

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

**Evaluating and Comparing Trajectory Tracking Control
Strategies for Mobile Robots: A Robotino Case Study.**

Sebastián Alejandro Vega Narváez

Mateo Sebastián Vasquez Guevara

Electrónica y Automatización

Trabajo de fin de carrera presentado como requisito

para la obtención del título de

Ingeniera en Electrónica

Quito, Noviembre de 2024

Colegio de Ciencias e Ingenierías

**HOJA DE CALIFICACIÓN
DE TRABAJO DE FIN DE CARRERA**

**Evaluating and Comparing Trajectory Tracking Control Strategies for Mobile Robots: A
Robotino Case Study**

**Sebastián Alejandro Vega Narváez
Mateo Sebastián Vasquez Guevara**

Tutor: Óscar Camacho, Ph.D.

Quito, Noviembre de 2024

© DERECHOS DE AUTOR

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en la Ley Orgánica de Educación Superior del Ecuador.

Nombres y apellidos: Sebastián Alejandro Vega Narváez

Código: 00207264

C.I.: 1719110478

Fecha: Quito, Noviembre de 2024

© DERECHOS DE AUTOR

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en la Ley Orgánica de Educación Superior del Ecuador.

Nombres y apellidos: Mateo Sebastián Vasquez Guevara

Código: 00213967

C.I.: 1726051079

Fecha: Quito, Noviembre de 2024

ACLARACIÓN PARA PUBLICACIÓN

Nota: El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en <http://bit.ly/COPETheses>

UNPUBLISHED DOCUMENT

Note: The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on <http://bit.ly/COPETheses>

RESUMEN

Este trabajo presenta un análisis comparativo del rendimiento de control de diversas estrategias de control para el seguimiento de trayectorias aplicadas al robot Festo llamado "Robotino" en varios escenarios. Se evaluaron tres enfoques basados en modelos: Proporcional-Integral (PI), Control de Modelo Genérico (GMC) y Control por Modos Deslizantes (SMC). Además, se implementó un enfoque libre de modelos y se comparó su rendimiento en diferentes trayectorias. Los resultados destacan las fortalezas y debilidades de cada estrategia de control, ofreciendo perspectivas sobre su aplicabilidad en el campo de la robótica.

Palabras clave: Seguimiento de Trayectorias, robots móviles, estrategias de control, evaluación de rendimiento.

ABSTRACT

This work presents a comparative analysis of the control performance of various control strategies for trajectory tracking applied to the Festo robot called a "Robotino" in multiple scenarios. We evaluated three model-based approaches: Proportional-Integral (PI), Generic Model Control (GMC), and Sliding Mode Control (SMC). In addition, we implement a model-free approach and compare its performance across different trajectories. The findings highlight the strengths and weaknesses of each control strategy, offering insights into their applicability to the robotics field.

Keywords: Trajectory Tracking, mobile robots, Control strategies, performance evaluation.

CONTENTS

1	Introduction	14
2	Control Strategies Fundamentals	16
2.0.1	Proportional Integral Control - PI	16
2.0.2	Generic Model Control - GMC	16
2.0.3	Sliding Model Control - SMC	17
2.0.4	Model Free Control- iPI	18
3	Robotino - Festo Didactic	20
3.0.1	Kinematic of omnidirectional wheels	20
3.0.2	Trajectory Tracking Control for Robotino	21
4	Results and discussions	23
4.0.1	Empirical models	23
4.0.2	Controllers tunings	24
4.0.3	Simulations	24
4.0.4	Experimental	29
4.0.5	Performance Indices	33
5	Conclusions	38
	Bibliography	39
6	APPENDIX	41

LIST OF FIGURES

2.1	iPI Model Free Schematic.	19
3.1	Robotino-Festo.	20
3.2	Schematic of Control for trajectory tracking of Robotino	22
4.1	Empirical model based on the simulation of X , Y and ϕ , respectively.	24
4.2	Empirical model is based on the experimental data of X , Y , and ϕ	25
4.3	Comparative responses of Simulation Circular Trajectory	26
4.4	Comparative Error of Simulation Responses for Circular Trajectory	27
4.5	Comparative responses of Simulation Lemniscate Trajectory	28
4.6	Comparative Error of Simulation Responses for Lemniscate Trajectory	29
4.7	Comparative responses of Simulation Square Trajectory	30
4.8	Comparative Error of Simulation Responses for Square Trajectory	31
4.9	Comparative responses of Experimental Circular Trajectory	32
4.10	Comparative Error of Experimental Responses for Circular Trajectory	33
4.11	Comparative responses of Experimental Lemniscate Trajectory	34
4.12	Comparative Error of Experimental Responses for Lemniscate Trajectory	35
4.13	Comparative responses of Experimental Square Trajectory	36
4.14	Comparative Error of experimental responses for Square Trajectory	37
6.1	Robotino Direction	41

LIST OF TABLES

4.1	Approximation of integral transfer functions for simulation.	23
4.2	Approximation of integral transfer functions for experimental	24
4.3	Controller tuning parameters for trajectory tracking Simulation	25
4.4	Controller tuning parameters for trajectory tracking Experimental	29
4.5	Comparison of Performance Parameters for Simulation and Experimental Circular Trajectory	34

4.6	Comparison of Performance Parameters for Simulation and Experimental Lemniscate Trajectory	35
4.7	Comparison of Performance parameters for Simulation and Experimental Square Trajectory	36

DEDICATORIA

A mi padre, Fernando, a mi madre, Patricia, a mi hermana, Paola, a mi hermano, Bryan, a mi abuelito, Rodrigo, a mi abuelita, Mery, y a mi novia, Sorin. Gracias por ser mi constante fuente de apoyo, amor e inspiración. Este logro es el reflejo de su esfuerzo, paciencia y confianza en mí. Cada uno de ustedes ha sido un pilar fundamental en mi vida, y este trabajo es para ustedes, mi familia, que siempre han estado a mi lado.

Sebastian A. Vega N.

Dedico esta tesis a mi papá, Néstor Vásquez, y a mi madre Yolanda Guevara, por su amor incondicional y su apoyo constante. A mi hermano Alejandro Vásquez y mi cuñada Nadia Valencia, por su comprensión y ánimo en cada paso del camino. A mi sobrino Julián, por ser una fuente de inspiración y alegría. Y a toda mi familia, por creer en mí y acompañarme siempre durante todo este largo proceso.

Mateo S. Vasquez G.

AGRADECIMIENTO

Quiero expresar mi más profundo y sincero agradecimiento a todas las personas que han sido esenciales en este proceso. A mi padre, Fernando, y a mi madre, Patricia, por su amor incondicional, su apoyo constante y por enseñarme a perseverar incluso en los momentos más difíciles. A mi hermana, Paola, y a mi hermano, Bryan, por estar siempre a mi lado, por su inquebrantable apoyo y por recordarme el verdadero valor de la familia. A mis abuelos, Rodrigo y Mery, por su cariño, sabiduría, fe y por ser un ejemplo constante de esfuerzo y dedicación. A mi novia, Sorin, por ser mi compañera incondicional en este viaje, por su comprensión y apoyo en cada paso del camino. Y a mi profesor y mentor, Oscar Camacho, por tu orientación, paciencia, y la confianza que has depositado en mí, así como por los valiosos consejos que no solo han sido cruciales para completar este trabajo, sino también para convertirme en el profesional e investigador que soy hoy. A todos ustedes, mi familia y Oscar, este logro es tan suyo como mío. Gracias por su apoyo constante, por su confianza y por inspirarme a seguir adelante. Los quiero a todos.

Sebastian A. Vega N.

AGRADECIMIENTO

Al concluir este trabajo de titulación, quiero expresar mi más sincero agradecimiento a todas las personas que hicieron posible su realización. En primer lugar, deseo agradecer a mi director de tesis, Oscar Camacho, por su guía y apoyo incondicional, y por permitirme realizar varias publicaciones que han enriquecido este proyecto. Su experiencia y dedicación han sido fundamentales para el desarrollo de esta investigación. Agradezco profundamente a Marco Herrera y Pablo Proaño, quienes aportaron valiosas ideas esenciales para llevar a cabo este trabajo. Su colaboración y conocimientos han sido de gran ayuda. De manera especial, quiero expresar mi gratitud a mi familia, quienes han sido mi pilar durante todo este proceso. A mi padre, Néstor Vásquez, y a mi madre, Yolanda Guevara, por inculcarme los valores de la perseverancia y el esfuerzo, y por su apoyo constante en cada etapa de mi vida. A mi hermano, Alejandro Vásquez, por ser un ejemplo de fortaleza y determinación, y a mi cuñada, Nadia Valencia, por su comprensión y apoyo en los momentos más difíciles. A mi sobrino, Julián, cuya alegría y energía siempre me han motivado a seguir adelante. Y a toda mi familia, por su amor incondicional y su constante aliento. Gracias a todos por creer en mí y por estar siempre a mi lado, brindándome el ánimo y la fuerza necesaria para alcanzar mis metas. Este logro es tanto mío como de ustedes, y no podría haberlo conseguido sin su apoyo inquebrantable.

Mateo S. Vasquez G.

CHAPTER 1

INTRODUCTION

The trajectory tracking problem (TTP) represents a central control challenge, where the goal is to match the output of a system to a dynamic reference over time (Medina et al., 2024, Morales et al., 2021). This issue is especially significant for Autonomous Mobile Robots (AMRs). In industrial settings, AMRs must maintain a fixed course, such as one delineated by rails or lines, and enhance their movement by traveling at a designated speed. Consequently, the path-following problem becomes a TTP, integrating positional and velocity control.

Several research efforts have examined the problem of trajectory tracking (TTP) in Nonlinear Mobile Robots (NMR) through a variety of control strategies like the Linear Quadratic Regulator (LQR) (Morales et al., 2018), nonlinear control (Yang et al., 2015), Model Predictive Control (MPC) (Künhe et al., 2005), and Sliding Mode Control (SMC) (Salinas et al., 2018). These approaches based on models rely heavily on the precision of the system's model and parameters. Although analytical models can help to gain insight, incorporation of nonlinearities makes them more complex and increases computational requirements (Medina et al., 2024). The construction of a comprehensive model is difficult due to the complexity of the system and the need for accurate parameters (Smith and Corripio, 2005). Consequently, phenomenological models can result in complex controllers that are challenging to implement, limiting their applicability in industrial contexts.

Instead of phenomenological models, empirical models such as First-Order Plus Dead Time (FOPDT) models serve as a good substitute for control design (Capito et al., 2016, Guevara et al., 2019). These models are particularly advantageous for analyzing and designing numerous industrial process control applications by accurately depicting the fundamental dynamic behaviors of processes (Smith and Corripio, 2005).

This work conducts a comparative evaluation of different control strategies employed for a mobile robot known as the "Robotino" across several scenarios. Explore three model-based techniques: proportional-integral (PI), generic model control (GMC), and sliding mode control (SMC). Furthermore, it incorporates and contrasts a model-free approach developed by Flies (Fliess and Join, 2013) on various trajectories.

This article is structured in the following manner. Section 2 offers a theoretical foundation and initial knowledge concerning controllers. Section 3 gives a concise introduction to the Robotino-Festo robot. In Section 4, the detailed design of various controllers is discussed. Section 5 provides results from numerical simulations and real-world experiments, evaluating the effectiveness, practical use, and constraints of the controllers studied. Lastly, Section 6 wraps up the work.

CHAPTER 2

CONTROL STRATEGIES FUNDAMENTALS

This section introduces the fundamental aspects of the controllers employed in this study. It begins with a presentation of three model-based options, followed by an approach that explores a model-free proposal. The details are as follows.

2.0.1 Proportional Integral Control - PI

A PI controller calculates the control signal $u(t)$ as a combination of two distinct terms: proportional and integral. Each term addresses a different aspect of the system's behavior, enabling the controller to provide an effective and balanced response to the error signal $e(t)$, which is defined as the difference between the reference R and the system output y ($e(t) = R - y$) (Camacho et al., 2020). This error signal represents the deviation between the desired setpoint and the actual process variable. The controller law is expressed in equation (2.1).

$$u(t) = K_P(e(t) + \frac{1}{\tau_i} \int e(t) dt) \quad (2.1)$$

where $K_P = \frac{2}{K\tau_c}$ (K is the gain of the system) and $\tau_i = 2\tau_c$ (τ_c is an adjustment parameter)(Camacho et al., 2020).

2.0.2 Generic Model Control - GMC

Model-based controllers (MBC) use a mathematical representation of the system they control. These models capture system dynamics, often through differential equations or state-space representations, allowing the controller to anticipate system responses to inputs and disturbances (Ogunnaike and Ray, 1994).

GMC is a model-based controller designed to simplify the implementation of control, especially in nonlinear systems(Lee and Sullivan, 1988, Ogunnaike and Ray, 1994). GMC uses a dynamic model of the system and a target function that specifies the desired behavior. One

of GMC's distinguishing features is that it does not require model familiarization or long-term prediction, which reduces computational complexity, making it suitable for a range of industrial applications. The dynamic of a process is presented by equation (2.2)

$$\frac{dy}{dt} = f(y, u, d) \quad (2.2)$$

y is the output of the process that is determined by an input (u), uncertainty (d) and a function (f).

$$\frac{dy_r}{dt} = (K_1(R - y) + K_2 \int (R - y) dt) \quad (2.3)$$

Here y_r represents the trajectory for the process output to gradually converge to its target value. R , by following a pre-established path, referred to as the reference trajectory y_r . This trajectory is defined by a particular mathematical equation that directs the process output toward its desired state in a controlled manner. When we equate, equations (2.2) in (2.3). The control law expressed for an integrated system in equation (2.4):

$$u(t) = \frac{1}{K} (K_1 e(t) + K_2 \int e(t) dt) \quad (2.4)$$

Where K_1 and K_2 are values proposed by the user.

2.0.3 Sliding Model Control - SMC

Sliding mode control is a technique that utilizes a discontinuous control law to drive the system's state towards a specific surface in the state space (Utkin et al., 2020). Once the system reaches this surface, it slides along it toward the desired equilibrium point. This strategy is renowned for its robustness against uncertainties and external disturbances due to the discontinuous nature of the control action, which keeps the system on the sliding surface. This ensures high accuracy and stability, particularly in nonlinear systems and those with incomplete models (Herrera et al., 2023).

The surface proposed in equation (2.5) and the final control law for integrate model at (2.6)

$$\sigma(t) = e + \lambda_1 \int e dt \quad (2.5)$$

where $\lambda_1 = \frac{1}{4}(\frac{t_c + \tau}{t_c \tau})^2$, t_c is an adjustment parameter.

$$u(t) = \frac{1}{K}(\dot{R} + \lambda_1 e) + K_D \frac{\sigma}{|\sigma| + \delta}. \quad (2.6)$$

2.0.4 Model Free Control- iPI

It refers to a more advanced approach that merges the principles of proportional and integral control with intelligent techniques, such as adaptation and learning. These controllers are designed to improve performance in systems that may be non-linear or exhibit uncertainties. The first-order local process model for a Single Input and Single Output, proposed by Fliess is expressed by equation (2.8):

$$\dot{y}(t) = F(t) + \alpha u(t) \quad (2.7)$$

Where $\dot{y}(t)$ is the desire reference, $F(t)$ is an online estimate of the uncertainties of the process and $\alpha > 0$ is the parameter proposed by the user. For a PI controller with the model free model is expressed by:

$$u(t) = \frac{1}{\alpha}(-\hat{F}(t) + \dot{R}(t) + K_{1Pi}e(t) + K_{2Ii} \int e(t) dt) \quad (2.8)$$

Here K_{1Pi} and K_{2Ii} are the gain values of the PI controller from iPI, determined through trial and error.

In Fig. 2.1 is the schematic of model-free iPI proposed by Precup for estimate $\hat{F}(t)$ of the process (Precup et al., 2017).

The derivative blocks used to obtain the reference estimates ($R(s)$ and $\hat{F}(t)$) are expressed by equations (2.9) and (2.10), respectively. This is the derivative equation multiplied by a low-pass filter (Precup et al., 2017).

$$H_{Lp1}(s) = \frac{K_{Lp1}s}{T_{Lp1}s + 1} \quad (2.9)$$

$$H_{Lp2}(s) = \frac{K_{Lp2}s}{T_{Lp2}s + 1} \quad (2.10)$$

CHAPTER 3

ROBOTINO - FESTO DIDACTIC

This section offers a concise summary of the Robotino mobile robotics platform, which is employed for both research and educational purposes (Weinert and Pensky, 2011, Fuentes et al., 2021). Robotino features an omnidirectional drive, sensors, interfaces, and modular extensions, allowing for versatile use; see Fig. 3.1. Custom application programming is feasible through multiple programming environments that utilize its core components and dynamic behavior.



Figure 3.1: Robotino-Festo.

3.0.1 Kinematic of omnidirectional wheels

Omnidirectional wheels are a type of wheel design that allows a vehicle to move in any direction without changing its orientation. These wheels achieve this capability through the arrangement of rollers mounted at a 120-degree angle relative to the wheel's axis, which enables independent motion along both the forward-backward and sideways axes. The kinematics of omnidirectional wheels is characterized by the ability to simultaneously generate motion in multiple directions, offering significant advantages in maneuverability and precision in robotics and mobile platforms. Understanding the kinematics of these wheels is crucial for designing systems that can exploit their full range of motion in various applications, including

automated vehicles, robotic arms, and other systems requiring high maneuverability in confined spaces (Gutiérrez Ramón, 2016).

The omnidirectional wheel kinetic for one wheel velocity is represented by equation (3.1).

$$rv(t) = -v_x(t)\sin(\sigma) + v_y(t)\cos(\sigma) + R\dot{\phi}. \quad (3.1)$$

The three velocity are given by equation (3.2)

$$\begin{bmatrix} r_{xref}(t) \\ r_{yref}(t) \\ r_{\phi ref}(t) \end{bmatrix} = \frac{1}{r} \begin{bmatrix} -a(\beta_0) & b(\beta_0) & R \\ -a(\beta_1) & b(\beta_1) & R \\ -a(\beta_2) & b(\beta_2) & R \end{bmatrix} \begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \end{bmatrix} \quad (3.2)$$

where $a = \sin$ and $b = \cos$.

Let r be the radius of each wheel, β the angle of each wheel relative to a reference axis (usually in the forward direction), and R the radius from the center of the robot's chassis to the center of each wheel.

To determine the robot's position relative to the setpoint, one must calculate the inverse kinematic matrix from equation (3.2) to derive the robot's position, as shown in equation (3.3).

$$\begin{bmatrix} r_x(t) \\ r_y(t) \\ r_\phi(t) \end{bmatrix} = \begin{bmatrix} b(\theta(t)) & a(\theta(t)) & 0 \\ a(\theta(t)) & b(\theta(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \frac{1}{r} \begin{bmatrix} -a(\beta_0) & b(\beta_0) & R \\ -a(\beta_1) & b(\beta_1) & R \\ -a(\beta_2) & b(\beta_2) & R \end{bmatrix} \begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \end{bmatrix} \quad (3.3)$$

where $\theta(t)$ is the robotino's rotation angle (Gutiérrez Ramón, 2016).

3.0.2 Trajectory Tracking Control for Robotino

In this section, we describe the procedure for implementing trajectory tracking control. Fig. 3.2 shows a schematic of how each part of the system works in terms of control. Before starting, several factors must be considered when controlling the trajectory of a robot. In this case, we

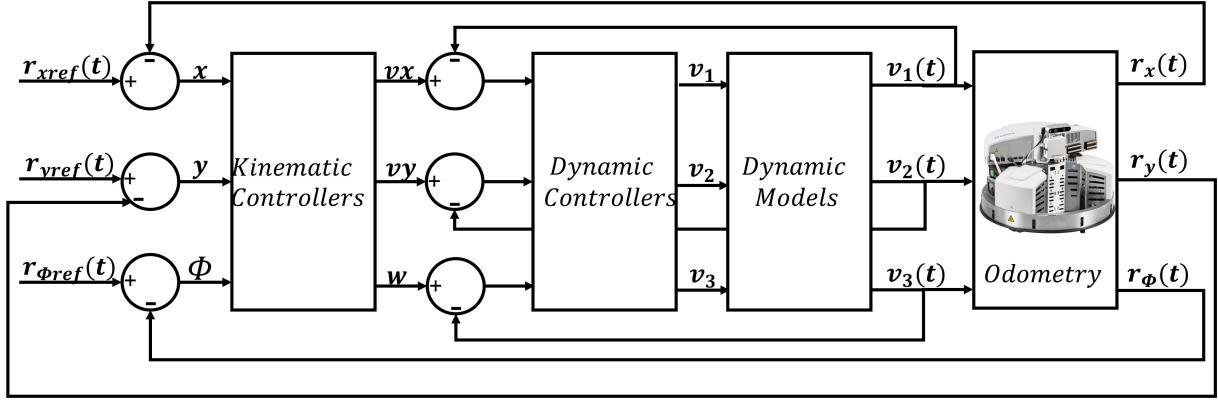


Figure 3.2: Schematic of Control for trajectory tracking of Robotino

focus on the issue of controlling the RPM of each motor, as well as the control of the robot's position.

A cascaded control scheme has been implemented for Robotino's RPM control. The inner loop of this system includes dynamic controllers, each of which is based on a PID controller. On the other hand, a setpoint for these controllers is provided by the outer loop, which uses a kinematic controller. This kinematic controller receives feedback on the trajectory reference and the current position of Robotino. The robot's position is obtained through the odometer, which calculates the position based on velocity, using the equation 3.3. For the control of the outer loop, different types of controllers will be applied, each offering distinct advantages, including robustness and speed.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.0.1 Empirical models

In this section, we present the empirical models that approximate the robotino's behavior at different positions. To obtain the transfer functions, we used MATLAB's `ident` toolbox, which applies system identification techniques to estimate the transfer functions from experimental input-output data. These models help characterize the robot's dynamics and are essential for simplifying control design and analysis across various operating conditions.

Transfer function approximation for simulation

The integrated approximation models for x , y , and ϕ are presented in Table 4.1, based on the simulation system identification results.

Table 4.1: Approximation of integral transfer functions for simulation.

Variable	Simulation Model
X	$\frac{3.8}{s}e^{-0.13s}$
Y	$\frac{2.5}{s}e^{-0.12s}$
ϕ	$\frac{2.1}{s}e^{-0.1s}$

Fig. 4.1 illustrates the actual response of the robotino model along with the approximation model. It can be seen that the results closely match the operating point.

Transfer function approximation for experimental model

The integrated approximation models for x , y , and ϕ are presented in Table 4.2, based on the system identification results from experimental data.

Fig. 4.2 shows the real response of the robotino model alongside the approximation model. It is evident that the results closely align with the operating point.

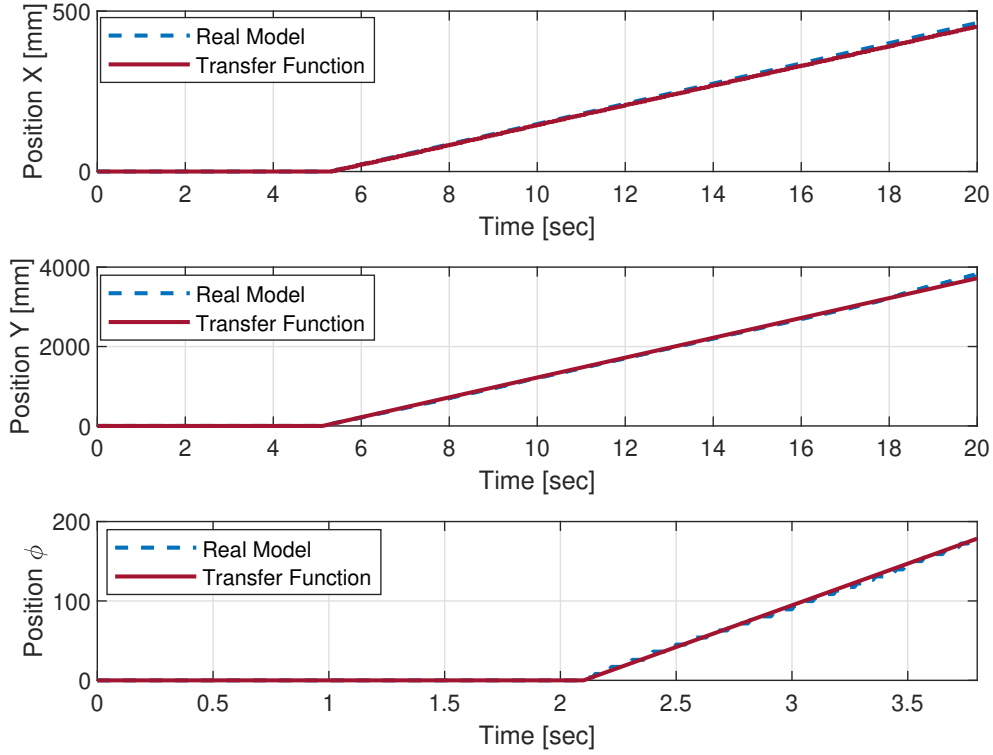


Figure 4.1: Empirical model based on the simulation of X , Y and ϕ , respectively.

Table 4.2: Approximation of integral transfer functions for experimental

Variable	Experimental Model
X	$\frac{0.31}{s}e^{-0.25s}$
Y	$\frac{0.32}{s}e^{-0.3s}$
ϕ	$\frac{0.31}{s}e^{-0.3s}$

4.0.2 Controllers tunings

In this section, Tables 4.3 and 4.4 are shown, detailing the adjustment parameters obtained for each control strategy, implemented both in simulation and through experimental validation, respectively. The adjustment process involved trial and error adjustments until obtaining acceptable performance indices, such as response time, stability, and ISE.

4.0.3 Simulations

This section presents the results obtained from the comparison of the controllers in the simulation of the trajectories.

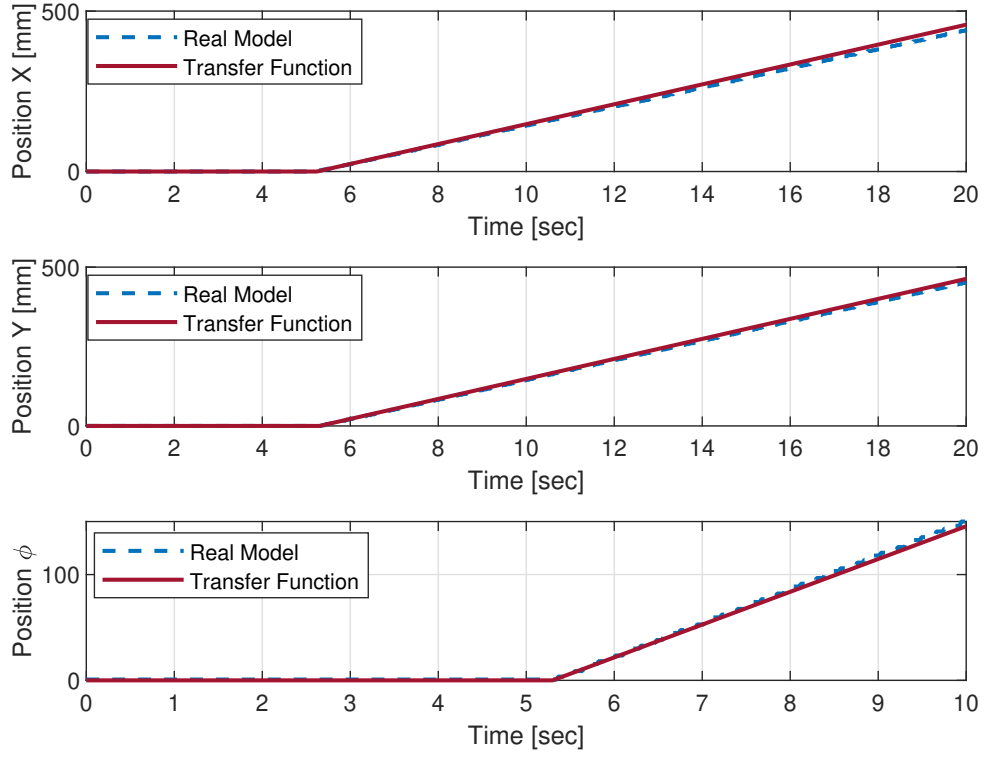


Figure 4.2: Empirical model is based on the experimental data of X , Y , and ϕ .

Table 4.3: Controller tuning parameters for trajectory tracking Simulation

Parameters	PI			GMC			SMC			iPI		
	x	y	ϕ	x	y	ϕ	x	y	ϕ	x	y	ϕ
K_p	2.5	2.0	2.5	-	-	-	-	-	-	-	-	-
K_I	0.6	0.5	0.6	-	-	-	-	-	-	-	-	-
K_1	-	-	-	6.0	3.0	4.0	-	-	-	-	-	-
K_2	-	-	-	2.0	5.0	1.5	-	-	-	-	-	-
λ_1	-	-	-	-	-	-	4.0	3.6	4.0	-	-	-
t_c	-	-	-	-	-	-	0.6	0.3	0.3	-	-	-
K_D	-	-	-	-	-	-	0.3	1.2	0.7	-	-	-
δ	-	-	-	-	-	-	1.9	2.0	1.9	-	-	-
α	-	-	-	-	-	-	-	-	-	30.0	30.0	30.0
K_{P_i}	-	-	-	-	-	-	-	-	-	2.5	2.0	2.5
K_{I_i}	-	-	-	-	-	-	-	-	-	0.6	0.5	0.6
K_{LP1}	-	-	-	-	-	-	-	-	-	$1.0e^{-5}$	$1.0e^{-5}$	$1.0e^{-4}$
K_{LP2}	-	-	-	-	-	-	-	-	-	1.0	1.0	1.0
T_{LP1}	-	-	-	-	-	-	-	-	-	1.0	1.0	1.0
T_{LP2}	-	-	-	-	-	-	-	-	-	1.0	1.0	1.0

Circular trajectory

In Fig. 4.3, all controllers show a fairly uniform response to the reference. However, small differences are observed at the moment they reach the reference. This is due to the initial point set for the robot to perform the circular trajectory.

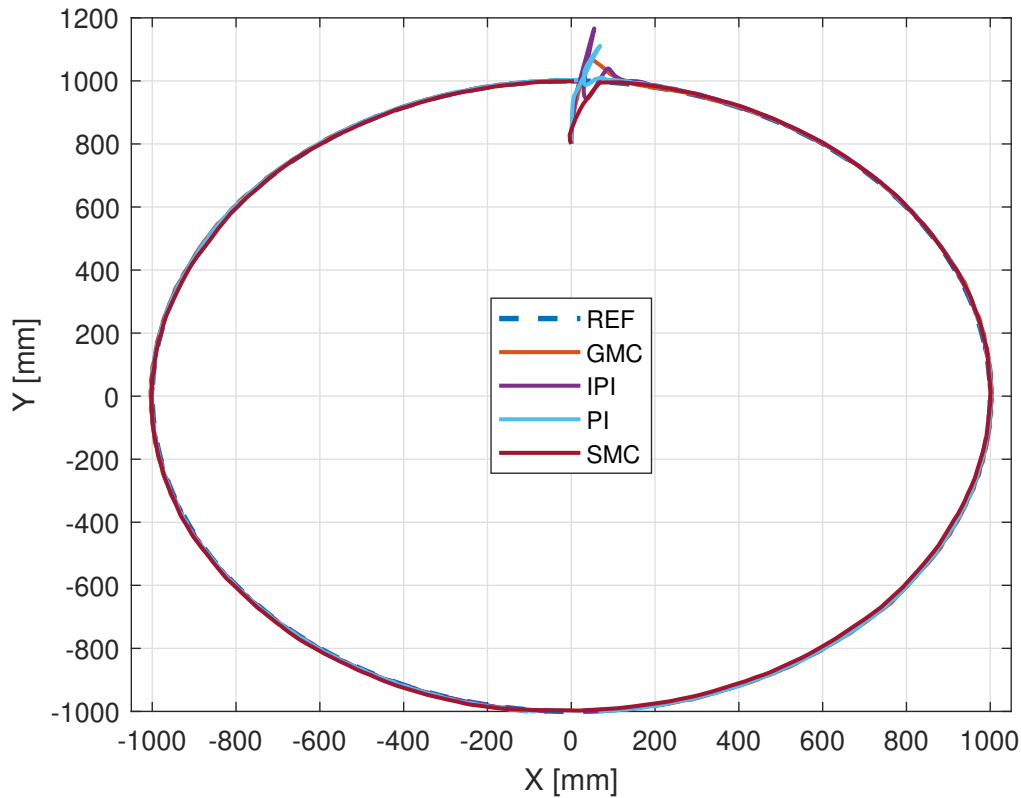


Figure 4.3: Comparative responses of Simulation Circular Trajectory

The main difference is that, when using a conventional controller such as the PI, there is an overshoot before reaching the reference, which is also seen in the iPI and GMC controllers. It is worth mentioning that, when using a Model-Free controller, it approximates the reference to the output. This causes the controller to react more quickly when large changes occur, leading to behavior similar to GMC but with a smoother response than that of the iPI. It can be highlighted that the iPI controller follows the reference more accurately after the overshoot. On the other hand, when using a robust controller like the SMC, a smoother response is observed compared to the other controllers.

In the error graph for the circle in the simulation, shown in Fig. 4.4, the weaknesses of the controller are highlighted in more detail. The overshoot occurs in the y error, specifically when the system starts from a point other than zero, causing an increase in the error. In contrast, in the

ϕ result, there is no significant difference between the controllers. However, by analyzing how the controllers attempt to reach the reference after the change, it becomes clear that the SMC responds much more smoothly.

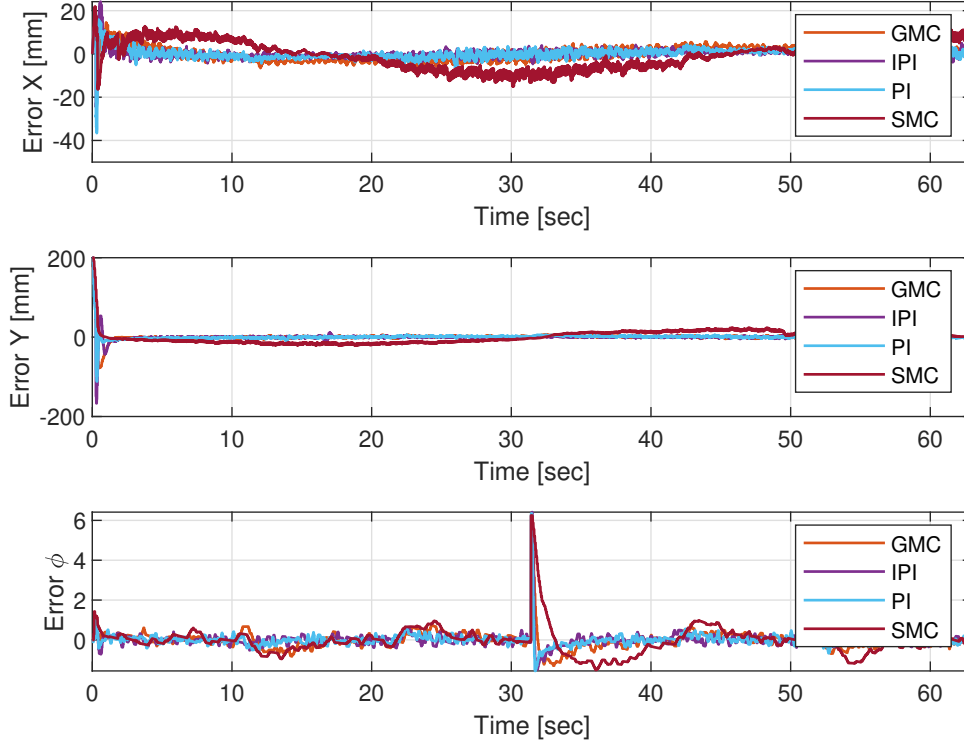


Figure 4.4: Comparative Error of Simulation Responses for Circular Trajectory

Lemniscate Trajectory

The results shown in Fig. 4.5 represent the responses of the controllers. It can be seen that all the controllers provide a satisfactory response when generating the trajectory. The main difference lies in the precision, where the iPI demonstrates the best accuracy, followed by the GMC and the SMC, with the PI controller showing the least precision. This is particularly evident in the curvature, where the PI controller deviates more significantly from the reference.

The error plots show in Fig. 4.6 a similar behavior across all controllers, but there is a noticeable difference with the SMC due to its robustness, leading to this particular response. However, its actions are much smoother. It is also evident that the iPI controller exhibits a much more abrupt response when reaching the reference, followed by the GMC.

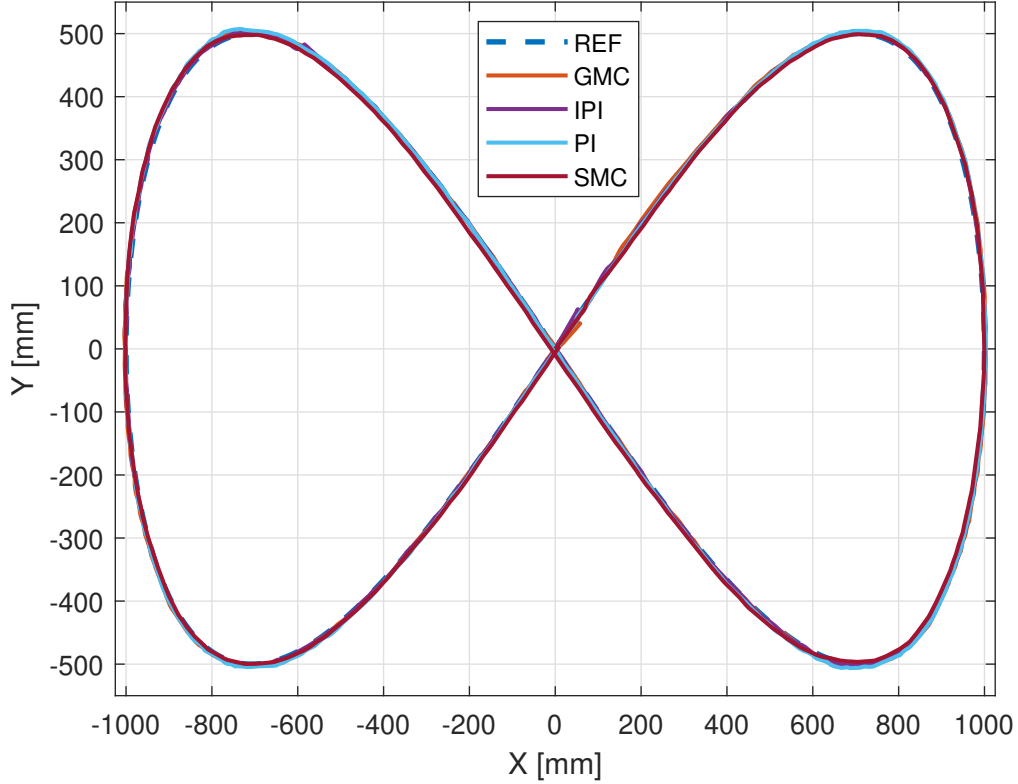


Figure 4.5: Comparative responses of Simulation Lemniscate Trajectory

Square Trajectory

Fig. 4.7 shows the square trajectory generated in the simulation. Unlike the experimental results, the simulation reveals that the controllers initially oscillate before settling on the reference, although these oscillations are smaller compared to those observed in the experiment. Furthermore, the 90-degree corners are traced with significant precision, which is not the case in the experimental setup. Nevertheless, the behavior of all the controllers remains quite similar, with the system following the reference nearly perfectly.

When analyzing the error behavior at each position in Fig. 4.8, it can be observed that the controllers aim to maintain the error close to zero in the X direction, except for the SMC, which exhibits significant oscillations and does not stabilize. In the Y direction, all controllers keep the error at zero, although there are minor oscillations at the start as the controllers work to reach the reference. Finally, in the ϕ direction, the error behavior is similar to the other simulation errors, with noticeable kicks occurring due to the changes in the trajectory angle.

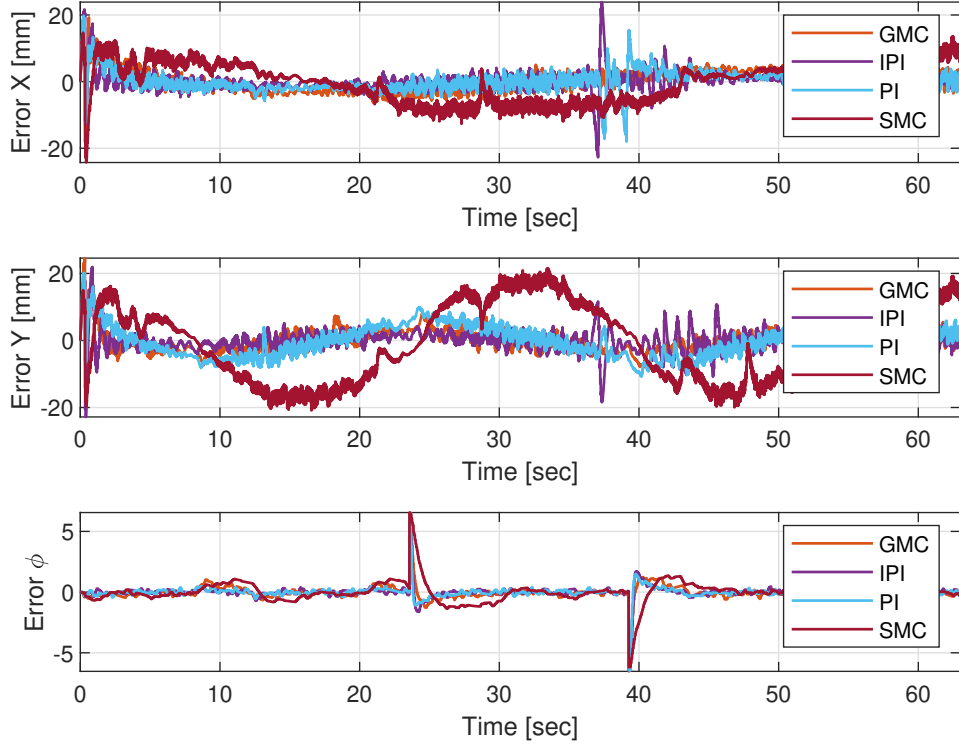


Figure 4.6: Comparative Error of Simulation Responses for Lemniscate Trajectory

4.0.4 Experimental

Table 4.4: Controller tuning parameters for trajectory tracking Experimental

2*Parameters	PI			GMC			SMC			iPI		
	x	y	ϕ	x	y	ϕ	x	y	ϕ	x	y	ϕ
K_p	1.0	1.2	2.5	-	-	-	-	-	-	-	-	-
K_I	0.6	1.0	0.6	-	-	-	-	-	-	-	-	-
K_1	-	-	-	3.0	3.0	1.0	-	-	-	-	-	-
K_2	-	-	-	2	4.0	1.5	-	-	-	-	-	-
λ_1	-	-	-	-	-	-	2.4	3.5	2.63	-	-	-
t_c	-	-	-	-	-	-	0.6	0.4	0.5	-	-	-
K_D	-	-	-	-	-	-	5.0	4.7	5.8	-	-	-
δ	-	-	-	-	-	-	1.13	1.3	1.2	-	-	-
α	-	-	-	-	-	-	-	-	-	30	30	30
K_{P_i}	-	-	-	-	-	-	-	-	-	1.0	1.2	2.5
K_{I_i}	-	-	-	-	-	-	-	-	-	0.6	1.0	0.6
K_{LP1}	-	-	-	-	-	-	-	-	-	$1.0e^{-5}$	$1.0e^{-5}$	$1.0e^{-4}$
K_{LP2}	-	-	-	-	-	-	-	-	-	1.0	1.0	1.0
T_{LP1}	-	-	-	-	-	-	-	-	-	1.0	1.0	1.0
T_{LP2}	-	-	-	-	-	-	-	-	-	1.0	1.0	1.0

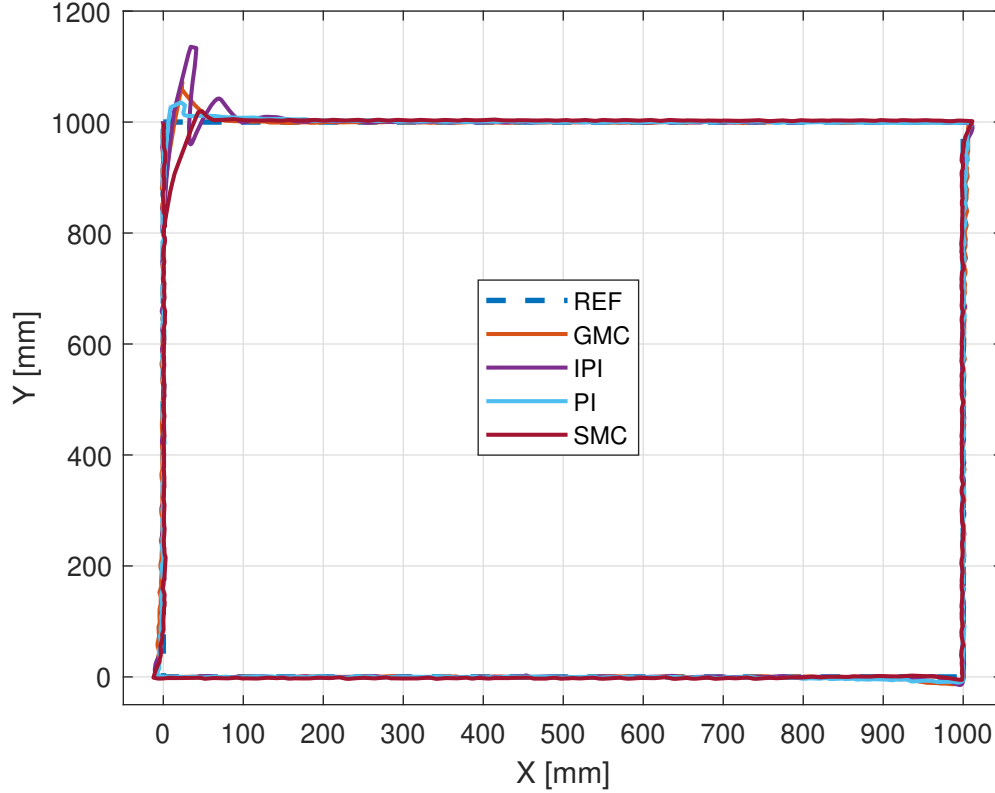


Figure 4.7: Comparative responses of Simulation Square Trajectory

Circular trajectory

The Fig. 4.9 shows the experimental response of trajectory tracking for a circular path using different controllers. The graph compares the performance of GMC, iPI, PI, and SMC regarding the reference trajectory.

GMC reaches the reference quickly but shows noticeable oscillations around the upper section of the circle. iPI follows the reference closely with minimal oscillations, indicating stable performance. PI also follows the reference well, with slightly more oscillations compared to iPI. SMC shows the best performance, closely following the reference trajectory with the least number of oscillations.

The error in variables X , Y , and ϕ for each controller acting on the circular path is shown in Fig. 4.10. Initially, all controllers experience overshoot as the path does not start at zero. The iPI mitigates the error faster and with less oscillation compared to the other controllers, stabilizing

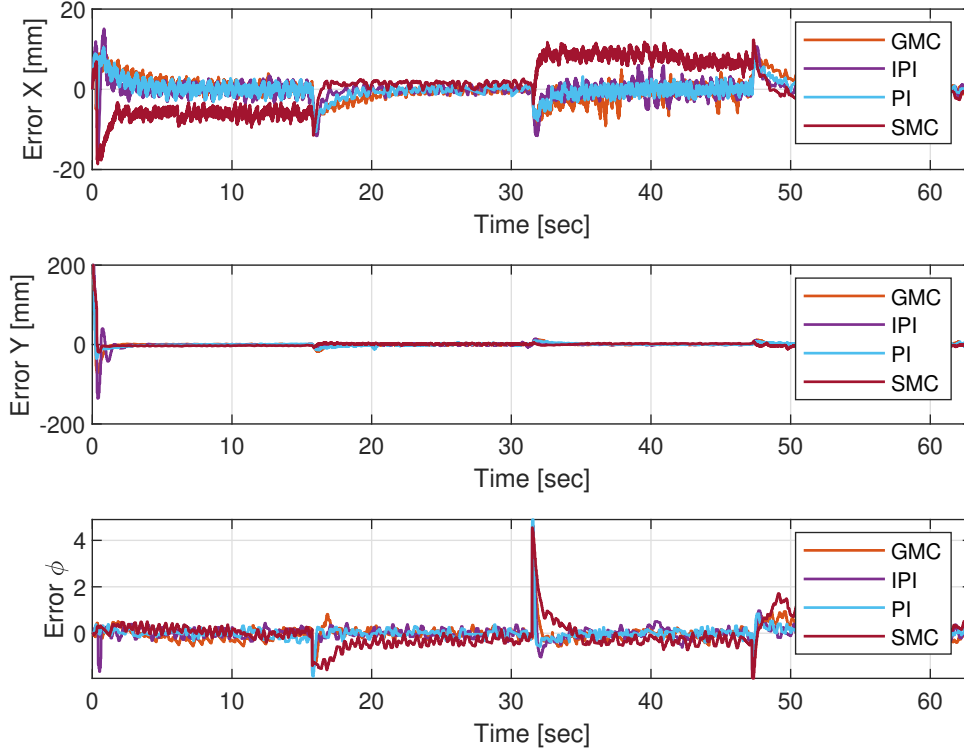


Figure 4.8: Comparative Error of Simulation Responses for Square Trajectory

closer to zero. In contrast, the PI controller exhibits the highest error. In the Y direction, all controllers reach zero error, but the SMC has the best performance. For ϕ , the iPI maintains the lowest error, even though all controllers aim for zero. Additionally, around the 30-second mark, a disturbance occurs due to the robot's motors and the surface it operates on.

Lemniscate Trajectory

Fig. 4.11 shows the response of the controllers for the lemniscate path. At this point, the SMC controller quickly reaches the reference with the least possible oscillation, while the PI controller has trouble staying on the reference. Although it manages to follow it, it does not fully reach it. However, the iPI and GMC controllers exhibit intermediate behavior.

Analyzing the error comparison of each controller in Fig. 6.1, it is clear that both directions and the angle ϕ aim to approach zero. However, in the case of ϕ , certain oscillations occur at different sample times because of the robot's motor, which responds with small jolts. Additionally, the GMC controller exhibits the error closest to zero, while the PI controller remains the farthest

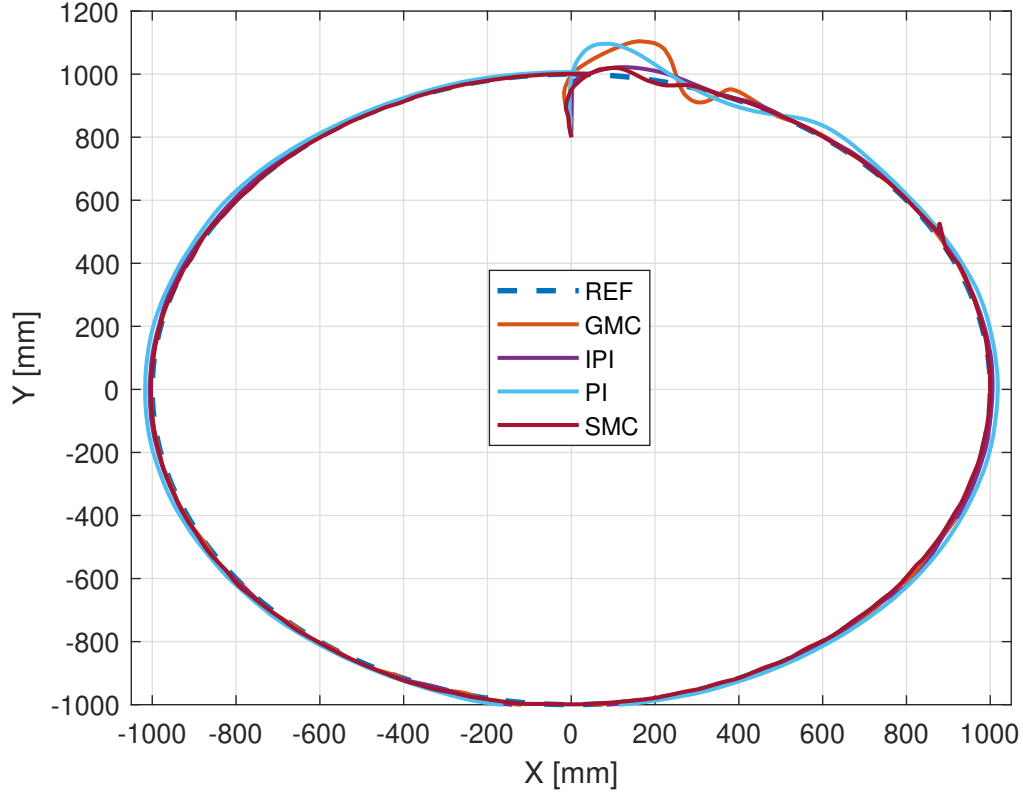


Figure 4.9: Comparative responses of Experimental Circular Trajectory

from it.

Square Trajectory

Fig. 4.13 illustrates the controller responses to a square trajectory, where problems are observed when reaching the reference point, as the trajectory does not start at zero. Furthermore, the GMC experiences significant oscillations at the beginning. Concerning the corners, the SMC can navigate the curves smoothly, while the other controllers exhibit pronounced curvature at the 90-degree angles. As in previous cases, the PI controller proved to be the least effective for this application.

Fig. 4.14 shows that all controllers strive for zero error, despite initially exhibiting a significant overshoot. In this case, all controllers eventually stabilize around zero, although the GMC shows considerable oscillations in the ϕ variable, and the PI controller takes longer to reach zero in the X direction.

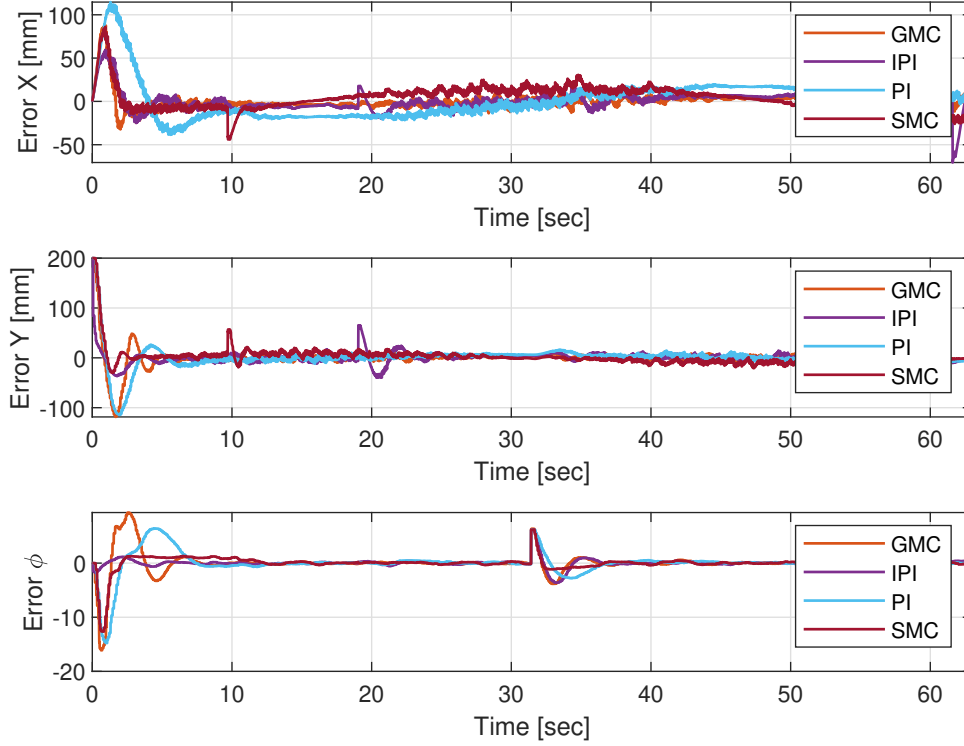


Figure 4.10: Comparative Error of Experimental Responses for Circular Trajectory

4.0.5 Performance Indices

The Table 4.5 compares the performance of four controllers for the circular trajectory, using Integral Squared Error (ISE) for the x , y , and ϕ coordinates in both simulation and experimental conditions.

In the simulation results (ISE_{sim}), the PI controller shows the lowest errors in x and ϕ , but the highest error in y . GMC has higher errors across all coordinates compared to PI, with significant errors in y . SMC exhibits the largest errors in both x and y , due to its robust nature, but performs better in terms of angular precision (ϕ). iPI shows similar performance to PI in x and ϕ , but has a higher error in y . For the experimental results (ISE_{exp}), PI shows the highest errors, especially in x and y , indicating lower performance in real-world conditions. GMC, while performing better than PI, still has high y -coordinate errors. SMC performs well in ϕ , but still has significant errors in x and y . iPI shows the best angular precision with a low ISE_{ϕ} , but still has considerable errors in the x and y coordinates.

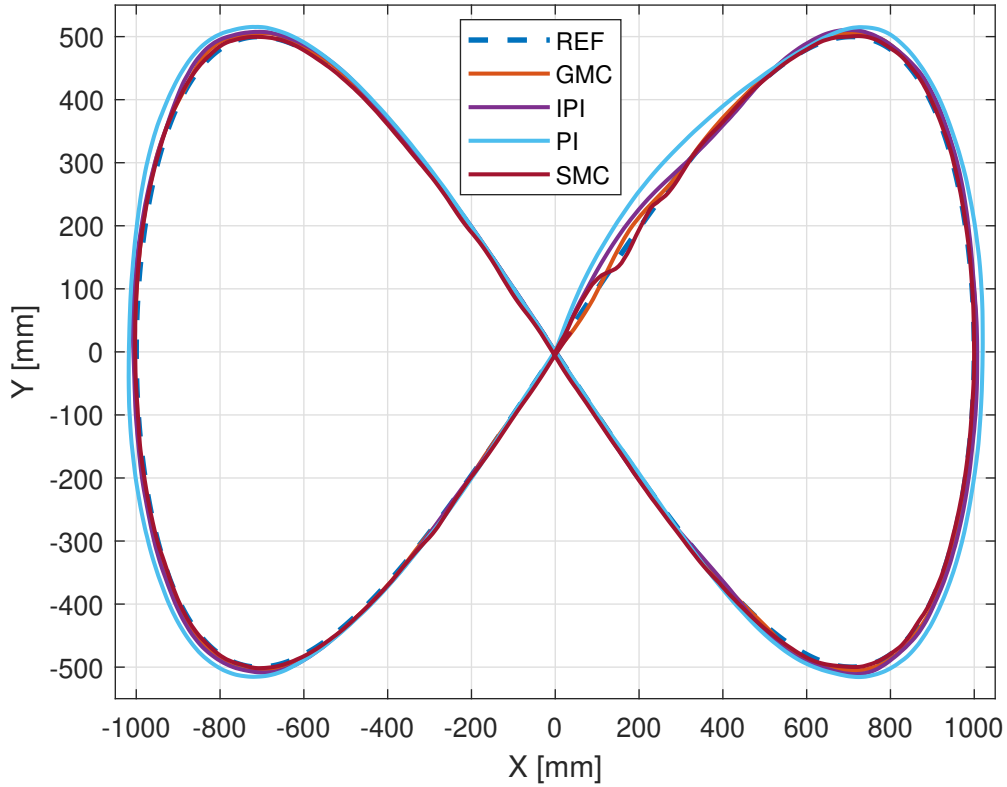


Figure 4.11: Comparative responses of Experimental Lemniscate Trajectory

Table 4.5: Comparison of Performance Parameters for Simulation and Experimental Circular Trajectory

Controllers	ISE _{sim}			ISE _{exp}		
	x	y	ϕ	x	y	ϕ
PI	282.2	5743.0	7.9	3.2×10^4	3.6×10^4	302.5
GMC	687.1	9455.0	14.3	1.3×10^4	3.6×10^4	339.4
SMC	4986.0	9166.0	32.1	1.6×10^4	2.3×10^4	124.0
iPI	325.7	1.0×10^4	9.3	7874.0	9080.0	37.5

The Table 4.6 compares the performance of four controllers for the Lemniscate trajectory in both simulation and experimental conditions, using the ISE for the x , y , and ϕ coordinates.

In the simulation results, the PI controller shows the lowest errors in x and ϕ , but higher errors in y . GMC offers a balanced performance, while SMC exhibits higher errors, especially in the x -coordinate, due to its robust nature. iPI performs similarly to PI in x and y , but with higher error in ϕ . For the experimental results, PI shows significantly higher errors, particularly in x and y , indicating poorer real-world performance. GMC performs better experimentally, though still with higher y -errors. SMC exhibits the lowest x coordinate error, but struggles with

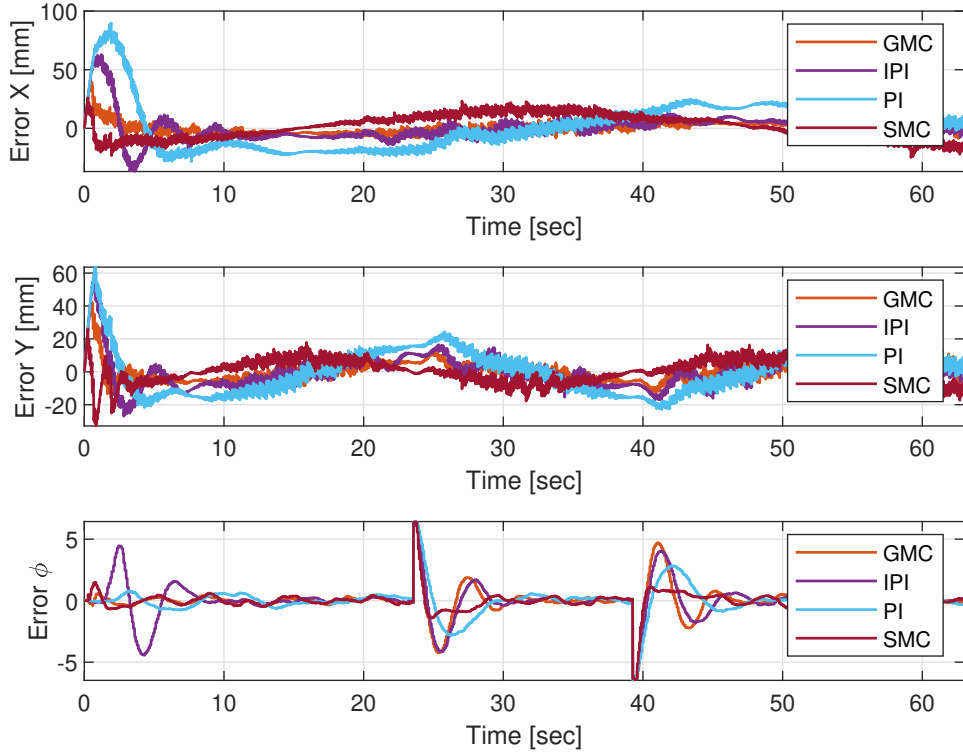


Figure 4.12: Comparative Error of Experimental Responses for Lemniscate Trajectory

a higher angular (ϕ) error. iPI has competitive performance in x and y , but a notable increase in ϕ error.

Table 4.6: Comparison of Performance Parameters for Simulation and Experimental Lemniscate Trajectory

Controllers	ISE _{sim}			ISE _{exp}		
	x	y	ϕ	x	y	ϕ
PI	616.5	1476.0	16.7	2.9×10^4	1.2×10^4	89.8
GMC	730.0	711.8	24.5	1637.0	2417.0	94.2
SMC	2.5×10^4	3819.0	38.7	6497.0	2912.0	44.1
iPI	596.5	653.3	17.6	8282.0	5974.0	140.1

The table compares the performance of four controllers for the square trajectory, using ISE for the x , y , and ϕ coordinates under both simulation and experimental conditions.

In the simulation results, the PI controller shows the lowest errors in x and ϕ , but the highest error in y . GMC has higher errors than PI, particularly in y , with relatively higher errors in all coordinates. SMC performs poorly in the coordinates x and y , but still manages to maintain reasonable angular precision (ϕ). iPI has a performance similar to that of PI in x and ϕ , but

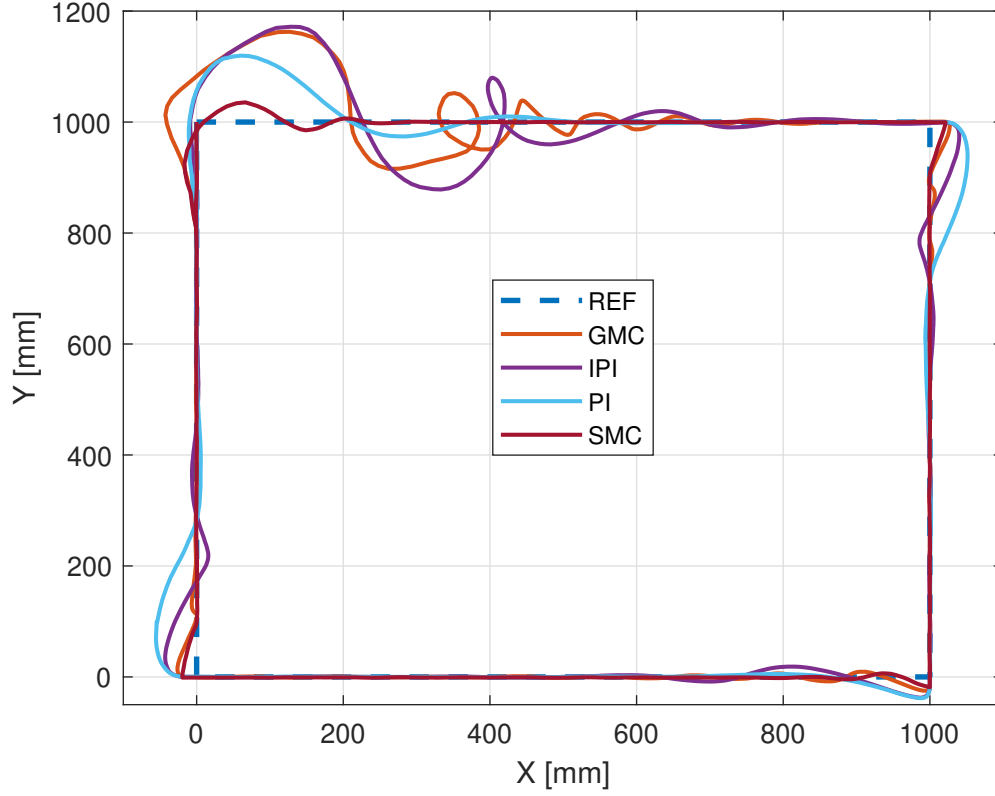


Figure 4.13: Comparative responses of Experimental Square Trajectory

with significantly higher errors in y . For the experimental results, PI shows the highest errors in all coordinates, especially in x and y , reflecting poor real-world performance. GMC performs better than PI, but its y -coordinate errors are considerably high. SMC shows better performance than GMC in the x and y coordinates, while still maintaining a relatively low error in ϕ . iPI, although showing good angular precision, has significantly higher errors in x and y , affecting its overall performance.

Table 4.7: Comparison of Performance parameters for Simulation and Experimental Square Trajectory

Controllers	ISE _{sim}			ISE _{exp}		
	x	y	ϕ	x	y	ϕ
PI	202.7	5162.0	4.9	3.0×10^4	4.3×10^4	475.1
GMC	429.7	6949.0	8.5	1.1×10^4	7.4×10^4	1575.0
SMC	1861	8278.0	23.3	5429.0	2.6×10^4	189.6
iPI	319.9	1.3×10^4	6.9	2.5×10^4	8.1×10^4	587.3

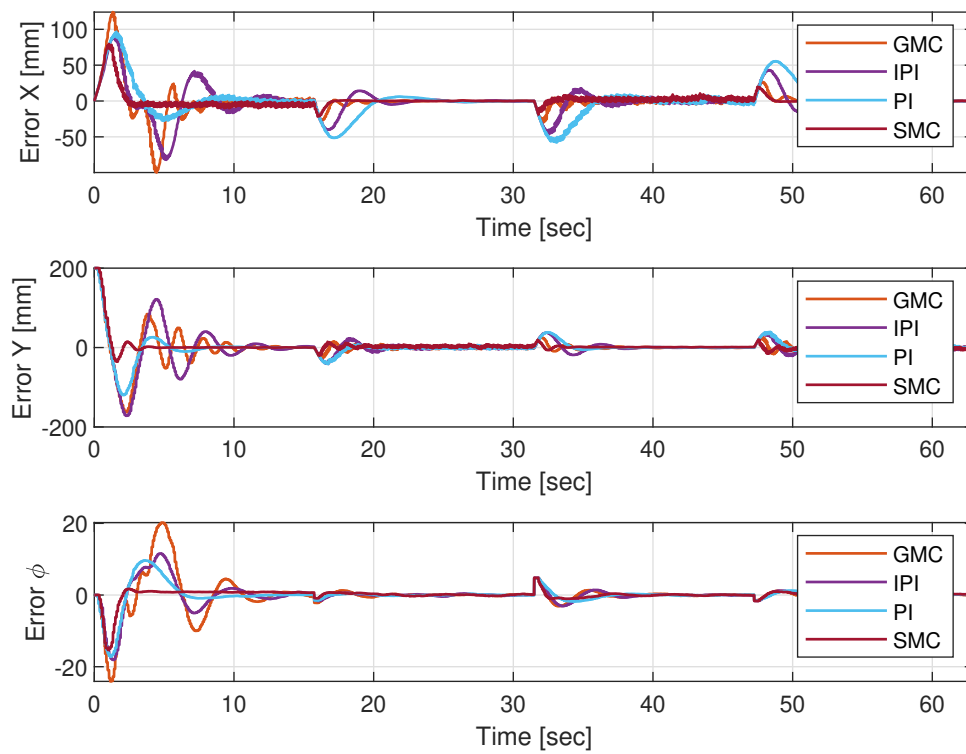


Figure 4.14: Comparative Error of experimental responses for Square Trajectory

CHAPTER 5

CONCLUSIONS

This study compared control strategies for trajectory tracking on the Festo Robotino robot. Each approach has specific benefits and drawbacks depending on the trajectory and context. The PI controller is simple to implement, but struggles with precision on complex paths. The GMC improves tracking accuracy, but requires more computation. SMC is robust against disturbances, but can cause chattering. The model-free method is flexible and suitable for high model uncertainty or scenarios that need adaptability.

The main difficulties encountered during the study included issues with connecting the Robotino to MATLAB, a problem exacerbated by the fact that the MATLAB version used was from 2008. This older version led to compatibility issues, impacting the overall performance and calibration of the system. Moreover, discrepancies between the results obtained from simulations and actual hardware tests were observed. These differences can be attributed to several factors, including errors in the robot's calibration, the inherent limitations of the software and hardware, and the challenges posed by real-world conditions that could not be fully replicated in simulations.

This comparative analysis offers useful insights for developing trajectory-tracking systems for mobile robots, especially in changing environments, with the Robotino Festo serving as a suitable option for academic applications. Despite the challenges faced, the study contributes valuable knowledge that can guide future improvements and optimization of robotic control systems, ensuring better accuracy and reliability in practical applications.

BIBLIOGRAPHY

- Camacho, O., Rosales, A., and Rivas, F. (2020). *Control de Procesos*. Escuela Politécnica Nacional, 1 edition.
- Capito, L., Proaño, P., Camacho, O., Rosales, A., and Scaglia, G. (2016). Experimental comparison of control strategies for trajectory tracking for mobile robots. *International Journal of Automation and Control*, 10(3):308–327.
- Fliess, M. and Join, C. (2013). Model-free control. *International journal of control*, 86(12):2228–2252.
- Fuentes, M., Castillo, O., and Cortés-Antonio, P. (2021). Review of fuzzy control for path tracking in the robotino system. *Recent Advances of Hybrid Intelligent Systems Based on Soft Computing*, pages 205–215.
- Guevara, L., Camacho, O., Rosales, A., Guevara, J., and Scaglia, G. (2019). A linear algebra controller based on reduced order models applied to trajectory tracking for mobile robots: an experimental validation. *International Journal of Automation and Control*, 13(2):176–196.
- Gutiérrez Ramón, X. (2016). Trajectory control design of a mobile robot with computer vision. B.S. thesis, Universitat Politècnica de Catalunya.
- Herrera, M., Benítez, D. S., Pérez, N., Di Teodoro, A., and Camacho, O. (2023). A novel hybrid control approach with pso optimization for processes with long time-delay. In *2023 IEEE Colombian Conference on Applications of Computational Intelligence (ColCACI)*, pages 1–6.
- Künhe, F., Gomes, J., and Fetter, W. (2005). Mobile robot trajectory tracking using model predictive control. In *II IEEE latin-american robotics symposium*, volume 51, page 5.
- Lee, P. and Sullivan, G. (1988). Generic model control (gmc). *Computers & Chemical Engineering*, 12(6):573–580.

- Medina, L., Guerra, G., Herrera, M., Guevara, L., and Camacho, O. (2024). Trajectory tracking for non-holonomic mobile robots: A comparison of sliding mode control approaches. *Results in Engineering*, 22:102105.
- Morales, L., Herrera, M., Camacho, O., Leica, P., and Aguilar, J. (2021). Lamda control approaches applied to trajectory tracking for mobile robots. *IEEE Access*, 9:37179–37195.
- Morales, S., Magallanes, J., Delgado, C., and Canahuire, R. (2018). Lqr trajectory tracking control of an omnidirectional wheeled mobile robot. In *2018 IEEE 2nd Colombian Conference on Robotics and Automation (CCRA)*, pages 1–5. IEEE.
- Ogunnaike, B. A. and Ray, W. H. (1994). Process dynamics, modeling, and control. (*No Title*).
- Precup, R.-E., Radac, M.-B., Roman, R.-C., and Petriu, E. M. (2017). Model-free sliding mode control of nonlinear systems: Algorithms and experiments. *Information Sciences*, 381:176–192.
- Salinas, L. R., Santiago, D., Slawiński, E., Mut, V. A., Chavez, D., Leica, P., and Camacho, O. (2018). P+ d plus sliding mode control for bilateral teleoperation of a mobile robot. *International Journal of Control, Automation and Systems*, 16:1927–1937.
- Smith, C. A. and Corripio, A. B. (2005). *Principles and practices of automatic process control*. John wiley & sons.
- Utkin, V., Poznyak, A., Orlov, Y. V., and Polyakov, A. (2020). *Road map for sliding mode control design*. Springer.
- Weinert, H. and Pensky, D. (2011). Mobile robotics in education and student engineering competitions. In *IEEE Africon'11*, pages 1–5. IEEE.
- Yang, H., Fan, X., Shi, P., and Hua, C. (2015). Nonlinear control for tracking and obstacle avoidance of a wheeled mobile robot with nonholonomic constraint. *IEEE Transactions on Control Systems Technology*, 24(2):741–746.

CHAPTER 6

APPENDIX

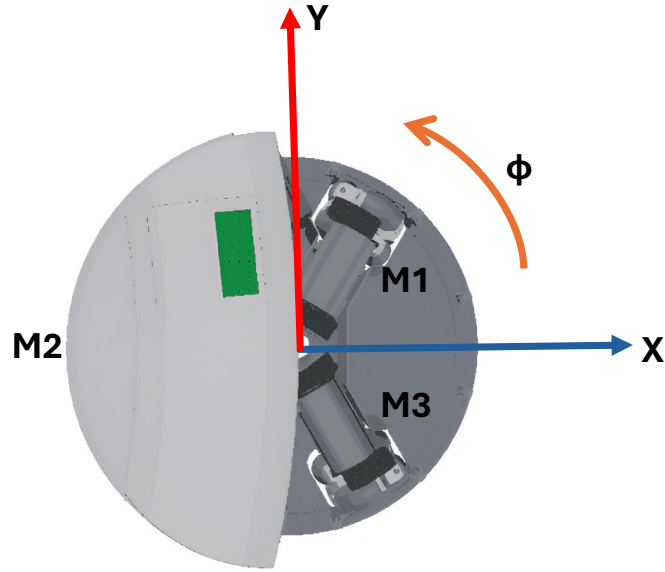


Figure 6.1: Robotino Direction

For the results, the controllers were compared using different types of trajectories, which are detailed below:

Circular:

$$x_{ref} = 1000 \sin\left(\frac{t}{10}\right) \quad (6.1)$$

$$y_{ref} = 1000 \cos\left(\frac{t}{10}\right) \quad (6.2)$$

$$\phi_{ref} = \tan^{-1}(x_{ref}, y_{ref}) \quad (6.3)$$

Lemniscate:

$$x_{ref} = 1000 \sin\left(\frac{t}{10}\right) \quad (6.4)$$

$$y_{ref} = 1000 \cos\left(\frac{t}{10}\right) \sin\left(\frac{t}{10}\right) \quad (6.5)$$

$$\phi_{ref} = \tan^{-1}(x_{ref}, y_{ref}) \quad (6.6)$$

Square:

$$x(t) = \begin{cases} \frac{1000}{T}t & \text{if } 0 \leq t < T \\ 1000 & \text{if } T \leq t < 2T \\ 1000 - \frac{1000}{T}(t - 2T) & \text{if } 2T \leq t < 3T \\ 0 & \text{if } 3T \leq t < 4T \end{cases} \quad (6.7)$$

$$y(t) = \begin{cases} 1000 & \text{if } 0 \leq t < T \\ 1000 - \frac{1000}{T}(t - T) & \text{if } T \leq t < 2T \\ 0 & \text{si } 2T \leq t < 3T \\ \frac{1000}{T}(t - 3T) & \text{if } 3T \leq t < 4T \end{cases} \quad (6.8)$$

$$\phi(t) = \begin{cases} 0 & \text{si } 0 \leq t < T \\ \frac{\pi}{2} & \text{si } T \leq t < 2T \\ \pi & \text{si } 2T \leq t < 3T \\ \frac{3\pi}{2} & \text{si } 3T \leq t < 4T \end{cases} \quad (6.9)$$

Where T is the simulation time.