transfer function of the actuator G_{act} and of the yaw-rate G_r that were defined in chapter 1.3. $G_{rational}$ is defined

as the rational part of G_0. At this point, the nominal model representation is the following:

$$G_0 = G_r \cdot G_{act} = G_{rational} \cdot e^{-\tau s} \tag{4.12}$$

If we include an uncertainty to the time delay, that we call δ and that considers the time variation from its nominal value, it is possible to write the uncertain model as:

$$G_u = G_{rational} \cdot e^{-(\tau+\delta)s} = G_{rational} \cdot e^{-\tau s} \cdot e^{-\delta s} = G_0 \cdot e^{-\delta s} \tag{4.13}$$

and if we represent G_u as an input multiplicative uncertainty model, according to equation 4.1, it becomes:

$$G_u = G_0 \cdot e^{-\delta s} = G_0(1 + W \cdot dG) \tag{4.14}$$

where $e^{-\delta s} = 1 + W \cdot dG$, while dG is the uncertainty matrix bounded in norm by 1, and W is the weighting function. The choice of W is crucial and it is tuned according to equation 4.3: it is the upper-bound of all the realizations of the uncertainty ([22]). Therefore:

$$\left| \frac{\tilde{G}(j\omega) - G_0(j\omega)}{G_0(j\omega)} \right| \leq W(j\omega) \qquad \forall\, \omega,\ \forall\, dG \tag{4.15}$$

where $\tilde{G}$ is a particular realization of G_u.

Calling Rr the rational regulator of the Smith-predictor defined according to the transfer function 2.3 and tuned using the "classical approach", it has been possible to build the inner-loop system using the `sysic` tool [22]. The following code shows the strategy adopted to get the complementary sensitivity transfer function T_r of the yaw-rate closed loop:

```
Delay = exp(-tau*s);
systemnames = 'Rr Delay G_rational G0';
inputvar = '[r]';
outputvar = '[G0]';
input_to_G0 = '[Rr]';
input_to_Rr = '[r-G0-G_rational+Delay]';
input_to_Delay = '[G_rational]';
input_to_G_rational = '[Rr]';
sysoutname = 'Tr';
cleanupsysic = 'yes';
sysic;
```

where `Delay` $= e^{-\tau s}$ is the time delay transfer function. This representation shows a way

to build the Smith predictor model without running into algebraic problems.

Recalling inequality 4.15, it is possible to express $W(s)$ as a function of a value σ_{max} and therefore $\tau_{max} = \tau + \sigma_{max}$ such that:

$$\left| \frac{e^{-(\tau+\delta)s} - e^{-\tau s}}{e^{-\tau s}} \right| \leq \left| \frac{e^{-\tau_{max} s} - e^{-\tau s}}{e^{-\tau s}} \right| = W(s) \qquad \forall\, \omega,\ \forall\, \delta \tag{4.16}$$

The objective is now to find the higher value of τ_{max} such that the small-gain theorem guarantees stability, id est equation 4.9 holds:

$$|W(s) \cdot T_r(s)| \leq 1 \tag{4.17}$$

The value is looked for through an iterative process in which at each step τ_{max} is increased, and a bode plot is shown to check if the magnitude of $W \cdot T_r$ is lower than 1. Note that $W(s)$ is approximated by a 4th-order Padé model in order to make the function rational. A value of $\tau_{max} \approx 1.4\,\tau$ has been found and verified with the Bode plot of fig. 4.3. It shows that the yaw-rate loop with its Smith-predictor is robust up to delay variations of 40% of its nominal value. It should be remembered that τ_{act} was estimated considering all the time required for the control command to be actually applied. With a view to improvement it would be interesting to estimate the variability of this parameter during the operation of the Microcar, it would give an estimation of how much actual margin is currently available for this control architecture.

4.3. Delay and tire stiffness-coefficient uncertainties in the structured case

In this final analysis, we attempt to study the stability of the entire system in the face of uncertainties regarding the time delay together with the stiffness coefficients of the tires. The latter play an important role in the vehicle model, their uncertainty could significantly alter its dynamic behavior. In this case, since we have several uncertainties to consider in a single set, we structure the matrix with a diagonal shape as described in eq. (4.10). As previously mentioned, in this way it is possible to set a constraint directly on each uncertain parameter and avoid obtaining overly conservative results.

The uncertainty of each parameter is set arbitrarily to be bounded in norm by 10% of its nominal value:

```
tau_u = ureal('tau',param_car.tau,'Percentage',10);
```

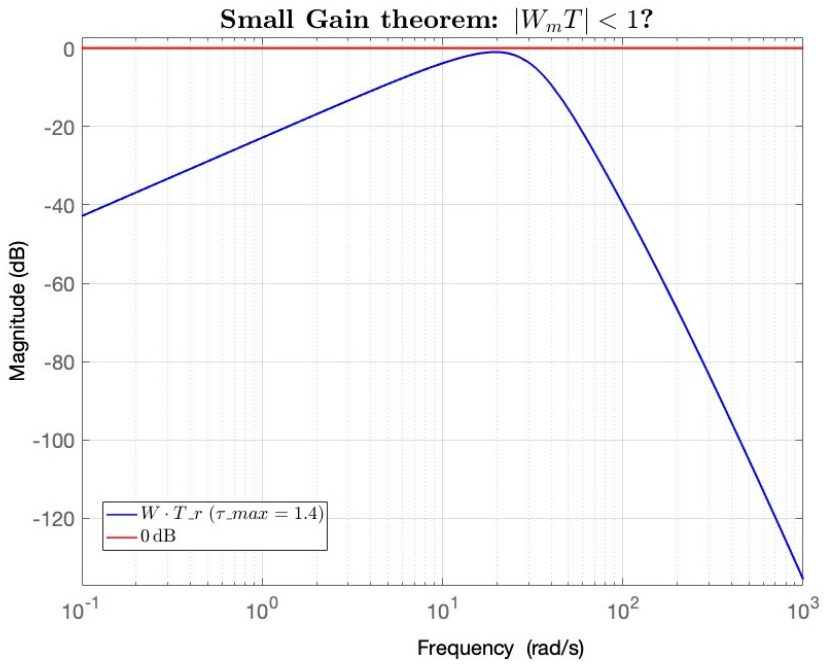


Figure 4.3: Bode plot: Small-gain theorem with $\tau_{max} = 1.4$

```
C_alpha_f2_u = ureal('C_alpha_f2',param_car.C_alpha_f2,'Percentage',10);
C_alpha_f1_u = ureal('C_alpha_f1',param_car.C_alpha_f1,'Percentage',10);
C_alpha_f0_u = ureal('C_alpha_f0',param_car.C_alpha_f0,'Percentage',10);
C_alpha_r2_u = ureal('C_alpha_r2',param_car.C_alpha_r2,'Percentage',10);
C_alpha_r1_u = ureal('C_alpha_r1',param_car.C_alpha_r1,'Percentage',10);
C_alpha_r0_u = ureal('C_alpha_r0',param_car.C_alpha_r0,'Percentage',10);
```

with the command `ureal` it is possible to create uncertain real values for each parameter, and then the tire stiffness coefficients are employed by the linear model in the same way as described in chapter 1:

```
% Parameter definition - Cornering and long tire stiffness:
C_alpha_f_u = (C_alpha_f2_u*V^2+C_alpha_f1_u*V+C_alpha_f0_u);
C_alpha_u = (C_alpha_r2_u*V^2+C_alpha_r1_u*V+C_alpha_r0_u);
```

$C_{\alpha f-u}$ and $C_{\alpha r-u}$ are therefore used to obtain the yaw-rate G_{r-u} and sideslip rate $G_{\dot{\beta}-u}$ transfer function using exactly the same procedure of chapter 1. G_{r-u} and $G_{\dot{\beta}-u}$ are uncertain models whose nominal value is identical to eqs. (1.28) and (1.29). The actuator model is defined using the same rational part of the actuator transfer function described in equation 1.27, while the delay term is modeled by a Padé approximation of the 2th order using the uncertain parameter `tau_u`. In this case, the command `pade(tau_u,2)` is not usable due to uncertainty, so the approximant has been described analytically according

to [13]:

$$e^{-\tau_u s} = \frac{1 - \frac{\tau_u s}{2} + \frac{\tau_u^2 s^2}{12}}{1 + \frac{\tau_u s}{2} + \frac{\tau_u^2 s^2}{12}} \tag{4.18}$$

The interconnected system is built according to the scheme already seen in Figure 1 using the `sysic` tool:

```
% Generalized plant
systemnames = 'Re R Delay G_rational Gact_u Gr_u Gbdot_u G_ecog';
inputvar = '[r]';
outputvar = '[G_ecog]';
% Build inner loop model
input_to_Gact_u = '[R]';
input_to_Gr_u = '[Gact_u]';
input_to_Gbdot_u = '[Gact_u]';
% Build Smith-predictor
input_to_R = '[r+Re-Gr_u-G_rational+Delay]';
input_to_Delay = '[G_rational]';
input_to_G_rational = '[R]';
% Build outer-loop model
input_to_G_ecog = '[Gr_u+Gbdot_u]';
% Build Lead-lag
input_to_Re = '[-G_ecog]';
sysoutname = 'CLu';
cleanupsysic = 'yes';
sysic;
```

where R_e is the Lead-Lag regulator, R_r is the rational term of the Smith predictor and is used together with the terms Delay and $G_{rational}$ to create the Smith predictor. G_{ecog} is a transfer function defined as V/s^2.

The stability of the system `CLu` is verified by the Matlab command `robstab()`:

```
[stabmarg,wcu,infors] = robstab(CLu,opts)
```

It computes the robust stability margin according to Theorem 4.4, and provided the following result:

```
Computing bounds... Points completed: 47/47
Computing peak...  Percent completed: 100/100
System is robustly stable for the modeled uncertainty.
 -- It can tolerate up to 301% of the modeled uncertainty.
 -- There is a destabilizing perturbation amounting to 305% of the modeled uncertainty.
```

```
 -- This perturbation causes an instability at the frequency 18.2 rad/seconds.
 -- Sensitivity with respect to each uncertain element is:
      15% for C_alpha_f0. Increasing C_alpha_f0 by 25% decreases the margin by 3.75%.
      19% for C_alpha_f1. Increasing C_alpha_f1 by 25% decreases the margin by 4.75%.
      2% for C_alpha_f2. Increasing C_alpha_f2 by 25% decreases the margin by 0.5%.
      0% for C_alpha_r0. Increasing C_alpha_r0 by 25% decreases the margin by 0%.
      0% for C_alpha_r1. Increasing C_alpha_r1 by 25% decreases the margin by 0%.
      14% for C_alpha_r2. Increasing C_alpha_r2 by 25% decreases the margin by 3.5%.
      71% for tau. Increasing tau by 25% decreases the margin by 17.8%.

stabmarg =
    LowerBound: 3.0064
           UpperBound: 3.0467
    CriticalFrequency: 18.2219

wcu =
    C_alpha_f0: -1.3758
    C_alpha_f1: 8.1274
    C_alpha_f2: -0.3034
    C_alpha_r0: -3.7839
    C_alpha_r1: 8.7694
    C_alpha_r2: 2.1786
           tau: 0.1264
```

The results indicate that the lower bound guarantees stability up to 301% of the model uncertainty. Since the uncertainty for each parameter is bounded in norm up to 10% of its nominal value, the system remains stable even if the value of each parameter changes by $\frac{301}{100} \cdot \frac{10}{100} = 30.1\%$. Compared to what has previously been inferred from Theorem 4.3, now it is the lower bound to provide guarantee of stability because the robust stability margin is the inverse with respect to the SSV. Figure 4.4 reports a comparison between a nominal simulation and a perturbed one, in which all the parameters have been altered by approximately 30% of their nominal values. In this condition, the vehicle performance deteriorates even though its run remains stable, thus supporting the robustness analysis.

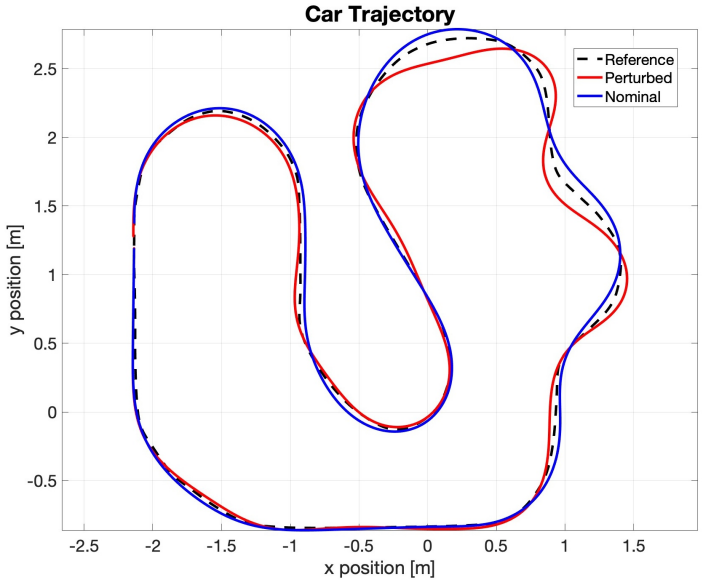
Instead, the upper bound shows that with a 30.5% variation the system is unstable, and the corresponding values for which this instability occurred are summarized in `w_cu`.

From the sensitivity results it emerges that the delay uncertainty is the one that most influences the stability of the entire system. It has a 71% impact compared to the other uncertainties, which have a decidedly minor role. In any case, the latter cannot be over-

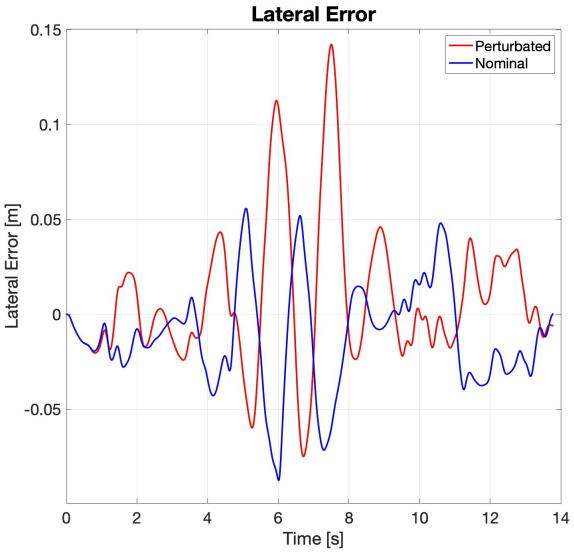
looked, especially with regard to the front wheels. In fact, the uncertainty margin of the time delay is reduced from 40% of the previous case, where only time-delay uncertainty was considered, to 30%. Furthermore, it can be noted that only some parameters that model the dynamic behavior of the wheels have a significant role on the stability of the vehicle lateral model.

However, it must also be considered that uncertainties should be set with more precise limits. In this thesis, the choice was arbitrary and dictated by the desire to demonstrate the most significant elements for stability. There was no prior, realistic analysis of the actual model uncertainties. In general, considering too many uncertainties, as well as using unrealistic ranges, could yield overly conservative results that would be too restrictive in practice. As highlighted in [23], the μ-analysis is a worst-case analysis and may include scenarios that would be unlikely to occur in practice.

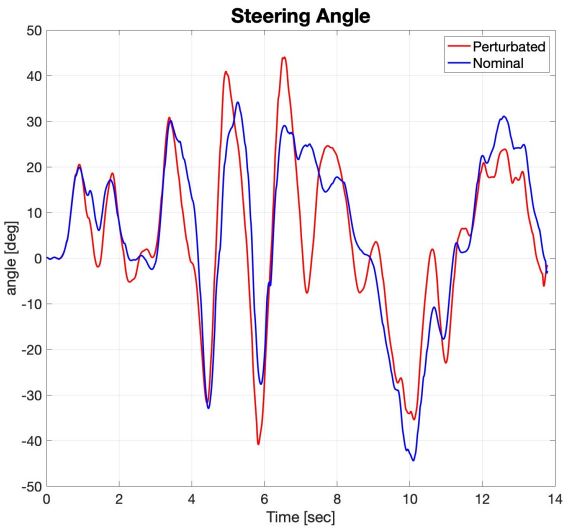
An analysis of this type is significant because it highlights the sensitive elements of the entire architecture. From a research and development perspective, it shows which parameters or models have the greatest impact on control stability. This is achieved not only qualitatively, but also effectively with numerical values. Having a numerical feedback allows us to intervene and improve both the Microcar and the platform, ensuring that the parameters always remain within the allowed range. By adapting and improving the architecture model, the numerical feedback may also allow to re-design the sensitivity percentages according to specific needs.



(a) Trajectory



(b) Lateral error



(c) Steering angle

Figure 4.4: Comparison between nominal and perturbed ($\approx 30\%$) simulations - Longer path

5 | Conclusions and future developments

The work of this thesis aims to implement A Non-Optimized Preview Control strategy on a small vehicle. As described in the introduction (section 0.2), this architecture has been developed and presented in [6] for evasive maneuvers. In this work, however, attention is focused on the performance of a vehicle following a complete path and, above all, on the critical issues that the platform and the Microcar introduce into this architecture. It is recalled in fact, as described in section 1.2.1, that vehicle control is managed entirely remotely from an external computer connected to the cameras of a motion capture system. Although this approach may simplify many aspects (such as the vehicle's hardware, the communication between the platform and vehicles) and make the platform more versatile for different research objectives, it introduces a significant actuator time delay. In section 1.2.2 it is shown that the Microcar's actuator time-delay estimate is just under 2 tenths of second. It is a significant value that can compromise the performance and stability of the control.

It has been shown that the choice of using a Smith predictor is functional. In theory, it allows both improving the performance and calibrating the preview term on the reference in a more precise, analytical and simple way. It does not need any optimization procedure, but only an accurate estimate of the time-delay.

On a practical level, both in simulation and in Microcar running tests, satisfactory results have been obtained (chapter 3), in particular on the lateral error when testing the vehicle on articulated and complex tracks. The control demonstrated robustness even under speed variations. For example, in a particular testing case the vehicle started from standstill rather than at the equilibrium speed V, but comparable results were obtained. The greatest limitation was caused by steering angle saturation, where the control architecture was not designed to predict this behavior. The Microcar inevitably deviates from the desired trajectory, increasing the lateral error. During the test, the external controller proved capable of intervening and returning the vehicle to the original track, although

it was not designed to compensate for saturation issues and excessively abrupt changes in lateral error. This is an issue that generally does not occur in everyday driving, but in certain specific situations (such as off-road driving or during hairpin turns or parking maneuvers) it could be critical. Furthermore, a discrepancy was observed in certain situations between the results obtained in the simulation and the actual performance of the Microcar. As discussed in chapter 3, this is a question that deserves future analysis and further study. However, some considerations and hypotheses have been put forward that might facilitate future development.

Although Smith predictor is sensitive to model uncertainties and in particular time-delay variations, these tests have shown that the system is stable and also robust to possible estimation inaccuracies. The parameter estimation was done with a slightly different Microcar than the one used during the tests, but beyond this last aspect which cannot have much scientific weight, a robustness analysis brought some interesting results (chapter 4). If the inaccuracy involves only the time-delay, and not other parameters nor the model itself (compared to the real vehicle system), a tolerability of $0.4 \cdot \tau = 0.4 \cdot 0.1818 \approx 70\,[\text{ms}]$ on the delay variation with respect to its nominal value is possible in order to still have a stable system. If the uncertainties involve other parameters or any dynamics of the model this margin decreases. Indeed, in case of tire-stiffness coefficient inaccuracies together with time-delay uncertainty (both assuming a range of 10% with respect to their nominal value), we observed that the guarantee of stability reduces to a range of 30% of their nominal value. It means a margin of $54\,[\text{ms}]$ for the entire time delay. It would be interesting to compare these results with a ping analysis of the local Wi-Fi network. Furthermore, it must be considered that the robustness analysis is based on a linearized model around its equilibrium point. Its results are approximated values and in the future, it would be interesting to implement tools that can actually test these estimates and verify their reliability.

Bibliography

[1] A. Artuñedo, M. Moreno-Gonzalez, and J. Villagra. Lateral control for autonomous vehicles: A comparative evaluation. *Annual Reviews in Control*, 57:100910, 2024. ISSN 1367-5788. doi: https://doi.org/10.1016/j.arcontrol.2023.100910. URL `https://www.sciencedirect.com/science/article/pii/S1367578823000743`.

[2] N. Birla and A. Swarup. Optimal preview control: A review. *Optimal Control Applications and Methods*, 36(2):241–268, 2015. doi: https://doi.org/10.1002/oca.2106. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/oca.2106`.

[3] P. Bolzern, R. Scattolini, and N. Schiavoni. *Fondamenti di controlli automatici*. McGraw-Hill Education, Milano, 4 edition, 2015. ISBN 978-8838668821.

[4] A. M. Borrell. *LPV Lateral Control of Autonomous and Automated Vehicles*. Phd thesis, Université Grenoble Alpes and Universitat Politècnica de Catalunya - BarcelonaTech, 2023. URL `https://theses.hal.science/tel-04842226`. NNT: 2023GRALT067.

[5] M. Buchanan. Autonomous cars and the long road ahead. *Nature Physics*, 21(1): 2–2, 2025. doi: 10.1038/s41567-024-02749-w. URL `https://doi.org/10.1038/s41567-024-02749-w`.

[6] M. Corno, A. Gimondi, G. Panzani, F. Roselli, A. Alessandretti, and S. M. Savaresi. A non-optimization-based dynamic path planning for autonomous obstacle avoidance. *IEEE Transactions on Control Systems Technology*, 31(2):722–734, March 2023. ISSN 1558-0865. doi: 10.1109/TCST.2022.3196880.

[7] F. T. Council. Robotaxis at a crossroads, 20 years after darpa. `https://www.forbes.com/councils/forbestechcouncil/2025/02/13/robotaxis-at-a-crossroads-20-years-after-darpa/`, 2025. Accessed on June 21, 2025.

[8] G. E. Dullerud and F. Paganini. *A *course in robust control theory : a convex approach / Geir E. Dullerud, Fernando Paganini*. Springer, New York [etc, 2000. ISBN 978-03-87989-45-7.

[9] Gipsa-Lab. Picture of the microcar platform, 2025. URL https://www.gipsa-lab.grenoble-inp.fr/en.

[10] G. M. Hoffmann, C. J. Tomlin, M. Montemerlo, and S. Thrun. Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing. In *2007 American Control Conference*, pages 2296–2301, 2007. doi: 10.1109/ACC.2007.4282788.

[11] D. Kapsalis, O. Sename, V. Milanes, and J. J. Molina. A reduced lpv polytopic look-ahead steering controller for autonomous vehicles. *Control Engineering Practice*, 129:105360, 2022. ISSN 0967-0661. doi: https://doi.org/10.1016/j.conengprac.2022.105360. URL https://www.sciencedirect.com/science/article/pii/S0967066122001915.

[12] U. Kiencke and L. Nielsen. *Automotive Control Systems: For Engine, Driveline, and Vehicle*. Springer, 2005. doi: 10.1007/b137654.

[13] L. Magni and R. Scattolini. *Advanced and Multivariable Control*. Pitagora, Bologna, Dec. 2014. ISBN 978-8837119058.

[14] W. F. Milliken and D. L. Milliken. *Race car vehicle dynamics*. SAE International, Warrendale, 1995.

[15] Orr Danon. As adas market advances, is autonomy still the right goal? https://www.eetimes.com/as-adas-market-advances-is-autonomy-still-the-right-goal/, 2021. News Analysis — Accessed on June 21, 2025.

[16] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on Intelligent Vehicles*, 1(1):33–55, March 2016. ISSN 2379-8904. doi: 10.1109/TIV.2016.2578706.

[17] A. Pressley. *Elementary Differential Geometry*. Springer, London, 2001. ISBN 978-1-85233-152-8.

[18] R. Rajamani. *Vehicle Dynamics and Control*. Mechanical Engineering Series. Springer, 2006. doi: 10.1007/0-387-28823-6. URL https://doi.org/10.1007/0-387-28823-6.

[19] F. Roselli, M. Corno, S. M. Savaresi, M. Giorelli, D. Azzolini, A. Irilli, and G. Panzani. H∞ control with look-ahead for lane keeping in autonomous vehicles. In *2017 IEEE*

Conference on Control Technology and Applications (CCTA), pages 2220–2225, 2017. doi: 10.1109/CCTA.2017.8062781.

[20] SAE International. Sae j3016™: Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles (april 2021 revision). `https://www.sae.org/standards/content/j3016_202104/`, 2021. Accessed on June 21, 2025.

[21] C. W. Scherer. Theory of robust control. URL `https://www.imng.uni-stuttgart.de/mst/files/RC.pdf`. MSc Course "Robust Control", MSc Course "Linear Matrix Inequalities in Control".

[22] O. Sename. Robust control of mimo systems, 2025. URL `https://oliviersename.fr/docs/robust/robust_control_2025.pdf`. MSc Course "Robust Control".

[23] S. Skogestad and I. Postlethwaite. *Multivariable feedback control : analysis and design / Sigurd Skogestad, Ian Postlethwaite*. John Wiley & Sons, Chichester, 2. ed edition, 2005. ISBN 978-04-7001-168-3.

[24] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, and P. Mahoney. Stanley: The robot that won the darpa grand challenge. *J. Field Robotics*, 23:661–692, 01 2006.

[25] M. Tomizuka. *The Optimal Finite Preview Problem and Its Application to Man-Machine Systems*. Ph.d. thesis, Massachusetts Institute of Technology, 1974.

[26] Wikipedia contributors. Curvature — wikipedia, the free encyclopedia. `https://en.wikipedia.org/wiki/Curvature`, 2024. URL `https://en.wikipedia.org/wiki/Curvature`. Accessed: 2025-06-26.

[27] Wikipedia contributors. Waymo — wikipedia, the free encyclopedia. `https://en.wikipedia.org/wiki/Waymo`, 2025. Accessed on June 21, 2025.

[28] S. Xu and H. Peng. Design, analysis, and experiments of preview path tracking control for autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 21(1):48–58, Jan 2020. ISSN 1558-0016. doi: 10.1109/TITS.2019.2892926.

List of Figures

List of Tables