



MOTORVEHICLE
UNIVERSITY OF
EMILIA-ROMAGNA

LEADER-FOLLOWER AGV CONTROL

Applied Automatic Control
Advanced Automotive Engineering
Motorvehicle University of Emilia Romagna
A.Y. 2023-2024

Francesco Bianco, Giovanni Paternò, Filippo Tullio

February 6, 2025

Contents

1	Introduction	1
1.1	Motivations	1
1.2	Contributions	3
1.3	State of art and literature comparison	4
1.4	Organisation of the manuscript	5
1.5	List of the symbols	6
2	AGVs leader-follower control	7
2.1	Problem Formulation	7
2.2	Model Analysis	9
2.2.1	Kinematic Model	9
2.2.2	Sensors	12
2.2.3	Linearization	15
2.2.4	Control problem formalization	19
2.3	Linear Model Analysis	20
2.3.1	Open Loop Dynamics	20
2.4	Control	27
2.4.1	Reachability	27
2.4.2	Integral Action	28
2.4.3	Observability	30
2.5	Proposed Solution	32
2.5.1	Optimal Control Model	33
2.5.2	Selection of Q and R matrices	34
2.5.3	Solution of the Optimal Control Problem	36
2.5.4	Contribution of K_S and K_I	38
2.5.5	Design of K_S and K_I	40
2.5.6	Observer and Kalman filter	48
2.5.7	Tuning procedure of K_O	51
2.5.8	Feedforward	56

3 Application	57
3.1 Simulator scheme and description	57
3.1.1 Linear Plant	58
3.1.2 State Feedback with integral Action	58
3.1.3 Observer	59
3.2 Real Plant simulation	59
3.2.1 Path selection	59
3.2.2 Tuning of the Real Plant	60
3.2.3 Real Plant simulation results	62
3.3 Obstacle detection and stopping strategy	63
3.3.1 Obstacle Trajectory	64
3.3.2 Control Strategy Implementation	64
3.3.3 Test and Validation	65
3.3.4 Stopping Strategy simulation and results	69
4 Conclusions and further investigation	74

Abstract

Autonomous Guided Vehicles (AGVs) have become vital in enhancing operational efficiency in industrial sectors, especially within automotive manufacturing, where streamlined logistics are crucial. This project focuses on developing a leader-follower AGV configuration, where the follower AGV relies on sensor-based systems to track and replicate the movements of the leading AGV. This method ensures synchronized, automated transport of materials across the factory floor, minimizing the need for manual handling and optimizing internal workflows. The use of onboard sensors allows the follower AGV to maintain a safe and precise distance from the leader, adapting to dynamic environments such as shifting layouts or the presence of obstacles. The proposed system is designed to ensure accuracy and reliability in path-following while addressing the challenges of varying industrial conditions, thus improving the overall efficiency of material handling within the automotive production process.

Chapter 1

Introduction

1.1 Motivations

The increasing demand for more flexible, efficient, and scalable automation systems in manufacturing has significantly driven the development of advanced logistics solutions, particularly in the automotive industry, where precision, speed, and safety are of paramount importance. As production lines evolve and become more complex, traditional methods of material handling are often no longer sufficient. These traditional systems tend to result in inefficiencies, delays and increased operational costs, making the integration of automated systems such as Automated Guided Vehicles (AGVs) crucial for maintaining smooth production processes and reducing the risk of human error. In this context, AGVs offer a transformative solution by automating the transport of materials and components across various stages of production, while significantly enhancing both the speed and precision of operations. Moreover, AGVs bring additional advantages in terms of improving workplace safety, by reducing the need for human interaction with potentially dangerous machinery and equipment. AGVs, particularly those in a leader-follower configuration, present a highly effective solution to the logistical challenges faced by modern manufacturing plants. In this setup, one AGV leads the way, establishing a path that the second vehicle follows closely behind. This dynamic requires careful and sophisticated path-planning strategies to ensure that the two vehicles operate smoothly without interference, even in environments with multiple moving vehicles. To achieve this, the following AGV typically moves along a predetermined route map that mirrors the path of the leading vehicle, enabling precise coordination of their movements. The leader-follower approach allows for

synchronized transport of goods, without the need for fixed infrastructure, offering immense flexibility in factory layouts. This level of flexibility is particularly valuable in dynamic manufacturing environments, where production needs and facility layouts are constantly changing. By automating the transport of components between different production stages, AGVs contribute to streamlining workflows and reducing the reliance on human labor for repetitive tasks, which can further enhance both productivity and safety. Moreover, in more advanced manufacturing plants, such as those in the automotive industry, AGVs are increasingly being used not just for transport but also as platforms for vehicle assembly. This innovative approach allows AGVs to autonomously transport vehicle parts between different assembly stations, minimizing the space needed to move vehicles through the production process. By using AGVs as mobile platforms, manufacturers can optimize the available factory space, enabling more efficient use of the area while also increasing the space available for operators to carry out assembly tasks. This system allows for a more organized and efficient layout, with fewer interruptions in the production process, improving overall throughput and reducing the time required to complete each vehicle.

In environments where multiple AGVs are in operation, ensuring effective coordination is critical. While individual AGVs can often be programmed with simple path-planning algorithms, the complexity increases when multiple vehicles share the same space. To avoid collisions and optimize the flow of traffic, coordination strategies must be employed, which may involve real-time communication between AGVs and central control systems. Additionally, AGVs are often constrained to move along predefined routes, referred to as a route map, to ensure they stay within designated areas and avoid bottlenecks. These predefined paths help to organize the movement of AGVs in busy industrial settings, where there may be multiple vehicles moving simultaneously.

At Mercedes-Benz's Türk truck assembly plant, KUKA's KMP 1500 AGVs are employed to autonomously transport large truck cabs between assembly stations, dramatically improving the efficiency and flexibility of the production process while reducing the amount of space required for logistics. These AGVs operate in dynamic conditions, coordinating their movements with other vehicles in real time, which not only streamlines the transport of components but also enhances the flexibility of the entire production system. This example highlights the potential for AGVs to change radically factory logistics, paving the way for smart, autonomous manufacturing environments. Developing advanced control mechanisms for leader-follower AGVs is crucial to ensuring they can effectively operate in such demand-



Figure 1.1: AGVs in a plant

ing settings, as improving their coordination, adaptability and precision will allow for the reduction of logistical bottlenecks and the maximization of productivity, setting the stage for the future of fully autonomous and highly efficient production lines.

1.2 Contributions

The primary objective of this project is to develop a control system for two Automated Guided Vehicles (AGVs) working in tandem in a leader-follower configuration. The follower AGV is tasked with maintaining accurate synchronization with the leader AGV’s path in a dynamic manufacturing environment. Rather than focusing on the physical characteristics of the AGVs themselves, this project emphasizes the design and implementation of the control system. The system’s mathematical model was thoroughly analyzed using eigenvalue techniques to assess its stability, followed by linearization at a key point—specifically along a path with minimal curvature—to facilitate the creation of a closed-loop control system. This system ensures the follower consistently stays on course relative to the leader’s trajectory, compensating for various disturbances encountered along the path. In the leader-follower scenario, the follower AGV must continuously adjust its path to account for changes in curvature, environmental conditions and other dynamic factors. The control approach implemented relies on the use of industrial camera sensors, which deliver real-time feedback on the leader’s position and movement. This enables the follower AGV to adjust its trajectory accordingly, ensuring precise tracking of the leader’s movements. To enhance the stability and responsiveness of the system, the design integrates state feedback control,

with additional integral action and an observer to handle disturbances, while feedforward control was excluded as further described in 2.5.8. The path selected for the test scenarios was designed to simulate real-world conditions in an industrial setting, featuring various turns with different curvature radii. This allowed for the thorough testing of the leader-follower control system's robustness across a range of conditions, ensuring the follower AGV could reliably track the leader while maintaining operational stability. The control parameters were fine-tuned using optimal control analysis to calibrate the state feedback and observer, ultimately ensuring both AGVs could operate efficiently and safely within the constraints of the environment.

1.3 State of art and literature comparison

The development of Automated Guided Vehicles (AGVs) began in the 1950s, with early models like Barrett Electronics' "Guide-O-Matic" relying on overhead wires for guidance. AGVs saw large-scale adoption in the 1970s when Volvo introduced them to streamline material handling in automotive production [1]. Over time, as manufacturing demands for flexibility grew, the leader-follower configuration emerged, where a follower AGV autonomously tracks a leader AGV along a shared route, enabling synchronized transport without the need for fixed pathways.

Today, leader-follower AGV systems are crucial for enhancing logistics efficiency and adaptability, particularly in dynamic settings like automotive manufacturing. Modern AGVs incorporate advanced navigation systems such as laser-guided technology, inertial sensors and vision-based tracking that allow them to follow both static and adaptive paths. Decentralized control enables follower AGVs to respond instantly to the leader's movements, ensuring smooth coordination on the factory floor.

A major advancement in controlling these systems has been the integration of model predictive control (MPC), enabling AGVs to predict and adjust their paths in real-time to maintain safe distances and avoid obstacles, even in complex layouts. Robust control methods enhance accuracy by minimizing the effects of sensor noise and environmental disruptions. An example of these technologies in action is KUKA's KMP 1500 AGVs at the Mercedes-Benz assembly plant, which use omnidirectional mobility, LiDAR and inertial measurement units (IMUs) for precise navigation. Cooperative localization ensures that follower AGVs continuously update their position relative to the leader, providing synchronized and reliable performance essential for today's dynamic production environments.

1.4 Organisation of the manuscript

Chapter 1 introduces the project, including references to real-world implementations within the automotive industry. Additionally, a comprehensive list of symbols used throughout the work is provided to support the understanding of equations, theoretical analysis and control configurations.

Chapter 2 presents the physical and mathematical modeling of the problem, covering system properties and developing a state-space representation. This chapter also includes a thorough model analysis, with linearization, open-loop eigenvalue configuration and the control design process—from reachability and observability analysis to the implementation and tuning of an integral action, state feedback stabilizer and observer design.

Chapter 3 provides an overview of our Simulink blocks, tested first on a simple hypothetical trajectory and then on a more realistic industrial-like trajectory used by AGVs. Finally, as a beyond, we explored AGV control in the presence of nearby obstacles, enhancing navigation strategies for more complex scenarios.

Chapter 4 presents our conclusions on the project and further investigations.

1.5 List of the symbols

Symbol	Definition	Reference System	Units
x_f	Position of the follower AGV in the x direction	Inertial reference frame	meters (m)
y_f	Position of the follower AGV in the y direction	Inertial reference frame	meters (m)
x_l^f	Position of the leader AGV wrt the follower in the x direction	Follower reference frame	meters (m)
y_l^f	Position of the leader AGV wrt the follower in the y direction	Follower reference frame	meters (m)
θ_f	Orientation (angle) of the follower AGV	Angle w.r.t. X-axis (global)	radians (rad)
v_f	Linear velocity of the follower AGV	Inertial reference frame	meters per second (m/s)
ω_f	Angular velocity of the follower AGV	Inertial reference frame	radians per second (rad/s)
x_l	Position of the leader AGV in the x direction	Inertial reference frame	meters (m)
y_l	Position of the leader AGV in the x direction	Inertial reference frame	meters (m)
θ_l	Orientation (angle) of the leader AGV	Angle w.r.t. X-axis (global)	radians (rad)
v_l	Linear velocity of the leader AGV	Inertial reference frame	meters per second (m/s)
ω_l	Angular velocity of the leader AGV	Inertial reference frame	radians per second (rad/s)
P_f^I	Position of the follower AGV in the inertial frame	Inertial reference frame	meters (m)
P_l^I	Position of the leader AGV in the inertial frame	Inertial reference frame	meters (m)
P_l^f	Position of the leader AGV relative to the follower	Follower's local frame	meters (m)
R_{FI}	Rotation matrix from inertial to follower's frame	-	unitless
x_R	Reference relative x position between leader and follower	Inertial reference frame	meters (m)
y_R	Reference relative y position between leader and follower	Inertial reference frame	meters (m)
$S(\omega_f)$	Skew-symmetric matrix for angular velocity ω_f	-	unitless
x_{rel}	Position of the obstacle wrt the follower AGV in the x direction	Follower reference frame	meters (m)
y_{rel}	Position of the obstacle wrt the follower AGV in the y direction	Follower reference frame	meters (m)
d_O	Distance between the obstacle and the follower AGV	Follower reference frame	meters (m)

Table 1.1: List of Symbols

Chapter 2

AGVs leader-follower control

2.1 Problem Formulation

To establish a foundation for the control problem, it is essential to first examine and define the primary components of the system model. This model serves as the basis for the entire control strategy formulation:

$$\dot{\mathbf{x}} = f(x, u, w) \quad \mathbf{x}(t_0) = x_0 \quad (2.1)$$

$$\mathbf{y} = h(x, u, w) \quad (2.2)$$

$$\mathbf{e} = h_e(x, u, w) \quad (2.3)$$

where

- \mathbf{x} is the State, $\mathbf{x} \in \mathbb{R}^n$
- \mathbf{d} represents Disturbances, $\mathbf{d} \in \mathbb{R}^{l_d}$
- \mathbf{r} represents References, $\mathbf{r} \in \mathbb{R}^{l_r}$
- \mathbf{u} are the Controls, $\mathbf{u} \in \mathbb{R}^p$
- \mathbf{y} is the Measurements, $\mathbf{y} \in \mathbb{R}^q$
- \mathbf{e} represents Goals, $\mathbf{e} \in \mathbb{R}^{l_m}$
- $\boldsymbol{\nu}$ represents Noises, $\boldsymbol{\nu} \in \mathbb{R}^q$

where $n, l_d, l_r, p, q, l_m \in \mathbb{N}$.

Finally, the exogenous vector \mathbf{w} is defined as:

$$\mathbf{w} = \begin{bmatrix} \mathbf{d} \\ \boldsymbol{\nu} \\ \mathbf{r} \end{bmatrix} \quad (2.5)$$

In order to solve the control problem, we need to lay some fundamental assumptions:

1. \mathbf{w} is not observable, so our system needs to be robust against disturbances and reach the goals despite them.
2. \mathbf{d} has to be bounded.
3. The reference vector \mathbf{r} is known, including all the first order derivatives.
4. If \mathbf{d} is bounded, it implies that all the other values are bounded.
5. If $\mathbf{u} \in \mathbb{R}^p$ and $\mathbf{e} \in \mathbb{R}^{l_m}$, we want $p \geq l_m$. In fact, if the relationship $p \geq l_m$ holds, then we can always find a solution for our control problem.
6. \mathbf{e} is readable from \mathbf{y} . Therefore, it must exist E such that $\mathbf{e} = E(\mathbf{y})$.

These assumptions are fundamental for the formalization of the whole control problem.

2.2 Model Analysis

2.2.1 Kinematic Model

The kinematic model describes the motion of Automated Guided Vehicles (AGVs) in a leader-follower configuration. In this configuration, one AGV, designated as the leader, navigates along a predefined path while the follower AGV adjusts its position to maintain a specified distance and alignment relative to the leader. The model assumes that each AGV operates under non-holonomic constraints, meaning that it can only move forward or backward and cannot directly change its lateral position without turning. In this project, according to Figure 2.1 below, we define:

- the state vector of the follower AGV as

$$\mathbf{x} = \begin{bmatrix} x_l^f \\ y_l^f \\ \theta_f \end{bmatrix} \quad (2.6)$$

- the control vector as:

$$\mathbf{u} = \begin{bmatrix} \omega_f \\ v_f \end{bmatrix} \quad (2.7)$$

- the exogenous vector as:

$$\mathbf{w} = \begin{bmatrix} \theta_l \\ v_l \\ \nu \\ x_R \\ y_R \end{bmatrix} \quad (2.8)$$

- the output vector as:

$$\mathbf{y} = \begin{bmatrix} x_l^f + \nu \\ y_l^f + \nu \end{bmatrix} \quad (2.9)$$

- the controlled output vector as:

$$\mathbf{e} = \begin{bmatrix} x_l^f + \nu - x_R \\ y_l^f + \nu - y_R \end{bmatrix} \quad (2.10)$$

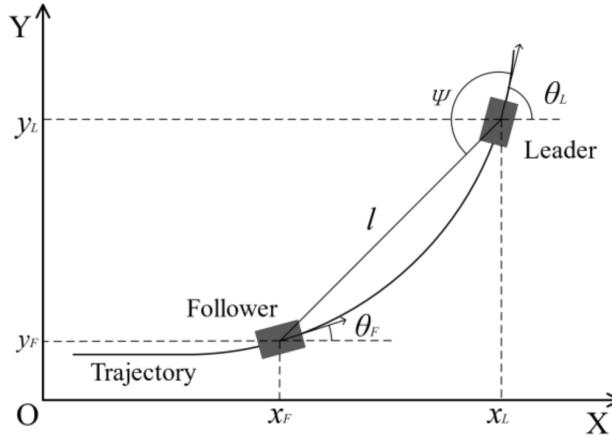


Figure 2.1: AGV leader-follower scheme

The position of the leader with respect to the follower in the inertial reference frame is expressed as:

$$l = \mathbf{P}_1^I - \mathbf{P}_f^I \quad (2.11)$$

where \mathbf{P}_1^I and \mathbf{P}_f^I represent the positions of the leader and follower in the inertial frame, respectively. To transform coordinates from the inertial reference frame to the follower's reference frame, we use the rotation matrix:

$$\mathbf{R}_{FI} = \begin{bmatrix} \cos(\theta_f) & \sin(\theta_f) & 0 \\ -\sin(\theta_f) & \cos(\theta_f) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.12)$$

The relative position of the leader with respect to the follower is expressed as:

$$\mathbf{P}_1^f = \mathbf{R}_{FI}(\mathbf{P}_1^I - \mathbf{P}_f^I) \quad (2.13)$$

Taking the time derivative of P_l^f , we obtain:

$$\dot{\mathbf{P}}_1^f = \frac{\partial \mathbf{R}_{FI}}{\partial \theta_f}(\mathbf{P}_1^I - \mathbf{P}_f^I)\dot{\theta}_f + \mathbf{R}_{FI}(\dot{\mathbf{P}}_1^I - \dot{\mathbf{P}}_f^I) \quad (2.14)$$

And substituting $\dot{\theta}_f = \omega_f$, the first term of the derivative can be computed:

$$\frac{\partial \mathbf{R}_{FI}}{\partial \theta_f} (\mathbf{P}_l^I - \mathbf{P}_f^I) \omega_f = \begin{bmatrix} -\sin(\theta_f) & \cos(\theta_f) & 0 \\ -\cos(\theta_f) & -\sin(\theta_f) & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_l^f \\ y_l^f \\ 0 \end{bmatrix}^I \omega_f \quad (2.15)$$

Taking advantage of the fact that the derivative of a rotation matrix can be expressed as the product of a skew-symmetric matrix \mathbf{S} and the rotation matrix itself, we express the time derivative of the relative position as:

$$\frac{\partial \mathbf{R}_{FI}}{\partial \theta_f} (\mathbf{P}_l^I - \mathbf{P}_f^I) \omega_f = \begin{bmatrix} 0 & \omega_f & 0 \\ -\omega_f & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \cos(\theta_f) & \sin(\theta_f) & 0 \\ -\sin(\theta_f) & \cos(\theta_f) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_l^f \\ y_l^f \\ 0 \end{bmatrix}^I \quad (2.16)$$

Recognizing that $\mathbf{R}_{FI} \cdot [x_l^f, y_l^f, 0]^T = \mathbf{P}_l^f$, the position of the leader with respect to the follower, and substituting it in 2.14 we derive the final expression for the velocity of the leader relative to the follower:

$$\dot{\mathbf{P}}_l^f = \mathbf{S}(\omega_f) \mathbf{P}_l^f + \mathbf{R}_{FI} \left(\begin{bmatrix} \cos(\theta_l) \\ \sin(\theta_l) \\ 0 \end{bmatrix} v_l - \begin{bmatrix} \cos(\theta_f) \\ \sin(\theta_f) \\ 0 \end{bmatrix} v_f \right) \quad (2.17)$$

Through additional simplifications, we obtain the following expression for the relative velocity:

$$\begin{aligned} \dot{\mathbf{P}}_l^f &= \omega_f \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_l^f - \begin{bmatrix} \cos(\theta_f) & \sin(\theta_f) & 0 \\ -\sin(\theta_f) & \cos(\theta_f) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta_f) \\ \sin(\theta_f) \\ 0 \end{bmatrix} v_f + \mathbf{R}_{FI} \begin{bmatrix} \cos(\theta_l) \\ \sin(\theta_l) \\ 0 \end{bmatrix} v_l = \\ &= \begin{bmatrix} y_l^f \\ -x_l^f \\ 0 \end{bmatrix} \omega_f - \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} v_f + \mathbf{R}_{FI} \begin{bmatrix} \cos(\theta_l) \\ \sin(\theta_l) \\ 0 \end{bmatrix} v_l \end{aligned} \quad (2.18)$$

Finally, this results in the system of model equations:

$$\begin{cases} \dot{x}_l^f = -v_f + y_l^f \omega_f + v_l \cos(\theta_l - \theta_f) \\ \dot{y}_l^f = -x_l^f \omega_f + v_l \sin(\theta_l - \theta_f) \\ \dot{\theta}_f = \omega_f \end{cases} \quad (2.19)$$

This can also be formulated in a more compact form by introducing disturbances as follows:

$$\begin{cases} \dot{x}_l^f = -v_f + y_l^f \omega_f + d_1(v_l, \theta_l, \theta_f) \\ \dot{y}_l^f = -x_l^f \omega_f + d_2(v_l, \theta_l, \theta_f) \\ \dot{\theta}_f = \omega_f \end{cases} \quad (2.20)$$

2.2.2 Sensors

To complete the system plant, it is crucial to first ensure the installation of the necessary sensors. As anticipated before, the error \mathbf{e} must be readable from the output \mathbf{y} of the measurement sensors, therefore those measurement vectors have to be defined according to the sensors which are embedded inside the system. First of all, let's recall the goal vector:

$$\mathbf{e} = h_e(x, u, w) = \begin{bmatrix} x_l^f + \nu - x_R \\ y_l^f + \nu - y_R \end{bmatrix} \quad (2.21)$$

The primary goals of the control system, as effectively represented by \mathbf{e} , are to:

- reduce $h_{e1} = x_l^f + \nu - x_R$, i.e. the longitudinal tracking error: the deviation of the follower's longitudinal position from the desired reference position determined by the leader;
- reduce $h_{e2} = y_l^f + \nu - y_R$, i.e. the lateral tracking error: the deviation of the follower's lateral position from the desired reference position determined by the leader.

Not all errors carry the same weight or importance within the system. Specifically, the error h_{e1} in the x-direction is less critical than the error h_{e2} in the y-direction. In fact, in an industrial environment, a small offset in the x-direction, where the follower is positioned slightly behind the expected position with respect to the leader, is typically not a major concern as it rarely interferes with the operational space or other equipment. On the

other hand, a deviation in the y-direction can be far more problematic, potentially causing collisions with obstacles or other elements within the plant, which could lead to disruptions or damage. Therefore, precise control over the y-position is essential for ensuring both safety and operational efficiency. To achieve the desired level of precision, two potential solutions were considered: a camera-based system or a GPS-based system. However, the decision was made to implement a camera-based solution because in an indoor environment, GPS would not provide the necessary accuracy, as signals can be disrupted or weakened by walls and other structural obstacles, leading to positioning errors. In contrast, a camera system offers higher precision for real-time tracking and positioning, making it more suitable for the confined space of an industrial setting, where accurate and reliable navigation is essential.

BASLER a2A5320-23ucPRO camera

To track our measurements and achieve the goal of replicating the trajectory defined by the leader, we selected the Basler ace 2 R camera (model a2A5320-23ucPRO), equipped with the Sony IMX542 sensor.



Figure 2.2: BASLER a2A5320-23ucPRO camera

This high-performance camera provides the necessary precision and reliability for real-time tracking, ensuring accurate position measurement and trajectory replication. The Sony IMX542 sensor offers excellent image quality and fast processing, making it an ideal choice for capturing detailed data in dynamic environments, such as those encountered in industrial settings. It costs €1.579 (excluded fees) per unit.

Specifications

General Specifications

a2A5320-23ucPRO	
Resolution (H x V Pixels)	5328 × 3040 (full resolution) 5320 × 3032 (default resolution)
Resolution	16.1 MP
Sensor Type	Sony IMX542-AAQJ-C Progressive scan CMOS Global shutter
Sensor Format	1.1"
Effective Sensor Diagonal	16.8 mm
Pixel Size (H x V)	2.74 × 2.74 μm
Frame Rate (at Default Settings)	22.3 fps 23.6 fps (Device Link Throughput Limit mode set to Off)
Product Family	ace 2 R 
Mono / Color	Color

Figure 2.3: BASLER a2A5320-23ucPRO camera

By visiting the [BASLER website](#), the specifications for this camera and its sensor can be found, as illustrated in Figure 2.3.

We assume that the sensor introduces a precision error of 15 mm at a detection distance of 1 meter, which corresponds to the reference distance between the leader and follower AGVs. Although this value is not explicitly stated in the sensor's datasheet, we have chosen it as a conservative estimate, considering a worst-case scenario that might be more demanding than the actual expected error. By adopting this value, we ensure that the system design accounts for potential inaccuracies, providing a safety margin and ensuring reliable performance under varying conditions. This assumption also allows for more robust controller tuning and system performance, in line with the

project's goals of ensuring stability and robustness in automatic control.

HONEYWELL HMC5883L magnetometer

One possibility for improving the system would have been to introduce a second type of sensor in addition to the cameras: the magnetometer. However, we decided against this approach. The main reason for this decision is that in an industrial environment, characterized by a surface filled with metallic objects, a magnetometer would not provide reliable performance. The magnetic fields generated by metals and machinery within the plant can significantly interfere with the magnetometer's ability to accurately measure the angle between the leader and follower AGVs. Magnetometers rely on detecting variations in the Earth's magnetic field to determine orientation, but in the presence of electromagnetic interference from metal objects and non-uniform magnetic fields produced by machinery, the readings can become distorted or inaccurate. Furthermore, the system was tested in open loop both with and without the magnetometer, and it was found to remain fully observable even without the sensor. This demonstrated that the magnetometer was unnecessary, enabling a reduction in overall costs. The decision to exclude the magnetometer is further explained in subsection 2.3.1, where the Open-Loop dynamics of the system are treated.



Figure 2.4: HONEYWELL HMC5883L magnetometer

2.2.3 Linearization

The initial step involves linearizing the system, which simplifies its control. This process relies on determining stationary operating points $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{w}^*)$

representing the states where the system is intended to operate. By focusing on these points, we can identify particular trajectories that allow the system to be modeled with adequate accuracy as follows:

$$\begin{aligned}\dot{\mathbf{x}} &= f(x, u, w) \quad \mathbf{x}(t_0) = x_0 \\ \mathbf{y} &= h(x, u, w) \\ \mathbf{e} &= h_e(x, u, w)\end{aligned}\tag{2.22}$$

where $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{w}^*)$ represents the stationary operating point, which serves as the reference point for calculating the linearized trajectories as:

$$\begin{aligned}\dot{\mathbf{x}}^* &= f(x^*, u^*, w^*) \quad \mathbf{x}^*(t_0) = x_0^* \\ \mathbf{y}^* &= h(x^*, u^*, w^*) \\ \mathbf{e}^* &= h_e(x^*, u^*, w^*)\end{aligned}\tag{2.23}$$

At this phase, attention must be given to the deviations arising during the system's linearization. A key aim is to identify an operating point that ensures these deviations remain minimal. The fluctuations relative to the chosen point can be expressed as:

$$\begin{aligned}\dot{\tilde{\mathbf{x}}} &= \dot{\mathbf{x}} - \dot{\mathbf{x}}^* = f(x, u, w) - f(x^*, u^*, w^*) \quad \tilde{\mathbf{x}}(t_0) = \mathbf{x}_0 - \mathbf{x}_0^* \\ \tilde{\mathbf{y}} &= \mathbf{y} - \mathbf{y}^* = h(x, u, w) - h(x^*, u^*, w^*) \\ \tilde{\mathbf{e}} &= \mathbf{e} - \mathbf{e}^* = h_e(x, u, w) - h_e(x^*, u^*, w^*)\end{aligned}\tag{2.24}$$

By leveraging $\tilde{\mathbf{u}} = \mathbf{u} - \mathbf{u}^*$ and $\tilde{\mathbf{w}} = \mathbf{w} - \mathbf{w}^*$, the perturbation $\dot{\tilde{\mathbf{x}}}$ can be rewritten as:

$$\begin{aligned}\dot{\tilde{\mathbf{x}}} &= \dot{\mathbf{x}} - \dot{\mathbf{x}}^* = f(\tilde{x} + x^*, \tilde{u} + u^*, \tilde{w} + w^*) - f(x^*, u^*, w^*) = \\ &= \frac{\partial f}{\partial \mathbf{x}} \Big|_{\substack{x=x^* \\ u=u^* \\ w=w^*}} \cdot \tilde{\mathbf{x}} + \frac{\partial f}{\partial \mathbf{u}} \Big|_{\substack{x=x^* \\ u=u^* \\ w=w^*}} \cdot \tilde{\mathbf{u}} + \frac{\partial f}{\partial \mathbf{w}} \Big|_{\substack{x=x^* \\ u=u^* \\ w=w^*}} \cdot \tilde{\mathbf{w}} + o(\|\tilde{x}^2\|, \|\tilde{u}^2\|, \|\tilde{w}^2\|)\end{aligned}\tag{2.25}$$

The derivation of equation (2.25) relies on the application of Taylor series expansion. By considering $f(x, u, w)$ and expanding it using the Taylor series around the chosen operating point, we arrive at the following expression:

$$\begin{aligned}f(x, u, w) &= f(x^*, u^*, w^*) + \frac{\partial f}{\partial \mathbf{x}} \cdot (\mathbf{x} - \mathbf{x}^*) + \frac{\partial f}{\partial \mathbf{u}} \cdot (\mathbf{u} - \mathbf{u}^*) + \frac{\partial f}{\partial \mathbf{w}} \cdot (\mathbf{w} - \mathbf{w}^*) + \\ &\quad \frac{\partial^2 f}{\partial \mathbf{x}^2} \cdot (\mathbf{x} - \mathbf{x}^*)^2 + \frac{\partial^2 f}{\partial \mathbf{u}^2} \cdot (\mathbf{u} - \mathbf{u}^*)^2 + \frac{\partial^2 f}{\partial \mathbf{w}^2} \cdot (\mathbf{w} - \mathbf{w}^*)^2 + \dots\end{aligned}\tag{2.26}$$

By disregarding all terms beyond the 2nd order, equation (2.26) can be substituted into the first equation of (2.24). This substitution results in equation (2.25). Following the same approach used for $\dot{\tilde{x}}$, the variables \tilde{y} and \tilde{e} can also be linearized as follows:

$$\begin{aligned}\tilde{y} &= \frac{\partial h}{\partial \mathbf{x}} \cdot \tilde{x} + \frac{\partial h}{\partial \mathbf{u}} \cdot \tilde{\mathbf{u}} + \frac{\partial h}{\partial \mathbf{w}} \cdot \tilde{\mathbf{w}} + o(\|\tilde{\mathbf{x}}^2\|, \|\tilde{\mathbf{u}}^2\|, \|\tilde{\mathbf{w}}^2\|) \\ \tilde{e} &= \frac{\partial h_e}{\partial \mathbf{x}} \cdot \tilde{x} + \frac{\partial h_e}{\partial \mathbf{u}} \cdot \tilde{\mathbf{u}} + \frac{\partial h_e}{\partial \mathbf{w}} \cdot \tilde{\mathbf{w}} + o(\|\tilde{\mathbf{x}}^2\|, \|\tilde{\mathbf{u}}^2\|, \|\tilde{\mathbf{w}}^2\|)\end{aligned}\quad (2.27)$$

Now, thanks to the introduction of the following matrices, the system can be expressed in a more compact and simplified form:

$$\begin{aligned}\mathbf{A} &= \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\substack{x=x^* \\ u=u^* \\ w=w^*}} & \mathbf{B}_1 &= \left. \frac{\partial f}{\partial \mathbf{u}} \right|_{\substack{x=x^* \\ u=u^* \\ w=w^*}} & \mathbf{B}_2 &= \left. \frac{\partial f}{\partial \mathbf{w}} \right|_{\substack{x=x^* \\ u=u^* \\ w=w^*}} \\ \mathbf{C} &= \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\substack{x=x^* \\ u=u^* \\ w=w^*}} & \mathbf{D}_1 &= \left. \frac{\partial h}{\partial \mathbf{u}} \right|_{\substack{x=x^* \\ u=u^* \\ w=w^*}} & \mathbf{D}_2 &= \left. \frac{\partial h}{\partial \mathbf{w}} \right|_{\substack{x=x^* \\ u=u^* \\ w=w^*}} \\ \mathbf{C}_e &= \left. \frac{\partial h_e}{\partial \mathbf{x}} \right|_{\substack{x=x^* \\ u=u^* \\ w=w^*}} & \mathbf{D}_{1e} &= \left. \frac{\partial h_e}{\partial \mathbf{u}} \right|_{\substack{x=x^* \\ u=u^* \\ w=w^*}} & \mathbf{D}_{2e} &= \left. \frac{\partial h_e}{\partial \mathbf{w}} \right|_{\substack{x=x^* \\ u=u^* \\ w=w^*}}\end{aligned}\quad (2.28)$$

Thus, the resulting linearized model is:

$$\begin{cases} \dot{\tilde{x}} = \mathbf{A} \cdot \tilde{x} + \mathbf{B}_1 \cdot \tilde{\mathbf{u}} + \mathbf{B}_2 \cdot \tilde{\mathbf{w}} \\ \tilde{y} = \mathbf{C} \cdot \tilde{x} + \mathbf{D}_1 \cdot \tilde{\mathbf{u}} + \mathbf{D}_2 \cdot \tilde{\mathbf{w}} \\ \tilde{e} = \mathbf{C}_e \cdot \tilde{x} + \mathbf{D}_{1e} \cdot \tilde{\mathbf{u}} + \mathbf{D}_{2e} \cdot \tilde{\mathbf{w}} \end{cases}\quad (2.29)$$

In order to consider those matrices (2.28) as constant in time, we need to impose the existence of special triplet $(\tilde{\mathbf{x}}^*, \tilde{\mathbf{u}}^*, \tilde{\mathbf{w}}^*)$ where $f(x^*, u^*, w^*) = 0 \implies \mathbf{x}^*, \mathbf{u}^*, \mathbf{w}^*$ are constant, defining our matrices time invariant. The set $(\mathbf{x}^*, \mathbf{u}^*, \mathbf{w}^*)$ is called *equilibrium triplet* and the associated system is the 'LTI' (Linear time invariant). The matrices in the linear model are all calculated in $(\mathbf{x}^*, \mathbf{u}^*, \mathbf{w}^*)$. At this stage, it is necessary to define the equilibrium triplet at which the system will be linearized. This must be done in such a way that the following condition is satisfied:

$$f(x^*, u^*, w^*) = \begin{bmatrix} \dot{x}_l^{f*} \\ y_l^{f*} \\ \dot{\theta}_f^* \end{bmatrix} = 0 \quad (2.30)$$

Putting equations (2.30) into (2.19), the equilibrium equations of the system are derived:

$$\begin{cases} 0 = -v_f^* + y_l^{f*}\omega_f^* + v_l(\cos(\theta_l - \theta_f^*)) \\ 0 = -x_l^{f*}\omega_f^* + v_l(\sin(\theta_l - \theta_f^*)) \\ 0 = \omega_f^* \end{cases} \quad (2.31)$$

When attempting to solve the system, we realized that our system is under-determined, as it has more unknowns than equations. As a result, some parameters have been imposed by us. The solution to the system yields the following results:

- the third equation is already solved and gives $\omega_f^* = 0$
- from the second equation two possible conditions arise:
 - $v_l = 0$, which is a trivial condition and not useful for our purposes, as it simply states that if both AGVs are stationary, they are in equilibrium, which is already implied
 - $\sin(\theta_l - \theta_f^*) = 0 \rightarrow \theta_l - \theta_f^* = 0 \rightarrow \theta_l = \theta_f^*$, which leads to the conclusion that the two AGVs must move along parallel paths to maintain equilibrium
- from the first equation, by substituting the previously obtained results, we get: $v_f^* = v_l$.

Subsequently, \mathbf{A}_{star} , \mathbf{B}_{1star} and \mathbf{B}_{2star} matrices of the linearized system are obtained through the **Jacobian Matlab** function applied on the symbolic expressions of such matrices, then evaluated in the equilibrium triplet. As a consequence, \mathbf{A}_{star} and the other matrices are directly related to the chosen equilibrium point:

$$\mathbf{A}_{star} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -0.1 \\ 0 & 0 & 0 \end{bmatrix} \quad (2.32)$$

$$\mathbf{B}_{1star} = \begin{bmatrix} 0 & -1 \\ -1 & 0 \\ 1 & 0 \end{bmatrix} \quad (2.33)$$

$$\mathbf{B}_{2star} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.34)$$

The complete set of matrices obtained by the linearization process and the parametric ones are attached in Appendix 4.

2.2.4 Control problem formalization

Taking into account the assumptions introduced in section 2.1, we can claim that:

- $\forall \tilde{\mathbf{x}}(t_0)$, it happens that $\tilde{\mathbf{x}}(t)$, $\tilde{\mathbf{u}}(t)$, $\tilde{\mathbf{y}}(t)$ and $\tilde{\mathbf{e}}(t)$ are bounded $\forall t \geq 0$. Therefore, this implies that the errors are bounded for every initial condition, ensuring that the system is robust and stable.
- Assuming that $\dot{\mathbf{d}} = 0$, meaning the disturbance \mathbf{d} is constant, and $\nu = 0$, we have $\limsup_{t \rightarrow +\infty} \|\tilde{\mathbf{e}}(t)\| = 0$. This ensures the control system's performance, with the disturbance remaining constant and the noise equal to zero.

2.3 Linear Model Analysis

The aim of this section is to analyze the behavior of the linearized plant in absence of any type of control system; such condition is known as *Open-loop*. It is important to understand if the system is self-stable or tends to drift away from equilibrium when the initial conditions not align with the equilibrium triplet. To do this it is necessary to study the eigenvalues and eigenvectors of the \mathbf{A}_{star} matrix, which are strictly linked to the dynamics of the whole system, as the boundedness of trajectories is closely linked to the eigenvalues of such matrix:

$$\dot{\tilde{\mathbf{x}}} = \mathbf{A}_{star}\tilde{\mathbf{x}} \quad (2.35)$$

To make the notation lighter, we will neglect the *tilde* symbol referring to the set of linearized coordinates.

2.3.1 Open Loop Dynamics

The goal here is to study the dynamics of the system by performing a change of coordinates to transform the original system into an easier (i.e. to be analyzed) one. This result can be obtained transforming our system into the *Jordan Canonical form* exploiting the *Jordan Transformation*.

The first step is to calculate the **eigenvalues** λ_i of the \mathbf{A}_{star} matrix of the linear system from equation (2.36), discarding the trivial solution as we are only interested in those vectors \mathbf{v} whose direction does not change over time. The eigenvalues $\lambda_i \in \mathbb{C}$ of equation (2.36) are the solutions to the polynomial equation (2.37), which is referred to as the *characteristic polynomial*. Finally, for each λ_i it is possible to find all the corresponding **eigenvectors** v_i from equation (2.37):

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0 \quad (2.36)$$

$$(\mathbf{A} - \lambda\mathbf{I})\mathbf{v} = 0 \quad (2.37)$$

In our system, the three eigenvalues λ_i are determined by the matrix \mathbf{A}_{star} , which results from the linearization process. Any changes in the equilibrium conditions, represented by the equilibrium triplet $(\mathbf{x}^*, \mathbf{u}^*, \mathbf{w}^*)$, will lead to a modification of the \mathbf{A}_{star} matrix and, consequently, a change in the eigenvalues. That's why it's important to mention that the equilibrium conditions have been selected based on the following assumptions:

- Leader and follower have the same linear speed ($v_l = v_f = 0.1m/s$)

- The two AGVs are aligned or parallel and horizontal ($\theta_l = \theta_f = 0$ rad)
- Null angular speed of the follower AGV ($w_f = 0$)

Eigenvalues of \mathbf{A}_{star} matrix have been studied through the *Matlab* command **eig** leading to:

$$\begin{aligned}\lambda_1 &= 0, \\ \lambda_2 &= 0, \\ \lambda_3 &= 0\end{aligned}\tag{2.38}$$

In this situation, there exists a single eigenvalue with algebraic multiplicity $\alpha = 3$. To study the geometric multiplicity, kernel of \mathbf{A}_{star} has been computed through *Matlab* **null** function, resulting in two independent vectors. This indicates that the geometric multiplicity (g_i) is less than the algebraic multiplicity (a_i) and so a chain of eigenvectors is needed to find all the solutions.

$$(A - \lambda_i I)v_{1,1,1} = 0\tag{2.39}$$

$$\begin{bmatrix} 0 - 0 \cdot 1 & 0 & 0 \\ 0 & 0 - 0 \cdot 1 & -0.1 \\ 0 & 0 & 0 - 0 \cdot 1 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 \\ -0.1 \cdot s_3 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \rightarrow s_3 = 0\tag{2.40}$$

$$v_{1,1,1} = \begin{bmatrix} s_1 \\ s_2 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}\tag{2.41}$$

$$v_{1,2,1} = \begin{bmatrix} s_1 \\ s_2 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}\tag{2.42}$$

Between the two vectors found, the chain continues using the second one, as the first would lead to an algorithm stop. So, moving on with the chain:

$$(A - \lambda_i I)v_{1,1,2} = v_{1,2,1}\tag{2.43}$$

$$\begin{bmatrix} 0 - 0 \cdot 1 & 0 & 0 \\ 0 & 0 - 0 \cdot 1 & -0.1 \\ 0 & 0 & 0 - 0 \cdot 1 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 \\ -0.1 \cdot s_3 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \rightarrow s_3 = -10 \quad (2.44)$$

$$v_{1,1,2} = \begin{bmatrix} 0 \\ 0 \\ s_3 \end{bmatrix} \rightarrow \begin{bmatrix} 0 \\ 0 \\ -10 \end{bmatrix} \quad (2.45)$$

Collecting then the eigenvalues in the \mathbf{V} matrix, one for each column, we obtain:

$$\mathbf{V} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -10 \end{bmatrix} \quad (2.46)$$

Moreover, it can be demonstrated that: if there are multiple eigenvalues with zero real part (and none with positive real part), the system is unstable if at least one of these eigenvalues has geometric multiplicity less than its algebraic multiplicity [8]. Consequently, for this system, a combined behavior is expected: two components will remain constant and a third will diverge linearly to infinite. To confirm this, the behavior of the system in open loop has been analyzed on *Matlab*:

```

A=A_star;
B=zeros(3,1);
C=[1,0,0;
   0,1,0;
   0,0,0];
D = zeros(3, 1);

t = linspace(0, 100, 10000);
u = 0*t;
x0 = rand(3, 1);

sys = ss(A, B, C, D); |
[y, t, x] = lsim(sys, u, t, x0);

```

Figure 2.5: Matlab script for Open-Loop analysis

To only study the “free” system response, **B** and **D** matrix have been set to zero. In a first open-loop simulation, the **C** matrix was set as a 3x3 identity matrix to observe all states, assuming the use of a camera and a magnetometer to measure x_l^f, y_l^f and θ_f . Subsequently, the system was analyzed in open loop with a modified matrix **C** defined as in Figure 2.5. It was observed that, in both cases, the system remained fully observable (a detailed observability analysis is presented in the following sections). Consequently, it was concluded that the use of a magnetometer could be avoided. This is particularly relevant because, in an industrial environment with AGVs, obtaining precise measurements from a magnetometer would be challenging due to interference caused by the presence of large amounts of iron in the surroundings. Moreover, the simulation has been performed with random initial conditions.

In the end, the system response has been plotted confirming the expectations:

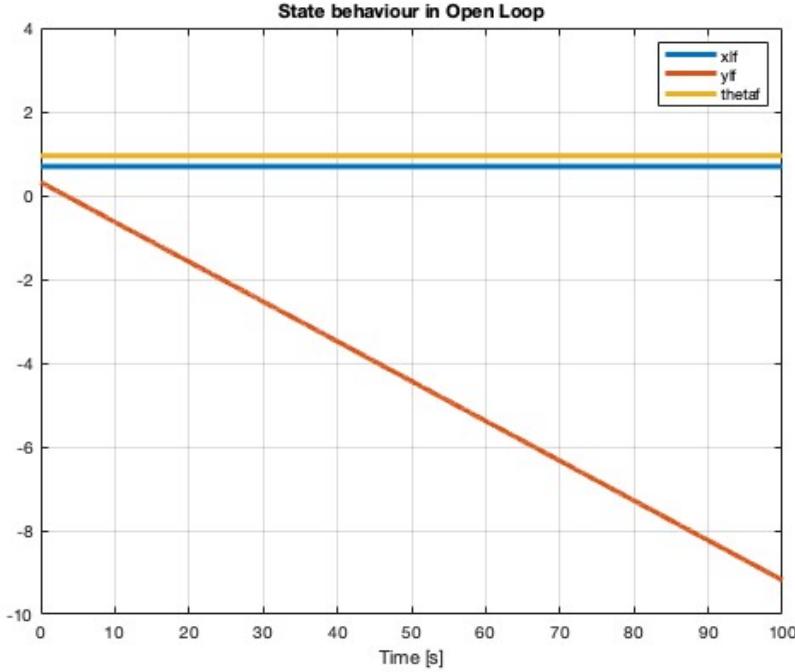


Figure 2.6: Open loop analysis

As expected the system is not BIBS due to not strictly negative eigenvalues. Therefore, we need to implement a control system that creates a close loop system and a Stabilizer, with the aim of making our system BIBS stable. Moreover, thanks to the Integral Action, we can make the already convergent behaviours to have a faster dynamics.

Now, in order to investigate our system dynamics we can exploit the already found eigenvalues and eigenvectors to perform a change of coordinates. This is done in order to have a system composed of independent sub-systems.

We can write the Jordan canonical form as:

$$\mathbf{J} = \mathbf{V}^{-1} \mathbf{A} \mathbf{V} \quad (2.47)$$

The Jordan block \mathbf{J} obtained through the transformation is an alternative

representation of the matrix \mathbf{A} :

$$\mathbf{J} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad (2.48)$$

Starting from the original system $\dot{\mathbf{x}} = \mathbf{Ax}$ and considering the transformation matrix $\mathbf{T} = \mathbf{V}^{-1}$, the system in the new coordinates can be written as $\mathbf{z} = \mathbf{T}\mathbf{x}$ and so it can be written that:

$$\dot{\mathbf{z}} = \mathbf{T}\dot{\mathbf{x}} = \mathbf{T}\mathbf{A}\mathbf{x} = \mathbf{V}^{-1}\mathbf{A}\mathbf{V}\mathbf{z} = \mathbf{J}\mathbf{z} \quad (2.49)$$

Therefore, the system can be studied according to:

$$\dot{\mathbf{z}} = \mathbf{J}\mathbf{z}, \quad \mathbf{z}(t_0) = \mathbf{z}_0, \quad (2.50)$$

$$\mathbf{z}(t) = e^{\mathbf{J}t}\mathbf{z}(t_0) \quad (2.51)$$

And in our case:

$$e^{\mathbf{J}t} = \mathbf{I} + \mathbf{J}t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix} \quad (2.52)$$

And so:

$$\begin{bmatrix} z_1(t) \\ z_2(t) \\ z_3(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} z_1(0) \\ z_2(0) \\ z_3(0) \end{bmatrix} = \begin{bmatrix} z_1(0) \\ z_2(0) + z_3(0)t \\ z_3(0) \end{bmatrix} \quad (2.53)$$

In the original coordinates, this relationship is represented by:

$$\mathbf{x} = \mathbf{T}^{-1}\mathbf{z} = \mathbf{V}\mathbf{z}. \quad (2.54)$$

Now, in order to understand how each mode affects our system, it is possible to analyze the product between the normalized eigenvector matrix (2.55) and the new coordinates behavior, since the original system behavior is given by (2.54). The normalized eigenvalues matrix \mathbf{V}_n is:

$$\mathbf{V}_n = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{bmatrix} \quad (2.55)$$

To facilitate visualization and understand how each mode influences the dynamics of individual components, we can plot a spider chart for each variable:

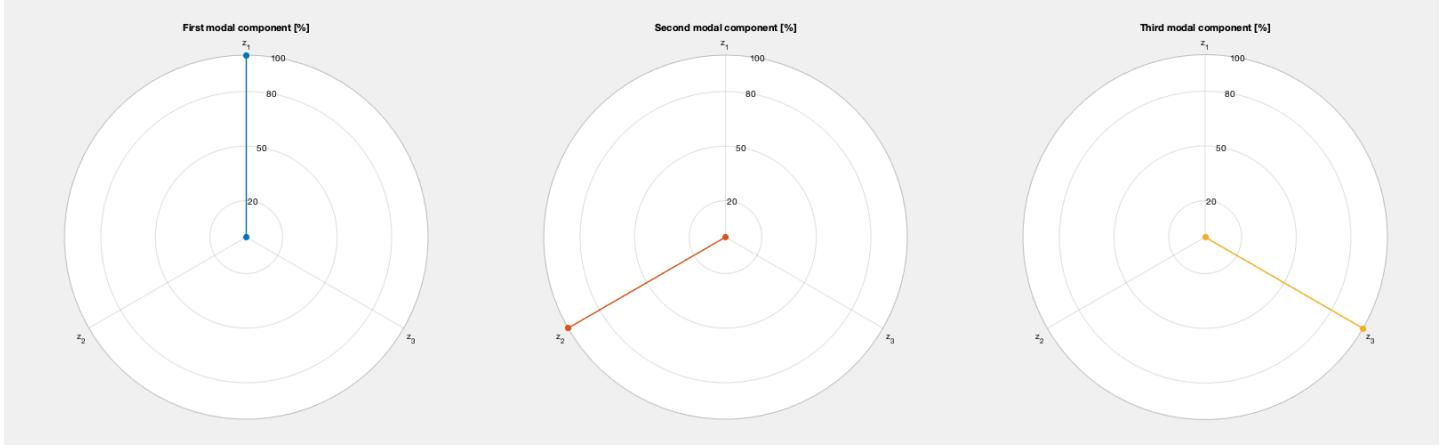


Figure 2.7: Modes spider charts

From the obtained results, some considerations can be made about how the modes influence the state components in your leader-follower AGV system:

- **First state component (x_l^f):** the relative distance between the leader and follower along the x -axis is influenced solely by the first mode (z_1). This implies that any initial deviation or perturbation in this mode directly affects x_l^f and no contributions come from z_2 or z_3 .

The same considerations can be applied to the second and third state components.

This structure suggests that each mode corresponds to specific aspects of the dynamics and controlling the modes can directly impact the respective state components.

2.4 Control

2.4.1 Reachability

In this section, a reachability study has been conducted. Starting from the initial problem $\dot{\tilde{x}} = \mathbf{A}\tilde{x} + \mathbf{B}_1\tilde{u}$, it is now necessary to analyze deeply the matrix \mathbf{A} . In fact, through its analysis we can discover which parts of \mathbf{A} can be modified by the control and which can not.

The subspace of \mathbb{R}^n called *reachability set* \mathbf{R} represents all the states that can be reached starting from the origin. The reachability matrix is defined as:

$$\mathbf{R} = \begin{bmatrix} \mathbf{B}_1 & \mathbf{AB}_1 & \mathbf{A}^2\mathbf{B}_1 & \dots & \mathbf{A}^{n-1}\mathbf{B}_1 \end{bmatrix} \quad (2.56)$$

For this specific system, reachability matrix \mathbf{R} is built as:

$$\mathbf{R} = \begin{bmatrix} \mathbf{B}_1 & \mathbf{AB}_1 & \mathbf{A}^2\mathbf{B}_1 \end{bmatrix} \quad (2.57)$$

Where $\mathbf{R} \in \mathbb{R}^{3 \times 6}$ because the dimension of the state vector \mathbf{x} is $n = 3$ and that of the control vector \mathbf{u} is $p = 2$.

The concept of reachability of the system is strongly linked to the calculation of the rank of \mathbf{R} ; in fact, the system is said to be *fully reachable* if \mathbf{R} is full rank. In this model, $n = 3$ and to claim full reachability of the system, $\text{rank}(\mathbf{R}) = 3$ is required. To verify this, the following *Matlab* code has been computed using the `ctrb` *Matlab* function to build the \mathbf{R} matrix:

```
if rank(R) == size(A_star,1)
    disp ('The system is Fully reachable')
else
    disp ('The system is NOT fully reachable')
end
```

Figure 2.8: Matlab script for reachability analysis

Running the code, the feedback is:

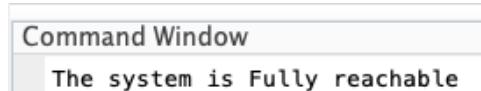


Figure 2.9: Result of reachability analysis

A fully reachable system is the most lucky case because it means that the control system can modify and control all the eigenvalues of the matrix \mathbf{A} . Furthermore, thanks to Automatic Control theory, one can claim that $\exists \mathbf{K}_S$ matrix such that $(\mathbf{A} + \mathbf{B}_1 \mathbf{K}_S)$ is *Hurwitz*. Consequently, there exist a state feedback control law which makes the analyzed system BIBS stable in closed loop; this is a very important result as it guarantees that the state of the system will always remains bounded in every moment and condition. Once the existence of the stabilizer matrix \mathbf{K}_S has been proved, the state feedback control law can be written as follow:

$$\tilde{\mathbf{u}} = \mathbf{K}_S \tilde{\mathbf{x}} \quad (2.58)$$

and substituting (2.58) inside the initial linearized system, we get:

$$\dot{\tilde{\mathbf{x}}} = (\mathbf{A} + \mathbf{B}_1 \mathbf{K}_S) \tilde{\mathbf{x}} \quad (2.59)$$

2.4.2 Integral Action

The action of a stabilizer matrix \mathbf{K}_S allows us to obtain a stable BIBS system, but this is not enough: an *Integral Action* is needed in order to drive the regulated output $\tilde{\mathbf{e}}$ to zero in the presence of unknown but constant bounded exogenous $\tilde{\mathbf{w}}$.

The error equation can be represented as:

$$\tilde{\mathbf{e}} = \mathbf{C}_e \tilde{\mathbf{x}} + \mathbf{D}_{1e} \tilde{\mathbf{u}} + \mathbf{D}_{2e} \tilde{\mathbf{w}} \quad (2.60)$$

In order to develop an integral action, a new variable $\boldsymbol{\eta}$ is introduced, so that:

$$\dot{\boldsymbol{\eta}} = \tilde{\mathbf{e}} \quad (2.61)$$

Expanding the system to include this integral action leads to the definition of an extended state \mathbf{x}_e as:

$$\mathbf{x}_e = \begin{bmatrix} \tilde{\mathbf{x}} \\ \boldsymbol{\eta} \end{bmatrix} \quad (2.62)$$

The dynamics of the extended system is described by:

$$\dot{\mathbf{x}}_e = \begin{bmatrix} \dot{\tilde{\mathbf{x}}} \\ \dot{\boldsymbol{\eta}} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C}_e & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}} \\ \boldsymbol{\eta} \end{bmatrix} + \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{D}_{1e} \end{bmatrix} \tilde{\mathbf{u}} + \begin{bmatrix} \mathbf{B}_2 \\ \mathbf{D}_{2e} \end{bmatrix} \tilde{\mathbf{w}} \quad (2.63)$$

In compact form:

$$\dot{\mathbf{x}}_e = \bar{\mathbf{A}} \mathbf{x}_e + \bar{\mathbf{B}}_1 \tilde{\mathbf{u}} + \bar{\mathbf{B}}_2 \tilde{\mathbf{w}} \quad (2.64)$$

where:

$$\bar{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C}_e & \mathbf{0} \end{bmatrix}, \quad \bar{\mathbf{B}}_1 = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{D}_{1e} \end{bmatrix}, \quad \bar{\mathbf{B}}_2 = \begin{bmatrix} \mathbf{B}_2 \\ \mathbf{D}_{2e} \end{bmatrix} \quad (2.65)$$

$\bar{\mathbf{A}}$, $\bar{\mathbf{B}}_1$ and $\bar{\mathbf{B}}_2$ matrices are reported in Appendix 4.

To verify the reachability of the extended system, the following *Matlab* code has been computed using the `ctrb` *Matlab* function to build the \mathbf{R}_e matrix:

```
R_e = ctrb(Ae, Be);
rango_R_e=rank(R_e);
if rango_R_e == size(Ae,1)
    disp('The extended system is Fully reachable')
else
    disp('The extended system is NOT Fully reachable')
end
```

Figure 2.10: Matlab script for reachability analysis

Running the code, the feedback is:

Figure 2.11: Result of reachability analysis

And so, there exists at least one matrix $\bar{\mathbf{K}}$ such that $(\bar{\mathbf{A}} + \bar{\mathbf{B}}_1 \bar{\mathbf{K}})$ is *Hurwitz*. By defining the control vector $\tilde{\mathbf{u}}$ as:

$$\tilde{\mathbf{u}} = \bar{\mathbf{K}} \tilde{\mathbf{x}}_e \quad \text{with} \quad \bar{\mathbf{K}} = \begin{bmatrix} \mathbf{K}_S & \mathbf{K}_I \end{bmatrix} \quad (2.66)$$

Placing (2.66) inside (2.63) one gets:

$$\begin{aligned} \tilde{\mathbf{u}} &= \begin{bmatrix} \mathbf{K}_S & \mathbf{K}_I \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}} \\ \boldsymbol{\eta} \end{bmatrix} \\ \dot{\mathbf{x}}_e &= (\bar{\mathbf{A}} + \bar{\mathbf{B}}_1 \bar{\mathbf{K}}) \mathbf{x}_e + \bar{\mathbf{B}}_2 \tilde{\mathbf{w}} \end{aligned} \quad (2.67)$$

where \mathbf{K}_S and \mathbf{K}_I are respectively the stabilizer and integral matrices. $\mathbf{K}_S \in \mathbb{R}^{p \times n}$ and $\mathbf{K}_I \in \mathbb{R}^{p \times lm}$, thus in our system \mathbf{K}_S is 2×3 and \mathbf{K}_I 2×2 matrix.

2.4.3 Observability

Up to this point, the state $\tilde{\mathbf{x}}$ of the system has been assumed to be known. However, this assumption does not always hold in practice, making it necessary to introduce a method to approximate the state when $\tilde{\mathbf{x}}$ is not directly accessible. It is necessary to introduce a mechanism that, in the absence of $\tilde{\mathbf{x}}$, gathers all the available measurements and control inputs to construct an accurate approximation of the state, enabling the generation of $\tilde{\mathbf{y}}_O$.

It is important to note that $\tilde{\mathbf{x}}_O$ cannot be computed by directly inverting the equation $\tilde{\mathbf{y}}(t) = \mathbf{C} \cdot \exp(\mathbf{A} \cdot t) \cdot \tilde{\mathbf{x}}_O$, as \mathbf{C} is not a square matrix. A key challenge arises from the potential presence of *non observable* states: these are states, denoted as $\tilde{\mathbf{x}}_0$, that produce $\tilde{\mathbf{y}} = 0$ and thus belong to the kernel of \mathbf{C} . This presents a significant issue as the control system is unable to distinguish these states from the origin.

To analyze and identify the non observable states of the system, the concept of *observability* must be introduced. However, it is first necessary to specify the expected measurements that can be obtained through the chosen sensors:

- the position of the follower AGV on x-axis, denoted as x_{lf}
- the position of the follower AGV on y-axis, denoted as y_{lf}

The number of measurements is less than the number of states to be estimated, which means that the system must exhibit certain *observability* properties to enable full state reconstruction. Given the number of states $n = 3$, the observability matrix is defined as follows:

$$\mathbf{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \end{bmatrix} \quad (2.68)$$

where $\mathbf{O} \in \mathbb{R}^{6 \times 3}$.

A very important matrix has been defined, as it allows the recognition of the indistinguishable states from the origin by the observer. By the Automatic Control theory, a LTI system is said to be completely observable if $\dim(\ker(\mathbf{O})) = \{0\}$ (i.e. if it contains only the origin). So, by looking at the $\text{rank}(\mathbf{O})$, if the $\text{rank}(\mathbf{O}) = n$, then (\mathbf{A}, \mathbf{C}) is fully observable. To verify this, the following *Matlab* code has been computed exploiting the `obsv` *Matlab* function to build the \mathbf{O} matrix:

```

O = obsv(A_star, C);

if rank(O) == n
    disp('The system is Fully observable');
else
    disp('The system is NOT Fully observable');
end

```

Figure 2.12: Matlab script for observability analysis

Running the code the feedback is:



The image shows a screenshot of a Matlab Command Window. The title bar says "Command Window". Inside the window, the text "The system is Fully observable" is displayed in a monospaced font.

Figure 2.13: Result of the observability analysis

This proves that $\text{rank}(\mathbf{O}) = n = 3$, making our system fully observable. In this way, there isn't the problem of non distinguishable states. Furthermore, if (\mathbf{A}, \mathbf{C}) is fully observable, there $\exists \mathbf{K}_O$ matrix such that $(\mathbf{A} - \mathbf{K}_O \mathbf{C})$ is *Hurwitz*.

2.5 Proposed Solution

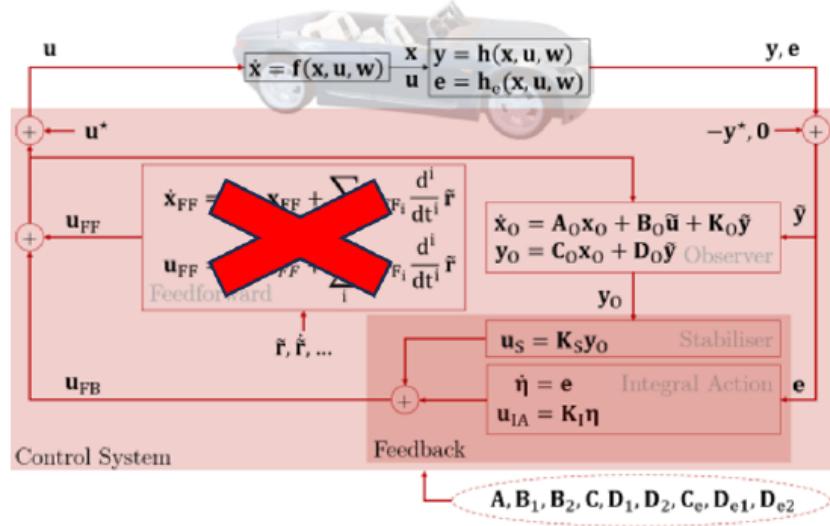


Figure 2.14: Plant and control system scheme

As clearly illustrated in Figure 2.14 above, the AGV is modeled as a plant characterized by inputs, disturbances, states and outputs. The plant is controlled through two different control contributions: *State Feedback* and *Integral Action*.

To generate the plant control input \mathbf{u} , since it is impossible to know exactly the state \mathbf{x} , which can be seen as an internal plant variable, an *Observer* block is needed in order to generate a system approximation \mathbf{y}_O to send as input to control blocks. To follow the state \mathbf{x} , *Observer* block need as input both the control \mathbf{u} and the system measurement \mathbf{y} .

Subsequently, the observer's output and the error \mathbf{e} are exploited by the *Feedback* block to compute the control action \mathbf{u}_{FB} . This control action ensures that both the state \mathbf{x} and the error \mathbf{e} remain bounded over time, thereby maintaining system stability. Additionally, the *integral action* incorporated within the feedback loop enhances the robustness of the control system. As shown, the *Feedforward* block has not been computed in our system for reasons that will be further explained later.

2.5.1 Optimal Control Model

The matrices \mathbf{K}_S , \mathbf{K}_I and \mathbf{K}_O can be determined by solving the Optimal Control Problem. This approach seeks to achieve the best trade-off between minimizing the error and the cost of the control actions. Specifically, the goal is to steer the error to zero while balancing the desire for rapid convergence with the need to limit the effort required by the controls, which might involve accepting a small residual error.

Let us start from the system:

$$\dot{\tilde{\mathbf{x}}} = \mathbf{A}\tilde{\mathbf{x}} + \mathbf{B}_1\tilde{\mathbf{u}} \quad (2.69)$$

$$\tilde{\mathbf{h}}_e = \mathbf{C}_e\tilde{\mathbf{x}} + \mathbf{D}_{1e}\tilde{\mathbf{u}} \quad (2.70)$$

The influence of the exogenous term $\tilde{\mathbf{w}}$ has been disregarded, enabling the design of a control law that remains robust against unknowns.

Then, let us recall the dynamics of the *extended state* \mathbf{x}_e as:

$$\dot{\tilde{\mathbf{x}}}_e = \bar{\mathbf{A}}\tilde{\mathbf{x}}_e + \bar{\mathbf{B}}_1\tilde{\mathbf{u}} \quad (2.71)$$

In order to find the most suitable matrices, that dynamics (2.71) is modified by adding an additional parameter $\alpha \geq 0$, whose purpose is to make the system less stable by modifying the eigenvalues of the \mathbf{A} matrix, thus making them less negative. This means that if the control law is able to manage a “less stable” system, it will surely be able to control the real one.

Then, the formula becomes:

$$\dot{\tilde{\mathbf{x}}}_e = \bar{\mathbf{A}}\tilde{\mathbf{x}}_e + \bar{\mathbf{B}}_1\tilde{\mathbf{u}} + \alpha\mathbf{I}\tilde{\mathbf{x}}_e \quad (2.72)$$

The meaning of α can be proved by recalling the change of coordinates (2.49): inside this analysis, the subscript “e” is neglected as this procedure can be applied to any linear system, not only the extended one:

$$\begin{aligned} \dot{\tilde{\mathbf{x}}} &= (\mathbf{A} + \alpha\mathbf{I})\tilde{\mathbf{x}} + \mathbf{B}\tilde{\mathbf{u}} \\ \dot{\mathbf{z}} &= \mathbf{T}\dot{\tilde{\mathbf{x}}} = \mathbf{T}(\mathbf{A} + \alpha\mathbf{I})\tilde{\mathbf{x}} + \mathbf{T}\mathbf{B}\tilde{\mathbf{u}} \\ &= (\mathbf{T}\mathbf{A}\mathbf{T}^{-1} + \alpha\mathbf{T}\mathbf{I}\mathbf{T}^{-1})\mathbf{z} + \bar{\mathbf{B}}\tilde{\mathbf{u}} \\ &= (\mathbf{J} + \alpha\mathbf{I})\mathbf{z} + \bar{\mathbf{B}}\tilde{\mathbf{u}} \end{aligned} \quad (2.73)$$

From $(\mathbf{J} + \alpha\mathbf{I})$, it can be observed that the eigenvalues shift closer to the imaginary axis, indicating that they become less negative. This implies a system that is less stable compared to the actual one. Consequently, the

parameter α serves to enhance the system's robustness against model uncertainties.

Moreover, a fake output ε is introduced as a linear combination of the extended state $\tilde{\mathbf{x}}_e$ and the control $\tilde{\mathbf{u}}$:

$$\varepsilon = \mathbf{C}_\varepsilon \tilde{\mathbf{x}}_e + \mathbf{D}_\varepsilon \tilde{\mathbf{u}} \quad (2.74)$$

Where the elements of ε are not necessarily measurable.

The *cost function* J is defined as:

$$J = \int_0^{+\infty} (\varepsilon^\top \mathbf{Q} \varepsilon + \tilde{\mathbf{u}}^\top \mathbf{R} \tilde{\mathbf{u}}) dt \quad (2.75)$$

where $J > 0$ belongs to scalar numbers. J has to be minimized by the most suitable control $\tilde{\mathbf{u}}$ which solves the Optimal Control Problem.

To have a well posed problem, the following conditions must be satisfied:

- $\mathbf{J} > 0$
- $\mathbf{Q} \succeq 0$ (semi-positive defined).

Increasing \mathbf{Q} keeps the error $\varepsilon(t)$ close to the origin, but at a higher cost. Setting $\mathbf{Q} = 0$ implies no concern about the error, resulting in zero cost but stability is not guaranteed. On the other hand, $\mathbf{Q} < 0$ leads to $\varepsilon(t) \rightarrow \infty$, which means the system would generate energy instead of consuming it to achieve the goal.

- $\mathbf{R} \succ 0$ (positive defined).

This ensures that the control is always penalized. It represents the cost of the control $\tilde{\mathbf{u}}$, therefore, it cannot be zero, which would stand for “the control comes for free”. The higher the magnitude of \mathbf{R} , the more $\tilde{\mathbf{u}}$ is considered expensive: then, if $\mathbf{R} = 0$, it means that a control law can be designed with an infinite magnitude. This cannot be feasible due to the power that has to be provided to the actuators inside the system, so the control must be penalized placing \mathbf{R} positive defined.

2.5.2 Selection of \mathbf{Q} and \mathbf{R} matrices

The selection of \mathbf{Q} and \mathbf{R} is related to the study case. In the general case, $\mathbf{Q} \in \mathbb{R}^{n_\varepsilon \times n_\varepsilon}$ where n_ε is the dimension of the extended state \mathbf{x}_e , while $\mathbf{R} \in \mathbb{R}^{p \times p}$ where p is the dimension of the control \mathbf{u} .

Taking \mathbf{C}_e as the identity matrix and \mathbf{D}_e as a null matrix, we can state that, in our case, $\mathbf{Q} \in \mathbb{R}^{5 \times 5}$ and $\mathbf{R} \in \mathbb{R}^{2 \times 2}$ and these matrices can be built manually considering that:

- the higher \mathbf{Q} , the more we care about how far we are from reaching the zero. Thus, the element $\varepsilon_{i,\max}$ has to be low.
- the higher \mathbf{R} , the less we use our control; therefore, the control will be less aggressive.

The matrices are:

$$\mathbf{Q} = \frac{1}{n_\varepsilon} \begin{bmatrix} \frac{1}{(\varepsilon_{1,\max})^2} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{(\varepsilon_{2,\max})^2} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{(\varepsilon_{3,\max})^2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{(\varepsilon_{4,\max})^2} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{(\varepsilon_{5,\max})^2} \end{bmatrix} \quad (2.76)$$

$$\mathbf{R} = \frac{1}{p} \begin{bmatrix} \frac{1}{(u_{1,\max})^2} & 0 \\ 0 & \frac{1}{(u_{2,\max})^2} \end{bmatrix} \quad (2.77)$$

where $\varepsilon_{i,\max}$ for $i = 1:3$ represents the maximum allowable error for every component of the state variable x_i and $\varepsilon_{i,\max}$ for $i = 4:5$ represents the maximum allowable error in the integral of the errors η_i .

The extended vector is composed in this way:

$$\mathbf{x}_e = \begin{bmatrix} x_l^f \\ y_l^f \\ \theta_f \\ \eta(1) \\ \eta(2) \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} \omega_f \\ v_f \end{bmatrix} \quad (2.78)$$

- $\varepsilon_{1,\max}$: maximum allowable error on x-axis [m]
- $\varepsilon_{2,\max}$: maximum allowable error on y-axis [m]
- $\varepsilon_{3,\max}$: maximum allowable error on angular orientation [rad]
- $\varepsilon_{4,\max}$: maximum allowable error on $\eta(1)$ [m]
- $\varepsilon_{5,\max}$: maximum allowable error on $\eta(2)$ [m]
- $u_{1,\max}$: control on angular speed of the follower [rad/s]
- $u_{2,\max}$: control on follower's speed [m/s]

2.5.3 Solution of the Optimal Control Problem

After analyzing the structure and significance of the matrices, the next step is to address the Optimal Control problem. The goal of the preceding analysis was solely to minimize the cost. However, to guarantee the system's stability and to design the most appropriate control input $\tilde{\mathbf{u}}$, the following assumptions are established:

- $(\bar{\mathbf{A}}, \mathbf{C}_\varepsilon)$ is detectable $\iff \ker(\mathbf{O}) = 0$;
- $(\bar{\mathbf{A}}, \bar{\mathbf{B}}_1)$ is stabilizable $\iff \text{rank}(\mathbf{R}) = n_\varepsilon$;
- $\tilde{\mathbf{u}}^T \mathbf{R} \tilde{\mathbf{u}} \succ 0$ (\mathbf{R} is positive definite);
- $\varepsilon^T \mathbf{Q} \varepsilon \succeq 0$ (\mathbf{Q} is semi-positive definite).

Recalling (2.72), (2.74) and (2.75), the system becomes:

$$\begin{cases} \dot{\tilde{\mathbf{x}}}_e = \bar{\mathbf{A}} \tilde{\mathbf{x}}_e + \bar{\mathbf{B}}_1 \tilde{\mathbf{u}} + \alpha \mathbf{I} \tilde{\mathbf{x}}_e \\ \varepsilon = \mathbf{C}_\varepsilon \tilde{\mathbf{x}}_e + \mathbf{D}_\varepsilon \tilde{\mathbf{u}} \\ J = \int_0^{+\infty} (\varepsilon^T \mathbf{Q} \varepsilon + \tilde{\mathbf{u}}^T \mathbf{R} \tilde{\mathbf{u}}) dt \end{cases} \quad (2.79)$$

The goal is to find \mathbf{u}^* such that J is minimised and the closed-loop system is BIBS stable. The matrix \mathbf{C}_ε has been chosen as the identity matrix, while \mathbf{D}_ε has been set equal to zero (null matrix) in order to ensure the observability of the pair $(\bar{\mathbf{A}}, \mathbf{C}_\varepsilon)$. Furthermore, this choice simplifies the tuning of the matrices \mathbf{Q} and \mathbf{R} because ε is not function of the control $\tilde{\mathbf{u}}$, as can be seen from (2.74) with $\mathbf{D}_\varepsilon = \mathbf{0}$.

Let us now introduce the *Hamiltonian function*:

$$H(\mathbf{x}_e, \tilde{\mathbf{u}}, \lambda) = \varepsilon^T \mathbf{Q} \varepsilon + \tilde{\mathbf{u}}^T \mathbf{R} \tilde{\mathbf{u}} + \lambda^T ((\bar{\mathbf{A}} + \alpha \mathbf{I}) \mathbf{x}_e + \bar{\mathbf{B}}_1 \tilde{\mathbf{u}}) \quad (2.80)$$

where $\lambda \in \mathbb{R}^n$ and it is called *co-state*. The Hamiltonian function leads to an optimal solution \mathbf{u}^* if two necessary, but not sufficient, conditions are defined:

$$1) \quad \frac{d\lambda}{dt} = - \left[\frac{\partial H}{\partial x_e} \right]^T \quad (2.81)$$

$$2) \quad \frac{\partial H}{\partial \tilde{\mathbf{u}}} |_{\tilde{\mathbf{u}}=\mathbf{u}^*} = 0 \quad (2.82)$$

Starting from 2) and replacing (2.74), we get:

$$\mathbf{u}^* = -\frac{1}{2} \bar{\mathbf{R}}^{-1} (\bar{\mathbf{B}}_1^T \lambda + 2 \mathbf{D}_\varepsilon^T \mathbf{Q} \mathbf{C}_\varepsilon) \mathbf{x}_e \quad (2.83)$$

where

$$\bar{\mathbf{R}} = (\mathbf{D}_\varepsilon^T \mathbf{Q} \mathbf{D}_\varepsilon + \mathbf{R}) \quad (2.84)$$

Defining $\lambda = 2\mathbf{S}\mathbf{x}_e$, where \mathbf{S} is called the *stabilizing solution*, we obtain:

$$\mathbf{u}^* = -\bar{\mathbf{R}}^{-1} (\bar{\mathbf{B}}_1^T \mathbf{S} + \mathbf{D}_\varepsilon^T \mathbf{Q} \mathbf{C}_\varepsilon) \mathbf{x}_e = \begin{bmatrix} \mathbf{K}_S & \mathbf{K}_I \end{bmatrix} \mathbf{x}_e \quad (2.85)$$

This procedure allows us to retrieve the shape of the \mathbf{K} matrix. Imposing $\dot{\mathbf{S}} = 0$ and combining it with 1), the *Algebraic Riccati Equation* (ARE) is obtained:

$$\begin{aligned} \mathbf{S} \bar{\mathbf{B}}^{-1} \bar{\mathbf{B}}^T \mathbf{S} - \mathbf{S} (\bar{\mathbf{A}} + \alpha \mathbf{I} - \bar{\mathbf{B}} \bar{\mathbf{R}}^{-1} \mathbf{D}_\varepsilon^T \mathbf{Q} \mathbf{C}_\varepsilon) \\ - (\bar{\mathbf{A}} + \alpha \mathbf{I} - \bar{\mathbf{B}} \bar{\mathbf{R}}^{-1} \mathbf{D}_\varepsilon^T \mathbf{Q} \mathbf{C}_\varepsilon)^T \mathbf{S} \\ - \mathbf{C}_\varepsilon^T \mathbf{Q} [\mathbf{I} - \mathbf{D}_\varepsilon \bar{\mathbf{R}}^{-1} \mathbf{D}_\varepsilon^T \mathbf{Q}] \mathbf{C}_\varepsilon = 0 \end{aligned} \quad (2.86)$$

In *Matlab*, the ARE (2.86) is solved using the `icare` command, which must be compared with equation (2.86) to get the correct result:

$$(\mathbf{X}_m, \mathbf{K}_m, \mathbf{L}_m) = \text{icare}(\mathbf{A}_m, \mathbf{B}_m, \mathbf{Q}_m, \mathbf{R}_m, \mathbf{S}_m, \mathbf{E}_m, \mathbf{G}_m) \quad (2.87)$$

where:

- $\mathbf{X}_m = \mathbf{S}$;
- $\mathbf{A}_m = \bar{\mathbf{A}} + \alpha \mathbf{I}$;
- $\mathbf{Q}_m = \mathbf{C}_\varepsilon^T \mathbf{Q} \mathbf{C}_\varepsilon$;
- $\mathbf{R}_m = \bar{\mathbf{R}}$;
- $\mathbf{S}_m = -\mathbf{C}_\varepsilon^T \mathbf{Q} \mathbf{D}_\varepsilon$;
- $\mathbf{E}_m = \mathbf{I}$;
- $\mathbf{G}_m = \mathbf{0}$;
- $\mathbf{K}_m = -\mathbf{K} = -\begin{bmatrix} \mathbf{K}_S & \mathbf{K}_I \end{bmatrix}$;
- \mathbf{L}_m contains the eigenvalues of the closed-loop system.

The subscript “m” stands for *Matlab* matrices.

Considerations about `icare` and \bar{R} :

Concerning the `icare` function, if the previous hypothesis are not satisfied, this function has no solution: as the dimension of ε is chosen, the detectability of the couple $(\bar{\mathbf{A}}, \mathbf{C}_\varepsilon)$ and the stabilizability of the couple $(\bar{\mathbf{A}}, \bar{\mathbf{B}}_1)$ have to be verified.

Moreover, it is known that $\bar{\mathbf{R}}$ can be seen as the cost of the control. As shown inside its definition (2.84), this matrix has a contribution provided by \mathbf{Q} . Inside the previous analysis \mathbf{D}_ε has been set null, thus the influence of \mathbf{Q} is not present. Instead, if \mathbf{D}_ε is different from a null matrix, \mathbf{Q} has an impact on the cost of the control. Indeed, in order to avoid such a high $\bar{\mathbf{R}}$, its other contribution \mathbf{R} has to be limited. This way of thinking explains the choice of $\mathbf{D}_\varepsilon = 0$.

2.5.4 Contribution of K_S and K_I

The effects of each element of \mathbf{K}_S and \mathbf{K}_I and their design are explained in the definition of the control $\tilde{\mathbf{u}}$:

$$\tilde{\mathbf{u}} = \mathbf{K} \begin{bmatrix} \tilde{\mathbf{x}} \\ \boldsymbol{\eta} \end{bmatrix} = \begin{bmatrix} \mathbf{K}_S & \mathbf{K}_I \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}} \\ \boldsymbol{\eta} \end{bmatrix} = \mathbf{K}_S \tilde{\mathbf{x}} + \mathbf{K}_I \boldsymbol{\eta} \quad (2.88)$$

The control is expressed as a linear combination of the state variables $\tilde{\mathbf{x}}$ and $\boldsymbol{\eta}$, where the elements of the matrix \mathbf{K} define the influence each component of $\tilde{\mathbf{x}}$ and $\boldsymbol{\eta}$ has on the control law. If one element of \mathbf{K}_S is larger than the others, it indicates that it will have a greater impact on the control output, and the same applies to \mathbf{K}_I .

Furthermore, modifying the values of $\varepsilon_{i,\max}$ can adjust the effect of these matrices on the control: the smaller the value of $\varepsilon_{i,\max}$, the stronger the influence of these matrices on the control action.

By extending the matrices:

$$\mathbf{K}_S = \begin{bmatrix} k_{s11} & k_{s12} & k_{s13} \\ k_{s21} & k_{s22} & k_{s23} \end{bmatrix}, \quad K_S \in \mathbb{R}^{2 \times 3} \quad (2.89)$$

$$\mathbf{K}_I = \begin{bmatrix} k_{i11} & k_{i12} \\ k_{i21} & k_{i22} \end{bmatrix}, \quad K_I \in \mathbb{R}^{2 \times 2} \quad (2.90)$$

Thus, the control $\tilde{\mathbf{u}}$ becomes:

$$\tilde{\mathbf{u}} = \begin{bmatrix} \omega_f \\ v_f \\ \theta_f \end{bmatrix} = \begin{bmatrix} k_{s11} & k_{s12} & k_{s13} \\ k_{s21} & k_{s22} & k_{s23} \end{bmatrix} \begin{bmatrix} x_i^f \\ y_i^f \\ \theta_i \end{bmatrix} + \begin{bmatrix} k_{i11} & k_{i12} \\ k_{i21} & k_{i22} \end{bmatrix} \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} \quad (2.91)$$

The role of each k_{ij} in the gain matrices \mathbf{K}_S (State feedback) and \mathbf{K}_I (Integral action) has been examined by conducting simulations under identical initial conditions, varying the parameters ε and u to observe how the gains respond. The results revealed that certain elements, specifically $k_{s11}, k_{s22}, k_{s23}, k_{i11}$ and k_{i22} , remain zero regardless the values of ε and u . The zero gains indicate that the corresponding states do not influence the associated control inputs. These results are consistent with the specific roles of the gains, as k_{ij} determines the contribution of state x_j to control input u_i . Larger values imply a strong influence on the control input, which enables rapid corrections but increase sensitivity to noise, while smaller or zero values reflect reduced or no impact, leading to smoother but potentially slower adjustments.

- When the parameter ε_1 is reduced by an order of magnitude (from an initial value of 0.3), an increase in k_{s21} by an order of magnitude in absolute value is observed. However, when ε_1 is increased by factors of 10, 100, or 1000, the values of all elements of \mathbf{K}_S remain largely unchanged. This indicates that increasing ε_1 by several orders of magnitude has a minimal effect on the values of \mathbf{K}_S and, consequently, on the control performance.

The same observations can be applied to $\varepsilon_2, \varepsilon_3, \varepsilon_4, \varepsilon_5$. In each case, a significant reduction of ε_i leads to noticeable changes in the corresponding elements of the gain matrix, while increasing ε_i by orders of magnitude results in minimal variation in the gains and control response.

- Starting from a value of $u_{1,\max} = 0, 3$ and pushing it, it is possible to see that this leads to higher values of all elements of \mathbf{K}_S and \mathbf{K}_I . Especially, if we increase $u_{1,\max}$ the elements with the higher absolute values will be the ones related to ω_f and η_2 that are k_{s12} and k_{i12}
- About $u_{2,\max}$, starting from a value $u_{2,\max} = 1, 3$ a similar behaviour to $u_{1,\max}$ can be observed with the difference that the elements with

the higher absolute values will be k_{s21} and k_{i21} the ones related to v_f and η_1

- Conversely, as expected, decreasing alternatively the $u_{i,\max}$ and therefore the maximum allowable values of the controls ω_f and v_f , results in a proportional decrease in the elements of \mathbf{K}_S and \mathbf{K}_I by the same order of magnitude

2.5.5 Design of K_S and K_I

Once the way in which the elements of \mathbf{Q} and \mathbf{R} influence the ones of \mathbf{K}_S and \mathbf{K}_I has been investigated, the tuning of those matrices can be developed, so the values $\varepsilon_{i,\max}$ and $u_{i,\max}$ have to be chosen properly. The tuning has been performed on a closed loop fed directly with the system state \mathbf{x} to achieve the best possible performances. For the same reason also sensors noise interference has been neglected. It is important to emphasize that the goal of this project is to design a control system for an industrial AGV (follower) tasked with tracking a “leader” AGV by adjusting v_f and ω_f . Before proceeding with the tuning process, it is essential to define the trajectory of the leader which the follower must, as much as possible, replicate:

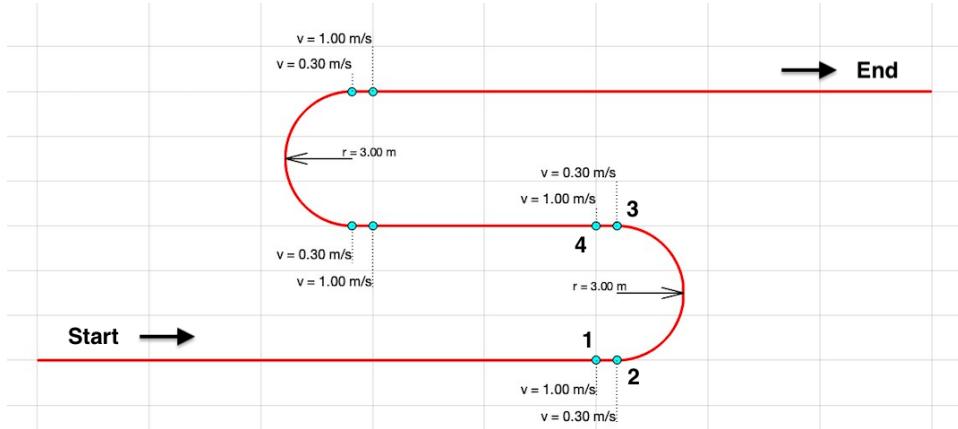


Figure 2.15: Leader’s path

As visible in Figure 2.15, the path is composed of a straight, a left turn (half circumference) with $R = 3m$, another shorter straight, a right turn also of $R = 3m$ and the final straight. The overall test lasts approximately 130s. This simulation duration is defined as a parameter in a *Matlab* function

and is automatically calculated as the sum of the time required to cover all individual segments. We can individuate 4 important positions (marked in the image) in which the AGV changes its speed:

- At the beginning of the path, the leader AGV starts from $v_f = 1 \text{ m/s}$, maintaining this speed until reaching point **(1)**. The straight segment lasts for 25 seconds. Once point **(1)** is reached, the AGV enters the curve and begins braking, reducing its speed from 1 m/s to 0.3 m/s . Using the initial speed, target speed, and deceleration, we calculated the time required to change velocity.
- At point **(2)**, the AGV starts the curve, maintaining a constant linear speed of 0.3 m/s and a constant angular speed. The angular speed is derived from the linear speed and the curve radius. The duration of the curve is determined using the speed in the curve and its radius.
- At point **(3)**, having exited the curve, the AGV begins to accelerate, gradually increasing its speed from 0.3 m/s to 1 m/s . Similarly to earlier, the time required to change velocity is calculated using the speed in the curve, the target speed, and the acceleration.
- Once at point **(4)**, the AGV has reached a speed of 1 m/s and continues at this velocity until the next deceleration.

We are looking for a reasonable balance between admissible errors $\varepsilon_{i,\max}$ and control strength $u_{i,\max}$, taking into account that what finally counts is the **Q/R** ratio. The choice of **R**, and consequently of $u_{1,\max}$ and $u_{2,\max}$, was based on an important consideration: making $u_{i,\max}$ too low would result in a very large **R**, thus discouraging the use of the controls as they become too “expensive.” On the other hand, excessively increasing the values of $u_{i,\max}$ would ultimately lead to overly aggressive control actions, potentially reaching unrealistic magnitudes.

For the choice of **Q**, we considered the specific objectives of industrial AGVs. In this context, it is crucial to impose stricter limits on $\varepsilon_{2,\max}$, associated with the error along the y -axis, as opposed to $\varepsilon_{1,\max}$ and $\varepsilon_{3,\max}$, which correspond to the error along the x -axis and the orientation θ , respectively. This reflects the importance of maintaining precise lateral alignment for effective AGV operation in industrial settings.

Before starting with the first attempt is worth highlighting that the condition $\alpha = 0$ will be maintained during the whole tuning process of **K_S** and **K_I**. The main reason for that is related to the fact that setting $\alpha \neq 0$ would

drastically increase the values in the gain matrices, potentially amplifying noise and leading to instability.

Firts test

For the first attempt, we started with nearly random values for \mathbf{Q} and \mathbf{R} , without focusing too much on their physical meaning, in order to better understand the system's behaviour:

$$\mathbf{Q} = \frac{1}{5} \cdot \begin{bmatrix} \frac{1}{10^2} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{0.1^2} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{0.1^2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{1^2} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{1^2} \end{bmatrix} \quad (2.92)$$

$$\mathbf{R} = \frac{1}{2} \cdot \begin{bmatrix} \frac{1}{0.2^2} & 0 \\ 0 & \frac{1}{0.01^2} \end{bmatrix} \quad (2.93)$$

It seems clear that the results will not be very good because we set a very small value for $u_{2,\max}$, resulting in weak control over v_f and a very large tolerance for the x -error.

The matrices \mathbf{K}_S and \mathbf{K}_I for the first test are:

$$\mathbf{K}_S = \begin{bmatrix} 0 & 1.5832 & -0.3583 \\ 0.1125 & 0 & 0 \end{bmatrix} \quad (2.94)$$

$$\mathbf{K}_I = \begin{bmatrix} 0 & 0.1265 \\ 0.0063 & 0 \end{bmatrix} \quad (2.95)$$

Anyway, the results given back by the 1st attempt are shown in the following Figure 2.16:

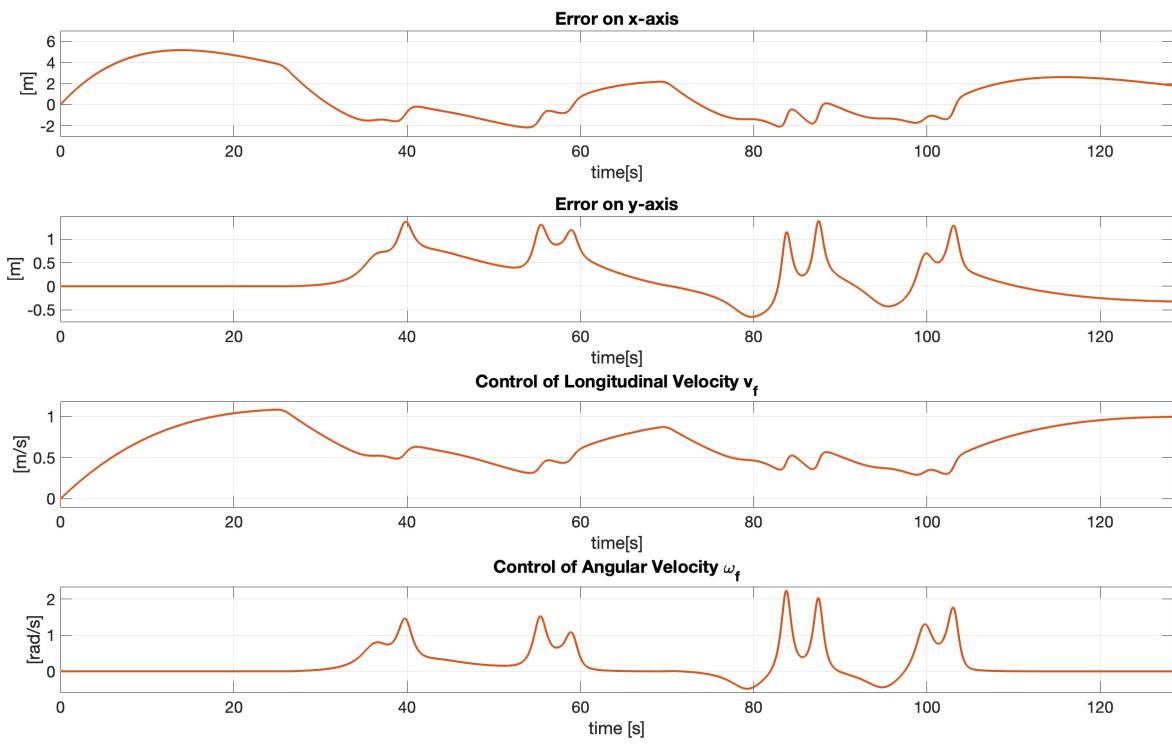


Figure 2.16: 1st test results

As can be seen, results are very poor! The maximum error on x-axis reaches, at its pick, a value of $\approx 6m$. Moreover, also the error on y-axis reaches values of $\approx 1.5m$: this is absolutely unacceptable!

Second test

Since the error along the x -axis is the most critical aspect of the first simulation, in the second test we reduce $\varepsilon_{1,\max}$ in order to improve the performance. Moreover, in order to improve also the result of y -error, we try to reduce the value of $\varepsilon_{5,\max}$ that is related to integral action on y reference error. For the second test, we are leaving the \mathbf{R} matrix unchanged:

$$\mathbf{Q} = \frac{1}{5} \cdot \begin{bmatrix} \frac{1}{0.3^2} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{0.1^2} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{0.1^2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{1^2} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{0.2^2} \end{bmatrix} \quad (2.96)$$

$$\mathbf{R} = \frac{1}{2} \cdot \begin{bmatrix} \frac{1}{0.2^2} & 0 \\ 0 & \frac{1}{0.01^2} \end{bmatrix} \quad (2.97)$$

The matrices \mathbf{K}_S and \mathbf{K}_I , for the second test are:

$$\mathbf{K}_S = \begin{bmatrix} 0 & 2.0125 & -0.1937 \\ 0.1144 & 0 & 0 \end{bmatrix} \quad (2.98)$$

$$\mathbf{K}_I = \begin{bmatrix} 0 & 0.6325 \\ 0.0063 & 0 \end{bmatrix} \quad (2.99)$$

The results given back by the 2nd attempt are shown below in Figure 2.17. It can be clearly observed that there are slight improvements in the error along the y -axis, while, despite the fact we reduced $\varepsilon_{1,\max}$ there has been almost no change in the error along the x -axis. This behavior is due to the fact that we still have a too low value of $u_{2,\max}$ and so the control is not

able to respect the constraint imposed on maximum allowable error along x -axis:

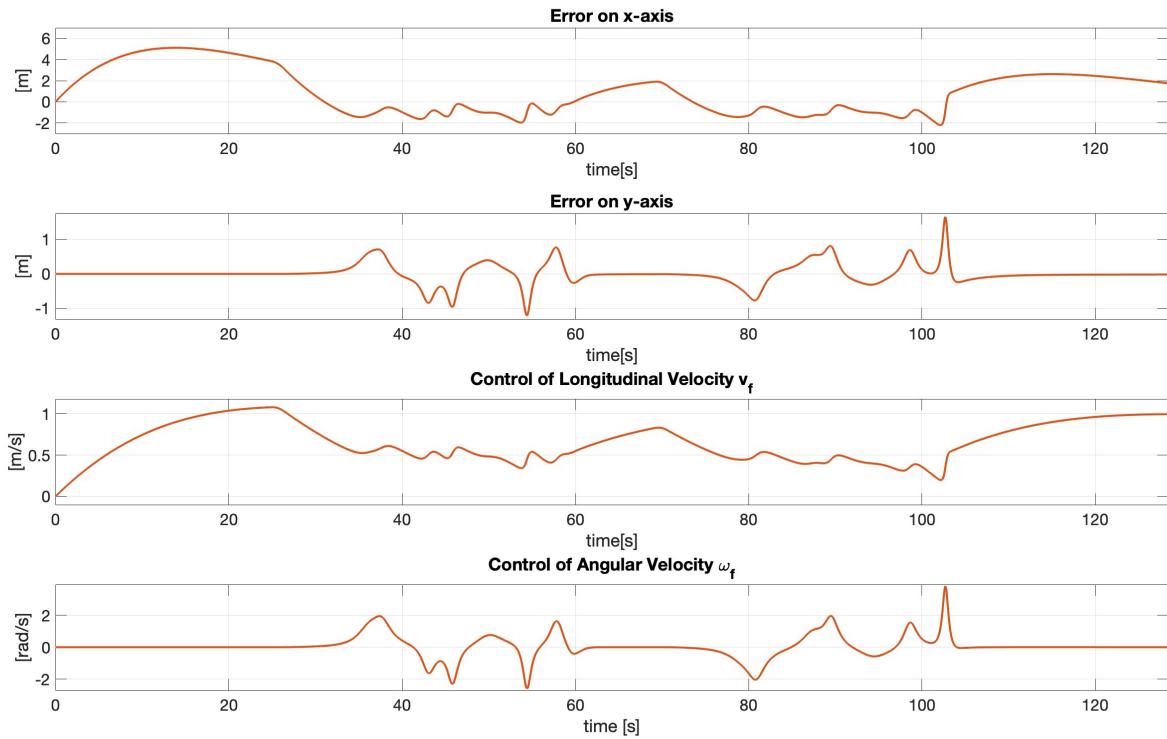


Figure 2.17: 2nd test results

Third test

In order to reduce the errors along both axes, we increase the values of $u_{i,\max}$, but without forgetting that these parameters are related to the controls that the actuator has to perform on v_f and ω_f , and so cannot be too large:

$$\mathbf{Q} = \frac{1}{5} \cdot \begin{bmatrix} \frac{1}{0.3^2} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{0.1^2} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{0.1^2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{1^2} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{0.2^2} \end{bmatrix} \quad (2.100)$$

$$\mathbf{R} = \frac{1}{2} \cdot \begin{bmatrix} \frac{1}{0.3^2} & 0 \\ 0 & \frac{1}{1.2^2} \end{bmatrix} \quad (2.101)$$

The matrices \mathbf{K}_S and \mathbf{K}_I , for the second test are:

$$\mathbf{K}_S = \begin{bmatrix} 0 & 2.8665 & -0.2433 \\ 2.8139 & 0 & 0 \end{bmatrix} \quad (2.102)$$

$$\mathbf{K}_I = \begin{bmatrix} 0 & 0.9847 \\ 0.7589 & 0 \end{bmatrix} \quad (2.103)$$

The resulting outcomes of the 3rd tuning test are shown in Figure 2.18:

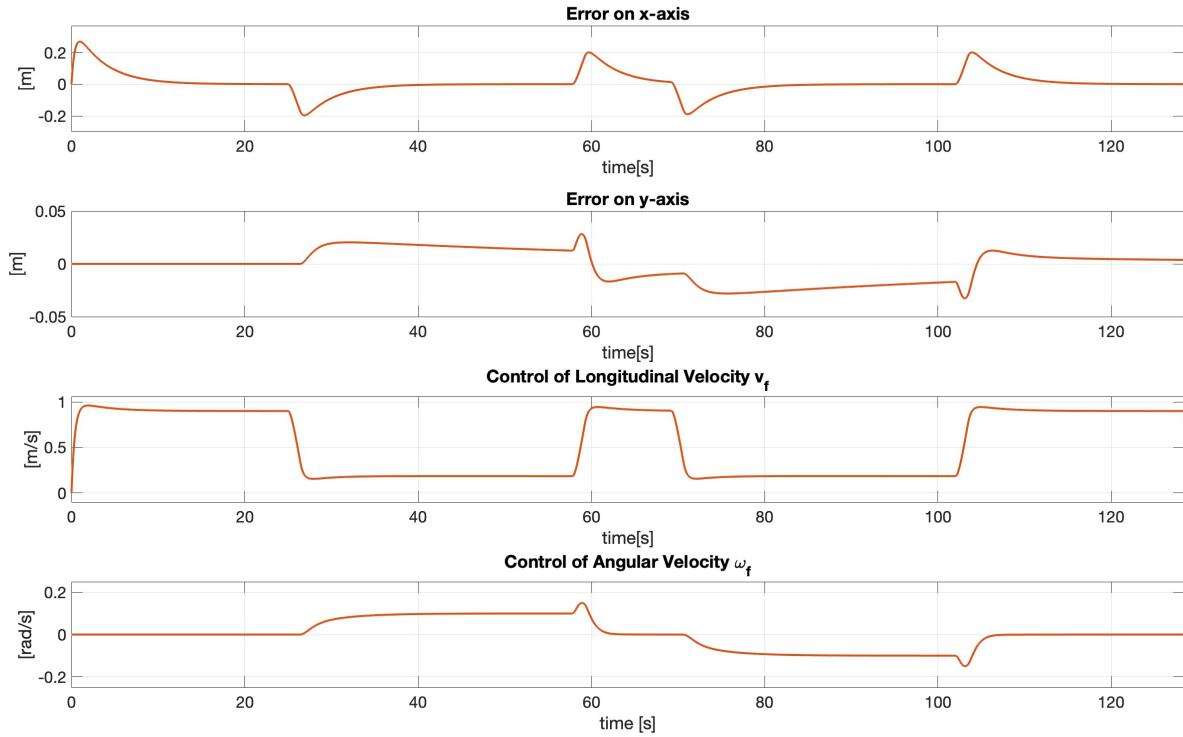


Figure 2.18: 3rd test results

The errors are limited both on the x and y axes. Notably, as shown in the Figure 2.18, greater accuracy has been maintained in limiting the error on the y-axis, which is considered more critical in industrial applications.

2.5.6 Observer and Kalman filter

As previously discussed in section 2.4.3, the state $\tilde{\mathbf{x}}$ cannot be directly accessed by the controller. Therefore, implementing a control law of the form $\tilde{\mathbf{u}} = \mathbf{K}_S \tilde{\mathbf{x}}$ requires a system that can estimate the state using sensor measurements from the plant. Additionally, section 2.4.3 demonstrated that the couple (\mathbf{A}, \mathbf{C}) is fully observable and this guarantees the existence of a gain matrix \mathbf{K}_O such that the matrix $(\mathbf{A} - \mathbf{K}_O \mathbf{C})$ is *Hurwitz*. As described in section 2.2.2 the plant has just one single sensor (camera) providing both the x and y distance between the leader and the follower AGV.

The observer matrix \mathbf{K}_O is built through the Kalman Filter: first, a dual system is introduced:

$$\begin{cases} \dot{\boldsymbol{\chi}} = \mathbf{A}^T \boldsymbol{\chi} + \mathbf{C}^T \boldsymbol{\nu} \\ \boldsymbol{\mu} = \mathbf{B}_2^T \boldsymbol{\chi} + \mathbf{D}_2^T \boldsymbol{\nu} \end{cases} \quad (2.104)$$

Dimensions of the vectors is the same as the primary system (2.4), even if there is no relationship between them:

$$\begin{aligned} \boldsymbol{\chi} &\in \mathbb{R}^n \\ \boldsymbol{\nu} &\in \mathbb{R}^q \\ \boldsymbol{\mu} &\in \mathbb{R}^r \end{aligned} \quad (2.105)$$

The Optimal Control 2.5.1 applied to the primary system in order to obtain \mathbf{K}_S and \mathbf{K}_I matrices can be now applied to the dual system (2.104) and exploited to get \mathbf{K}_{sd} matrix such that $\mathbf{A}^T + \mathbf{C}^T \mathbf{K}_{sd}$ is *Hurwitz*.

It can be proved that if the primary system is fully reachable, then the dual system is fully observable; moreover, if the primary system is fully observable, then the dual system is fully reachable. Consequently, the solution of the observability problem of the primary system \mathbf{K}_O will be the solution of the Optimal Control of the dual system, equal to $-\mathbf{K}_{sd}^T$. The optimal control applied to the dual system is:

$$\begin{cases} \dot{\boldsymbol{\chi}} = (\mathbf{A}^T + \alpha_d \mathbf{I}) \boldsymbol{\chi} + \mathbf{C}^T \boldsymbol{\nu} \\ \boldsymbol{\mu} = \mathbf{B}_2^T \boldsymbol{\chi} + \mathbf{D}_2^T \boldsymbol{\nu} \\ J = \int_0^{+\infty} (\boldsymbol{\mu}^T \mathbf{Q} \boldsymbol{\mu} + \boldsymbol{\mu}^T \mathbf{R} \boldsymbol{\mu}) dt \end{cases} \quad (2.106)$$

Then, the Algebraic Riccati Equation is addressed to minimize the cost function J using the `icare` function available in *Matlab*:

$$(\mathbf{X}_{md}, \mathbf{K}_{md}, \mathbf{L}_{md}) = \text{icare}(\mathbf{A}_{md}, \mathbf{B}_{md}, \mathbf{Q}_{md}, \mathbf{R}_{md}, \mathbf{S}_{md}, \mathbf{E}_{md}, \mathbf{G}_{md}) \quad (2.107)$$

where the matrices of the dual system have values:

$$\begin{aligned}
\mathbf{A}_d &= \mathbf{A}^\top, & \mathbf{B}_d &= \mathbf{C}^\top, & \mathbf{C}_d &= \mathbf{B}_2^\top, & \mathbf{D}_d &= \mathbf{D}_2^\top \\
\mathbf{A}_{md} &= \mathbf{A}_d + \lambda_d \mathbf{I} \\
\mathbf{B}_{md} &= \mathbf{B}_d \\
\mathbf{Q}_{md} &= \mathbf{C}_d^\top \mathbf{Q}_d \mathbf{C}_d \\
\mathbf{R}_{md} &= \bar{\mathbf{R}}_d \quad \text{and} \quad \bar{\mathbf{R}}_d = \mathbf{R}_d + \mathbf{D}_d^\top \mathbf{Q}_d \mathbf{D}_d \\
\mathbf{S}_{md} &= \mathbf{C}_d^\top \mathbf{Q}_d \mathbf{D}_d \\
\mathbf{E}_{md} &= \mathbf{I} \\
\mathbf{G}_{md} &= \mathbf{0}
\end{aligned} \tag{2.108}$$

The result is the observability matrix:

$$\mathbf{K}_O = \mathbf{K}_{md}^\top \tag{2.109}$$

This matrix can be placed inside the Luenberger Observer defined as:

$$\mathbf{A}_O = (\mathbf{A} - \mathbf{K}_O \mathbf{C}) \quad \mathbf{B}_O = \begin{bmatrix} \mathbf{B}_1 & \mathbf{K}_O \end{bmatrix} \tag{2.110}$$

It is important to clarify that equation 2.110, which defines the matrices \mathbf{A}_O and \mathbf{B}_O , is derived solely from the mathematical structure of the system and remains valid independently of the specific observer type. These matrices describe the general dynamics of the observer and are not influenced by the interpretations or roles of \mathbf{Q}_d and \mathbf{R}_d . This ensures that the implemented observer's structure is universally applicable and not restricted to cases where the Kalman filter is equivalent to the Luenberger observer. Therefore, the dynamics of the observer will be described by the following:

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}_O \hat{\mathbf{x}} + \mathbf{B}_O \begin{bmatrix} \tilde{\mathbf{u}} \\ \tilde{\mathbf{y}} \end{bmatrix} \tag{2.111}$$

$$\mathbf{y}_O = \mathbf{C}_O \hat{\mathbf{x}} + \mathbf{D}_O \begin{bmatrix} \tilde{\mathbf{u}} \\ \tilde{\mathbf{y}} \end{bmatrix} \tag{2.112}$$

Meaning and design of \mathbf{Q}_d and \mathbf{R}_d :

In our plant matrices \mathbf{Q}_d and \mathbf{R}_d have been defined as follow:

$$\mathbf{Q}_d = \begin{bmatrix} w_{1max}^2 & 0 & 0 & 0 & 0 \\ 0 & w_{2max}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.113)$$

$$\mathbf{R}_d = \begin{bmatrix} \sigma_{camx}^2 & 0 \\ 0 & \sigma_{camy}^2 \end{bmatrix} \quad (2.114)$$

Once the most appropriate sensors for the plant are selected, \mathbf{R}_d is fixed because it represents their standard deviations. Consequently, the values of \mathbf{Q}_d and λ_d can be adjusted in order to design the optimal observer \mathbf{K}_O .

Before fine-tuning the observer matrix \mathbf{K}_O , it is essential to analyze the role of the matrices \mathbf{Q}_d and \mathbf{R}_d . The matrix $\bar{\mathbf{R}}_d$, which is equal to \mathbf{R}_{md} , consists of two components: \mathbf{R}_d , the noise co-variance matrix that contains the standard deviations or uncertainties of the sensors as listed in their datasheets, and $\mathbf{D}_d^\top \mathbf{Q}_d \mathbf{D}_d$. The latter term represents a balance between the sensor noise and the speed in achieving goals.

The matrix \mathbf{S}_{md} is defined as $\mathbf{C}_d^\top \mathbf{Q}_d \mathbf{D}_d$ and its role can be understood as follows: if $\mathbf{S}_{md} \neq 0$, disturbances and noise are linked; however, in this analysis, \mathbf{S}_{md} is set to zero.

Furthermore, the matrix \mathbf{Q}_{md} , defined as $\mathbf{C}_d^\top \mathbf{Q}_d \mathbf{C}_d$, will be better understood by recalling the relationship between \mathbf{K}_{sd} and \mathbf{K}_O : \mathbf{K}_{sd} is obtained through $-\bar{\mathbf{R}}_d^{-1} (\mathbf{C}_d^\top \mathbf{Q}_d \mathbf{D}_d + \mathbf{B}_d^\top \mathbf{S}_{md})$. $\bar{\mathbf{R}}_d^{-1}$ tells us that the noisier the sensors, the lower they are used, in order to perceive their noise as low as possible.

The matrix \mathbf{Q}_{md} , being the product $\mathbf{C}_d^\top \mathbf{Q}_d \mathbf{C}_d$, suggests that as \mathbf{Q}_{md} increases, the uncertainties of the system rise, consequently increasing the influence of \mathbf{K}_{sd} . In conclusion, as \mathbf{Q}_{md} grows, \mathbf{S}_{md} also increases, since \mathbf{S}_{md} reflects the uncertainties within the system.

After conducting this analysis, the form of \mathbf{Q}_d becomes clearer: the plant has two disturbances, which means the matrix \mathbf{Q}_d contains two non-zero diagonal elements, corresponding to the maximum uncertainties of the disturbances that the system is capable of predicting.

2.5.7 Tuning procedure of K_O

For the observer tuning process, our primary objective is to achieve an effective trade-off between fast dynamic response and robust stability, while ensuring that the noise introduced remains within acceptable limits. To achieve this, we adopt an approach similar to the one used for tuning the state feedback (\mathbf{K}_S) and integral action (\mathbf{K}_I) matrices, ensuring consistency across the design process; with \mathbf{R}_d fixed, the focus shifts to tuning the two remaining parameters, λ_d and \mathbf{Q}_d . To provide a starting point that reflects the system's expected behavior, the initial conditions for the observer are defined as follows:

- $x_l^f = 1$ [m]
- $y_l^f = 0$ [m]
- $\theta_f = 0$ [rad]

First test

$$\mathbf{R}_d = \begin{bmatrix} 0.015^2 & 0 \\ 0 & 0.015^2 \end{bmatrix} \quad (2.115)$$

$$\mathbf{Q}_d = \begin{bmatrix} 0.1^2 & 0 & 0 & 0 & 0 \\ 0 & 0.1^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.116)$$

$$\lambda_d = 0.01 \quad (2.117)$$

In this initial attempt, we chose to begin with a conservative approach, using random but low values for the parameters. The results of the 1st observer tuning test are showed in Figure 2.19:

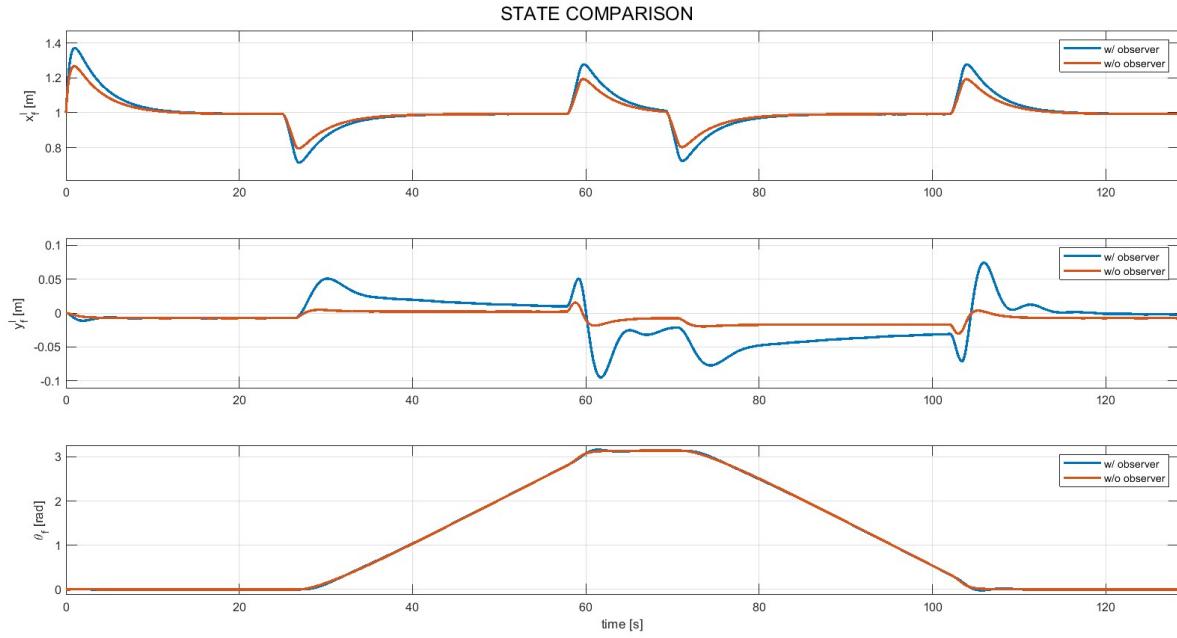


Figure 2.19: 1st test results

While the resulting errors are relatively small (almost acceptable, particularly for the follower's orientation angle) it is evident that the system's response can be improved and made faster. This indicates room for improvement by pushing on λ_d and \mathbf{Q}_d .

Second test

$$\mathbf{R}_d = \begin{bmatrix} 0.015^2 & 0 \\ 0 & 0.015^2 \end{bmatrix} \quad (2.118)$$

$$\mathbf{Q}_d = \begin{bmatrix} 10^2 & 0 & 0 & 0 & 0 \\ 0 & 10^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.119)$$

$$\lambda_d = 1 \quad (2.120)$$

In the second attempt, we applied more aggressive parameter tuning, which, as expected, reduced the amplitude of the errors and improved accuracy, but at the expense of increased noise and instability.

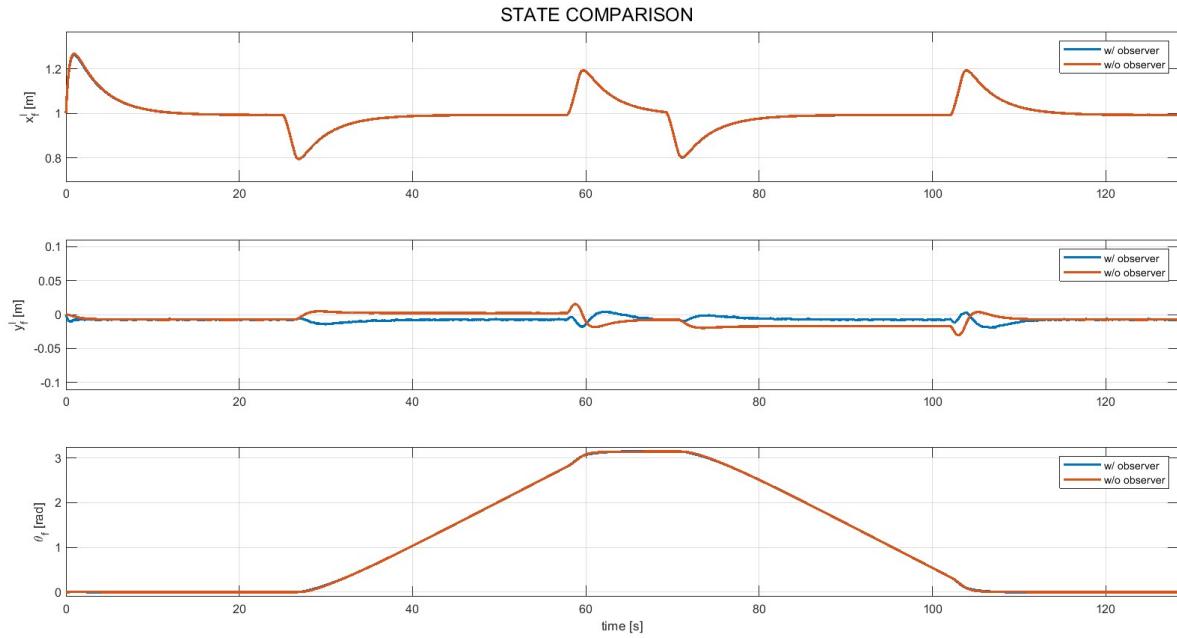


Figure 2.20: 2nd test results

The increase in noise and instability is most noticeable in the comparison of y_l^f (2.21), particularly at the beginning of turns, where the control becomes excessively sensitive to variations and noise.

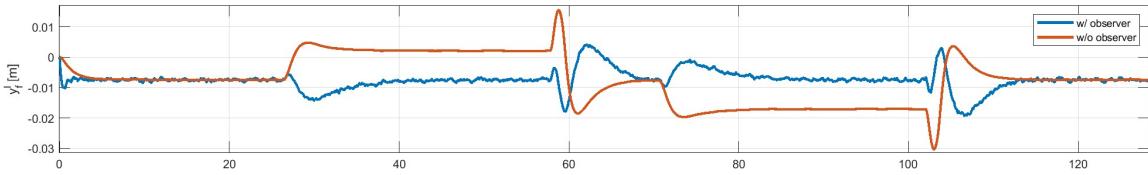


Figure 2.21: 2nd test results

Chosen configuration

To reduce the excessive noise of the previous test, we decreased the value of λ_d by one order of magnitude and simultaneously reduced the values of $w_{i,\max}$:

$$\mathbf{R}_d = \begin{bmatrix} 0.015^2 & 0 \\ 0 & 0.015^2 \end{bmatrix} \quad (2.121)$$

$$\mathbf{Q}_d = \begin{bmatrix} 5^2 & 0 & 0 & 0 & 0 \\ 0 & 1^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.122)$$

$$\lambda_d = 0.1 \quad (2.123)$$

After extensive testing with various parameter configurations, we identified this setup as the optimal compromise between error magnitude, response speed, and noisiness.

$$\mathbf{K}_O = \begin{bmatrix} 66.7667 & 0 \\ 0 & 33.6335 \\ 0 & -66.8670 \end{bmatrix} \quad (2.124)$$

The results of the 3rd observer tuning test are showed in Figure 2.22:

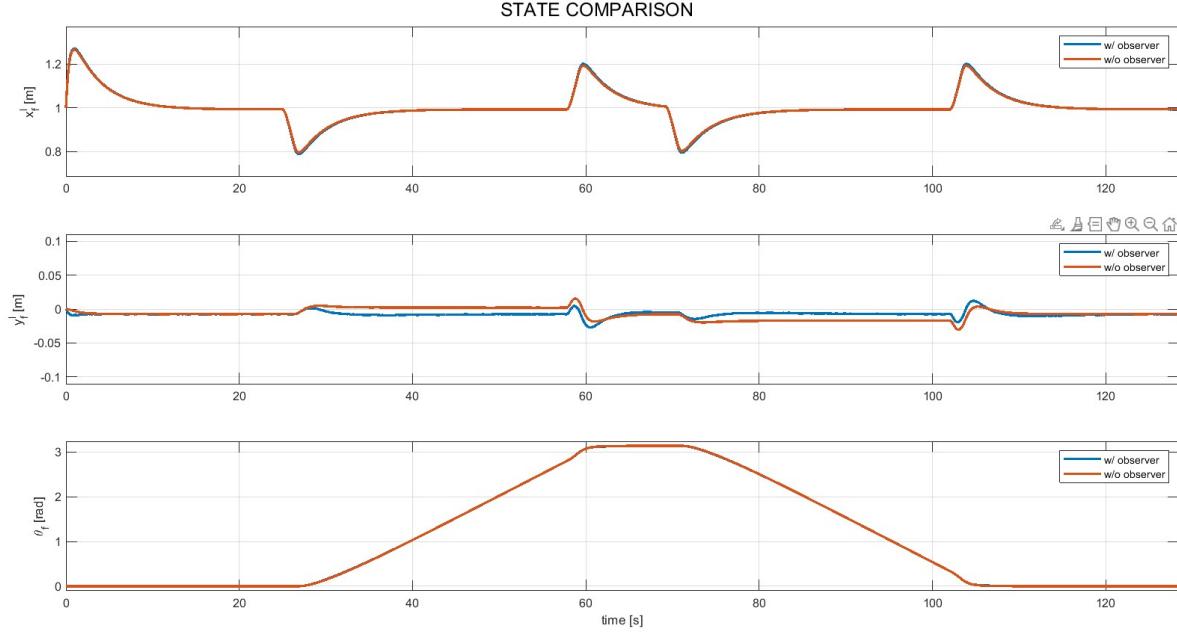


Figure 2.22: 3rd test results

Furthermore, we prioritized increasing $w_{1,\max}$ over $w_{2,\max}$, as it is closely related to the uncertainty in the leader's angle and has the most significant influence on y_l^f (2.23). This decision is supported by the results of the previous tests, which clearly demonstrate that y_l^f is the most critical state component in this tuning procedure.

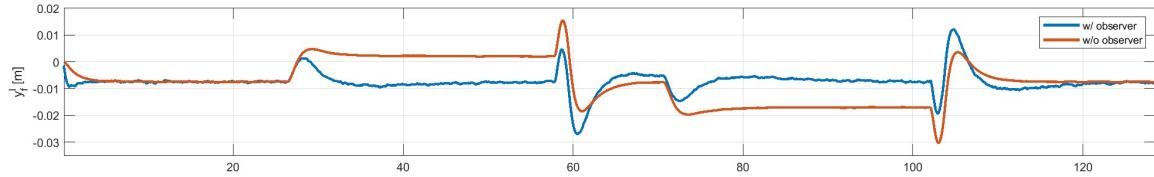


Figure 2.23: 3rd test results: zoom on y_l^f

Consideration about state estimation of the Observer

As shown in the graphs, regardless of the tuning of the observer parameters, there is always an error between the state estimated by the observer and the

state y_l^f . This error occurs along the trajectory curves and is due to the fact that the observer does not process the exogenous vector \mathbf{w} that contains the disturbances θ_l and v_l .

As a result, the observer is unable to account for the external disturbances that affect the system's dynamics during the path tracking. Although tuning the observer parameters may improve the accuracy of the state estimation, this error remains present due to the observer's inability to detect and compensate for these disturbances.

2.5.8 Feedforward

The decision to exclude a feedforward control strategy from the system design was primarily driven by the constant nature of the reference values and the specific characteristics of the problem.

In this AGV control system, the reference represents fixed distances in the x and y directions (1m and 0m, respectively). Since feedforward control is typically advantageous in handling dynamic variations in the reference, its inclusion was unnecessary. The existing feedback and integral control strategies are sufficient to manage disturbances and maintain the desired performance.

Additionally, the dynamic industrial environment introduces uncertainties such as unpredictable obstacles, variations in leader AGV trajectories, and sensor noise. In such settings, disturbances cannot be accurately predicted, which limits the effectiveness of feedforward control. This unpredictability reinforces the choice to rely solely on feedback and integral action, which are inherently better suited for handling unexpected variations in real time. By avoiding feedforward control, the system maintains a streamlined design while effectively addressing the disturbances and uncertainties characteristic of industrial environments. This decision ensures reliable operation, simplicity, and robustness without unnecessary increases in computational or implementation overhead.

Chapter 3

Application

3.1 Simulator scheme and description

To test and visualize the previously modeled and implemented system in *Matlab* we used the *Simulink* environment. The complete scheme is shown below:

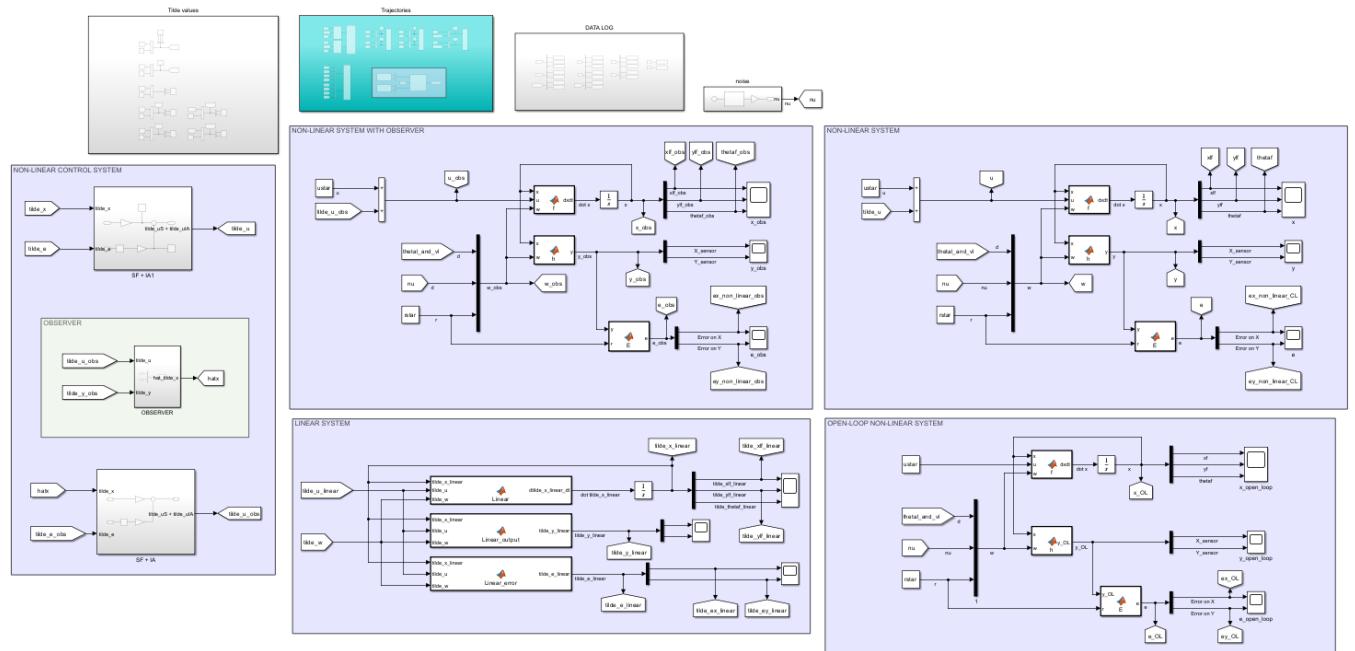


Figure 3.1: Simulink scheme

3.1.1 Linear Plant

The linearized plant was built using *Matlab* functions, where we implemented the matrices computed in section 2.2.3:

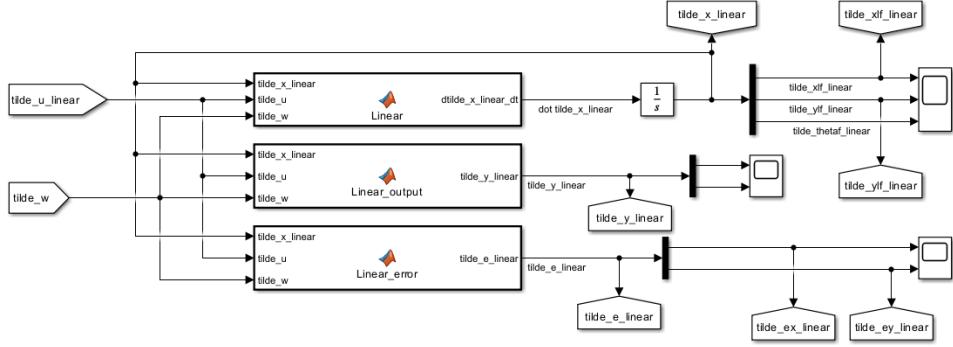


Figure 3.2: Linear plant scheme

3.1.2 State Feedback with integral Action

For the state feedback with integral action, we created a system utilizing the matrices \mathbf{K}_S and \mathbf{K}_I , which were designed in section 2.5.5:

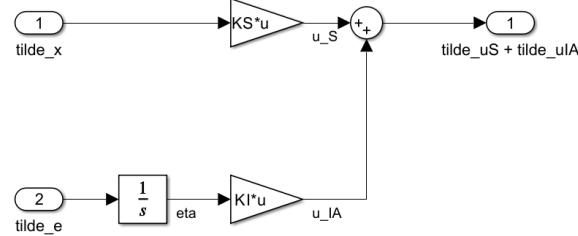


Figure 3.3: State feedback with integral action scheme

3.1.3 Observer

The observer was built using the state-space model, with matrices calculated based on the \mathbf{K}_O obtained from the `icare` function, as detailed in subsection 2.5.7:

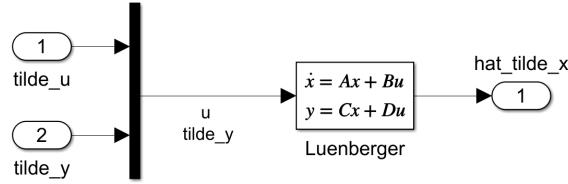


Figure 3.4: Observer scheme

3.2 Real Plant simulation

3.2.1 Path selection

After the modeling and tuning of the AGV control system in condition such as a double “U” turn trajectory (which is useful since a path that is quite often repeated in real operative condition), we designed a path that closely reflects a real industrial environment:

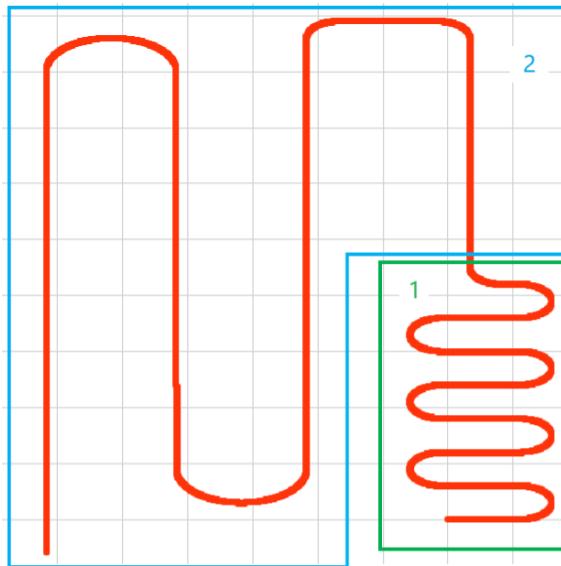


Figure 3.5: Leader’s realistic industrial path

This path consists of two distinct sections, designed to reflect the dual operational capabilities of modern AGVs, which can safely navigate human-shared areas (such as supermarkets within industrial plants, where goods are temporarily stored and slower speeds are necessary for safety and efficient picking) and operate at higher speeds in zones dedicated to fully automated processes, where no human presence is involved. Accordingly, the layout was divided into two zones: a **supermarket area** (zone 1 in figure 3.5) with straight-line speeds of $v = 0.6m/s$ and a curve radius of $r = 1.5m$, and an **automated area** (zone 2 in figure 3.5) with straight-line speeds of $v = 1m/s$ and a curve radius $r = 3m$. It is important to emphasize that the linear speed in curves is $v = 0.3m/s$ in both the supermarket area and the automated area, as this will play an important role in the considerations discussed in the next section.

3.2.2 Tuning of the Real Plant

After running several tests, we observed that the adjustments made in the sections 2.5.5 and 2.5.7 remain valid:

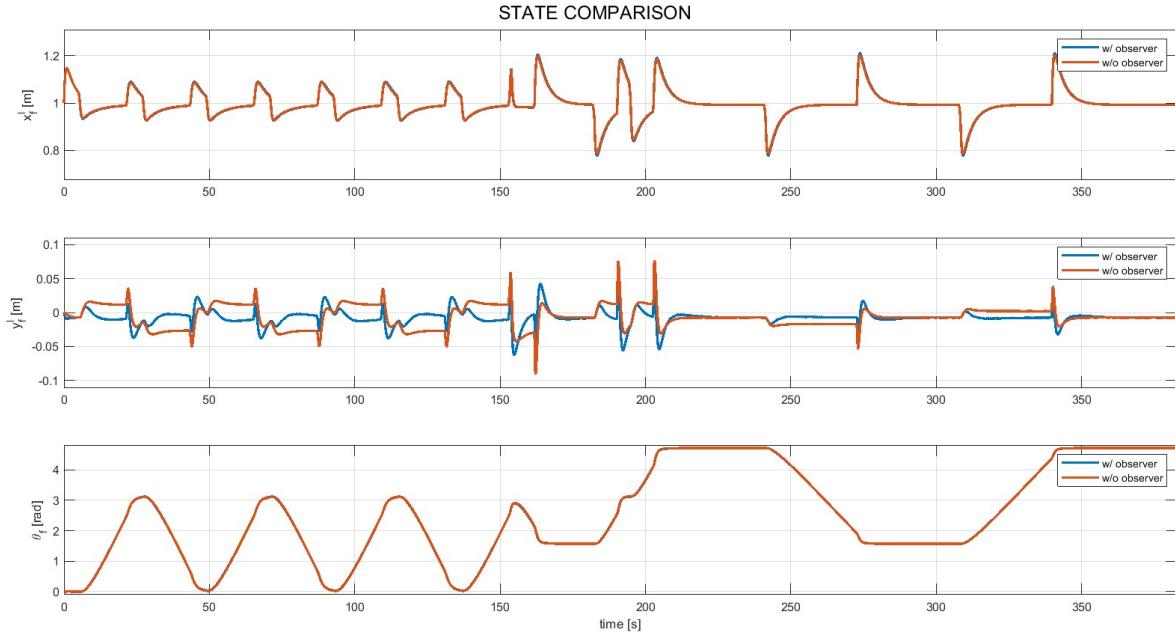


Figure 3.6: State comparison on real plant path with original tuning values

This makes sense, as the linearization (section 2.2.3) was performed under straight-moving conditions, making the curves the most critical sections; indeed, since the curves are traversed at the same speed as the path used during the tuning tests, the adjustments remain valid. Although the error is already acceptable, given the critical nature of the supermarket area due to the presence of operators and associated safety considerations, we decided to further refine the tuning of both the integral action and the observer, focusing particularly on this zone to ensure an even higher level of safety. Therefore, using the approach and considerations from the previous case and refining through multiple iterations, we arrived at the following parameters for the **integral action tuning**:

$$\mathbf{Q} = \frac{1}{5} \cdot \begin{bmatrix} \frac{1}{0.3^2} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{0.01^2} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{0.1^2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{1^2} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{0.1^2} \end{bmatrix} \quad (3.1)$$

$$\mathbf{R} = \frac{1}{2} \cdot \begin{bmatrix} \frac{1}{0.7^2} & 0 \\ 0 & \frac{1}{1.2^2} \end{bmatrix} \quad (3.2)$$

As discussed in the **\mathbf{Q}** and **\mathbf{R}** matrices selection section 2.5.2, $\varepsilon_{2,\max}$ represents the maximum error in the y component while $u_{1,\max}$ corresponds to the control input for the follower's angular speed.

A similar approach has been taken for the **observer tuning**. Since, as stated in section 2.5.6, $w_{1,\max}$ is closely related to the uncertainty in the leader's orientation angle, we decided to push this parameter further:

$$\mathbf{Q}_d = \begin{bmatrix} 10^2 & 0 & 0 & 0 & 0 \\ 0 & 1^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.3)$$

$$\lambda_d = 0.1 \quad (3.4)$$

$$\mathbf{K}_O = \begin{bmatrix} 66.7667 & 0 \\ 0 & 66.9667 \\ 0 & -133.5335 \end{bmatrix} \quad (3.5)$$

3.2.3 Real Plant simulation results

The plant described earlier was simulated in *Simulink*, yielding the following results:

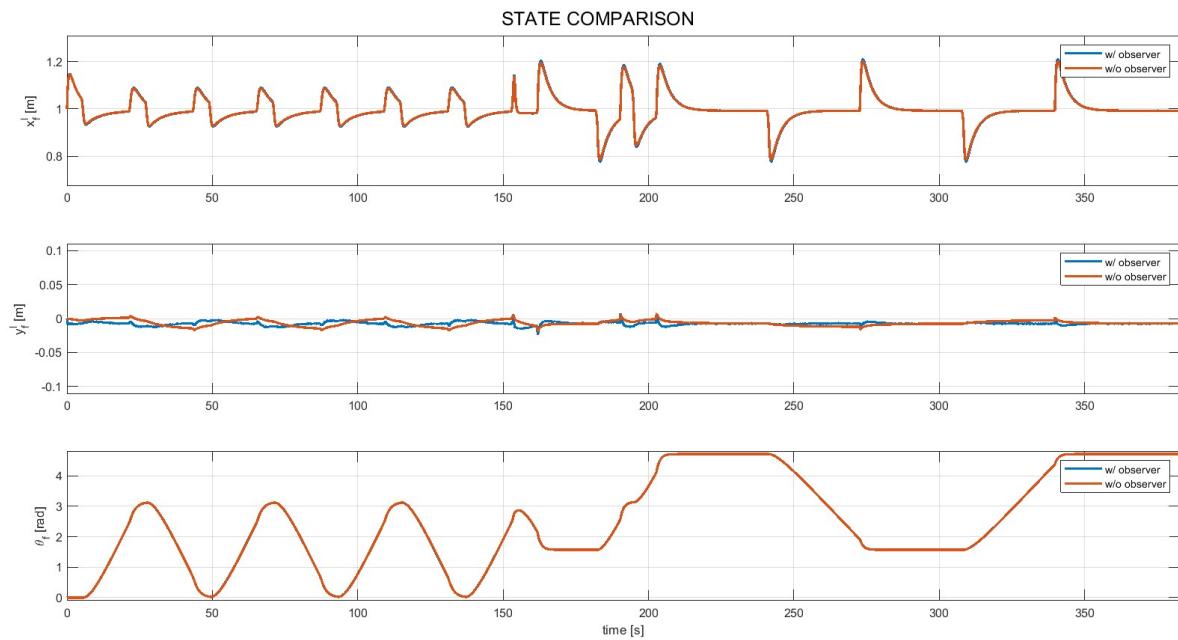


Figure 3.7: State comparison on real plant path with last tuning values

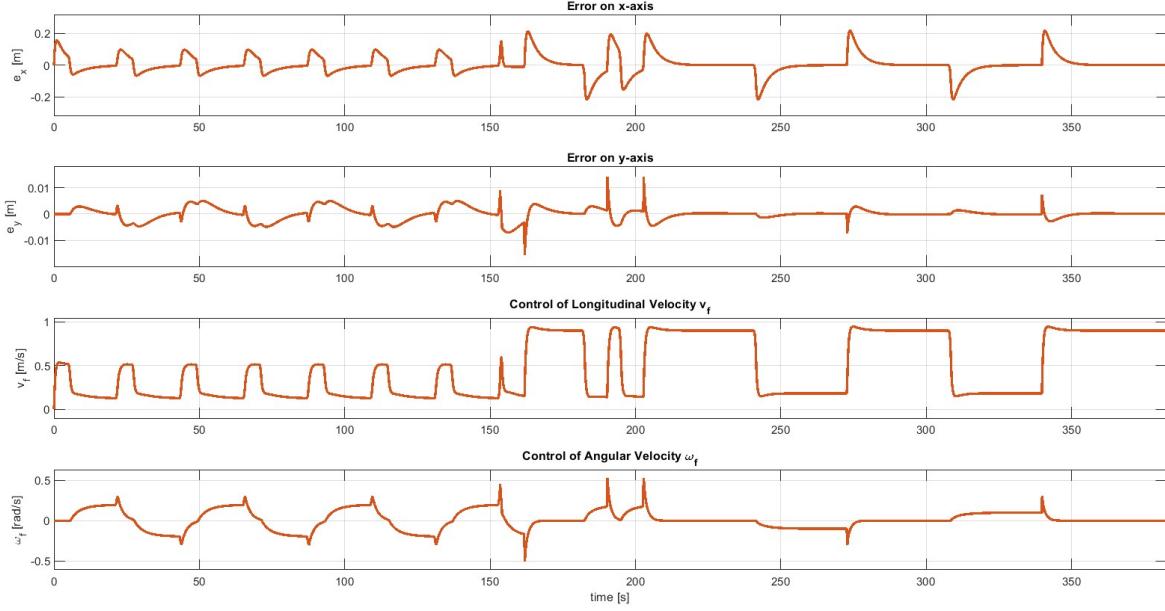


Figure 3.8: Errors and controls on real plant path with last tuning values

Following this final tuning, it is clear that, despite the slightly more demanding conditions for the actuators, the system performs effectively even in scenarios resembling real-world conditions. This is confirmed by the fact that the errors in both the x and y directions remain within a few centimeters, both in the supermarket area and the automated area.

3.3 Obstacle detection and stopping strategy

Given the high-risk nature of the AGV operating environment, which often includes scenarios where operators may inadvertently cross the AGV's path, we conducted an additional study to enhance safety. Specifically, we implemented and simulated a control strategy designed to handle such situations effectively. In this strategy, the AGV is programmed to stop immediately upon detecting an obstacle within a $0.7m$ radius. The vehicle remains stationary until the obstacle is no longer within this range, at which point it resumes its trajectory. This approach aims to mitigate the risks associated with human interaction and ensure safer operations in environments where

AGVs and humans share the workspace.

3.3.1 Obstacle Trajectory

Before implementing the strategy, it was necessary to define the trajectory of the dynamic obstacle. This was done in relative terms with respect to the AGV follower, allowing us to use the same sensor (described in the section 2.2.2) employed to detect the leader's position. The obstacle's trajectory was designed by keeping the relative distance along the x-axis between the follower and the obstacle constant, while the distance along the y-axis varies linearly over time. This approach enables precise control of the zero-crossing event along the y-axis in the simulation environment, allowing us to simulate crossings at different moments and under varying operating conditions.

```
function [x_rel, y_rel] = ObstacleTrajectory(t)
    % Trajectory of the obstacle relative to the follower
    x_rel = 0.5; % Constant (position along x relative to the follower)
    y_rel = 27 - t; % The obstacle moves along the y-axis in relation to the follower.
end
```

Figure 3.9: Obstacle trajectory code: the value of y_{rel} , defined as a quantity minus t , allows testing under different conditions

3.3.2 Control Strategy Implementation

The control strategy begins with comparing the distance between the AGV follower and the obstacle (calculated as $d_O = \sqrt{x_{\text{rel}}^2 + y_{\text{rel}}^2}$, where x_{rel} and y_{rel} represent the relative distances along the x - and y -axes, respectively) against a predefined threshold (set at 0.7m in this case).

This comparison generates a boolean output:

- **true (1):** if the distance is less than the threshold;
- **false (0):** if the distance is greater than or equal to the threshold.

The boolean output serves as the input for a switch that governs the control system's behavior:

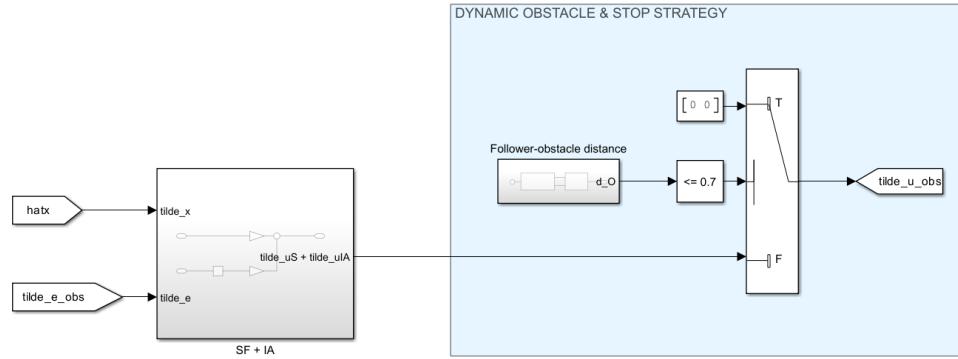


Figure 3.10: Control system blocks on Simulink

- when the output is **true**, the control inputs are set to zero, effectively stopping the AGV;
- when the output is **false**, the system continues operating as normal.

This mechanism ensures the AGV halts in the presence of nearby obstacles and resumes motion only when it is safe to proceed.

3.3.3 Test and Validation

The initial tests have been conducted with the leader's trajectory set as a double "U-turn":

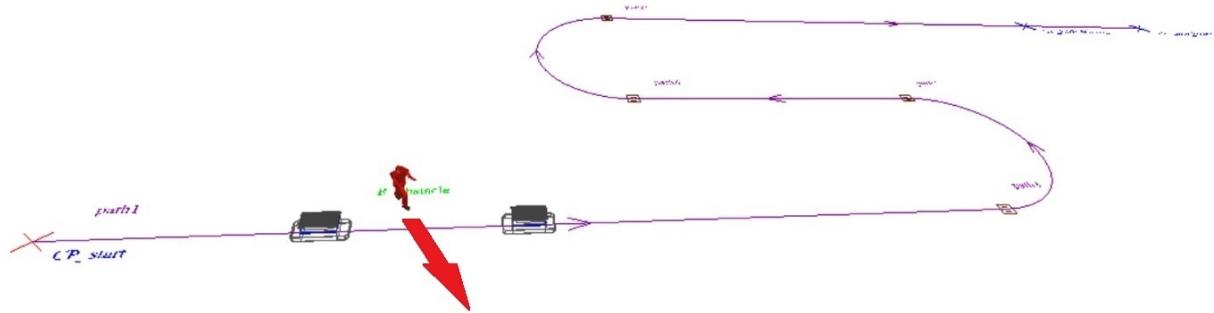


Figure 3.11: Automod representation of an obstacle crossing the trajectory

The first simulation focuses on straight-moving conditions, where the AGV was evaluated for its ability to stop and resume motion after detecting an obstacle:

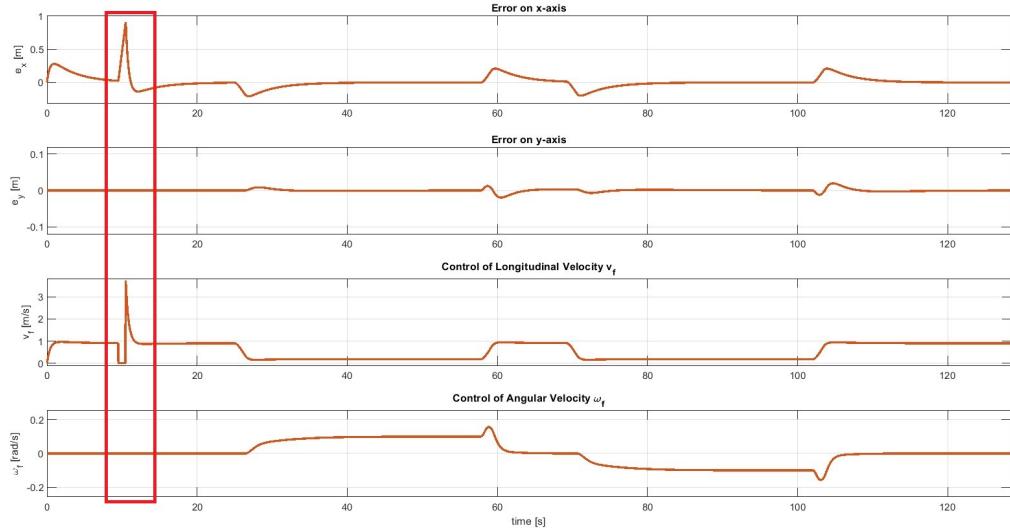


Figure 3.12: Errors and controls for the first test of the stop strategy on a straight section

In this test, the obstacle appears at $\approx 10s$ when the AGV follower is on a straight path (the point is highlighted in the Figure 3.12 by the red rectangle). As observed, the error on the y-axis remains at 0, while the error on the x-axis increases due to the follower stopping, which results in an increased distance between the leader and the follower.

The control on ω_f remains unchanged compared to tests without obstacles, whereas the control on v_f drops to zero in the presence of the obstacle and subsequently increases once the obstacle is no longer present, allowing the follower to catch up with the leader. A more detailed view of the v_f control is shown in the Figure 3.13:

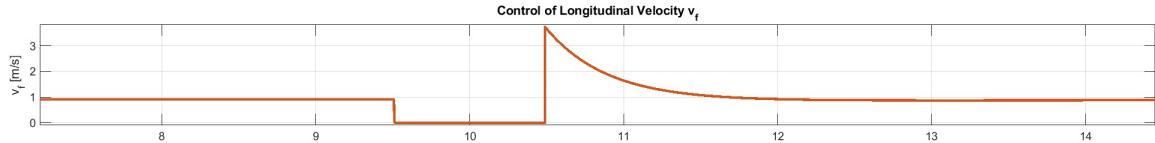


Figure 3.13: Zoom on the control on v_f

To further validate the system's robustness, we extended the analysis to curved trajectories. This scenario introduces additional challenges, such as managing angular velocity and maintaining path adherence, making it a crucial part of the AGV's overall performance assessment:

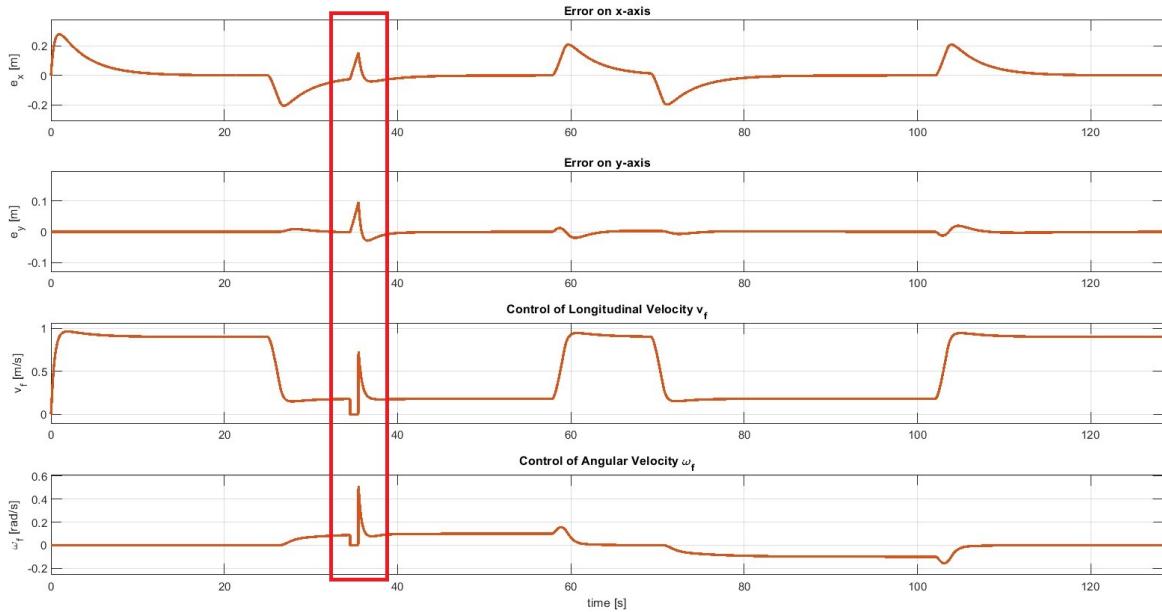


Figure 3.14: Errors and controls for the first test of the stop strategy during a turn

In the second test, since the follower is on a curve, both the error on the x-axis and the error on the y-axis increase when the follower stops due to the presence of the obstacle. As expected, both controls are necessary to allow the follower to catch up with the leader. A more detailed view of the v_f and ω_f controls is shown in 3.15 and 3.16:

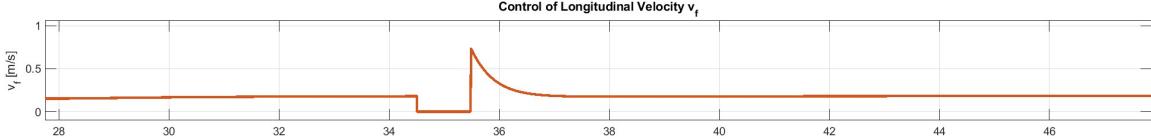


Figure 3.15: Zoom on the control on v_f

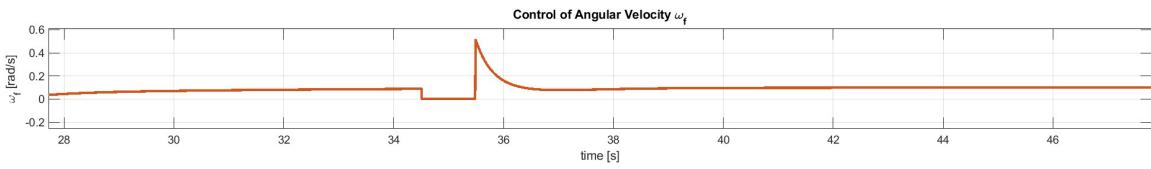


Figure 3.16: Zoom on the control on ω_f

Following these tests, two major areas for improvement were identified:

- **Physical behavior of controls:** The control responses did not accurately reflect real-world behavior. As shown in the Figures 3.13, 3.15 and 3.16, the switching of the safety control (when it activates and deactivates) was instantaneous. This unrealistic behavior was resolved by adding two “rate limiter” blocks to the angular and linear velocities, ensuring a smoother transition

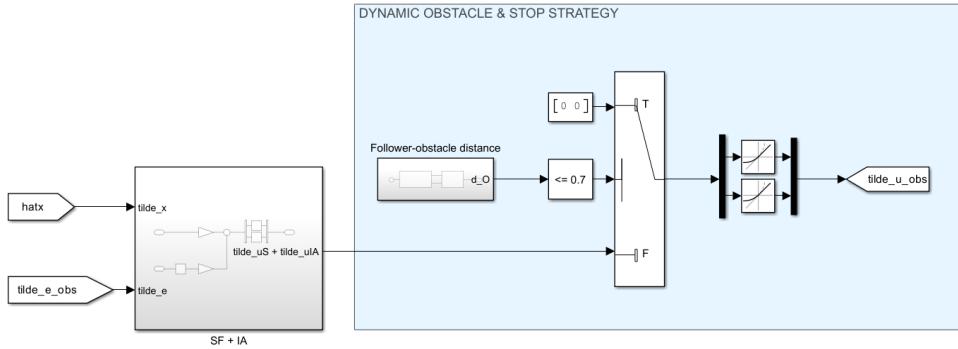


Figure 3.17: Control system blocks with rate limiter on *Simulink*

- **Actuator limits exceeded:** The actuators surpassed the safety threshold values we defined, specifically 0.5 rad/s for angular velocity (in absolute terms) and 1.5 m/s for linear velocity. To address this, we

added “saturation” blocks to the control inputs, constraining them to these predefined safety limits.

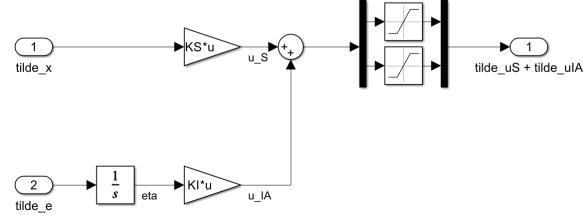


Figure 3.18: Control system blocks with saturation on *Simulink*

3.3.4 Stopping Strategy simulation and results

After implementing the above improvements, while the errors are slightly higher, the enhancements on the control side are evident:

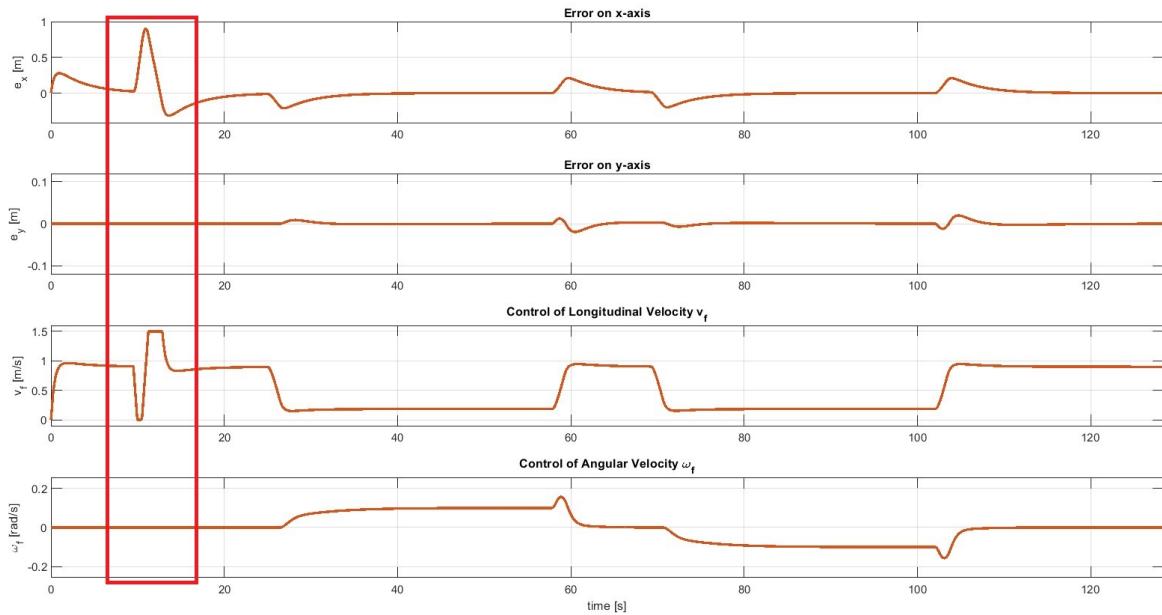


Figure 3.19: Errors and controls for the second test of the stop strategy on a straight section

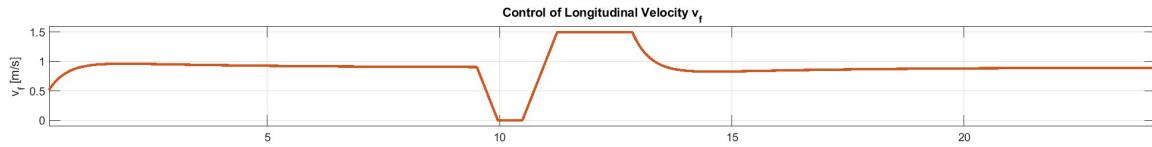


Figure 3.20: Zoom on the control on v_f

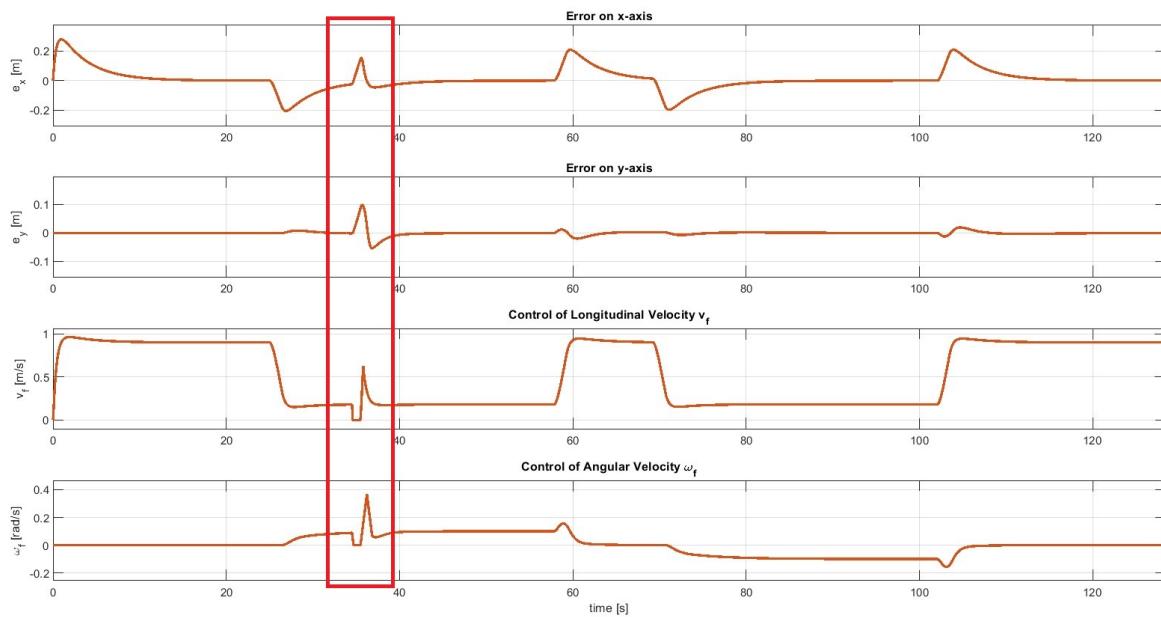


Figure 3.21: Errors and controls for the second test of the stop strategy during a turn

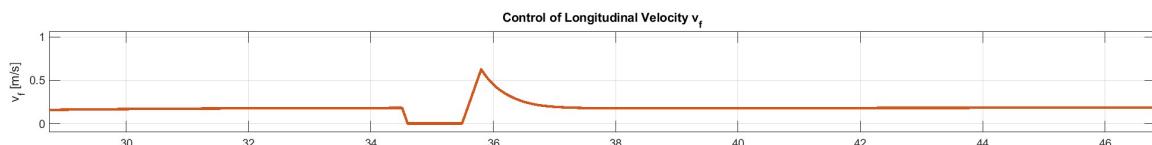


Figure 3.22: Zoom on the control on v_f

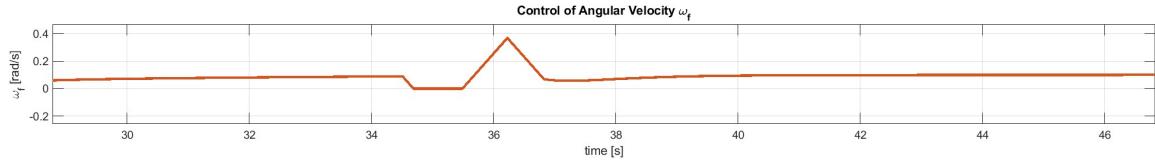


Figure 3.23: Zoom on the control on ω_f

In the images 3.20, 3.22 and 3.23, the effectiveness of the saturation and rate limiter blocks is clearly visible. These adjustments ensure that the controls respect the defined safety thresholds and transition smoothly when the safety control activates or deactivates.

After testing the leader's trajectory with a double "U-turn", the system is evaluated in the supermarket area, where the higher risk of incidents justifies the focus. The parameters from Section 3.3.3 are representative of the automated environment. The first simulation examines the AGV's performance during straight-line motion:

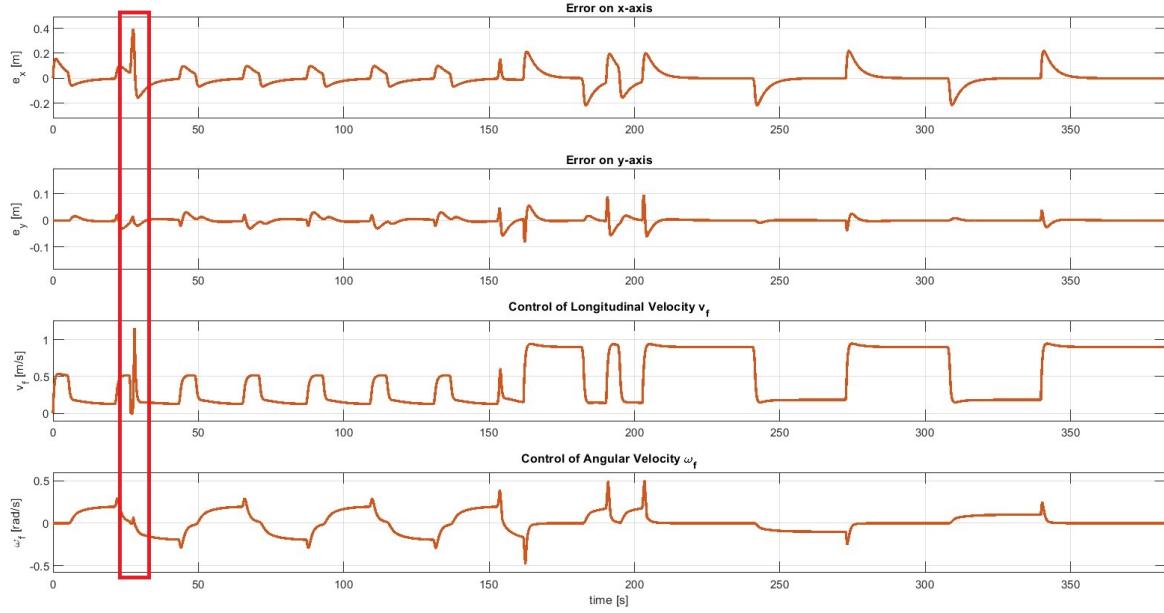


Figure 3.24: Errors and controls for the real plant test of the stop strategy on a straight section

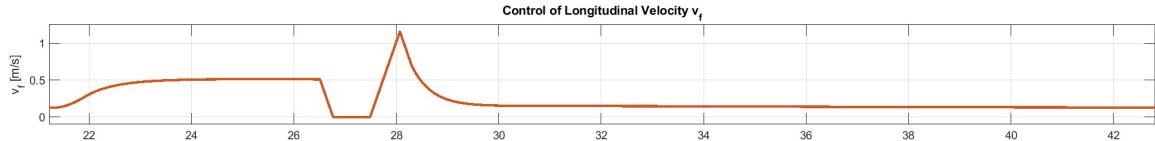


Figure 3.25: Zoom on the control on v_f

To complement the straight-line results, further simulations were conducted for curved sections of the real path. Similar to the approach used in the previous tests, these simulations assess the AGV's ability to navigate more dynamic scenarios:

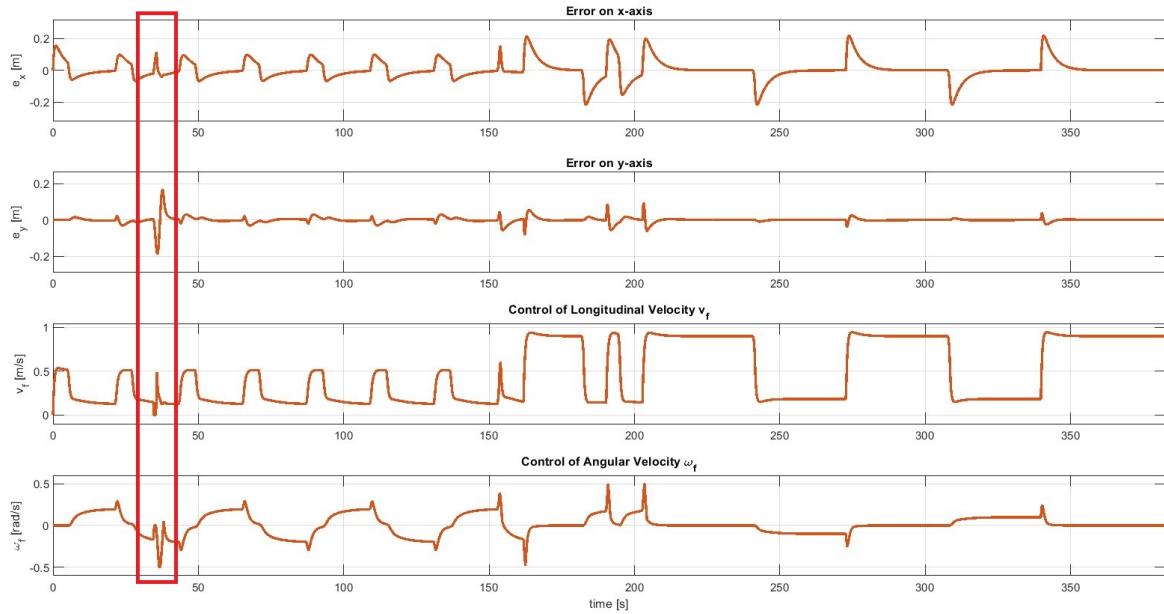


Figure 3.26: Errors and controls for the real plant test of the stop strategy during a turn

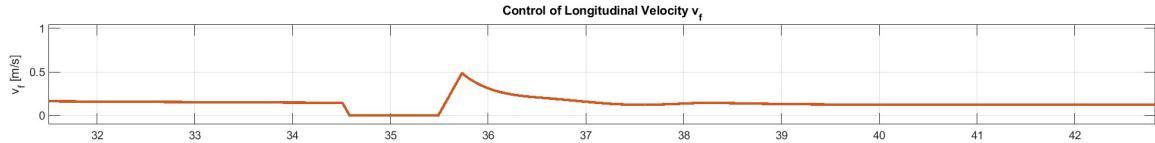


Figure 3.27: Zoom on the control on v_f

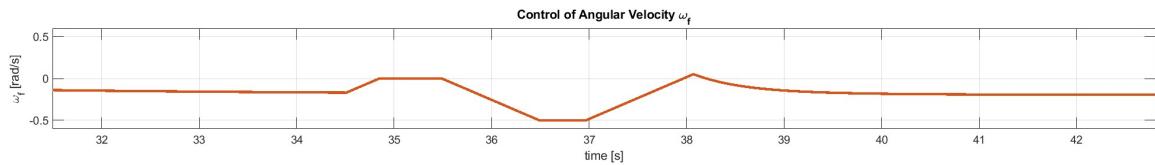


Figure 3.28: Zoom on the control on ω_f

In conclusion, the tests conducted on both straight-line and curved sections demonstrate that the AGV is capable of stopping effectively when an obstacle is detected within the safety range and can subsequently resume motion, re-engaging with the leader as required. This confirms the reliability of the control strategy implemented, ensuring that the AGV can navigate safely and maintain its trajectory even in environments with varying operational conditions.

Chapter 4

Conclusions and further investigation

This work has demonstrated the development and validation of a leader-follower control system for Automated Guided Vehicles (AGVs), specifically designed to ensure precise synchronization and adaptability in industrial environments. During the development of calculations, a trade-off between completeness and not too high complexity of the solution has been followed. For instance, in our model, no disturbances related to the floor were considered, assuming a perfectly smooth surface for the AGV's operating environment, being this a condition that, in industrial environments, cannot always be guaranteed.

The key conclusions include:

- **Effectiveness of the Integral Action:** The inclusion of an integral control element significantly enhanced tracking accuracy, especially in scenarios with persistent disturbances.
- **Accurate Path Following:** The follower AGV maintained a consistent and safe alignment with the leader's trajectory, even in complex paths with varying curvature radii.
- **Optimization of Control Parameters:** The optimal selection of control matrices \mathbf{Q} and \mathbf{R} ensured a balance between minimizing tracking errors and controlling effort, leading to an efficient and reliable system.
- **Obstacle Management:** The proposed obstacle detection and stopping strategies proved effective, demonstrating the system's ability to adapt to unexpected environmental changes.

In future work, it would be valuable to consider the disturbances originating from the floor or road surface, as these could have a significant impact on the performance of the AGV in real-world industrial environments.

Additionally, the integration of advanced sensor technologies, such as LiDAR or more sophisticated camera systems, could enhance the AGV's ability to detect obstacles and navigate through complex contexts. These sensors could provide more accurate and reliable data, improving the AGV's performance and safety in environments with varying conditions and obstacles.

Bibliography

- [1] F.R.E.D. AGV. *The Past, Present, and Future of AGVs*. URL: <https://www.fredagv.com/blog/the-past-present-and-future-of-agvs>.
- [2] Rafal Cupek et al. “Autonomous guided vehicles for smart industries—the state-of-the-art and research challenges”. In: *Computational Science-ICCS 2020: 20th International Conference, Amsterdam, The Netherlands, June 3–5, 2020, Proceedings, Part V 20*. Springer. 2020, pp. 330–343.
- [3] Qinhang Jiang and Hairong Yan. “Leader-follower Motion Control of Two Line Following AGVs using PID Algorithm”. In: *2022 IEEE 5th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*. Vol. 5. IEEE. 2022, pp. 445–449.
- [4] KUKA. *AGV solution for Body-In-White truck production*. URL: https://www.kuka.com/en-de/industries/solutions-database/2020/01/agv-solution_biw-trucks.
- [5] KUKA. *Intelligent intralogistics in production*. URL: <https://www.kuka.com/en-in/company/press/news/2020/11/intelligent-intralogistics-in-production>.
- [6] Nicola Mimmo. *Analysis and Design of Control Laws for Advanced Driver-Assistance Systems: Theory and Applications*. Springer, 2024.
- [7] Sergio Nuzzo. *Le dimensioni delle immagini digitali: risoluzione e definizione*. Sapere Scienza. 2021. URL: <https://www.saperescienza.it/rubriche/fotografia-digitale/le-dimensioni-delle-immagini-digitali-risoluzione-e-definizione-26-10-2021/>.
- [8] Paolo Rocco. *Controlli automatici - Complementi di automatica*. URL: <https://rocco.faculty.polimi.it/caut/Complementi%20di%20automatica.pdf>.

Appendix

Linear System Matrices

Linear system parametric matrices, obtained from (2.28):

$$\mathbf{A}_{star} = \begin{bmatrix} 0 & w_f & -v_l \sin(\theta_f - \theta_l) \\ -w_f & 0 & -v_l \cos(\theta_f - \theta_l) \\ 0 & 0 & 0 \end{bmatrix} \quad (1)$$

$$\mathbf{B}_{1star} = \begin{bmatrix} y_{lf} & -1 \\ -x_{lf} & 0 \\ 1 & 0 \end{bmatrix} \quad (2)$$

$$\mathbf{B}_{2star} = \begin{bmatrix} v_l \sin(\theta_f - \theta_l) & \cos(\theta_f - \theta_l) & 0 & 0 & 0 \\ v_l \cos(\theta_f - \theta_l) & -\sin(\theta_f - \theta_l) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3)$$

Corresponding numerical ones:

$$\mathbf{A}_{star} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -0.10 \\ 0 & 0 & 0 \end{bmatrix} \quad (4)$$

$$\mathbf{B}_{1star} = \begin{bmatrix} 0 & -1 \\ -1 & 0 \\ 1 & 0 \end{bmatrix} \quad (5)$$

$$\mathbf{B}_{2star} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (6)$$

The output is defined through the matrices reported below:

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (7)$$

$$\mathbf{D}_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (8)$$

$$\mathbf{D}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (9)$$

Concerning the error, subsequent matrices are taken into account:

$$\mathbf{C}_e = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (10)$$

$$\mathbf{D}_{1e} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (11)$$

$$\mathbf{D}_{2e} = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix} \quad (12)$$

Extended State Matrix

$$\bar{\mathbf{A}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{10} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (13)$$

$$\bar{\mathbf{B1}} = \begin{bmatrix} 0 & -1 \\ -1 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (14)$$

$$\bar{\mathbf{B2}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ \frac{1}{10} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix} \quad (15)$$

$$\mathbf{O} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -0.1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{O} \in \mathbb{R}^{6 \times 3} \quad (16)$$