



# Lane Keeping Assist Design

Master's Degree in Automotive Engineering

Loris Fonseca: s342696@studenti.polito.it

Nikoloz Kapanadze: s342649@studenti.polito.it

Maurizio Pio Vergara: s346643@studenti.polito.it

Giovanni Castagna: s333934@studenti.polito.it

a.y 2024/2025

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>LKA controller implementation</b>	<b>3</b>
2.1	PIDF . . . . .	3
2.1.1	Safe lateral distance tuning . . . . .	5
2.1.2	LQR tuning . . . . .	9
2.2	Stanley controller . . . . .	11
2.2.1	Stanley gain tuning . . . . .	11
<b>3</b>	<b>Stability test at different speeds and trajectories</b>	<b>13</b>
3.1	PIDF controller stability . . . . .	13
3.2	Stanley controller stability . . . . .	15
<b>4</b>	<b>Curvature radius misjudged by the driver</b>	<b>17</b>
<b>5</b>	<b>Conclusions</b>	<b>20</b>

# 1 Introduction

Lane Keeping Assist (LKA) is an advanced driver assistance system designed to keep the vehicle centered within the lane by detecting road boundaries and applying corrective steering actions. The system relies on a camera to identify lane markings and a feedback control strategy to determine the appropriate steering angle. This project aims to design and evaluate an LKA controller using a simplified linear vehicle model (Segel model), focusing on lateral and angular deviations from the lane center. The study begins with the implementation of a state-feedback controller (PIDF) based on the Linear Quadratic Regulator (LQR) approach. The state vector and the control input signal are deeply discussed in the following sections. Key performance metrics such as cross-track error and heading error are minimized by selecting appropriate control weighting matrices. The controller is then tested across different road curvatures and speeds to assess its stability and responsiveness under varying dynamic conditions.

## 2 LKA controller implementation

### 2.1 PIDF

After filling the MATLAB script *helperLKASetUp.m* with the data relative to the vehicle considered for this analysis and the LTI matrices of the linear Segel model, the first step of the project involves the design and implementation of a lane keeping controller using the Linear Quadratic Regulator (LQR) technique. Figure 1 shows the heading error  $e_h$  and the cross track error  $e_{ct}$  of a general vehicle which tries to follow a reference trajectory.

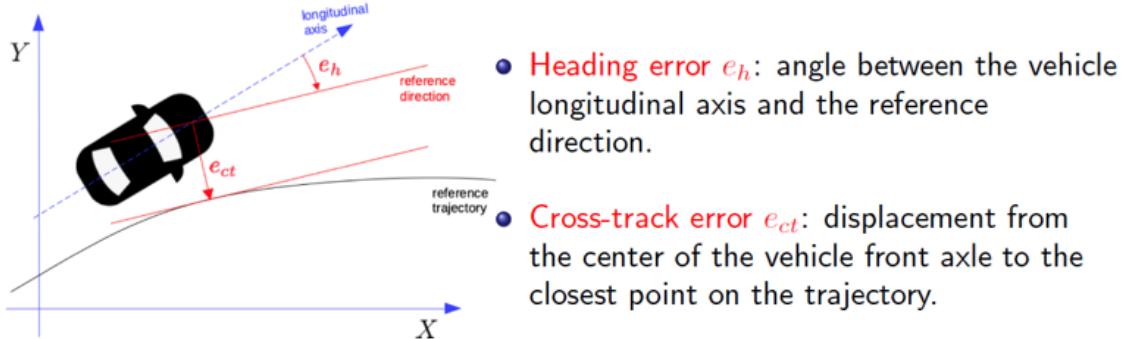


Figure 1: Heading and cross track errors

The controller is implemented using two PIDF controllers, each designed to minimize a specific error. However, since the system has only one control input, the steering angle, it is not possible to guarantee asymptotic convergence to zero for both errors simultaneously using two independent PIDFs. In fact, with a single actuation signal, only one of the two errors can be driven to zero at a time. For this reason, an LQR-based approach has been adopted to tune the two PIDFs, allowing a balanced control action and improved overall system performance. The controller structure is shown in Figure 2.

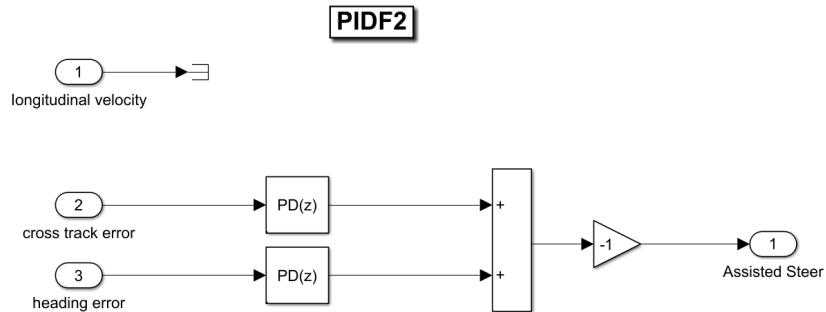


Figure 2: PIDF controller system for LKA

The PIDF control strategy combines proportional, integral, derivative, and filtering actions to shape the system response effectively. The proportional action (P) functions as a gain that scales the overall controller output in proportion to the current error. The integral action (I), when analyzed in the frequency domain using a Bode diagram, introduces a pole at the origin, resulting in a slope of  $-20$  dB/decade. This behavior is analogous to that of a low-pass filter and is essential for eliminating steady-state error, as it integrates past errors over time. In contrast, the derivative action (D) introduces a zero in the transfer function, corresponding to a  $+20$  dB/decade slope in the Bode plot. This contributes phase lead and improves the transient response by reacting to the rate of change of the error, effectively anticipating future behavior. However, the derivative term alone can lead to high-frequency amplification and noise sensitivity. For this reason, a low-pass filter is commonly applied to the derivative term to mitigate high-frequency gain. The filter is characterized by a parameter  $N$ , which determines the cutoff frequency. The value of  $N$  should be selected carefully: it must be sufficiently higher than the system's dominant poles to avoid interfering with the desired dynamics, yet not so high as to reintroduce noise amplification at high frequencies.

In order to tune both PIDF controllers the system is modeled as a linear time-invariant (LTI) system where the state vector is composed of the cross-track error  $e_{ct}$ , its derivative  $\dot{e}_{ct}$ , the heading error  $e_h$ , and its derivative  $\dot{e}_h$ . No integral action is considered, therefore the controller will be a PDF. The control input is the steering angle  $\delta$ , and an external disturbance is also considered to model environmental effects. The LTI system is the following:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{Ax}(t) + \mathbf{Bu}(t) + \mathbf{B}_d \dot{\psi}_{des}(t) \\ \begin{bmatrix} \dot{e}_{ct} \\ \ddot{e}_{ct} \\ \dot{e}_h \\ \ddot{e}_h \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2C_{\alpha f}+2C_{\alpha r}}{mV_x} & \frac{2C_{\alpha f}+2C_{\alpha r}}{m} & \frac{-2C_{\alpha f}l_f+2C_{\alpha r}l_r}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2C_{\alpha f}l_f-2C_{\alpha r}l_r}{I_z V_x} & \frac{2C_{\alpha f}l_f-2C_{\alpha r}l_r}{I_z} & -\frac{2C_{\alpha f}l_f^2+2C_{\alpha r}l_r^2}{I_z V_x} \end{bmatrix} \begin{bmatrix} e_{ct} \\ \dot{e}_{ct} \\ e_h \\ \dot{e}_h \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2C_{\alpha f}}{m} \\ 0 \\ \frac{2C_{\alpha f}l_f}{I_z} \end{bmatrix} \delta + \\ &\quad + \begin{bmatrix} 0 \\ -\frac{2C_{\alpha f}l_f-2C_{\alpha r}l_r}{mV_x} - V_x \\ 0 \\ -\frac{2C_{\alpha f}l_f^2+2C_{\alpha r}l_r^2}{I_z V_x} \end{bmatrix} \dot{\psi}_{des} \end{aligned}$$

All the parameters of the matrices  $A$ ,  $B$  and  $B_d$  are defined in the MATLAB script previously mentioned and are reported in Table 1.

Parameters	Units	Description
$m$	kg	Total mass of the vehicle
$V_x$	m/s	Longitudinal speed
$C_{\alpha f}$	N/rad	Side slip stiffness front axle
$C_{\alpha r}$	N/rad	Side slip stiffness rear axle
$l_f$	m	Distance front axle from gravity center
$l_r$	m	Distance rear axle from gravity center
$I_z$	kg·m <sup>2</sup>	Yaw moment of inertia

Table 1: Vehicle parameters

The longitudinal speed  $V_x$  is the one assigned for control design and it is a constant speed. The LQR framework minimizes the following cost function:

$$\int_0^\infty \mathbf{x}^T \mathbf{Q} \mathbf{x} + \delta^T \mathbf{R} \delta dt$$

where  $\mathbf{x}$  is the state vector,  $\mathbf{Q}$  is the state weighting matrix, and  $\mathbf{R}$  is the control weighting scalar. These matrices are selected based on physical considerations: maximum allowable deviations for lateral and angular errors, and acceptable control effort, i.e. steering angle. To tune the weighting matrices the following equations has been used:

$$\mathbf{Q} = diag([\frac{1}{e1MaxDev^2}, 0, \frac{1}{e2MaxDev^2}, 0]) \quad (1)$$

$$R = \frac{1}{deltaMaxDev^2} \quad (2)$$

where  $e1MaxDev$  represents the maximum cross track error allowable expressed in meters,  $e2MaxDev$  represents the maximum heading error allowable expressed in radians and  $deltaMaxDev$  represents the maximum steering angle as input, taking into account that typically the vehicle has a boundary limit around  $\pm 28^\circ$ . However, it is not desirable to have an angle that exceeds  $10^\circ$  for comfort reasons. If  $e1MaxDev$  is too small the vehicle could have an oscillating behavior on straight road due to a constant change of its trajectory in order to minimize the cross track error. The chosen values for defining the maximum allowable errors for this project are the following:

- $e1MaxDev = 1$
- $e2MaxDev = 7\frac{\pi}{180}$
- $deltaMaxDev = 7\frac{\pi}{180}$

Once the matrices are defined, the optimal gain matrix  $\mathbf{K}$  is computed using a MATLAB function as follow:

$$K = lqr(A, B, Q, R) \quad (3)$$

The resulting control law is a state feedback control which can be expressed as follows:

$$\delta = -Kx$$

To mitigate the amplification of the noise, a low-pass filter is often introduced in practical implementations, with a filter coefficient  $N$  carefully tuned to balance phase lead with noise attenuation. The filter coefficients  $N_1$  and  $N_2$  have been obtained with the following procedure:

$$\begin{aligned} \tau_{d1} &= \frac{K(2)}{K(1)} \\ N_1 &= \frac{10}{\tau_{d1}} \end{aligned} \quad (4)$$

$$\begin{aligned} \tau_{d2} &= \frac{K(4)}{K(3)} \\ N_2 &= \frac{10}{\tau_{d2}} \end{aligned} \quad (5)$$

Once the LQR tuning procedure is finished, it is possible to obtain the  $K$  vector of the control law previously discussed:

$$K = [0.1222, 0.0050, 1.1475, 0.0244]$$

### 2.1.1 Safe lateral distance tuning

This section presents the analysis of the results obtained through the tuning of the safe lateral distance parameter. As shown in Figure 3, this parameter is an input to the Controller block and defines the maximum lateral deviation from the lane center that triggers the activation of the Lane Keeping Assist (LKA) system. The goal of the tuning process is to identify a value that allows the controller to activate only when the vehicle is at risk of leaving the lane or is excessively close to the lane boundary, rather than continuously correcting even small deviations. At the same time, the tuning seeks to balance system responsiveness with energy efficiency. The goal is to understand if a trade off between responsiveness and energy efficiency exists, and for which safe lateral distance it happens. The most relevant variables to observe and study are the cross track and heading error, the side slip angle and the lateral acceleration, which can be also expressed in RMS. All tests were performed at a constant speed of  $22.2m/s$  on a predefined road profile featuring curves with a radius of 85 meters. The curvature radius perceived by the driver is set to match the actual road curvature, also at 85 meters, to ensure consistent testing conditions. An additional test was conducted with the LKA system completely disabled in order to highlight the differences in behavior and to assess the safety benefits offered by the control system.

Finally, a comparative test was carried out using the maximum safe lateral distance value, but with an alternative tuning of the gain vector  $K$ . This test emphasized the critical role of proper gain selection. When the gain vector is not adequately tuned, the vehicle exhibits a significantly higher side slip, confirming that optimal controller tuning is essential for maintaining both stability and comfort.

The following figures illustrate the Simulink implementation of the proposed Lane Keeping Assist control system.

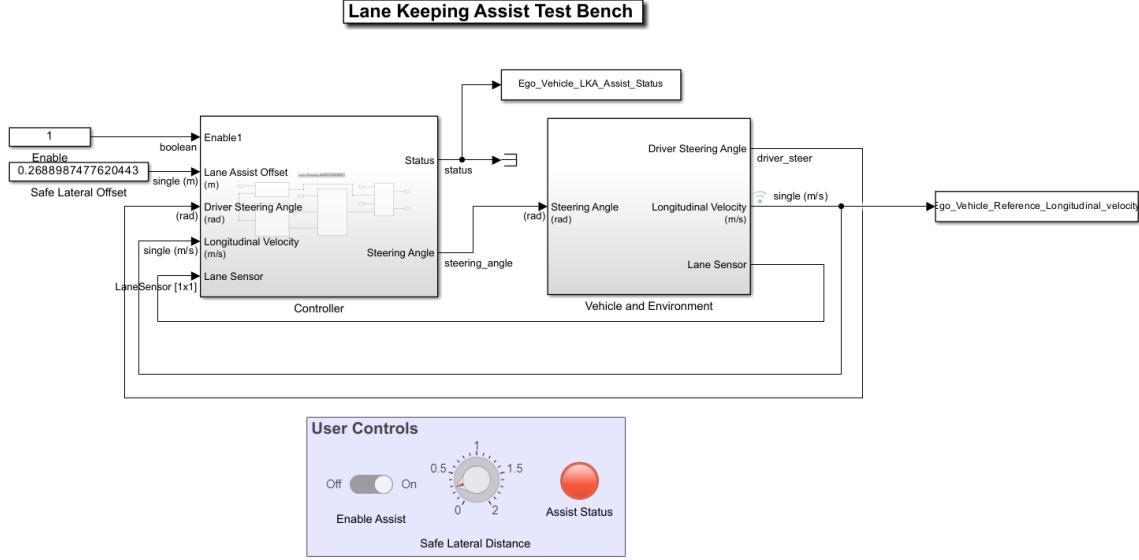


Figure 3: LKA complete model

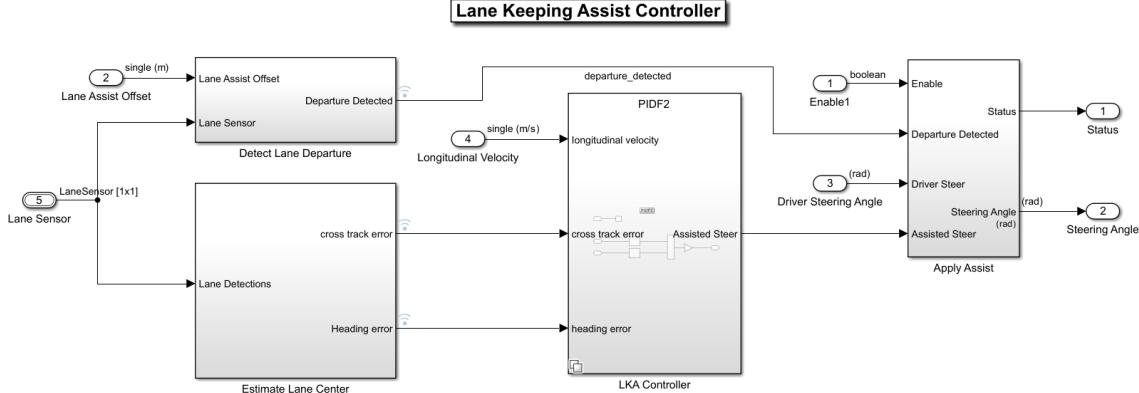


Figure 4: LKA controller implementation

In Figure 4 is possible to notice that the LKA Controller has three inputs, which are the longitudinal speed  $V_x$ , the cross track error  $e_{ct}$  and the heading error  $e_h$ . Moreover, the whole controller implementation needs some other inputs such as a lane detection which comes from the camera sensor and the lane assist offset which refers to the safe lateral distance to consider in order to activate the Lane Keeping Control over the driver. The following Figure 5 has been used as reference for the analysis because the vehicle keeps the required trajectory meanwhile cross track and heading error stay in a tight interval during the curvature, then converging to zero. The simulation has been carried out with a Safe Lateral Distance (SLD) equals to 1.75 m. The side slip angle stays within  $\pm 3^\circ$  and the lateral acceleration exploits a peak around 5 seconds, as well as the other quantities, due to the driver action during the curvature. The control system is always active during the trajectory as shown in the *Assist status* plot.

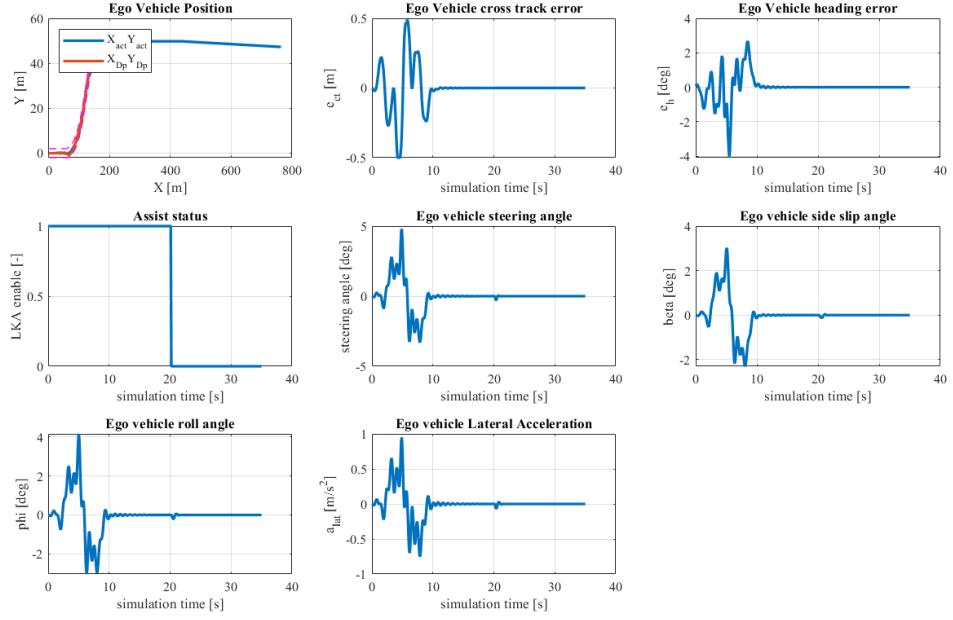


Figure 5:  $SLD = 1.75$  m

To better understand the effectiveness of the controller a comparison between two simulation is reported. In Figure 6 it is possible to notice the difference of the cross track error and the heading error that the vehicle exploits with either the Lane Keeping Assist enabled with a safe lateral distance (SLD) equals to 1.75 m or disabled.

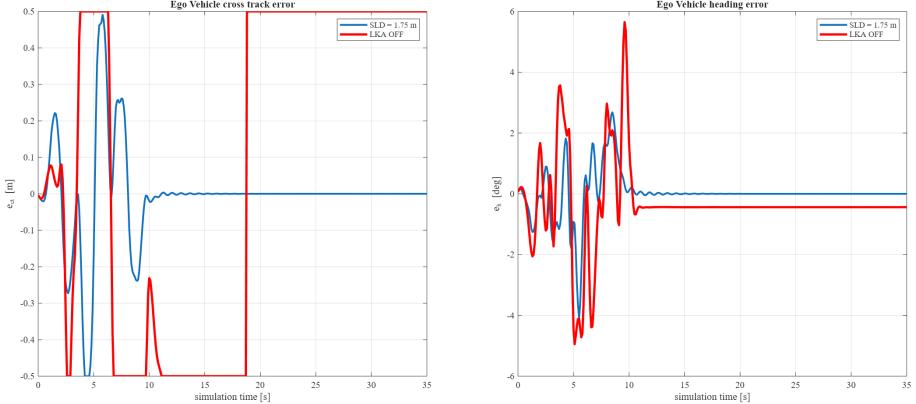


Figure 6: Comparison  $e_{ct}$  and  $e_h$ : LKA ON vs. OFF

The controller allows a great reduction of error peaks, keeping them around the zero, also converging to zero during the full operation period, right after the curvature imposed by the trajectory. It is interesting to notice that with the controller the side slip angle could be higher due to the the weighting matrices of the LQR. This happens because it has been chosen a controlled value for the steering angle. This limitation at high velocity and low radius of the curvature could be responsible of an understeering behavior of the vehicle, increasing the side slip peak. This is clearly noticeable in Figure 7. On the contrary, the controller helps  $\beta$  to converge to zero rapidly, as shown by the plot around the  $t = 10$ s, where the simulation with the LKA controller has an almost null side slip angle meanwhile the one with the controller disabled reaches a peak around  $\pm 2^\circ$ . Regarding the lateral acceleration the behavior is quite similar to the discussed for  $\beta$ , showing a robust and rapid convergence to zero.

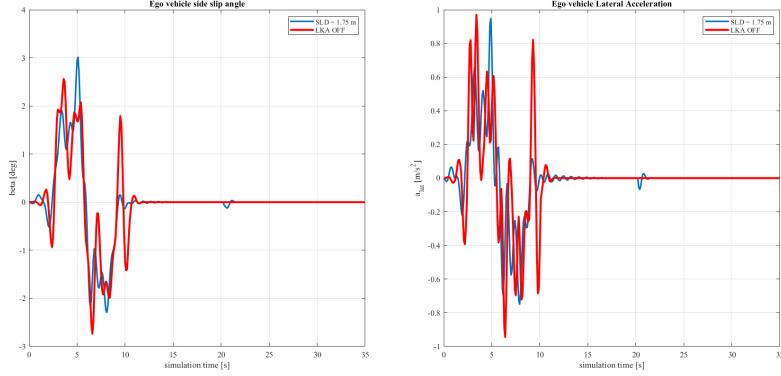


Figure 7: Comparison  $\beta$  and  $a_{lat}$ : LKA ON vs. OFF

A set of seven simulations with different Safe Lateral Distance (SLD) have been carried out and in Figure 8 it is possible to understand the behavior of the RMS value of the lateral acceleration as function of the SLD. The curve exhibits a non-linear behavior, making it essential to minimize lateral acceleration along with other key quantities such as the side slip angle, cross-track error, and heading error. This consideration has guided the selection of the Safe Lateral Distance for the LKA controller, depending on the vehicle's dynamic response. As will be discussed in later sections, at lower speeds it is possible to ensure both safety and comfort even with a reduced value of the Safe Lateral Distance. This is not always true when the vehicle is at high speed. In Figure 9 and 10 two simulation with lower SLD at same constant speed ( $V = 22.2 \text{ m/s}$ ) are reported. When the lateral distance is 1.70 m, the steering angle stays in the range of  $\pm 7^\circ$  and the side slip angle  $\beta$  in the range of  $\pm 3^\circ$ . The problem is more for the comfort rather than the safety because it is clearly noticeable in the plots regarding the errors that there is an oscillating behavior that leads to an oscillating trajectory, which is not so appropriate for the passengers. Moreover, the oscillations of the cross track error and the heading error have a non-null mean, which means that the car is not properly in the centre of the lane. In Figure 10 it is noticeable that the controller is continuously alternating the switch from ON state to OFF, as it shows the *Assist status* plot. This type of operation is not optimal because the vehicle steering angle input is continuously switching from the driver to the controller, leading to a discomfort behavior. In fact, the plots which refer to the errors show an even worse behavior with respect to the ones in Figure 9.

For that reason, it has been chosen an higher value for the safe lateral distance ( $SLD = 1.75 \text{ m}$ ) in order to guarantee the safety and responsiveness of the control even if it is not always energy optimised.

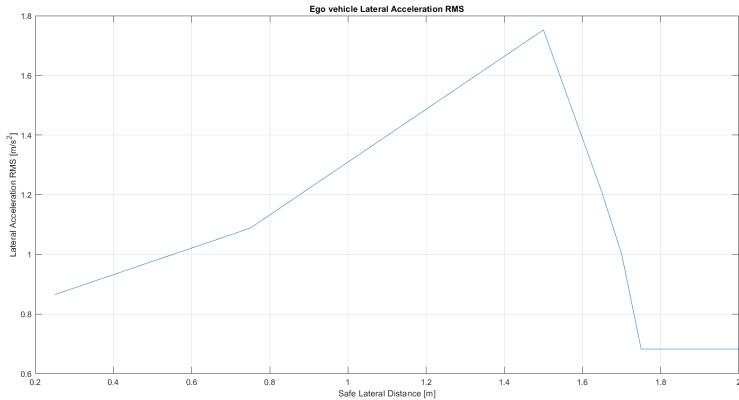


Figure 8:  $a_{lat}$  vs. SLD

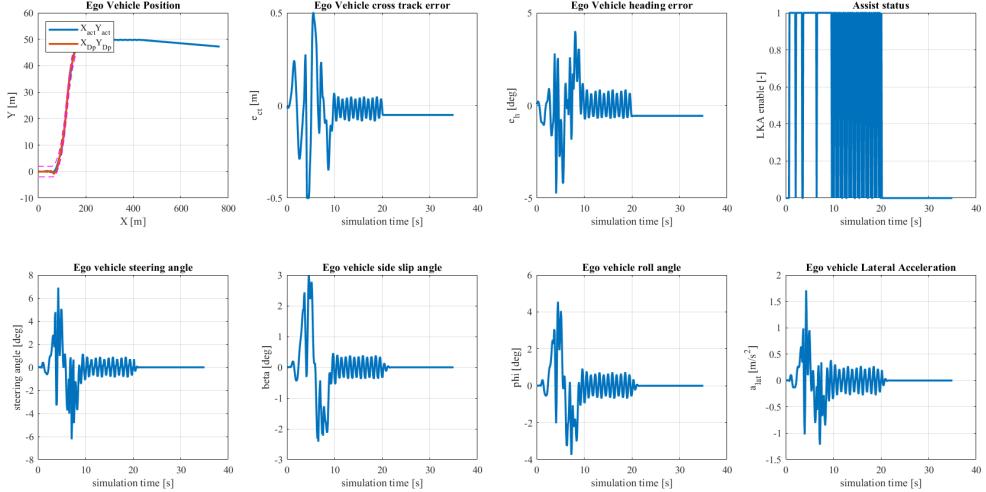


Figure 9:  $SLD = 1.70 \text{ m}$

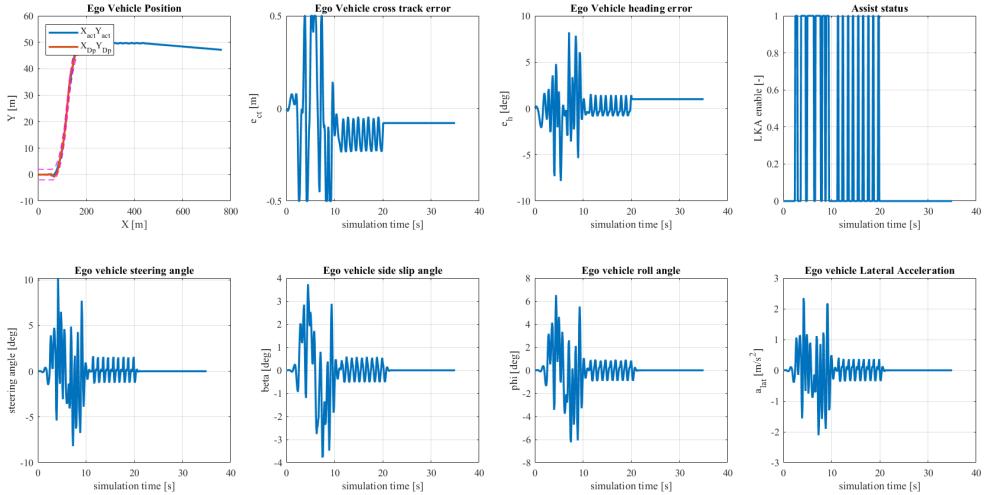


Figure 10:  $SLD = 1.50 \text{ m}$

### 2.1.2 LQR tuning

In this section the result obtained during the tuning procedure of the LQR are reported, showing the behavior of the vehicle with different weighting matrices  $\mathbf{Q}$  and  $\mathbf{R}$ , and following the same trajectory and with the same parameters and the chosen value of safe lateral distance ( $SLD = 1.75 \text{ m}$ ). Decreasing the error weight related to the heading error from  $7^\circ$  to  $5^\circ$  leads to an oscillating behavior of the vehicle not so comfortable for the passenger, as it is shown in Figure 11. In Figure 12 it is possible to notice that decreasing the error weight related to the input steering angle from  $7^\circ$  to  $5^\circ$  the vehicle exploits a good general behavior following the trajectory, but the side slip peak exceeds  $3^\circ$  and for that reason this configuration has not been chosen. In Figure 13 it is clearly shown a non proper behavior of the vehicle that must be avoided because the car does not follow the trajectory and the oscillating errors show a diverging curve. This simulation was carried out with a higher error weight of the steering angle, in particular  $deltaMaxDev = 10^\circ$ .

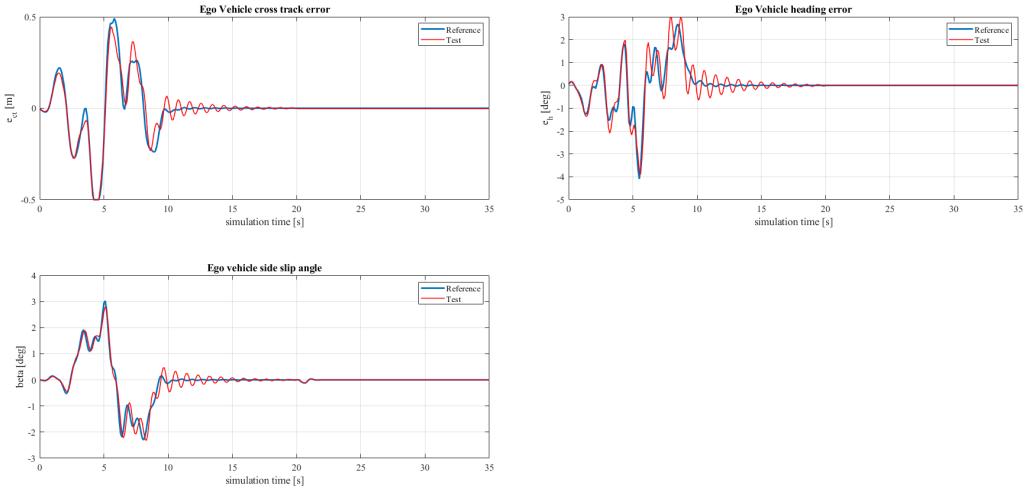


Figure 11: Reference K vs lower heading error weight

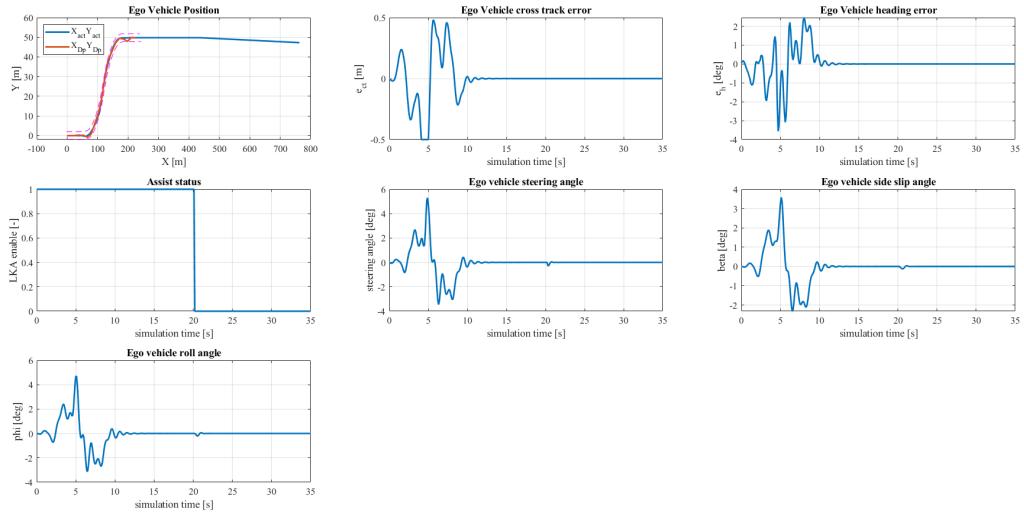


Figure 12: Low steering angle error weight

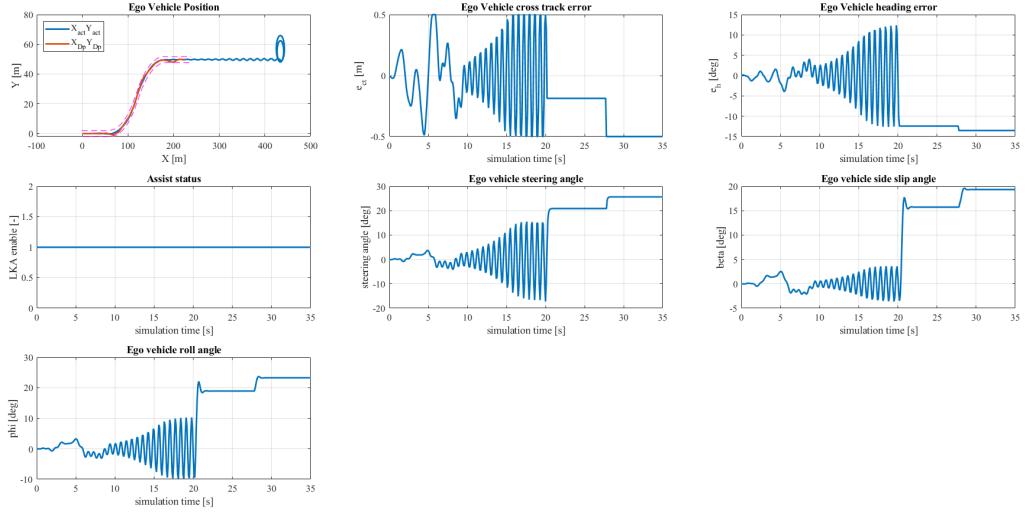


Figure 13: High steering angle error weight

## 2.2 Stanley controller

The Stanley controller is a widely adopted algorithm in autonomous vehicle navigation for performing lane keeping and path tracking tasks. It was originally developed by the Stanford Racing Team for their autonomous vehicle that won the 2005 DARPA Grand Challenge. The controller operates on a kinematic model and is designed to stabilize both the cross-track error and the heading error, ensuring that the vehicle converges to and follows a given reference path. It is designed as a traditional geometric path-tracking controller and it consists of a simple nonlinear feedback law. The primary aim of the Stanley controller is to guide the front axle of a vehicle along a predefined trajectory by adjusting the front steering angle. This is achieved through the minimization of the cross track error  $e_{ct}$  and the heading error  $e_h$ . The Stanley controller computes a desired steering command  $\delta_o$  based on the current heading error and cross track error using the following control law:

$$\delta_o = e_h + \text{atan} \left( \frac{K_s e_{ct}}{v_b + v_x} \right)$$

where  $\delta_o$  represent a preliminary and unsaturated steering angle,  $K_s$  is a positive gain parameter, the Stanley gain,  $v_x$  is the current longitudinal speed of the vehicle and  $v_b$  is a small constant added to prevent division by zero, typically set equal to one. To ensure the control command remains within feasible physical limits,  $\delta_o$  is saturated to the vehicle's maximum steering capability  $\delta_m$ . The examined vehicle steering angle  $\delta_f$  has been saturated to  $\delta_m = \pm 35^\circ$  as follows:

$$\delta_f = \text{sat}(\delta_o, \delta_m) \triangleq \begin{cases} \delta_m, & \delta_o \geq \delta_m \\ \delta_o, & -\delta_m < \delta_o < \delta_m \\ -\delta_m, & \delta_o \leq -\delta_m \end{cases}$$

The Stanley controller guarantees asymptotic convergence to the path under ideal kinematic conditions and has some beneficial advantages, as its simplicity with respect to the PIDF and the double PIDF tuned via LQR. For these reasons, it is easier to be implemented, with just one variable to be tuned,  $K_s$ . It demonstrates resilience to measurement noise due to the use of relative geometric relationships rather than relying on high-order derivatives. Moreover, it is computationally efficient, which is beneficial for embedded automotive systems. On the contrary, the Stanley controller does not account for vehicle dynamics, such as tire slip, inertia, and lateral forces. Therefore, its performance may degrade at higher speeds (over 60 km/h) or under aggressive maneuvers.

### 2.2.1 Stanley gain tuning

In this section different simulation are reported to show the response of the vehicle dynamics with the implementation of the Stanley controller depending on the Stanley gain  $K_s$ . In Figure 14 it is possible to notice a non-linear behavior of the RMS value of the lateral acceleration  $a_{lat}$  depending on the Stanley gain  $K_s$ . The minimization of variable  $a_{lat}$  is a key point to obtain a good behavior

of the vehicle, as well as other important variables such as the cross track error, the heading error and the side slip angle. The following parameters have been used to carry out the simulations:

- **Longitudinal speed** = 22.2 m/s

- **Curvature radius** = 85 m

- $v_b = 1$

- **Stanley gain:**

- $K_s = 0.6$

- $K_s = 0.7$

- $K_s = 1$

- $K_s = 1.6$

- $K_s = 1.65$

- $K_s = 1.7$

- $K_s = 3$

For the examined vehicle the optimal gain  $K_s$  is 1.6.

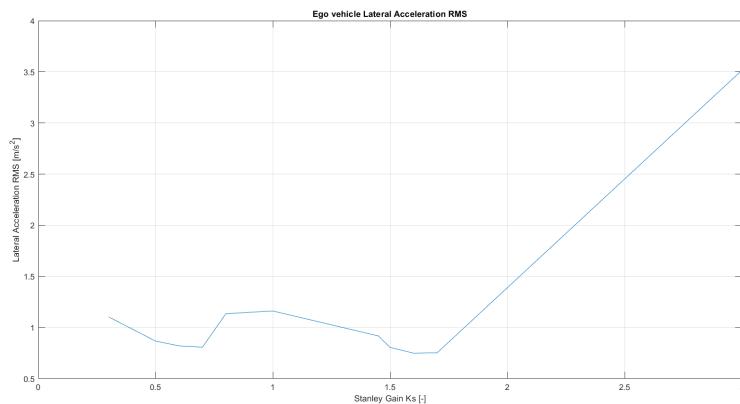


Figure 14: Stanley gain tuning

In Figure 15 the simulation results usign the tuned Stanley Gain are reported. The actual vehicle speed is not in the optimal range for the Stanley controller, which in its optimized configuration has a side slip angle  $\beta$  that exceeds the boundary limits of  $\pm 3^\circ$ . Even though, the general behavior of the vehicle is appropriate, with a steering angle that stays in the range  $\pm 7^\circ$  that has been previously chosen for the PIDF through the error weight  $deltaMaxDev$ . The cross track and the heading error around the  $t = 5$  s are oscillating but finally converging to zero around  $t = 10$  s. Decreasing the Stanley gain  $K_s$  leads to high peaks that exploit a non-proper behavior of the vehicle with a reduction in terms of safety and comfort, as visible in Figure 16.

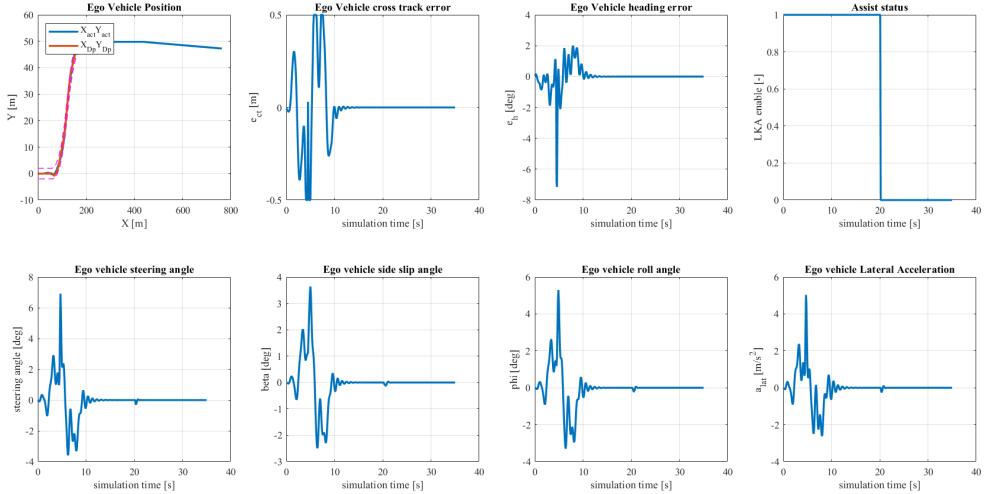


Figure 15:  $K_s = 1.6$

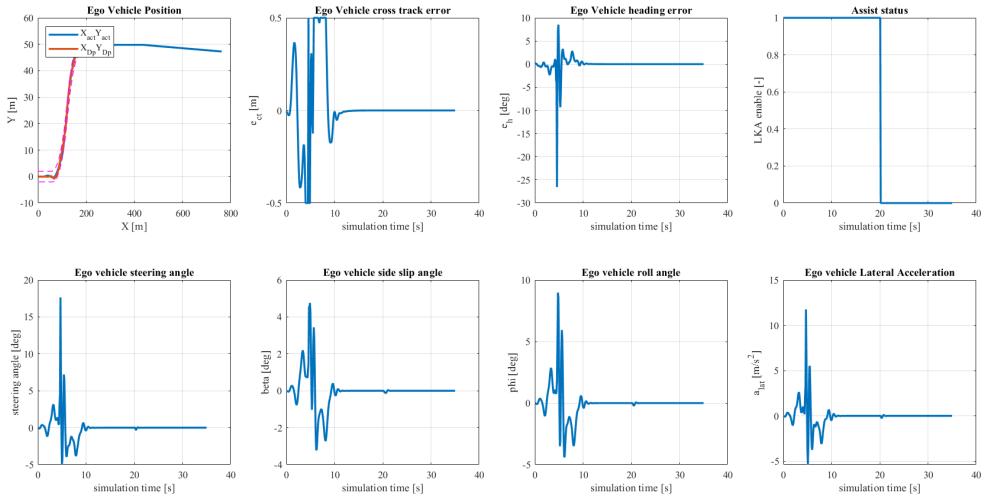


Figure 16:  $K_s = 1.0$

### 3 Stability test at different speeds and trajectories

The aim of this section is to perform multiple test analysis to evaluate the stability of the controllers by means of changing either the vehicle speed or the curvature radius. The vehicle speed could be easily changed in the MATLAB script *helperLKASetUp.m*, as it is a constant parameter. The curvature radius has been varied using the MATLAB script *createRoad.m*, which generates a road centerline path consisting of straight segments and curved turns based on a specified radius. Afterwards, the simulation scenario file *LKATestBenchScenario.mat* is updated by replacing its road centerline data with this newly generated path, saving the updated scenario in a new ".mat" file. Moreover, the safe lateral distance has been changed to observe how the optimal value depends on either the vehicle speed or the trajectory.

#### 3.1 PIDF controller stability

The results obtained in Figure 17 and in Figure 18 have been carried out by maintaining the same curvature radius that has been used for the previous simulations ( $R=85$  m), and changing the vehicle speed. The idea was to test the vehicle at urban speed ( $V=11.1$  m/s) and highway speed ( $27.7$  m/s), comparing to the previous tests constant speed ( $V=22.2$  m/s). The controller has been

tuned through the LQR procedure assuming a vehicle speed equal to 22.2 m/s and a curvature radius of 85 m.

By analyzing the results is possible to notice that it properly works for that speed, as well as for  $V = 11.1$  m/s. However, the problem comes at high speed, where the controller is not able to adjust the trajectory of the vehicle anymore, leading to both unsafe and uncomfortable behavior of the vehicle. This could be clearly seen in Figure 17 where both the cross track error and the heading error show high frequency and amplitude oscillations. It is clear that the lateral acceleration reached by this simulation with a vehicle speed  $V = 27.7$  m/s, visible in Figure 18, must be avoided.

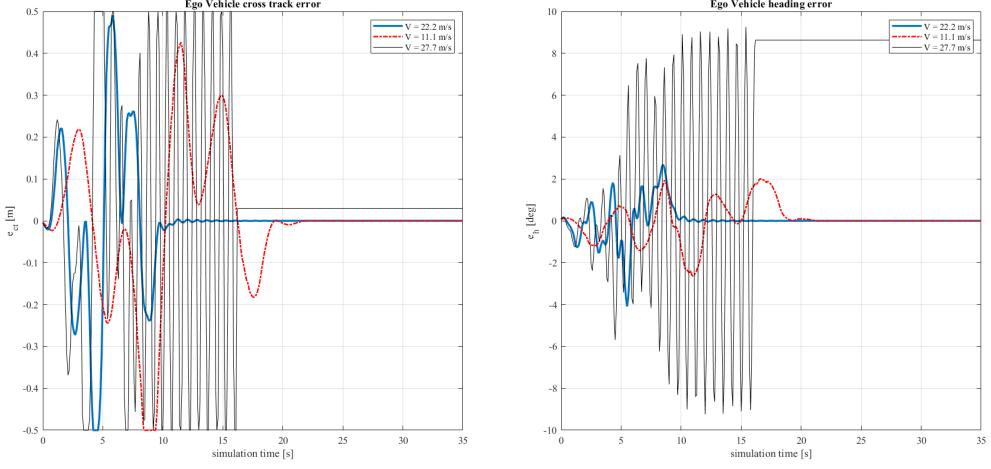


Figure 17: Comparison of  $e_{ct}$  and  $e_h$  errors with  $R = 85m$

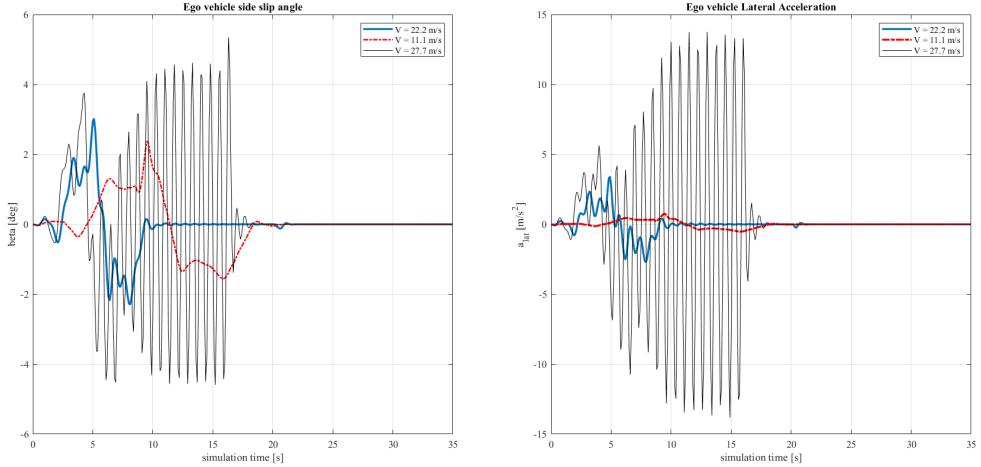


Figure 18: Comparison of  $\beta$  and  $a_{lat}$  with  $R = 85m$

Another simulation has been carried out by testing the same speeds with a higher curvature radius, imposed to be  $R = 285$  m. The considerations regarding the results are quite similar as before. It could be noticed in Figure 19 that the two lower speeds reach both comfort and safe behavior of the vehicle along the trajectory, keeping smoother curves and limited error variations thus showing an excellent results for the PIDF controller. On the opposite, the high speed test leads to large oscillations for all the variables under exam.

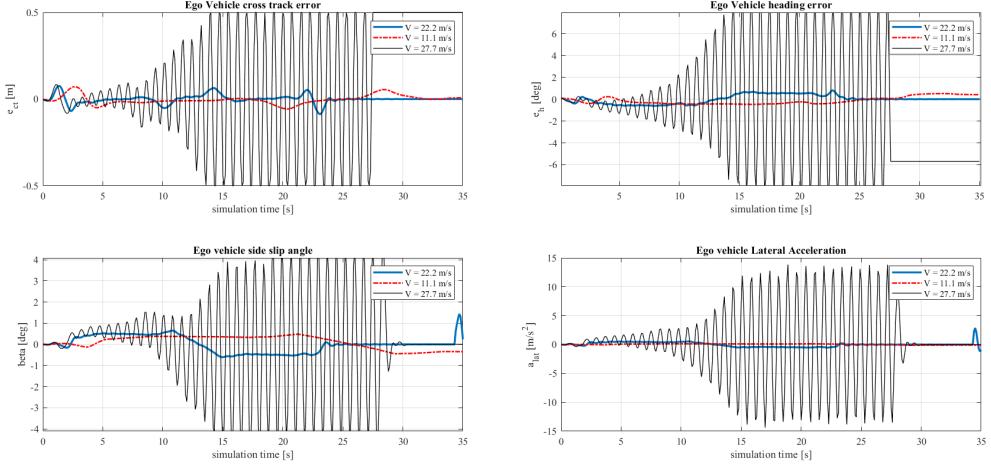


Figure 19: Comparison of  $e_{ct}$ ,  $e_h$ ,  $\beta$  and  $a_{lat}$  with  $R = 285m$

It has been tested also a very low value of curvature radius and it is possible to admit that a lower boundary limit for curvature radius exists. This is due to the responsiveness of either the controller or the steering actuators. In Figure 20 it is shown that the vehicle, even at urban speed ( $V = 11.1 \text{ m/s}$ ), is not able to perform the small radius trajectory.

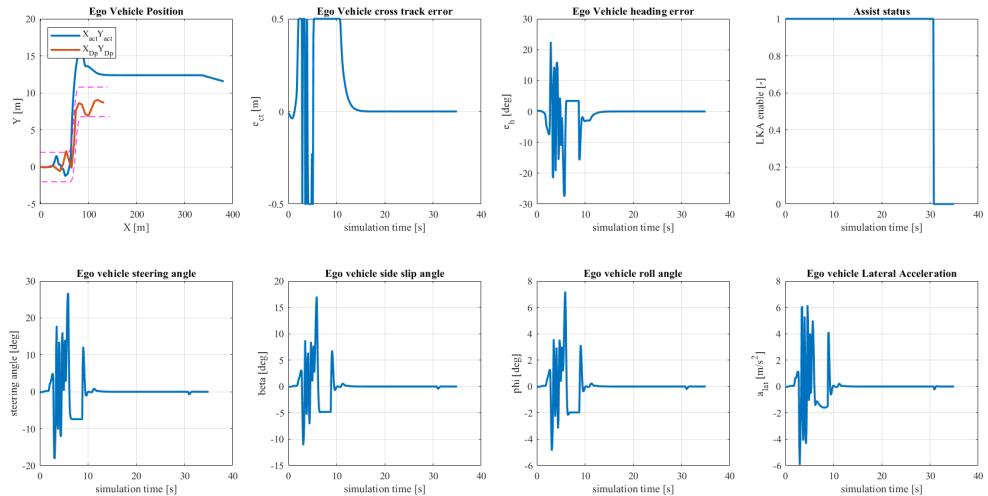


Figure 20: PIDF controller tested with  $R = 15 \text{ m}$  and  $V = 11.1 \text{ m/s}$

### 3.2 Stanley controller stability

Regarding the Stanley controller, a drastic difference of behaviors are obtained, as shown in Figure 21, which reports the results obtained from simulations carried out with a constant curvature radius of 85 m and the following set of longitudinal speeds:  $v_x = 11.1 \text{ m/s}$ ,  $v_x = 22.2 \text{ m/s}$  and  $v_x = 27.7 \text{ m/s}$ , respectively. Considering the boundary limits of  $\pm 3^\circ$  for  $\beta$ , the Stanley controller works quite well for low speed. However, some criticalities related to comfort could be present due to the heading error which converges to zero quite late with respect to the PIDF controller strategy. On the other hand, for higher speeds the controller shows a faster response with a quick convergence to zero of both the cross track and heading errors. However, higher amplitudes are present, which may compromise the safety of the vehicle.

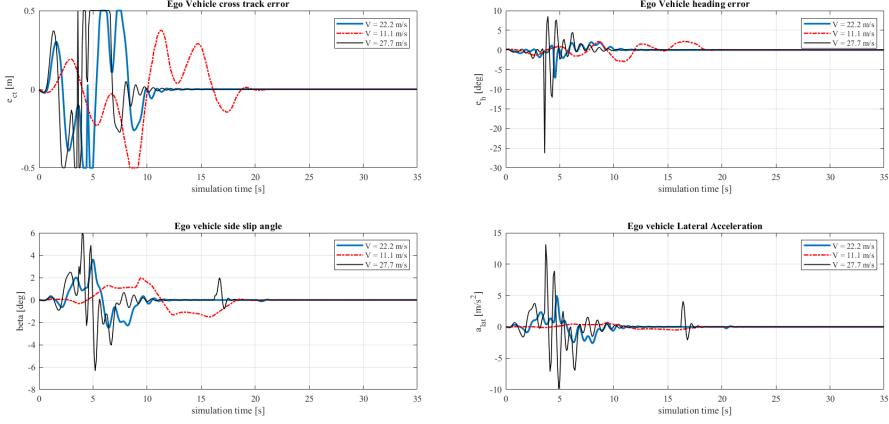


Figure 21: Stanley controller tested with  $R = 85\text{m}$  and different speeds

In addition, three different simulations were carried out with a constant curvature radius of  $R = 285\text{m}$  and varying vehicle speeds. As with the PIDF control strategy, the objective is to evaluate the vehicle's performance at urban speed ( $V = 11.1\text{ m/s}$ ) and highway speed ( $V = 27.7\text{ m/s}$ ), using the constant speed case ( $V = 22.2\text{ m/s}$ ) as a reference. The results are reported in Figure 22, where it can be observed that both the cross-track and heading errors are several orders of magnitude lower than those shown in Figure 21, where the curvature radius is  $R = 85\text{m}$ . This outcome is expected, as a larger curvature radius defines a trajectory that is easier for the vehicle to follow, keeping it closer to a steady-state condition. Small oscillations are still present, and the errors converge to zero only towards the end of the simulation, around 25 seconds. After that point, some peaks appear in both the  $\beta$  and  $a_{lat}$  plots. However, these are not related to the actual simulation, as the trajectory has already been completed.

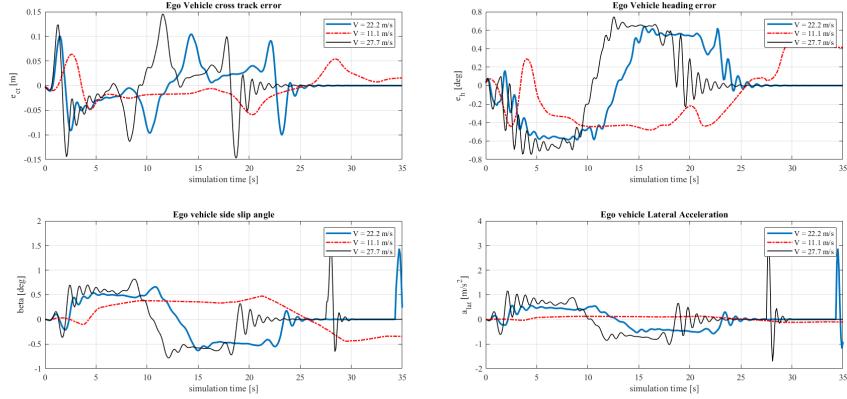


Figure 22: Stanley controller tested with  $R = 285\text{m}$  and different speeds

It is still interesting to notice that for very lower curvature radius the vehicle shows high instability. The simulation with a curvature radius  $R = 15\text{ m}$  and a vehicle speed  $V = 11.1\text{ m/s}$  has been reported in Figure 23. The controller is able to lead the errors to zero after 10 seconds, showing its general effectiveness, but it is not able to guarantee safety and comfort during a low radius steering manouvre. The side slip  $\beta$  oscillates and reaches a peak of  $15^\circ$ , which has to be avoided in order to keep the vehicle steady and between the lines.

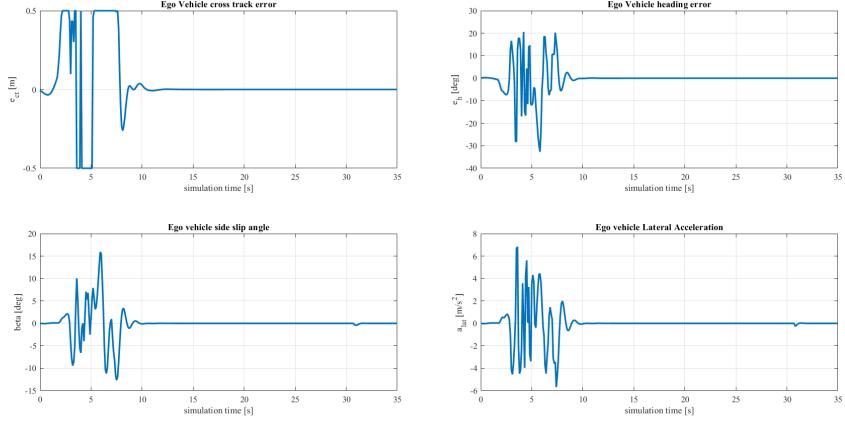


Figure 23: Stanley controller tested with  $R = 15\text{m}$  and  $V = 11.1\text{m/s}$

## 4 Curvature radius misjudged by the driver

It is particularly interesting to analyze the scenario in which the curvature radius perceived by the driver differs from that of the actual trajectory. In this section, the trajectory is kept constant with a curvature radius of  $85\text{m}$  and a constant vehicle speed of  $V = 22.2\text{m/s}$ , while varying the curvature radius provided by the driver. The simulation has been carried out both with the Lane Keeping Assist (LKA) controller enabled, using the previously tuned optimal value for the Safe Lateral Distance (SLD =  $1.75\text{m}$ ), and with the LKA system disabled. This allows for an analysis of the controller's performance under this critical condition. Figure 24 shows a snapshot from the Simulink driving scenario, illustrating the consequences of an incorrect curvature radius input by the driver when the LKA system is turned off. The vehicle drifts from the center lane into the right lane, resulting in a clearly unsafe and dangerous situation. In this case, the driver's input curvature radius, set via a MATLAB script, is  $84\text{m}$ . Although this is only a slight deviation from the actual trajectory radius, a significant impact on vehicle stability can be observed. As shown in Figure 25, both the cross-track error and the heading error fail to converge to zero at steady state, an undesirable condition for safe and reliable autonomous driving.

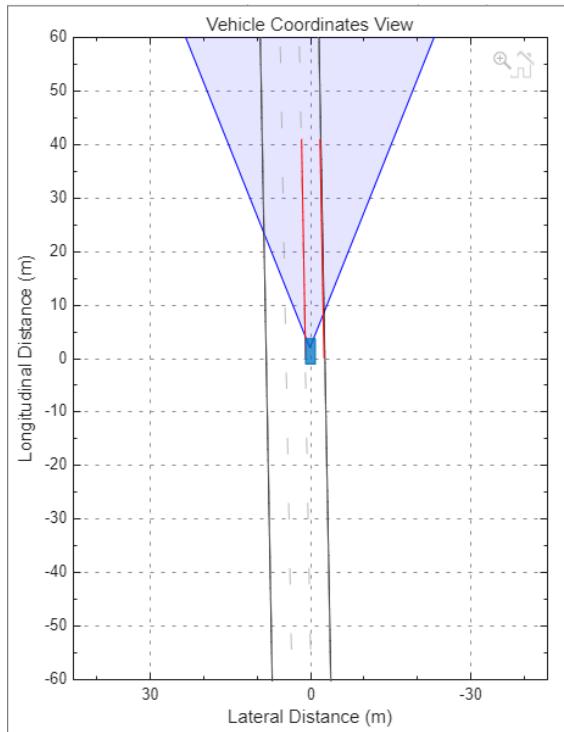


Figure 24: Misjudged radius simulation with LKA disabled - Driving Scenario

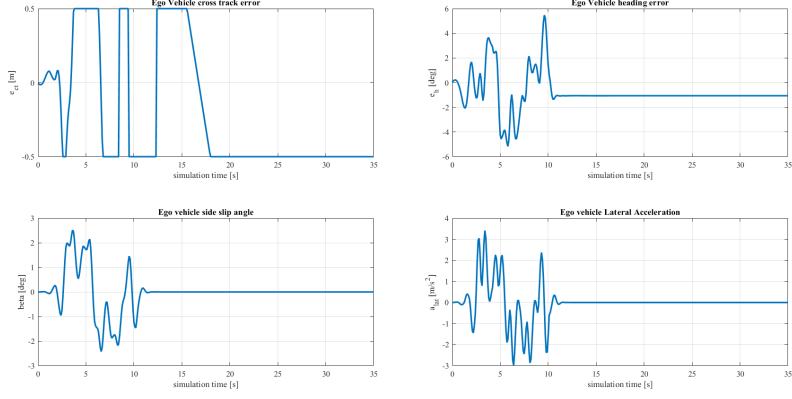


Figure 25: Misjudged radius simulation with LKA disabled - Errors,  $\beta$  and  $a_{lat}$

Using the same configuration for the driver and curvature radius, a simulation was performed with the Lane Keeping Assist system enabled, as shown in Figure 26. It is noteworthy that the LKA controller proves highly effective in this scenario, assisting the driver in keeping the vehicle on the correct path, following the intended trajectory, and leading to convergence of the tracking errors to zero at steady state. Although some oscillations are present, caused by the initial discrepancy in the curvature radius estimation, the vehicle is still able to safely follow the trajectory, demonstrating the robustness and utility of the LKA system in compensating for minor driver input errors.

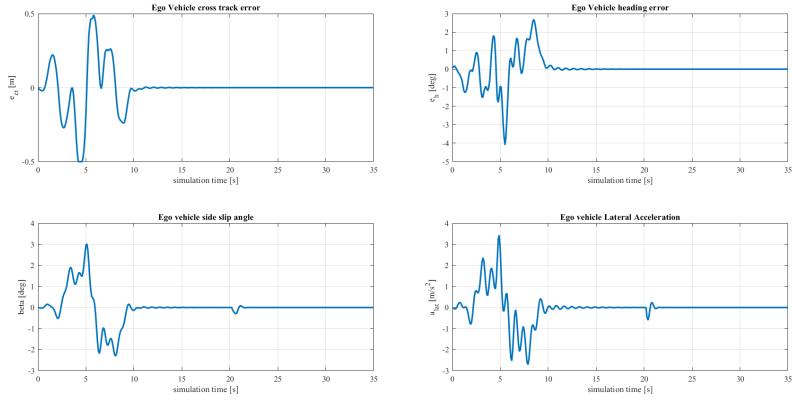


Figure 26: Misjudged radius simulation with LKA enabled - Errors,  $\beta$  and  $a_{lat}$

Moreover, an extreme scenario has been tested, in which the curvature radius perceived by the driver is  $R = 285$  m, while the actual curvature radius of the trajectory remains  $R = 85$  m. It is remarkable to observe that, even in this case, the Lane Keeping Assist system takes full control of the steering and effectively assists the driver in guiding the vehicle along the correct path, accurately following the intended trajectory. This result is illustrated in Figure 27.

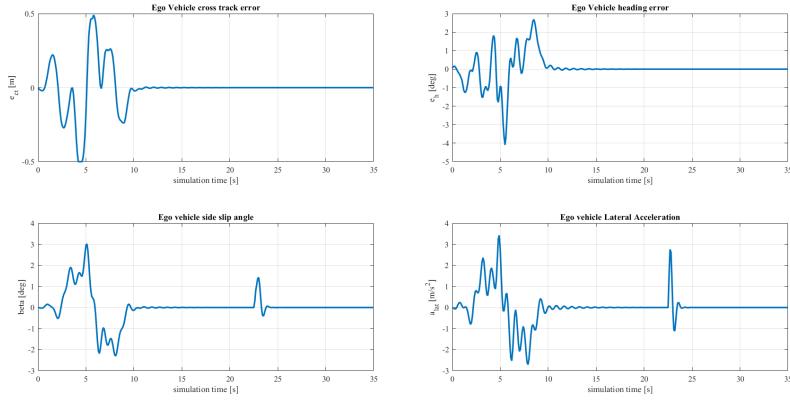


Figure 27: Misjudged radius simulation with LKA enabled - Errors,  $\beta$  and  $a_{lat}$

A similar simulation was performed using the Stanley controller in order to compare the reliability of both control strategies. In this scenario, the driver misjudges the curvature radius and approaches the curve assuming a smaller radius ( $R = 84\text{ m}$ ), while the actual trajectory has a curvature radius of  $R = 85\text{ m}$ . Despite this discrepancy, the Stanley controller is able to safely guide the vehicle along the desired trajectory, maintaining lane position as effectively as possible. Figure 28 presents a comparison between two simulations: one with the Lane Keeping Assist system disabled, and the other with the Stanley-based LKA controller enabled. The most notable observation is that, when the controller is active, the heading error converges to zero, unlike in the uncontrolled case, highlighting the enhanced safety and effectiveness of the Stanley LKA controller. Additionally, although the cross-track error exhibits a few irregular spikes, it ultimately converges to zero and shows overall better performance compared to the diverging behavior observed without the controller. It is also worth noting that other variables, such as the steering angle  $\delta$  and the side slip angle  $\beta$ , reach higher peak values when the controller is enabled. This is due to the system overriding the driver inputs to quickly and aggressively correct the vehicle's trajectory, ensuring adherence to the intended path.

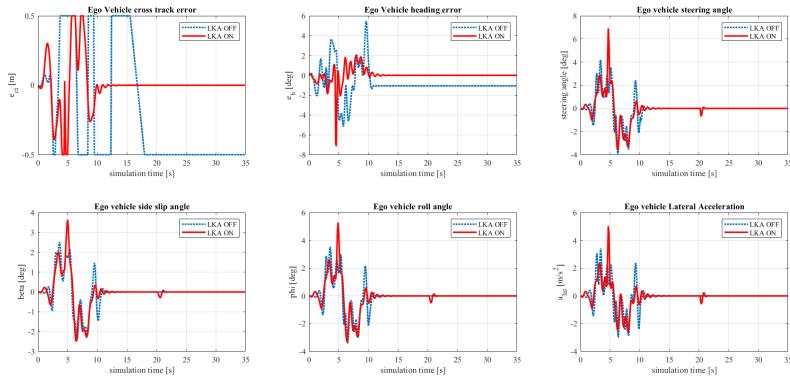


Figure 28: Misjudged radius simulation. Comparison: LKA disabled and LKA enabled

In Figure 29, two different case scenarios are presented in which the driver misjudges the curvature radius, either underestimating it ( $R = 84\text{ m}$ ) or overestimating it ( $R = 285\text{ m}$ ). It can be observed that, when the Lane Keeping Assist system is enabled and takes control of the steering, the driver's input is effectively disregarded, and the controller operates in a consistent manner regardless of the driver's estimation. The results of the two simulations are nearly identical and largely overlap throughout the trajectory. Minor differences appear only at the end of the simulation, after the driving scenario has concluded, and do not impact the overall system performance.

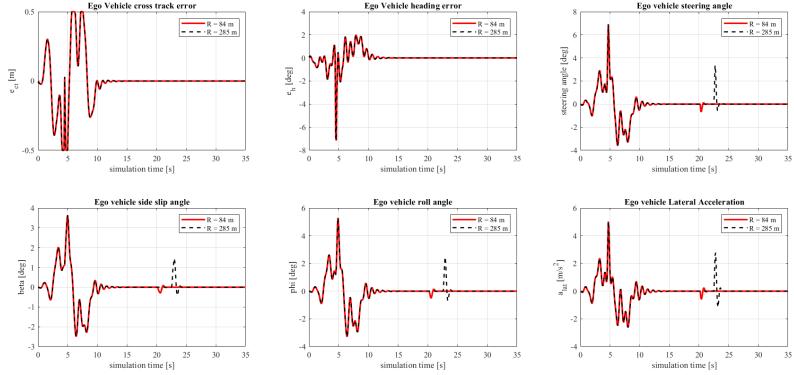


Figure 29: Misjudged radius simulation. Comparison  $R = 84\text{m}$  and  $R = 285\text{m}$

## 5 Conclusions

The comparative study between the PIDF controller, tuned via LQR, and the Stanley controller for Lane Keeping Assist revealed complex and scenario-dependent results. Initially, the PIDF controller appeared to outperform the Stanley controller in terms of tracking accuracy, error convergence, and comfort, especially at moderate speeds. Its model-based design and precise tuning of weighting matrices allowed for effective control over cross-track and heading errors, with limited steering effort and good dynamic stability. However, as the simulations progressed and the system was tested under more demanding conditions, particularly at higher vehicle speeds, the Stanley controller demonstrated superior robustness and stability. While its design is purely kinematic and simpler, the Stanley controller managed to maintain better lateral control and smaller error peaks in high-speed scenarios, where the PIDF began to show limitations. These included oscillatory behavior, reduced responsiveness, and higher lateral accelerations due to the static nature of its initial tuning. This outcome highlights a key insight: although model-based controllers like PIDF offer great flexibility and precision, their effectiveness strongly depends on operating conditions and tuning fidelity. In contrast, the Stanley controller, despite its simplicity, can outperform more complex alternatives in dynamic, high-speed environments, likely due to its geometric control structure and lower sensitivity to model inaccuracies or noise. In conclusion, there is no one-size-fits-all controller for LKA systems. The PIDF approach is valuable for precise and tunable control, especially at lower to medium speeds and in well-characterized scenarios. The Stanley controller, meanwhile, emerges as a surprisingly effective solution at higher speeds, offering robustness and simplicity, traits particularly desirable in real-world embedded systems and edge cases involving driver error or aggressive trajectories. These findings underscore the importance of evaluating control strategies not only in ideal conditions but across the full spectrum of expected use cases.