Final Project: Autonomous Car

Field and Service Robotics

Lorenzo Palma Xu P38000303
Gabriella Vuoto P38000273

July 2025

GitHub link:
`https://github.com/Pixel-0-1/Final_Project_FSR.git`
Videos link:
`https://drive.google.com/drive/folders/1RrM099Mb482fBJVtVMBMUR7hCdXt53cV?usp=drive_link`

Contents

1 Introduction

1.1 Autonomous Car

The concept of autonomous cars has gained increasing relevance and rapid advancement of mobile robotics, particularly in trajectory tracking and regulation applications. In the last decade, autonomous vehicle technologies have made significant strides, driven by their potential in transportation, logistics, and smart mobility. Their control and trajectory tracking pose complex challenges that continue to attract interest from both academic and industrial communities.

This work focuses on the control of autonomous ground vehicles, which are classified as non-holonomic mobile robotic systems. Unlike holonomic robots, which can move instantly in any planar direction, these vehicles are subject to non-holonomic constraints that limit their degrees of freedom—mainly due to their wheel configuration and steering mechanism. These constraints prevent direct lateral motion and on-the-spot rotation, adding considerable complexity to the control design process.

1.2 Bicycle Model

To model the vehicle, the study adopts the well-known "Bicycle Model", a simplified yet effective representation that captures the key non-holonomic characteristics of the system. The model considers a virtual two-wheeled vehicle: a front wheel for steering and a rear wheel for traction. It is governed by three state variables (x, y, and orientation θ) and two control inputs (linear velocity v and steering angle ψ), resulting in nonlinear kinematic equations that describe the system's motion in a 2D space.

The system parameters are:

- L: wheelbase (distance between wheel axes)
- (x, y): position of the reference point (typically the rear axle)
- θ: bicycle orientation with respect to the x-axis
- ψ: front wheel steering angle
- v: linear velocity

The fundamental kinematic equations are:

$$\begin{cases} \dot{x} = v\cos(\theta) \\ \dot{y} = v\sin(\theta) \\ \dot{\theta} = \frac{v}{L}\tan(\psi) \end{cases}$$

The bicycle is subject to the non-slip constraint:

- The rear wheel cannot slip laterally
- The front wheel follows the steering direction

The constraint is expressed as:

$$\dot{x}\sin(\theta) - \dot{y}\cos(\theta) = 0$$

The main goal of this work is to conduct a comparative analysis of various control strategies applied to this vehicle model, evaluating their performance, robustness and stability.
The study is structured in two main parts:

- **Regulation**: This section analyzes two classic regulators a Cartesian-based Regulator, and a Posture Regulatore, both based on Runge-Kutta numerical integration. These controllers aim to

drive the system from any initial state to a desired final configuration.

- **Trajectory Tracking**: The second section focuses on advanced trajectory tracking controllers. It includes traditional methods like Approximate Linear and Almost Non-linear approaches, Input-Output linearization and the Model Predictive Control (MPC) method. MPC is highlighted for its ability to optimize control actions over a finite horizon while accounting for system constraints and performance criteria. Its predictive nature allows it to dynamically adapt to changing operational conditions.

Ultimately, this work aims to provide a comprehensive understanding of the dynamic behavior of a non-holonomic robotic system modeled by the bicycle model, under the influence of different control and regulation strategies.

2 Regulators

2.1 Cartesian Regulator

The **Cartesian Regulator** represents a specific control technique for non-holonomic robotic systems that focuses exclusively on regulating the robot's position, leaving the final orientation free. This approach is particularly effective when the primary objective is to reach a specific Cartesian position (x, y) without strict constraints on the vehicle's orientation θ.

The Cartesian Regulator strategy for the bicycle model is based on defining a purely Cartesian position error:

$$\mathbf{e}_p = \begin{bmatrix} x_d - x \\ y_d - y \end{bmatrix}$$

where (x, y) represent the robot's current coordinates relative to the target point (x_d, y_d), which in this case is positioned at the origin of the reference frame.

The Cartesian controller for the unicycle model implements the following control laws:

- **Linear velocity**: $v = -k_1(x\cos\theta + y\sin\theta)$
- **Angular velocity**: $\omega = k_2(\text{atan2}(y, x) + \pi - \theta)$

where $k_1, k_2 > 0$ are the controller gains.
For the bicycle model, the angular velocity ω is converted to the steering angle ψ through the kinematic relationship:

$$\psi = \arctan\left(\frac{\omega \cdot L}{v}\right)$$

where L represents the vehicle's wheelbase. In the specific implementation, a value of $L = 3.5$ m is used, and the steering angle is limited to ± 60 to respect the system's physical constraints.
The 2^{nd} order Runge-Kutta linearization to obtain the feedback of the states is described as:

$$\begin{cases} x_{k+1} = x_k + v_k T_s \cos\left(\theta_k + \frac{\omega_k T_s}{2}\right) \\ y_{k+1} = y_k + v_k T_s \sin\left(\theta_k + \frac{\omega_k T_s}{2}\right) \\ \theta_{k+1} = \theta_k + \omega_k T_s \end{cases}$$

Generally v_k and ψ_k are reconstructed from the proprioceptive sensors such as the wheels' encoders, but considering the intrinsic uncertainty of the dynamic model, we can assume that the computed input already take in account odometry drifts and wheels slippage. The angular velocity is reconstructed instead by these two values, So that:

$$\begin{cases} v_k = v \\ \psi_k = \psi \\ \omega_k = \frac{v_k}{L}\tan(\psi_k) \end{cases}$$

2.1.1 Simulation Results Analysis

As observed from the trajectory plot (Figure 1), the bicycle proceeds to the desired pose along the path.

In Figure 2, it can be observed that the x-coordinate, starting from an initial position of approximately -100 m, shows exponential convergence toward zero. The trajectory exhibits a characteristic exponential decay shape, reaching values close to zero within the first 35-40 seconds of simulation. The y-coordinate, starting from a negative position of approximately -50 m, converges toward zero with dynamics similar to the x-coordinate. The convergence is monotonic and stable. The vehicle's orientation does not converge to zero. This behavior is typical of the Cartesian Regulator, which does not constrain the final orientation but uses it as a control variable to reach only the target position.

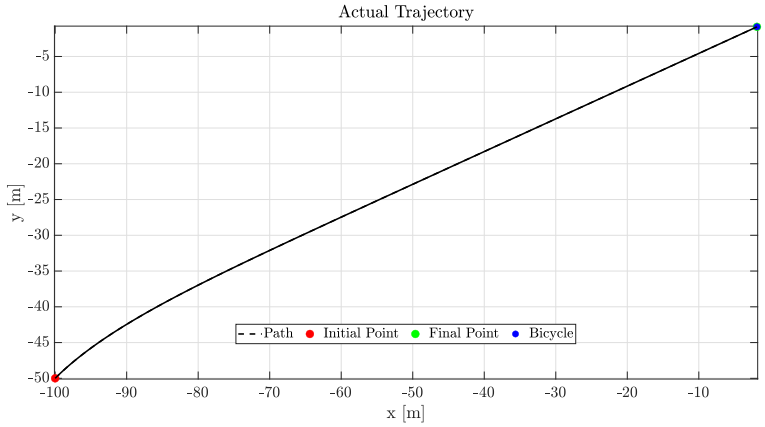

Figure 1: Actual Trajectory

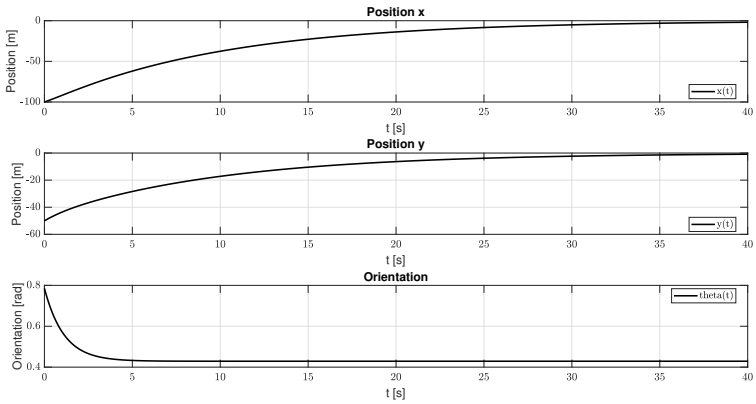

Figure 2: Bicycle's states

Figure 3 shows the errors. It can be observed that the errors e_x and e_y maintain excellent values (on the order of 10^{-6} m), demonstrating the effectiveness of the regulator for cartesian control. The error e_θ shown in the third graph simply represents the free evolution of the orientation of the system, which is not controlled by the Cartesian regulator. The oscillations observed are therefore normal and expected, since there is no active control on this variable.

Figure 4 represents the evolution of the linear velocity v and steering angle psi over time. The linear velocity presents high initial values (approximately 10 m/s) that decay exponentially toward zero; the velocity profile is always positive, indicating that the vehicle always moves forward toward the target; the decay shape is consistent with the control law $v = -k_1(x\cos\theta + y\sin\theta)$, which depends on the projection of the position error along the robot's sagittal axis. The steering angle (ψ) presents significant initial values (approximately -0.1 radians ≈ -5.7) that decay rapidly.

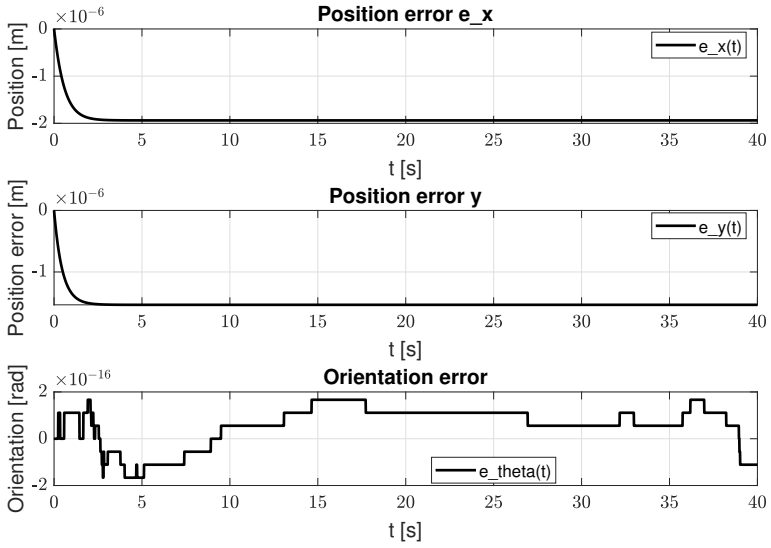

Figure 3: errors

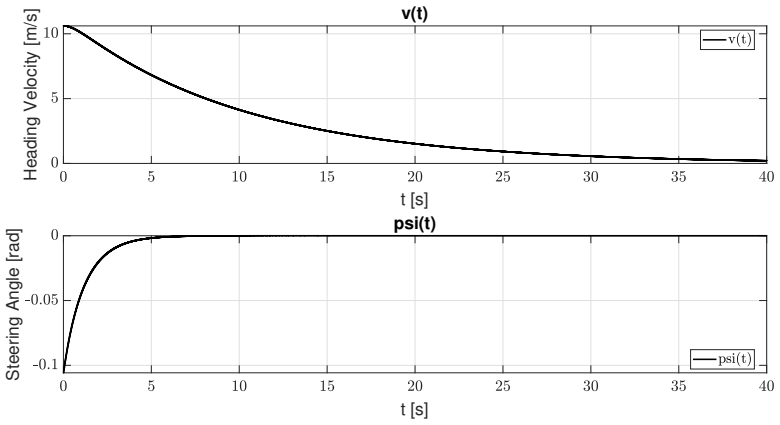

Figure 4: Heading velocity v and Steering angle ψ

2.1.2 Final Considerations

The graphs demonstrate the fundamental properties of the Cartesian Regulator:

1. **Asymptotically stable convergence**: The state variables (x, y) converge toward the desired values (target position and zero velocity)
2. **Non-holonomic behavior**: The system respects the kinematic constraints of the bicycle model, with an initial curve and then a straight trajectory of the terrestrial vehicle
3. **Energy efficiency**: The control inputs decay gradually, avoiding excessive system stress
4. **Robustness**: The controller maintains stable performance despite the non-linear nature of the system

This approach allows exploiting the simplicity of the Cartesian Regulator while maintaining the fidelity of the bicycle model in representing autonomous vehicle dynamics. However, as observed, there is no orientation control.
For this reason, the Posture regulator has been implemented.

2.2 Posture Regulator

This regulator, based on polar coordinates, with state feedback also obtained through the Runge-Kutta odometric localization method, aims to reach a desired pose and orientation final configuration.
First, the initial configuration is specified as:

$$q_i = [x_i \ y_i \ \theta_i] = \begin{bmatrix} -100 & -50 & \frac{\pi}{4} \end{bmatrix}^T$$

while, the regulation problem requires reaching:

$$q_f = [x_f \ y_f \ \theta_f] = [0 \quad 0 \quad -\frac{2\pi}{3}]^T$$

To express the problem in polar coordinates:

$$\begin{cases} \rho = \sqrt{x^2 + y^2} \\ \gamma = atan2(y, x) - \theta + \pi \\ \delta = \gamma + \theta \end{cases}$$

The control input for a bicycle model are defined as:

$$\begin{cases} v = k_1 \rho \cos\gamma \\ \psi = \arctan(\frac{\omega L}{v}) \end{cases}$$

with $\omega = k_2\gamma + k_1 \sin\gamma \cos\gamma\left(1 + k_3 \frac{\delta}{\gamma}\right)$ and $L = 3.5$ m the wheel base.

2.2.1 Simulation Results Analysis

For this simulation, the following gain values are chosen:

$$\begin{cases} k_1 = 0.2 \\ k_2 = 1 \\ k_3 = 1 \end{cases}$$

and the results are presented as follows:

As observed from the trajectory plot (Figure 5), the bicycle proceeds to the desired pose along a sort of S-shape path, that is characteristic of the polar coordinate-based control approach.

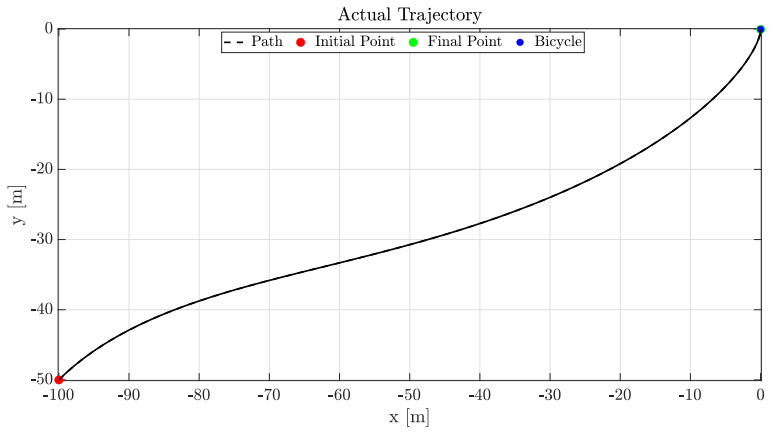

Figure 5: Actual Trajectory

The coordinates of the position x and y, and the orientation θ successfully converge to the desired final configuration (Figure 6), in approximately $35\,s$. The gradual and smooth convergence demonstrates the stability of the controller despite the presence of uncertainties in the system.

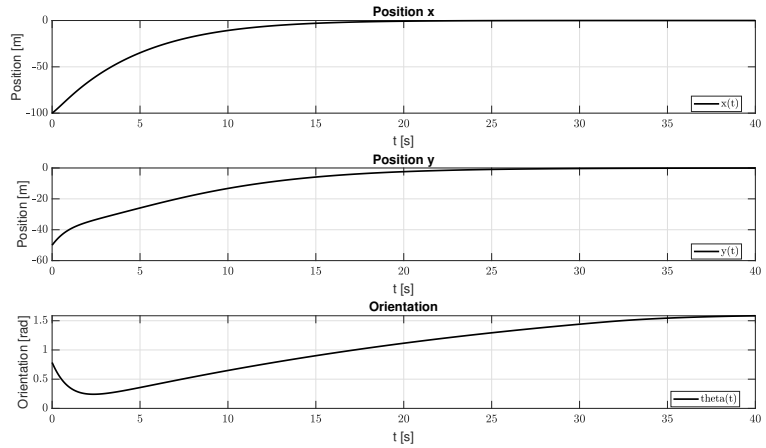

Figure 6: Bicycle's states

From Figure 7, we can see that the linear velocity converges to zero, indicating a successful regulation, while the steering angle converges to 60, which is the maximum limit considered for ψ.

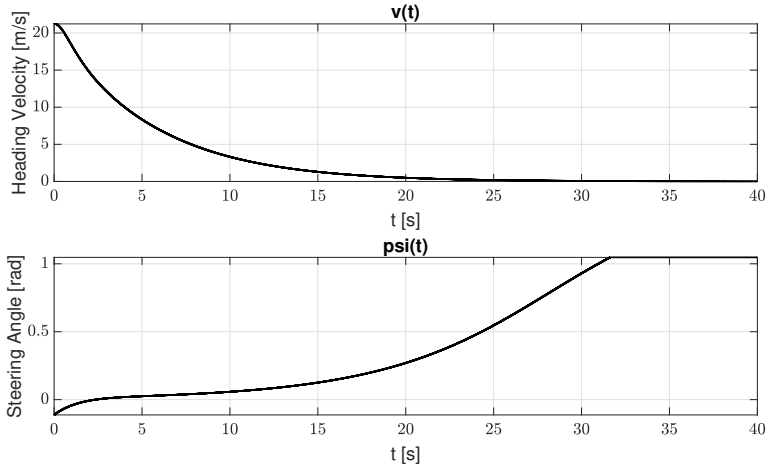

Figure 7: Heading velocity v and Steering Angle ψ

Finally, in Figure 8, we can observe that all three errors converge to zero in approximately 1 s, indicating that the overall performance of the posture regulator demonstrates robust and stable behavior, with the ability to handle also the uncertainties of the model.

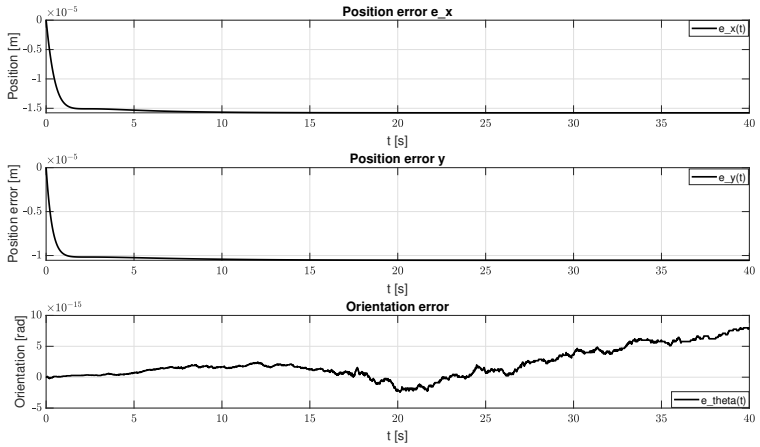

Figure 8: errors

2.2.2 Final Considerations

The simulation results demonstrate the fundamental properties of the Posture Regulator:

1. **Complete pose convergence**: Unlike the Cartesian Regulator, all three state variables (x, y, θ) converge to their desired values, ensuring precise positioning and orientation control
2. **Smooth control profiles**: The control inputs exhibit gradual variations without discontinuities (in the polar coordinates), reducing mechanical stress and energy consumption
3. **Stability and robustness**: The system maintains stable behavior throughout the convergence process, with no oscillations or instabilities
4. **Non-holonomic compliance**: The generated trajectories respect the kinematic constraints of the bicycle model, producing physically realizable motion
5. **Efficient convergence**: The controller achieves the desired pose in a reasonable time frame without excessive control effort

The Posture Regulator thus provides a complete solution for autonomous vehicle navigation tasks requiring precise pose control, making it particularly suitable for applications such as parking maneuvers, docking operations, and formation control where both position and orientation accuracy are essential. However, the drawback is that the kinematic model is not defined at the target point, and the controller back in the original coordinates (cartesian coordinates) is discontinuous at the origin.
In fact, due to **Brockett's necessary condition** any feedback control law that can regulate the full pose, must be necessarily discontinuous and/or time variant also for a bicycle.

3 Tracking Controllers

3.1 Approximate Linear Controller

The control of nonlinear robotic systems represents a significant challenge in the field of automation and mobile robotics. One of the fundamental techniques to address this complexity is the **approximate linearization** approach, which allows for simplifying the control problem through mathematical approximations valid in a neighborhood of the desired equilibrium point.
Approximate linearization is based on the principle of approximating the behavior of the nonlinear system in the vicinity of a specific operating point, typically when the tracking error is sufficiently small ($e(t) \approx 0$). This approach is particularly effective for robotic systems where it is desired to maintain the vehicle in proximity to a predetermined reference trajectory.
The technique relies on three key approximations that enable linearization of the system's kinematic model:

1. **Trigonometric approximation**: $\sin(e_3(t)) \approx e_3(t)$, valid for small orientation angles
2. **Elimination of coupling terms**: $-e_2(t)u_1(t) = 0$ and $e_1(t)u_2(t) = 0$
3. **Error system linearization**: the resulting system assumes the form of a linear time-varying system

These simplifications lead to a system of linear differential equations of the form:

$$\dot{e}(t) = A(t)e(t)$$

where $A(t)$ is a time-varying matrix that depends on the reference trajectory parameters.
The proposed controller implements a feedback strategy that acts directly on the tracking error components. The control law is structured as follows:

- **Longitudinal control**: $u_1(t) = -k_1 e_1(t)$, where $k_1 > 0$ is a proportional gain
- **Lateral control**: $u_2(t) = -k_2(t)e_2(t) - k_3 e_3(t)$, which combines corrections on lateral position and orientation

The parameter $k_2(t)$ is time-varying and depends on the reference trajectory characteristics, while k_1 and k_3 are positive constants.
In the examined case, the control point is shifted to the center of gravity the vehicle (x_g, y_g):

$$x_g = x + \frac{L}{2}\cos(\theta)$$

$$y_g = y + \frac{L}{2}\sin(\theta)$$

The error is defined as follows:

$$M = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$e = M \cdot e^*$$

where e^* is equal to:

$$e^* = \begin{bmatrix} y_{1d} - x_g \\ y_{2d} - y_g \\ \theta_d - \theta \end{bmatrix}$$

with:

$$y_{1d} = x_d + \frac{L}{2}\cos(\theta_d)$$

$$y_{2d} = y_d + \frac{L}{2}\sin(\theta_d)$$

the desired coordinates of the CoG.
After implementing the errors and control inputs in MATLAB, and converting the angular velocity to steering angle, the gains necessary for good tracking were defined:

$$k_1 = 2\zeta a$$
$$k_2 = (a^2 - w_d^2)/v_d$$
$$k_3 = k_1$$

where:

$$\zeta = 0.3$$
$$a = 0.6$$

3.1.1 Simulation Results Analysis

Figure 9 shows that the trajectory is a spline obtained thanks to three intermediate waypoints, where $q_i = [0 \quad 0 \quad 0]^T$ and $q_f = [163.80 \quad 113.40 \quad 17.49]^T$ was obtained randomly; while Figure 10 contains the desired values for both the states and the velocities, where the latter are bounded as:

$$\begin{cases} |v(t)| \leq 10 \text{ m/s} \\ |\omega(t)| \leq 2 \text{ rad/s} \end{cases}$$

In this simulation because the initial values didn't satisfy the constraints, they were scaled with a scalar factor k:

$$\begin{cases} T = 30.00 \text{ s} \\ v(t) = 17.89 \text{ m/s} \\ \omega(t) = 0.90 \text{ rad/s} \end{cases} \xrightarrow{k=1.79} \begin{cases} T = 53.66 \text{ s} \\ v(t) = 10.00 \text{ m/s} \\ \omega(t) = 0.50 \text{ rad/s} \end{cases}$$

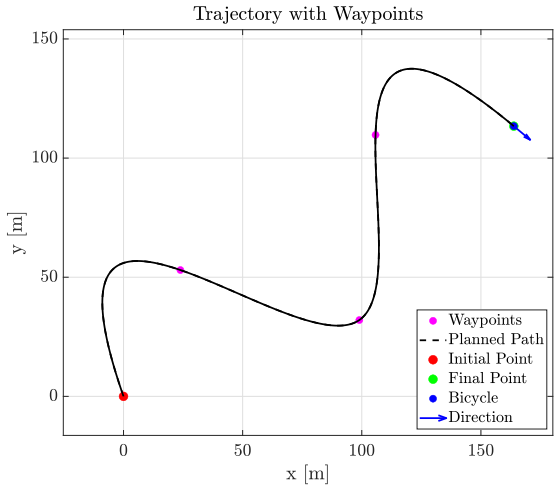

Figure 9: Desired Trajectory

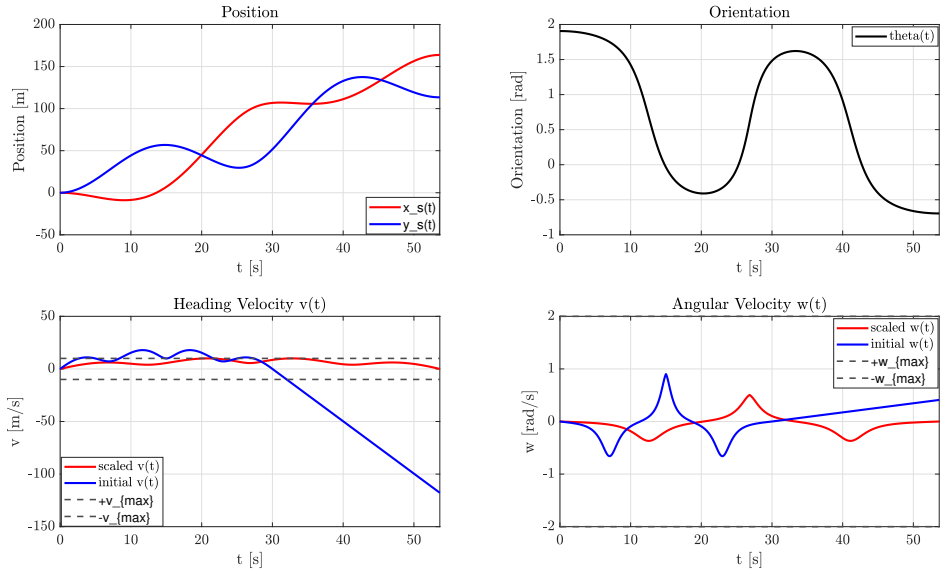

Figure 10: Desired States and Velocities

In Figure 11, it is evident that the controller does not perform optimally. The desired trajectory (red dashed line) is not well followed by the actual trajectory.

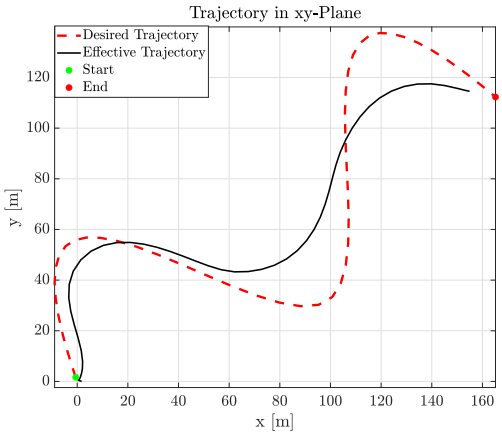

Figure 11: Trajectory

The lack of precision is particularly visible in the error plots (Figure 12), which show a maximum amplitude of 40-45 meters for position tracking. Also the orientation error is quite high and not negligible.

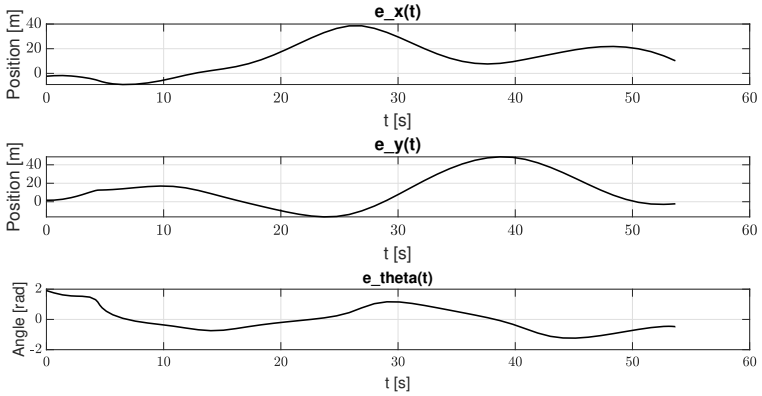

Figure 12: errors

From Figure 12 it is possible to notice that the control signals reveal interesting aspects:

- $v(t)$: velocity with peaks up to 23 m/s, it's a reasonable profile
- ψ (psi): steering angle with marked discontinuities in the transient and at the end of the tracking.

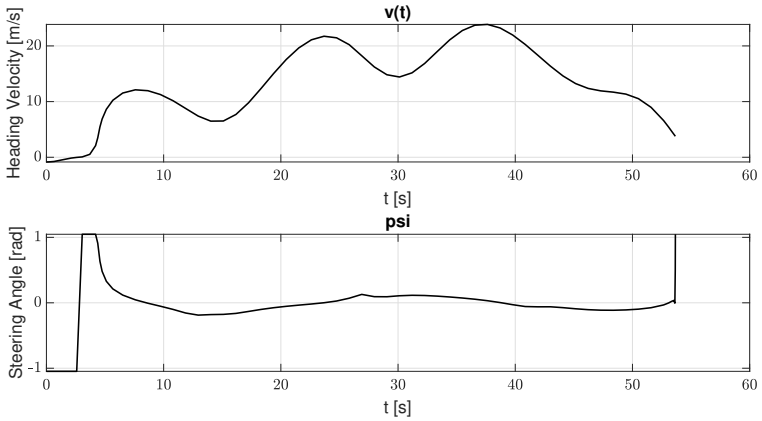

Figure 13: Control Input

3.1.2 Final Considerations

- **Discontinuities in steering control**: abrupt variations could cause actuation problems
- **Persistent errors**: the tracking performance remains suboptimal

This controller has not proven to be the best choice. At this point, the almost non-linear approach was tested as an alternative solution.

3.2 Almost Non-Linear Controller

This controller uses the same error vector as the approximate linear controller, but implements different control laws:

$$\begin{cases} u_1(t) = -k_1(v_d(t), \omega_d(t))e_1(t) \\ u_2(t) = -k_2(t)v_d(t)\frac{\sin e_3(t)}{e_3(t)}e_2(t) - k_3(v_d(t), \omega_d(t))e_3(t) \end{cases}$$

with $k_1(v_d(t), \omega_d(t)) > 0$ and $k_3(v_d(t), \omega_d(t)) > 0$ bounded positive gain, and $k_2 > 0$, in particular:

$$\begin{cases} k_1 = 0.05\left(\sqrt{v_d^2 + \omega_d^2}\right) \\ k_2(t) = 0.005 \\ k_3 = 0.1(|v_d| + |\omega_d|) \end{cases}$$

3.2.1 Simulation Results Analysis

The control inputs are computed using the same formulas as in the approximate linear controller case, and also the desired trajectory is the same.

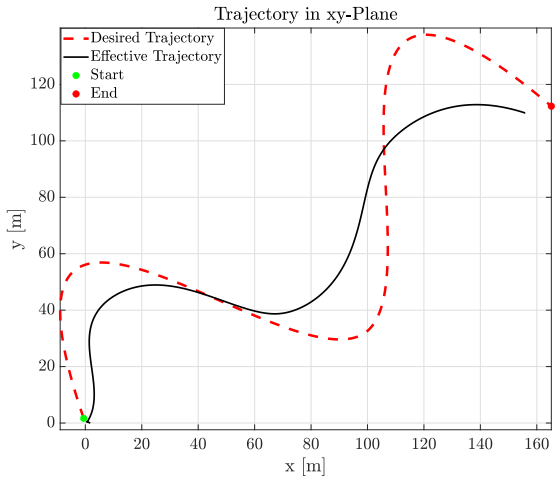

Figure 14: Actual and Desired Trajectory

In Figure 14 the trajectory appears quite similar to that produced by the linear controller, with a worse initial tracking but a smaller error on x-axis. Indeed, the black line (actual trajectory) does not faithfully follow the red dashed line (desired trajectory).

This lack of precision can also be observed in the error graphs (Figure 15). Specifically, the error along the x-axis remains within 20 m for most of the time, with a peak of approximately 30 m occurring around 30 seconds. The error along the y-axis presents a peak of 40 m, around 40 seconds. Also the error along θ is not well-contained.

Finally, regarding the control inputs in Figure 16, the heading velocity varies between $0 - 25$ m/s with a smooth trend, without excessive oscillations; the steering angle ψ remains within a reasonable range (± 0.2 rad/s), with gradual variations. This behavior is very similar to the approximate linear controller, but with no discontinuity of the steering angle at the end of tracking.

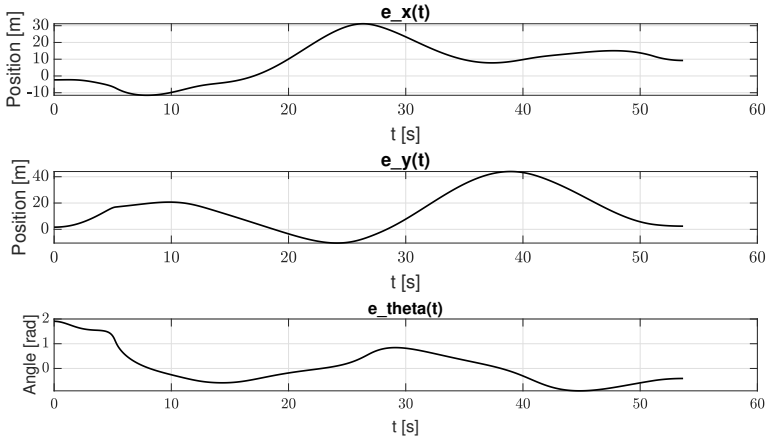

Figure 15: State errors

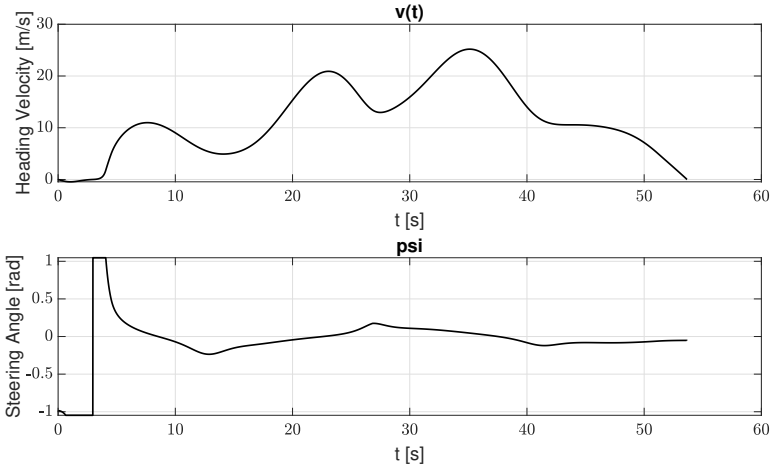

Figure 16: Heading velocity and Steering Angle

3.2.2 Final Considerations

- **Persistent errors**: the tracking performance remains suboptimal

In conclusion, the (Almost)-nonlinear controller does not demonstrate superiority over the linear controller. At this point, to try to improve the errors along the x,y axes, it was decided to implement the Input/Output controller.

3.3 Input/Output Linearization Controller

Given the previous trajectory, an I/O linearization control for the autonomous car has been used. To evaluate the performance of this control, the Center of Gravity has been considered, with $b = \frac{L}{2}$. Where L is the distance between the rear and front wheel, known as *wheelbase*: $L = (x_g, y_g)$.
The desired and the CoG coordinates are computed as:

$$\begin{cases} y_{1,d} = x_d + b\cos\theta \\ y_{2,d} = y_d + b\sin\theta \end{cases} \qquad \begin{cases} x_g = x + b\cos\theta \\ y_g = y + b\sin\theta \end{cases}$$

with x and y the coordinates of the center of the rear wheel. Then, the controller is designed as:

$$\begin{cases} u_1 = \dot{y}_{1,d} + k_x(y_{1,d} - x_g) \\ u_2 = \dot{y}_{2,d} + k_y(y_{2,d} - y_g) \end{cases}$$

control inputs for the unicycle are:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = T(\theta)^{-1} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \qquad \text{with} \qquad T(\theta) = \begin{bmatrix} \cos\theta & -b\sin\theta \\ \sin\theta & b\cos\theta \end{bmatrix}$$

with $T(\theta)$, which is invertible since $b \neq 0$.
But for the bicycle, there is the addition of the steering angle:

$$\psi = \arctan\left(\frac{\omega L}{2}\right)$$

3.3.1 Simulation Results Analysis

The controller demonstrates good overall performance. From Figure 17 It is possible to see that the actual trajectory (continuous line) follows the desired trajectory (dashed red line) very well, the system reaches the reference trajectory within reasonable time and no significant oscillations or instability are observed.
Gains k_x and k_y are chosen as:

$$k_x = 10 \qquad k_y = 30$$

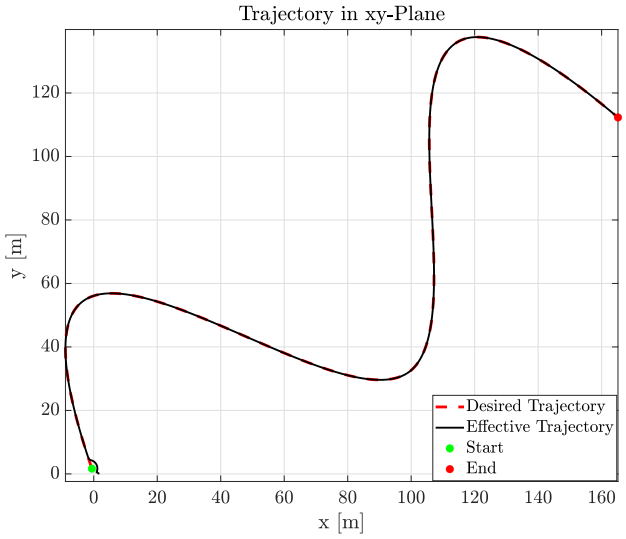

Figure 17: Desired and Actual Trajectory

The error plots shown in Figure 18, reveal several important characteristics: initial peaks (approximately ± 2 m for x and y and 2 rad for θ) reduce rapidly and converge to zero within the first 2 seconds. Therefore, the system demonstrates greater stability with quite aggressive, but fewer oscillations only in the first seconds for all the states that lead the controller to successfully track the desired position.

In theory, with this type of controller, it is not possible to directly control the orientation. However, the orientation error converges because it is implicitly controlled in the generation of this particular trajectory.

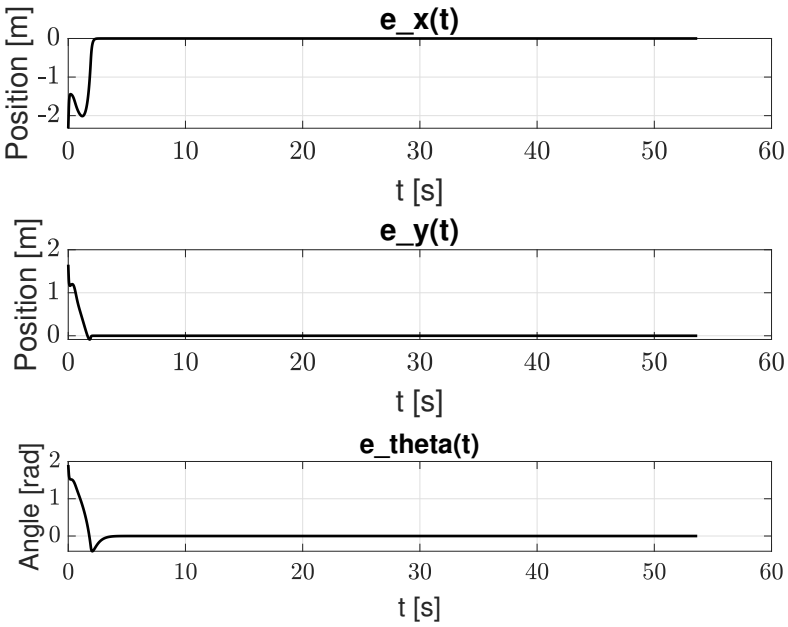

Figure 18: Errors of the states

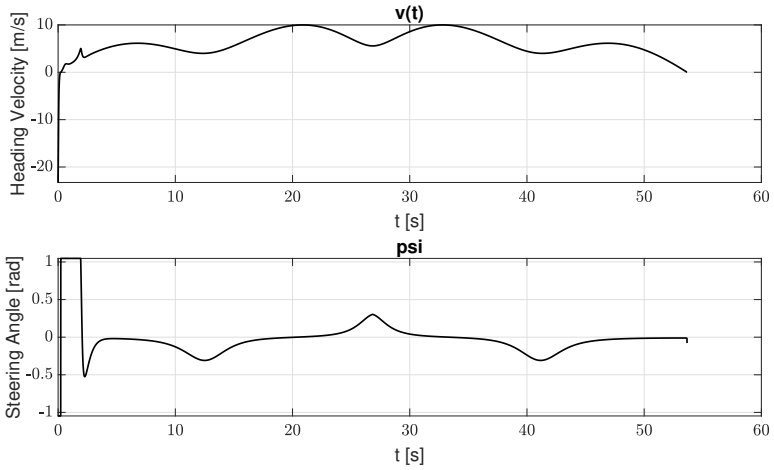

Figure 19: Control Inputs : Linear velocity v and Steering Angle ψ

Finally, Figure 19 shows the control input signals throughout the simulation. Concerning heading velocity, its average value is around 5 m/s and presents smooth and contained variations. Con-

cerning the Steering Angle, its values are within the interval $[-1, 1]$ radians, presenting gradual variations without many abrupt changes.

3.3.2 Final Considerations

The controller appears to be well-designed with the following key characteristics:

- **Acceptable transient response**: The controller achieves reasonable settling times and overshoot characteristics
- **Realistic and implementable control signals**: The control inputs remain within practical bounds and exhibit smooth behavior
- **Practically zero steady-state errors**: The system achieves excellent tracking accuracy in the long term

The input-output linearization controller shows excellent performance for bicycle trajectory tracking. The rapid convergence of position errors, combined with realistic and smooth control signals, demonstrates the effectiveness of the proposed approach. The controller successfully handles the nonlinear bicycle dynamics while maintaining stability and achieving precise trajectory following. In this particular case the orientation error was implicitly controlled due to this specific trajectory, but in general the I/O Linearization doesn't control it.
A counter-proof was obtained considering the same trajectory but with an orientation that wasn't tangent to the path anymore, but increased constantly (Figure 20).

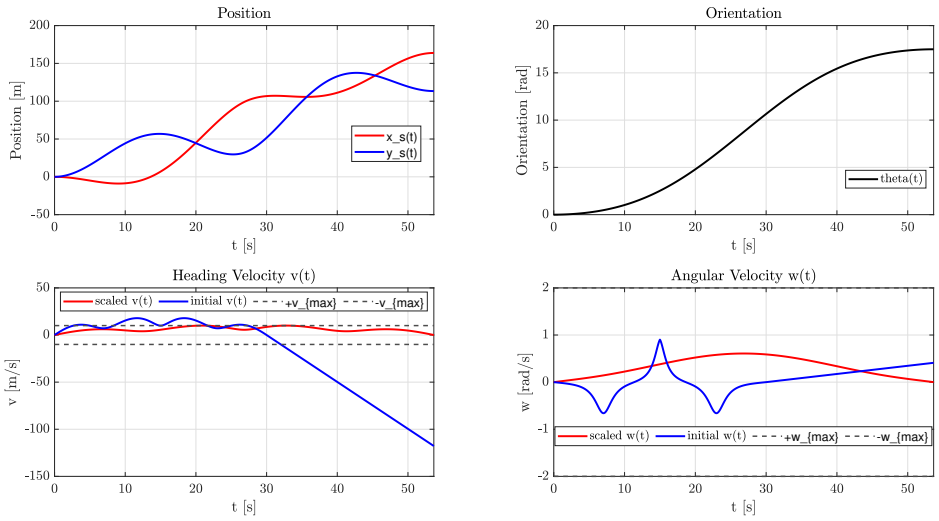

Figure 20: Desired states

In Figure 21 we can see that x and y errors still converge, but θ isn't controlled anymore.

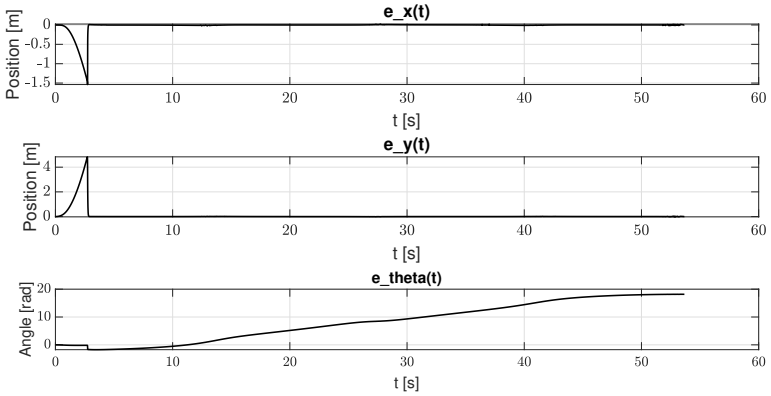

Figure 21: Errors of the states

While input-output control can effectively track reference trajectories, it lacks the ability to control the orientation and to anticipate and react to dynamic constraints such as obstacles. This limitation motivates the use of Model Predictive Control (MPC), which incorporates system dynamics and environmental constraints to generate safe and optimal control actions over a prediction horizon. Nonetheless, in this work, MPC was implemented in a simplified scenario without obstacles, in order to compare its performance with the input-output approach under similar conditions.

3.4 Model Predictive Control

The Model Predictive Control is an advanced control strategy that uses a discrete system model to predict future behavior and generate control input sequences over a control horizon, with the objective of minimizing a cost function over a prediction horizon.
MPC utilizes a mathematical model of the system (in this case, the vehicle dynamic model) to predict how the system will behave in future time steps.
In this case, it has been used an Extended Bicycle Model whose matrices are as follows:

$$\mathbf{A} = \begin{bmatrix} -\frac{2C_{af}+2C_{ar}}{mv_x} & 0 & -v_x - \frac{2l_fC_{af}-2l_rC_{ar}}{mv_x} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ -\frac{2l_fC_{af}-2l_rC_{ar}}{I_zv_x} & 0 & -\frac{2l_f^2C_{af}+2l_r^2C_{ar}}{I_zv_x} & 0 & 0 & 0 \\ 1 & v_x & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} \frac{2C_{af}}{m} & 0 \\ 0 & 0 \\ \frac{2l_fC_{af}}{I_z} & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{D} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} = \mathbf{0}_{3\times 2}$$

There are 6 states:

- $\dot{v}_y$: lateral acceleration
- θ: yaw angle
- $\dot{\theta}$: angular velocity
- y, x: position coordinates
- v_x: longitudinal velocity

and two control inputs:

- δ (delta): steering angle
- $throttle$: longitudinal acceleration

MPC solves an **optimization problem** at each time instant, with the following cost function:

$$J = \sum_{i=1}^{N_p} [Q_x(x_i - x_{ref,i})^2 + Q_y(y_i - y_{ref,i})^2 + Q_\theta(\theta_i - \theta_{ref,i})^2] + \sum_{j=1}^{N_c} [R_{u,\delta} \cdot \delta_j^2 + R_{u,throttle} \cdot throttle_j^2]$$

Subject to:

- Vehicle dynamic model
- Input constraints
- Initial conditions

Vehicle parameters are:

$$\begin{aligned} m = 165.9 \text{ kg} &\quad \text{Vehicle mass} \\ L = 3.73 \text{ m} &\quad \text{Vehicle length} \\ W = 1.88 \text{ m} &\quad \text{Vehicle width} \\ H = 1.2 \text{ m} &\quad \text{Vehicle height} \\ O_f = 0.9 \text{ m} &\quad \text{Front overhang} \\ O_r = 1.0 \text{ m} &\quad \text{Rear overhang} \\ l_f = 1.75 \text{ m} &\quad \text{Distance from CG to front axle} \\ l_r = 1.75 \text{ m} &\quad \text{Distance from CG to rear axle} \\ I_z = 2500 \text{ kg·m}^2 &\quad \text{Yaw moment of inertia} \\ C_{af} = 60000 \text{ N/rad} &\quad \text{Front tire cornering stiffness} \\ C_{ar} = 80000 \text{ N/rad} &\quad \text{Rear tire cornering stiffness} \end{aligned}$$

Control parameters are:

- **Prediction horizon** $N_p = 20$: the controller "looks ahead" 20 samples
- **Control horizon** $N_c = 7$: can modify controls only in the first 7 samples
- **Sampling period**: $step = 0.01s$

$$\begin{cases} Q_x = 10.0 & \text{weight for x position error} \\ Q_y = 6.0 & \text{weight for y position error} \\ Q_\theta = 1.0 & \text{weight for orientation error} \\ R_{u,\delta} = 0.05 & \text{weight for steering effort} \\ R_{u,throttle} = 0.15 & \text{weight for acceleration effort} \end{cases}$$

The MPC controller operates within the following physical constraints:

$$\begin{cases} \delta_{max} = 60 = \frac{\pi}{3} \text{ rad} \\ \delta_{min} = -60 = -\frac{\pi}{3} \text{ rad} \\ throttle_{max} = 5.0 \text{ m/s}^2 \\ throttle_{min} = -5.0 \text{ m/s}^2 \end{cases}$$

The input constraints can be expressed as:

$$\delta_{min} \leq \delta(k) \leq \delta_{max}$$
$$throttle_{min} \leq throttle(k) \leq throttle_{max}$$

for all $k = 0, 1, \ldots, N_c - 1$ within the control horizon.
The continuous-time state space representation is:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)$$

Where the system dynamics capture:

- Lateral dynamics (bicycle model): states 1-4
- Longitudinal dynamics: states 5-6

- Output selection: position (x, y) and orientation θ

The control strategy is:

1. **Prediction**: using the model, predicts the system evolution for the next N_p samples
2. **Optimization**: finds the optimal control sequence that minimizes the tracking error while respecting constraints
3. **Application**: applies only the first control of the sequence
4. **Repetition**: at the next step, repeats everything (receding horizon strategy)

3.4.1 Simulation Results Analysis

From the trajectory plot (Figure 22), it is possible to observe that the MPC controller demonstrates excellent tracking performance on the overall trajectory. Indeed, the almost perfect overlap between the desired trajectory (red dashed line) and the actual trajectory (black continuous line) indicates very effective control. The path, moreover, presents complex curves with variable curvature radii, successfully managed by the controller.

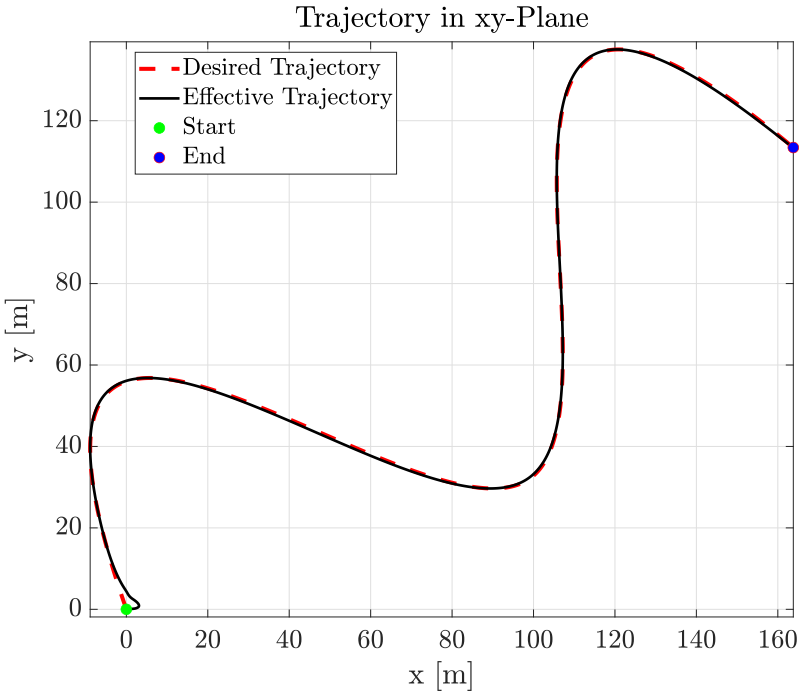

Figure 22: Desired and Actual Trajectory

From the error plots (Figure 23), it is possible to observe the behavior of the tracking errors for each state variable.
Regarding the error along the x-axis, there is a significant initial peak ($\sim$ 3 m) that rapidly decreases and stabilizes around 0 and -1 m for most of the simulation. This behavior suggests good correction capability after initial disturbances.
Concerning the error along the y-axis, the errors are contained within the interval 0 and -0.5 m, therefore significantly better compared to the X-axis.
Finally, the error along theta is initially quite high ($\sim$-2 rad $\approx$ -115°) and is then gradually corrected until it stabilizes around ± 0.2 rad after the first 10 seconds. The orientation correction requires more time compared to position errors but still does not result optimal.

In our case, the weight of theta was set equal to 1 because it was noticed that by increasingly raising the theta weight, there was a worsening of the y-dynamics. Given that theta, with these parameters, oscillates around ± 0.2 rad (11.5 degrees), an anomalous behavior of the vehicle is visible in the simulation where it slides laterally instead of following the trajectory with the correct orientation.

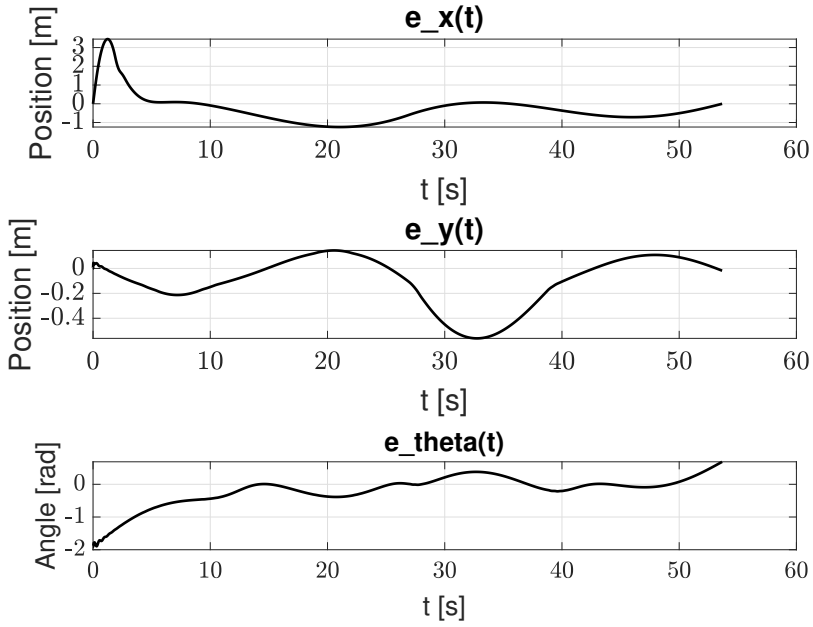

Figure 23: Errors of the states

From Figure 24, it is possible to observe the control inputs. With regard to the steering angle, it can be observed that the signal is free from oscillations except during the initial instants. Indeed, it exhibits a very well contained amplitude below the constraint of ± 60 after the small transient. Concerning the throttle, it presents cyclic variations that follow the trajectory geometry; in fact, it is possible to observe a positive acceleration (~ 0.15 m/s^2) during straight sections and significant deceleration (~ -0.5 m/s^2) in proximity to curves.

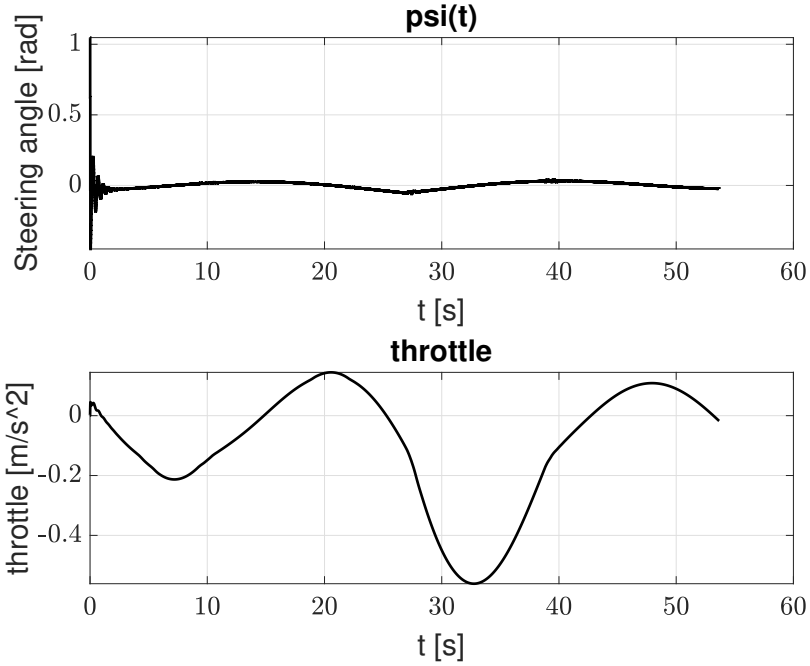

Figure 24: Control Input

3.4.2 Final Considerations

In conclusion, the MPC controller presents, as its main strength, effective trajectory tracking along the x and y axes, even if with slight initial oscillations. When comparing the obtained results with previously implemented controllers, it is evident that the MPC exhibits superior performance compared to the Approximate Linear and Almost-Nonlinear controllers. However, the inherent trade-off between y-position tracking and orientation (θ) control prevents it from being the ideal solution.
The controller demonstrates significant improvements in terms of:

- Overall trajectory adherence accuracy
- Convergence characteristics of position and orientation errors

Nevertheless, the compromise required between lateral position precision and orientation accuracy represents a fundamental limitation that must be addressed in future controller developments. This trade-off manifests as acceptable position tracking performance at the expense of suboptimal vehicle orientation control, resulting in the observed lateral sliding behavior.
Overall, the MPC controller represents a substantial advancement over previous control strategies, an improved position tracking w.r.t. the approximate linear and almost non linear; and a direct control in orientation w.r.t. the I/O linearization. Nonetheless, MPC offers clear advantages in scenarios involving noise, model uncertainties, or the need to handle sudden or unpredictable obstacles.

4 Conclusions

This study has conducted a comparative analysis of different control strategies applied to the bicycle model for autonomous vehicles, highlighting the distinctive characteristics of each approach and their practical applicability.

The analysis of regulators has shown how the **Cartesian Regulator** is effective for positional control but inadequate when orientation control is also required, while the **Posture Regulator** resolves this limitation by providing complete pose control, despite presenting discontinuities at the origin due to the intrinsic properties of non-holonomic systems.

Regarding trajectory tracking controllers, the **approximate linear controllers** have demonstrated insufficient performance with high tracking errors and discontinuities in control signals, proving unsuitable for applications requiring precision. In contrast, the **Input/Output linearization controller** has distinguished itself with superior performance, achieving rapid error convergence (within 2 seconds) and realistic control signals, establishing itself as the most reliable solution among the traditional approaches tested.
The **Model Predictive Control** has shown good tracking capability on complex trajectories and optimized actuator control, representing the most advanced and promising approach for future developments. However, an intrinsic trade-off has emerged between positional control precision and orientation, manifesting in lateral sliding behaviors of the vehicle. Despite this limitation, MPC offers a scalable architecture ideal for integrating dynamic constraints and obstacle management.

In conclusion, the study demonstrates that the choice of controller must be guided by specific operational conditions: the I/O controller represents the optimal solution for applications under ideal conditions, while MPC constitutes the preferable choice for real environments characterized by the presence of disturbances and obstacles, where its predictive capability and constrained optimization prove fundamental. The work therefore provides a solid foundation for selecting appropriate control strategies in different application contexts of the autonomous automotive sector.