الجمهورية الجزائرية الديمقراطية الشعبية
THE PEOPLE DEMOCRATIC REPUBLIC OF ALGERIA
وزارة التعليم العالي و البحث العلمي
THE MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
جامعة عمّار ثليجي بالأغواط
AMAR TELIDJI UNIVERSITY OF LAGHOUAT
كلية التكنولوجيا
FACULTY OF TECHNOLOGY
DEPARTMENT OF ELECTROTECHNIC

Master's dissertation

Domain :	Science and Technology
Field :	Automatic
Option :	Automatic and systems

Presented by:

- **Abdelali Ali**
- **Yagoubi Mehdi Aissa**

Title:

Applying Classical Control to Unicycle Robot Navigation

Mr. Rahmani Belkacem	***President***
Mr. Rayane Khaled	***Examinator***
Dr. Ameur Iyad	***Supervisor***
Mr. Brahimi Tahar	***Co-supervisor***

Academic Year : 2024/2025

Acknowledgements

In the name of Allah, the Most Gracious, the Most Merciful.

First and foremost, we thank Allah Almighty for granting us the strength, determination, and patience to complete this work.

We would like to express our deepest gratitude to our supervisor, **Dr. Ameur Iyad**, for their invaluable guidance, support, and encouragement throughout all stages of this thesis. Their constructive feedback and continuous motivation played a vital role in shaping our research.

Our sincere thanks also go to all the professors and staff members of the **Department of Electrotechnics,** whose dedication to teaching and research has been a constant source of inspiration.

To our families, we are forever grateful. Their unconditional love, support, and belief in us have always been our greatest strength.

We would also like to thank our classmates and friends who provided assistance, shared knowledge, and offered encouragement throughout this academic journey.

Finally, we hope this work will be a meaningful contribution to the field, and we ask Allah to make it beneficial and accepted.

Dedication

To those who taught me my first letters,

To those who instilled in me the values of perseverance and hard work,

To all who stood by me during moments of struggle,

To my beloved family, my greatest source of support,

To my esteemed professors, for their generous knowledge and guidance,

To my colleagues and friends, who shared with me the journey of learning and challenges,

To everyone who contributed with a word, advice, or sincere prayer,

I dedicate this humble work as a token of gratitude and appreciation.

Ali

Dedication

This work is dedicated to all those who lit the path of knowledge before me.

To my family, whose love and sacrifices have been my constant source of strength.

To the teachers and mentors who believed in me and guided me with wisdom.

To my friends, for their encouragement, laughter, and companionship throughout this journey.

To every soul who offered a kind word, a helping hand, or a silent prayer—thank you.

This achievement is as much yours as it is mine.

Mehdi

Abstract

This study aims to evaluate the performance of a Proportional-Integral-Derivative (PID) controller in guiding a mobile robot of the unicycle type along two reference trajectories: circular and figure-eight. The proposed approach starts with precise kinematic modeling of the robot, followed by the design and implementation of a PID control strategy. The effectiveness of the system is then assessed through detailed simulations conducted in the MATLAB environment.

Keywords: Mobile robots, PID controller, trajectory tracking, kinematic modeling, motion control, MATLAB, energy efficiency.

Résumé

Cette étude vise à évaluer les performances d’un contrôleur Proportionnel-Intégral-Dérivé (PID) dans le guidage d’un robot mobile de type unicycle le long de deux trajectoires de référence : circulaire et en forme de huit. L’approche proposée commence par une modélisation cinématique précise du robot, suivie de la conception et de la mise en œuvre d’une stratégie de commande PID. L’efficacité du système est ensuite évaluée à travers des simulations détaillées réalisées dans l’environnement MATLAB.

Mots-clés : Robots mobiles, contrôleur PID, suivi de trajectoire, modélisation cinématique, commande du mouvement, MATLAB, efficacité énergétique.

ملخص

تهدف هذه الدراسة إلى تقييم أداء المتحكم التناسبي-التكاملي-التفاضلي (PID) في توجيه روبوت متحرك من نوع أحادي العجلة (unicycle) على مسارين مرجعيين: دائري وشكل الرقم 8. تبدأ المنهجية المقترحة بنمذجة حركية دقيقة للروبوت، تليها تصميم وتنفيذ استراتيجية تحكم باستخدام PID. ثم يتم تقييم فعالية النظام من خلال محاكاة مفصلة باستخدام بيئة MATLAB.

الكلمات المفتاحية :الروبوتات المتنقلة، المتحكمPID ، تتبع المسار، النمذجة الحركية، التحكم في الحركة، MATLAB، كفاءة الطاقة.

Table of Contents

General Introduction

General Introduction

Mobile robotics has transformed numerous sectors, from industrial automation to service applications, by enabling autonomous navigation in dynamic environments. Among mobile robots, unicycle-type robots, characterized by two independently driven wheels and nonholonomic constraints, are widely studied for their simplicity and versatility. These robots are ideal for tasks requiring precise trajectory tracking, such as exploration, surveillance, and personal assistance, but their nonlinear dynamics pose significant control challenges.

The present thesis is structured into four main chapters, each addressing a key aspect of the study focused on the control of unicycle-type mobile robots.

The first chapter presents general concepts and foundational notions in robotics. It includes a historical overview and classification of mobile robots, with a particular focus on wheeled robots of the unicycle type.

The second chapter is dedicated to the kinematic modeling of unicycle-type mobile robots chapter explores the PID (Proportional-Integral) control technique. It discusses the theoretical background of PID control, compares open-loop and closed-loop systems, and justifies the use of PID control over alternative control strategies in the context of trajectory tracking.

The third chapter presents and analyzes the simulation results obtained using MATLAB. The effectiveness of the PID controller is evaluated across two different trajectory scenarios, and the performance is interpreted in detail.

Finally, the thesis concludes with a general conclusion that summarizes the main findings, highlights the scientific contributions, identifies limitations, and outlines future research directions.

This work lays a solid foundation for further developments in navigation of wheeled mobile robots and contributes to both theoretical understanding and practical implementation in mobile robotics.

Chapter I: Generalities

I.1. Introduction

The term "robot" often evokes images of mechanical systems performing industrial tasks, such as assembly or material handling. While stationary robots dominate factory automation, wheeled mobile robots remain underutilized despite their potential in applications like autonomous cleaning, mobility assistance, and intelligent navigation. This chapter introduces fundamental concepts in robotics, provides a brief historical overview, and compares mobile robots with traditional automated systems, emphasizing their functionality and autonomy.

I.2. Historical Background

The concept of robots has evolved over centuries, from early mechanical automata to modern intelligent systems. The term "robot" derives from the Slavic word "robota," meaning work or chore, and was popularized by Czech playwright Karel Čapek in his 1921 science fiction play R.U.R. (Rossum's Universal Robots). In the play, robots are artificial human-like beings, akin to modern cyborgs or androids. The term rapidly gained global recognition, though the idea of mechanical beings predates it.[2]

1. **Ancient and Medieval Times: The First Automata**

Myths and Legends: Greek mythology features Talos, a bronze automaton built by Hephaestus to guard Crete. Similar tales of mechanical servants appear in Chinese and Arabic literature.

2. **Renaissance and Pre-Industrial Period: The Age of Automata**

Leonardo da Vinci (1495): Designed a mechanical knight capable of simple movements, one of the earliest known humanoid robot concepts.

18th Century: Engineers like Jacques de Vaucanson created advanced automata, such as a mechanical duck that could move, eat, and simulate digestion.

3. **Industrial Revolution (19th Century)**

The advent of steam engines and mechanized production led to more sophisticated automated machines, used in industries like clockmaking and entertainment.

4. **20th Century: The Birth of Modern Robots**

1921: Czech writer Karel Čapek introduced the word "robot" in his play R.U.R. (Rossum's Universal Robots).

1940-1950: Norbert Wiener develops cybernetics, laying the foundation for automated control systems and artificial intelligence.

1956: George Devol and Joseph Engelberger invent Unimate, the first programmable industrial robot, revolutionizing manufacturing.

5. **Late 20th Century to Today: The Era of Artificial Intelligence**

1980-2000: Industrial robots expand, and humanoid robots like Honda's ASIMO (2000) emerge.

2010-2020: Advances in artificial intelligence, machine learning, and autonomous robotics drive innovations in medicine, space exploration, and household robotics (e.g., Sophia, Boston Dynamics' robots).

I.3.Robot Classification

Robots are classified into categories based on criteria such as physical characteristics, performance, functionality, and application. These classifications help researchers and engineers optimize robot design for specific tasks. Key classification criteria include:

- *Efficiency*: The ability to perform tasks with minimal resource consumption.
- *Precision:* The accuracy of task execution.
- *Speed:* The rate of task completion.
- *Flexibility:* The capacity to adapt to different tasks.
- *Adaptability*: The ability to operate in varying environments.

Due to this complexity, robotics experts, particularly those from organizations such as:

- Japan Industrial Robot Association (JIRA)
- Robotics Institute of America (RIA)
- French Association of Industrial Robotics (AFRI)

have developed classifications based on several criteria, including:

- Robot positioning mode
- Application field
- Locomotion mechanisms and kinematics

- Overall robot architecture
- Technological generation of the robot
- Size and operational capacity of the robot
- Type of controller used
- Integrated sensors for perception and interaction
- General design and mechanical structure

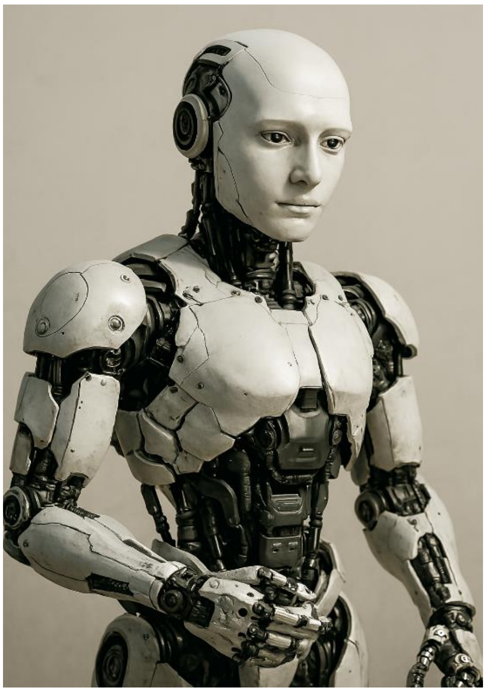
This structured classification facilitates the analysis and improvement of robot designs tailored to specific applications, fostering innovation in robotics technology [3].

I.4.Types of Robots

I.4.1. Humanoid Robots

Humanoid robots are among the most iconic, largely due to their presence in science fiction. They are distinguished by their human-inspired appearance, typically featuring a torso, head, two arms, and two legs. A well-known example is Honda’s Asimo robot.

Some of these robots replicate only part of the human body, such as Nexi, developed by MIT, which focuses primarily on facial expressions and social interactions. When a humanoid robot aims to mimic not only human appearance but also behavior, it is referred to as an android. A notable example is Actroid-DER, designed by the company Kokoro [4].



_ **Figure.1.1:** Humanoid robot

I.4.2. Industrial Robots

Industrial robots are primarily stationary machines, although some can be mounted on rails to extend their range of motion. These robots are designed to automate various tasks in factories and production lines, such as object handling (Pick and Place robots), welding, and painting .

Among the most common industrial robots are:

- ✓ *SCARA robots*: Known for high precision and speed.
- ✓ *Six-axis anthropomorphic robots*: Versatile robotic arms.
- ✓ *Delta robots*: highly valued for their speed and efficiency in sorting and packaging tasks [5].

Figure.1.2: Industrial robot

I.4.3. Wheeled Mobile Robots [6]

In the context of this study, the focus is placed on wheeled mobile robots, particularly those of the unicycle type. mobile robots can be classified based on their locomotion mechanisms and kinematic configurations. Among the various architectures, four main categories of wheeled robots are commonly encountered, primarily distinguished by the number and arrangement of wheels.

- **Unicycle-Type Robot**

The unicycle robot is equipped with two independently driven wheels, typically positioned on either side of the chassis, and may include one or more caster wheels to ensure mechanical stability. The center of rotation is located along the axis connecting the two drive wheels. Due to its non-holonomic constraints, the robot cannot move in a direction perpendicular to its driving wheels. As a result, its mobility is achieved through a combination of in-place rotations and straight-line movements, allowing it to reach target positions effectively.

- **Tricycle-Type Robot**

The tricycle-type robot consists of two fixed (non-steerable) wheels mounted on a common axle, and a steerable front wheel located along the robot's longitudinal axis. Its motion is controlled by the velocity of the fixed wheels and the steering angle of the front wheel. The instantaneous center of rotation is found at the intersection of the axle of the rear wheels and the axis of the steering wheel. Like the unicycle, this configuration also exhibits non-holonomic behavio**r**, restricting lateral motion.

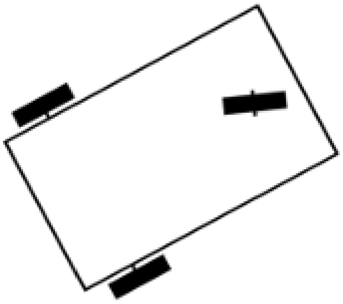


Figure.1.3 Tricycle-Type Robot

- **Car-Type Robot**

The car-type robot is a more stable variant of the tricycle model. It features four wheels, with two fixed rear wheels and two steerable front wheels, arranged on two parallel axles. Its stability is improved due to the additional support point. From a kinematic standpoint, it shares similar properties with the tricycle-type robot. In fact, it can be modeled equivalently by replacing the two front wheels with a single centrally located steering wheel, without altering the location of the center of rotation.

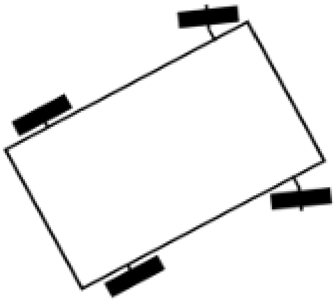


Figure.1.4 Car-Type Robot

- **Omnidirectional Robot**

An omnidirectional robot is capable of moving freely in any direction on the plane, regardless of its orientation. It is typically composed of three or more specialized wheels (such as Mecanum or Swedish wheels) arranged in an equilateral triangle or other suitable configurations. This setup grants the robot holonomic mobility, enabling precise and unrestricted navigation, particularly useful in constrained environments.

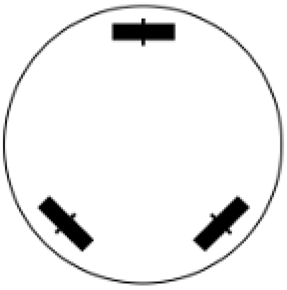


Figure.1.5 Omnidirectional robot

I.4.4.Aerial Robots :

Aerial robots, commonly known as Unmanned Aerial Vehicles (UAVs) or drones, are autonomous or remotely piloted systems designed for navigation in three-dimensional environments. These platforms have become indispensable in modern robotics due to their versatility, mobility, and adaptability across civilian and military applications.

1. **Sensing and Navigation:**
    - Equipped with advanced sensors (e.g., LiDAR, cameras, IMUs) for data acquisition, obstacle avoidance, and real-time positioning.
    - Rely on **GPS** and **computer vision** for autonomous path planning in dynamic environments.
2. **Applications:**

    - **Civilian:** Environmental monitoring, precision agriculture, infrastructure inspection, and search-and-rescue operations.
    - **Military:** Surveillance, reconnaissance, and payload delivery in high-risk zones.

3. **Technological Foundations:**
    - **Aerodynamics:** Ensures stable flight and energy efficiency.
    - **Control Systems:** PID, adaptive, or AI-based controllers for trajectory tracking and stability.
    - **Embedded Systems:** Onboard processors for real-time decision-making.
    - **Communication:** Robust data links for remote operation and swarm coordination.

Challenges and Advancements:

- **Autonomy:** Integration of **AI** (e.g., SLAM, deep learning) for fully autonomous missions.
- **Regulation:** Evolving policies for airspace safety and privacy concerns.
- **Energy Efficiency:** Development of hybrid propulsion systems and lightweight materials.

Aerial robotics represents a multidisciplinary field, merging mechanical engineering, control theory, and artificial intelligence to enhance mission reliability and operational scope [7]

Figure.1.6: Aerial robot

II.4.5. Surface and underwater Robots

Surface and underwater robotic systems, including Autonomous Surface Vehicles (ASVs), Autonomous Underwater Vehicles (AUVs), and Remotely Operated Vehicles (ROVs), represent critical advancements in aquatic automation. ASVs operate at the air-water interface for oceanographic monitoring, environmental surveillance, and maritime security, leveraging GPS navigation and multi-sensor fusion. AUVs conduct deep-sea exploration and infrastructure inspections with limited communication (acoustic modems) and pressure-resistant designs, while tethered ROVs enable high-precision tasks like offshore maintenance through real-time control. These systems integrate hydrodynamics, robust control strategies (e.g., PID for disturbance rejection), and advanced sensing (sonar, LiDAR) to address challenges such as energy constraints and turbid water navigation. Their interdisciplinary design—combining robotics, marine engineering, and AI—enhances safety and efficiency in hazardous or remote aquatic environments compared to traditional methods [8].

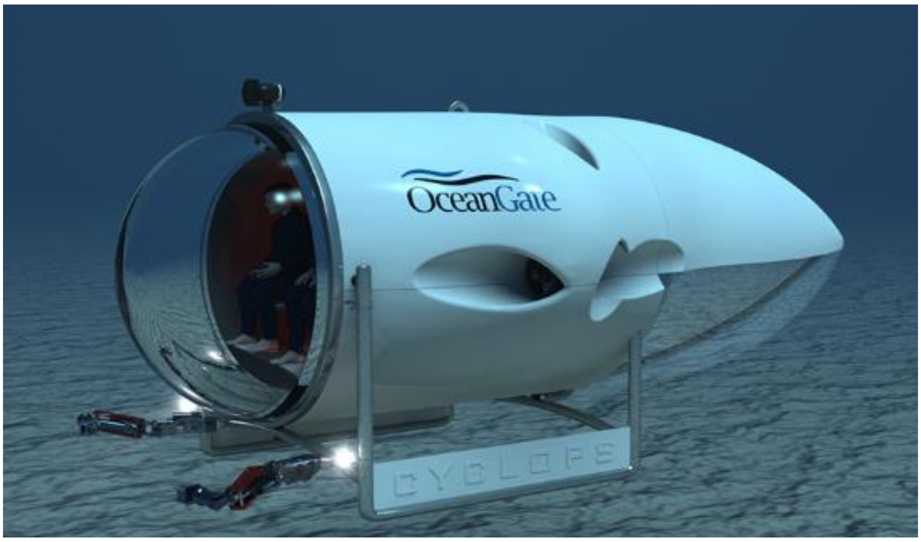


Figure.1.7: Underwater robot

II.4.6. Medical Robots

Medical robots integrate robotics and healthcare to enhance precision, reliability, and efficiency in clinical settings. These systems excel in minimally invasive surgery (e.g., the *da Vinci Surgical System*, which improves surgeon dexterity and patient outcomes), rehabilitation (e.g., robotic exoskeletons for motor recovery), and diagnostics. Their development combines biomechanics, control systems, biomedical engineering, and AI, ensuring real-time responsiveness and stringent safety standards. As the field advances, medical robotics promises to revolutionize healthcare delivery by increasing accessibility to high-quality treatments [9].

Figure.1.8 : Medical robot

I.5.Mechanical classification of robots navigation :

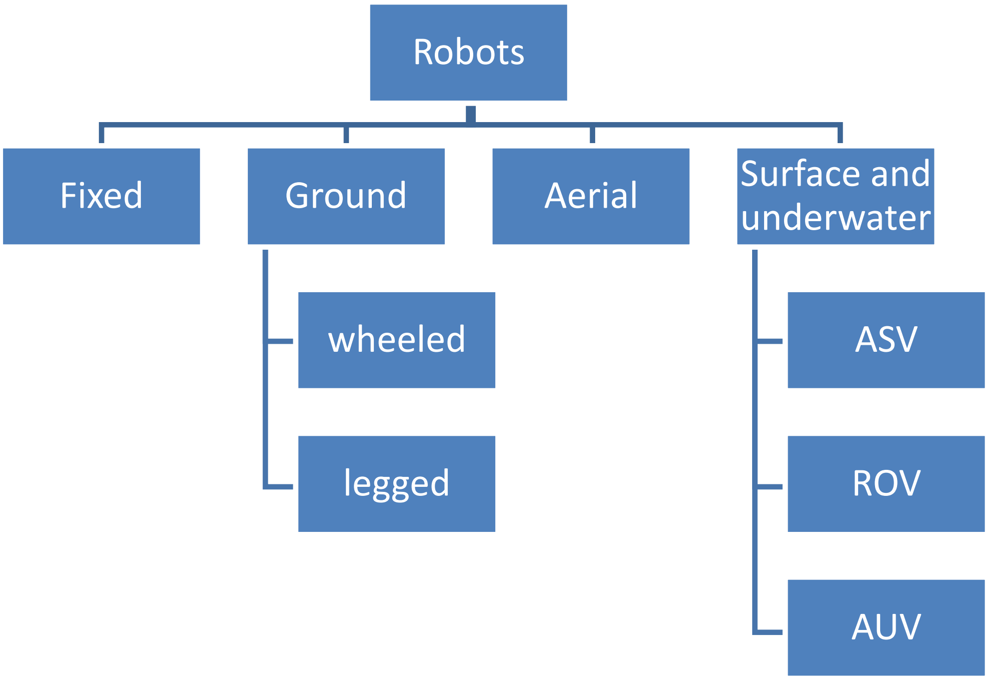


Figure.1.9. Mechanical classification of robots navigation

I.6. Conclusion :

This chapter has established a foundational framework for robotics by systematically examining its historical evolution, core definitions, and taxonomy. From early mechanical automata to contemporary AI-driven systems, robotics has progressed through innovations in engineering and computing. The classification of robots—spanning humanoid, industrial, mobile, aerial, aquatic, and medical systems—demonstrates their diverse applications and inherently multidisciplinary nature. By contextualizing mobile robotics within this broader landscape, the discussion sets the stage for focused exploration of unicycle-type robot modeling and control strategies in subsequent chapters. These insights are pivotal for addressing the technical challenges ahead.

Chapter II: PID Control of a Constrained Point in Unicycle Mobile Robots

chapter II: PID Control of a Constrained Point in Unicycle Mobile Robots

II.1.Introduction:

This Trajectory tracking for non-holonomic mobile robots is a key challenge in autonomous navigation. Unicycle-type robots are widely studied due to their simple kinematics and practical relevance. In many applications, the control objective is not the robot's center but a constrained point located in front—such as a sensor or tool.

This chapter focuses on applying a PID control law to track a desired trajectory by regulating the motion of this forward point. The control strategy uses polar coordinate errors to compute linear and angular velocities, ensuring the robot respects its non-holonomic constraints.

The method is simple, does not require dynamic linearization, and is proven stable through Lyapunov analysis. To enhance performance, PID gains are optimized using a Genetic Algorithm, making this approach efficient and robust for path tracking tasks.

[1–3] provide foundational insights into mobile robot modeling and control. This work extends the basic PID controller to track a circular path using feedback from a constrained point.

II.2. Kinematic Model

The unicycle model describes the robots motion as :

$$\dot{x} = v\cos\vartheta \qquad (1)$$

$$\dot{y} = v\sin\vartheta \qquad (2)$$

$$\dot{\vartheta} = \omega \qquad (3)$$

where (x, y) is the robot center position, θ is the orientation, v is the linear velocity, and ω is the angular velocity.

II.2.1. Non-Holonomic constraint

Non Unicycle robots are characterized by a non-holonomic constraint, which limits their ability to move sideways. This constraint arises from the differential-drive configuration, where the robot's motion is constrained to directions aligned with its heading. Mathematically, the constraint is expressed as:

$$\dot{y}\cos\theta - \dot{x}\sin\theta = 0 \qquad (4)$$

This condition implies that lateral velocity in the body frame must be zero, i.e., the robot cannot instantaneously change its position in the direction perpendicular to its current orientation.

To address this, we design a control law that indirectly respects the non-holonomic nature of the robot by:

- Defining a tracking point *P* located at a distance l_1 in front of the robot, such that:

$$x_p = x + l_1.\cos\theta \qquad (5)$$

$$y_p = y + l_1.\sin\theta \qquad (6)$$

- Controlling the robot's motion to ensure that this forward point *P* converges to a desired trajectory, rather than directly controlling the robot's center.
- Computing the velocity of point *P* using the transformation:

$$\begin{bmatrix}\dot{x}_p\\ \dot{y}_p\end{bmatrix} = \begin{bmatrix}\cos\theta & -l_1\sin\theta\\ \sin\theta & l_1\cos\theta\end{bmatrix}\begin{bmatrix}v\\ \omega\end{bmatrix} \qquad (7)$$

This mapping ensures that the control inputs (v, ω) are compatible with the non- holonomic structure of the unicycle model.

In this framework, we use a PID control law in polar coordinates between the con- strained point P and the reference trajectory. By regulating the distance ρ and angular error α of point P , the controller determines the linear and angular velocity inputs (*v, ω*) such that the robot respects its non-holonomic motion constraints while achieving accu- rate path tracking.

This approach effectively transforms the non-holonomic path-following problem into a setpoint regulation problem in error space, which is solvable using classical PID design. It avoids the need for dynamic feedback linearization or complex nonlinear controllers while still preserving the robot's motion feasibility.

II.3. Constrained tracking point

In many mobile robot applications, the control objective is not to navigate the robot's geometric center directly, but rather to control a point *P* located in front of the robot. This point may represent the location of a sensor (e.g., camera, LiDAR), a gripper, or simply a virtual point used to simplify path-following behavior.

Let *P* be defined at a distance l_1 ahead of the robot along its heading direction. Its position in global coordinates is:

$$x_p = x + l_1 . \cos\theta \qquad (8)$$

$$y_p = y + l_1 . \sin\theta \qquad (9)$$

Where (*x, y*) is the position of the robot's center and ϑ is its orientation.

Velocity Mapping via Jacobian Transformation

To control the motion of point P , we compute its velocity $(\dot{x}_p, \dot{y}_p)$ in terms of the robot's control inputs (*v, ω*):

$$\begin{bmatrix} \dot{x}_p \\ \dot{y}_p \end{bmatrix} = \begin{bmatrix} \cos\theta & -l_1 \sin\theta \\ \sin\theta & l_1 \cos\theta \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

The 2 × 2 matrix on the right-hand side is a Jacobian matrix that maps the robot's velocity inputs to the velocity of the constrained point P . This transformation arises by differentiating (x_p , y_p) with respect to time and applying the chain rule.

Why This Transformation is Necessary

The robot is underactuated and non-holonomic: it cannot move sideways, and the control inputs are limited to (*v, ω*). However, the desired trajectory is defined for point P , not for the robot's center.

To ensure that *P* follows the trajectory, we:

- Define the error between P and the reference trajectory.
- Use a PID controller to generate a desired velocity vector for point (${\dot{x}^{des}}_p$, ${\dot{y}^{des}}_p$)
- Use the inverse of the Jacobian to compute the control inputs (*v, ω*) that pro- duce these desired velocities.

This ensures that the robot moves in such a way that its constrained point P converges to the desired trajectory, while all movements remain physically feasible under the robot's kinematic constraints.

Interpretation of the Jacobian

The Jacobian :

$$J(\vartheta, l_1) = \begin{bmatrix} \cos\theta & -l_1 \sin\theta \\ \sin\theta & l_1 \cos\theta \end{bmatrix} \qquad (10)$$

can be interpreted as the coordinate transformation between the body velocities and the Cartesian velocity of point P . It embeds the geometric relationship between the robot's base and the forward tracking point, adjusting how the rotation ω contributes to linear motion at P .

This decoupling strategy enables classical feedback design (like PID) to be applied effectively to underactuated, non-holonomic systems.

II.4.Error definition

Let the tracking error be:

$$e_x = x_r - x_p \qquad (11)$$

$$e_y = y_r - y_p \qquad (12)$$

$$\rho = \sqrt{e^2{}_x + e^2{}_y} \qquad (13)$$

$$\alpha = arctan2(e_y, e_x) - \theta \qquad (14)$$

II.5.PID.Control.Law

In the proposed control approach, we regulate the motion of the unicycle robot such that a forward-constrained point *P* tracks a desired circular trajectory. The control is formulated directly in terms of two polar error variables:

- $\rho(t)$: the Euclidean distance between point *P* and the reference point on the circular path.
- $\alpha(t)$: the orientation error between the robot's heading and the line connecting point *P* to the reference.

These errors are computed in each iteration as :

$$\rho(t) = \sqrt{(x_r(t) - x_p(t))^2 + (y_r(t) - y_p(t))^2} \qquad (15)$$

$$\alpha(t) = arctan2(y_r(t) - y_p(t), x_r(t) - x_p(t)) \qquad (16)$$

The angle α is normalized to the range $[-\pi, \pi]$ to avoid discontinuities.

Controller Structure

We apply independent PID controllers to both ρ and α to compute the linear and angular velocity inputs of the unicycle robot:

$$v(t) = K^{\rho}{}_{p}\, \rho(t) + K^{\rho}{}_{i} \int_0^t \rho(\tau) d\tau + K^{\rho}{}_{d} \frac{d\rho}{dt} \qquad (17)$$

$$\omega(t) = K^{\alpha}{}_{p}\, \alpha(t) + K^{\alpha}{}_{i} \int_0^t \alpha(\tau) d\tau + K^{\alpha}{}_{d} \frac{d\alpha}{dt} \qquad (18)$$

This control structure is applied directly to the robot's motion commands (v, ω), without explicitly computing or inverting the Jacobian of the transformation from (v, ω) to the constrained point velocity.

Motivation and Effect of Each Term

- Proportional terms $(K^{\rho}{}_{p}, K^{\alpha}{}_{p})$Drive the robot forward and steer it toward the

reference by responding to the instantaneous tracking error. - $\rho(t)$ controls forward

progress toward the target. - $\alpha(t)$ aligns the heading to the direction of the target.

-Integral terms $(K^{\rho}{}_{i}, K^{\alpha}{}_{i})$: Help eliminate steady-state error in both distance and

orientation, especially if the robot reaches close to the trajectory but slightly off-center

or off-angle.

-Derivative terms $(K^{\rho}{}_{d}, K^{\alpha}{}_{d})$ Introduce damping by responding to the rate of error change, improving convergence speed and reducing overshoot.

Implementation Specifics from Simulation

- At each timestep, the robot updates its position using the unicycle model:

$$\dot{x} = v\cos\theta \ , \ \dot{y} = v\sin\theta \ , \dot{\theta} = \omega \qquad (19)$$

- The tracking point P is updated as:

$$x_p = x + l_1\cos\theta \ , \ y_p = y + l_1\sin\theta \qquad (20)$$

- The reference trajectory ($x_r(t)$, $y_r(t)$) is defined as a time-varying circular path.

- The control inputs (v, ω) are then applied to simulate the robot's next state.

Why This Works for Non-Holonomic Robots

Although the unicycle model is non-holonomic and cannot move sideways, this control strategy respects those constraints implicitly:

- The robot moves forward only via v and rotates via ω. - The PID feedback on ρ and α ensures that the robot turns and advances to bring point P closer to the reference without lateral motion .

- By tracking the forward point P rather than the center, the robot can converge more smoothly and effectively.

This makes the control law simple yet powerful, requiring no Jacobian inversion or dynamic model, while still achieving trajectory convergence for the constrained point.

II.6. Lyapunov Stability Analysis

To ensure that the proposed control law guarantees convergence of the constrained point P to the reference trajectory, we analyze the closed-loop system using Lyapunov theory .We define the candidate Lyapunov function :

$$V(\rho,\alpha) = \tfrac{1}{2}\rho^2 + \tfrac{1}{2}\theta^2 \qquad (21)$$

which is positive definite and radially unbounded in the error variables ρ and α. Taking its time derivative:

$$\dot{V} = \dot{\rho}\rho \ + \ \dot{\alpha}\alpha \qquad (22)$$

From the unicycle model and error kinematics, the derivatives of the errors are:

$$\dot{\rho} = -v\cos\theta \qquad (23)$$

$$\dot{\alpha} \ = \ \omega \ - \tfrac{v}{\rho}\sin\alpha \qquad (24)$$

Substituting the PID control laws for v and ω into the above (ignoring integral and derivative terms for initial analysis):

$$v = K_p\rho \qquad (25)$$

$$\omega = K^{\alpha}{}_{p}\alpha \qquad (26)$$

Now substitute into $\dot{V}$:

$$\dot{V} = -K^{\rho}{}_{p}\rho^2\cos\alpha + \alpha K^{\alpha}{}_{p}\alpha - \tfrac{K^{\rho}\rho}{\rho}\sin\alpha \qquad (27)$$

$$\dot{V} = -K^{\rho}{}_{p}\rho^2\cos\alpha + K^{\alpha}{}_{p}\alpha^2 - K^{\rho}{}_{p}\alpha\sin\alpha \qquad (28)$$

Assuming small α (near convergence), we linearize: $\cos\alpha \approx 1 \quad , \quad \sin\alpha = \alpha$

Then :

$$V = -K^{\rho}{}_{p}\rho^2 + (K^{\alpha}{}_{p} - K^{\rho}{}_{P})\alpha^2 \qquad (29)$$

This, the function $\dot{V} \ < \ 0$ is guaranteed when:

$$K^{\rho}{}_{p} > K^{\alpha}{}_{p} > 0 \qquad (30)$$

which ensures asymptotic stabilityof the tracking errors (ρ, α) to zero.

Note that the integral and derivative terms in the PID controller add robustness and improve convergence but are omitted here for clarity in the Lyapunov analysis.

II.7.PID Gain selection and tuning strategy

The choice of the PID gains (Kp, Ki, Kd) has a direct impact on the system's convergence rate, steady-state accuracy, and robustness to disturbances. Gain selection must respect the conditions required for Lyapunov stability, and can be guided by both theoretical and empirical considerations.

Theoritical Guidelines

From the Lyapunov analysis, we require: $K^{\rho}{}_{p} > K^{\alpha}{}_{p} > 0$

This ensures that the linear velocity command dominates the angular correction, avoiding instability due to excessive rotational gain.

Empirical Guidelines

Based on practical implementations in the literature and textbooks such as [1, 3], typical ranges are:

$K^{\rho}{}_{p} \in [1.0,4.0]$ $K^{\alpha}{}_{p} \in [0.5,2.5]$

$K^{\rho}{}_{i}, K^{\alpha}{}_{i} \in [0.005, 0.1]$ $K^{\rho}{}_{d}, K^{\alpha}{}_{d} \in [0.05, 1.0]$

These values are used in many simulation and real-world applications of differential- drive robots.

Simulation-Based Gain Optimization usin Genetic Algorithm

Rather than manually tuning the PID gains through trial-and-error, this work adopts a metaheuristic optimization strategy based on a Genetic Algorithm (GA). GA is well- suited for nonlinear and constrained search spaces and avoids local minima that can trap gradient-based methods.The cost function minimized by the GA evaluates the average tracking error of the constrained point P over a simulated trajectory. Specifically, the cost includes:

- The squared Euclidean distance between point P and the reference at each timestep:ρ^2.
- A penalty on heading error α^2 to ensure correct orientation. The objective function is:

$$J = \frac{1}{T}\int_0^T \rho(t)^2 + 0.1\,\alpha(t)^2 dt \qquad (31)$$

which was implemented numerically using forward Euler simulation of the unicycle model.

GA configuration :

-Search space : $[K^{\rho}{}_p \,, K^{\rho}{}_i \,, K^{\rho}{}_d \,, K^{\alpha}{}_p, K^{\alpha}{}_i \,, K^{\alpha}{}_d]$

-Bounds : $K_p \in [0.1\,,10]\,, K_i \in [0.001\,,0.5]\,, K_d \in [0.01\,,2]$

- Population size: 50

-Generations: 30

-Fitness function: Total tracking error on a figure-eight path over T = 60 *s*

-Progress: Visualized via live plot and progress bar per generation

Trajectory Used: To ensure rich excitation of both translational and rotational dynamics, a Figure-8 (lemniscate) path was used:

$$x_r(t) = A\,.\,sin\,\sin(\omega t) \quad , \quad y_r(t) = A\,.\sin(\omega t)\,\,.\cos(\omega t) \qquad (32)$$

With A=3m and $\omega = 0.2\,rad/s$

Final Gains obtained by Genetics Algorithm :

After convergence, the GA returned the following optimized gains that minimize the tracking cost:

$$K^{\rho}{}_{p} = 2.7439\ , \qquad K^{\rho}{}_{i} = 0.0402\ , \qquad K^{\rho}{}_{d} = 0.6646$$

$$K^{\alpha}{}_{p} = 9.5608\ , \qquad K^{\alpha}{}_{i} = 0.2317\ , \qquad K^{\alpha}{}_{d} = 0.1234$$

These gains resulted in stable and accurate tracking of the figure-8 path with minimal overshoot and fast convergence. The learned parameters respect the Lyapunov stability

requirement $K^{\rho}{}_{p} > K^{\alpha}{}_{p}$ and demonstrate a good balance between responsiveness (proportional gains), steady-state error elimination (integral gains), and damping (derivative gains)

II.8.Conclusion

This chapter presented a PID control strategy for unicycle-type robots that targets a constrained point located in front of the robot's center. The control law is implemented using polar error feedback and shown to be Lyapunov stable under certain gain conditions. Simulation-based gain optimization using a genetic algorithm validated the method's effectiveness for both circular and figure-8 trajectories.

Chapter III: Simulation Results and Performance Evaluation

Chapter III: Simulation Results and Performance Evaluation

III.1. Introduction

This chapter presents the simulation results obtained for the PID control of a unicycle mobile robot with a constrained forward tracking point. Two types of reference trajectories are tested: a circular path and a figure-eight path. The main objective is to assess the capability of the proposed controller in ensuring precise tracking under non- holonomic constraints and actuator saturation. The evaluation focuses on the behavior of the tracking point, the accuracy in the *x* and *y* directions, and the smoothness of the control signals generated in both cases.

III.2.Simulation Setup

All simulations are conducted in MATLAB. The robot is modeled kinematically as a unicycle with a constrained point P located at a fixed distance l_1 in front of its center. The PID controller regulates the motion of point P using polar coordinate errors between this point and the reference trajectory. The linear velocity is bounded by $vmax$ = 1.0 m/s, and the gains are tuned using a Genetic Algorithm as described in Chapter **??**. The controller is implemented in discrete time with a sampling interval of $\Delta t = 0.01$ s over a total simulation time of 75 seconds. The reference trajectories used in the simulations are a circle and a figure-eight, chosen to test the controller under both steady and varying curvature conditions.

III.3 Circular Trajectory Results

Figure 3.1 shows the result of the robot tracking a circular path of radius R = 5 meters. The actual trajectory of point P is compared with the reference trajectory. The robot succeeds in following the circular path with high precision after a short transient phase. The tracking error reduces quickly and remains negligible over time, which confirms the effectiveness of the PID controller in compensating the offset between the robot's center and the constrained point. The shape of the path is preserved and exhibits continuous curvature throughout the tracking duration.

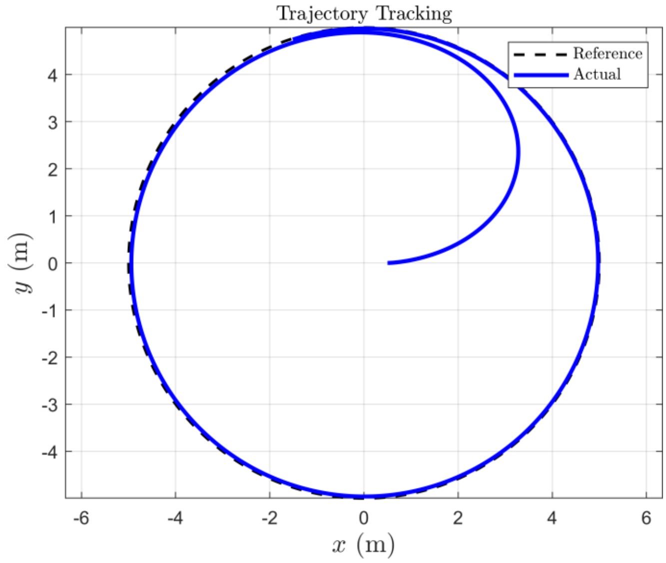


Figure 3.1: Reference vs. actual path of point P for a circular trajectory.

To better understand the performance in the Cartesian space, Figure 3.2 presents the time evolution of the xp and yp coordinates, compared to their respective reference values. Both components show accurate tracking behavior. The initial deviation is quickly corrected, and the signals match the reference closely for the remainder of the simulation. No oscillations or divergence are observed, indicating well-tuned derivative action that helps dampen the initial transient. The control loop ensures fast error attenuation in both directions while maintaining bounded responses.

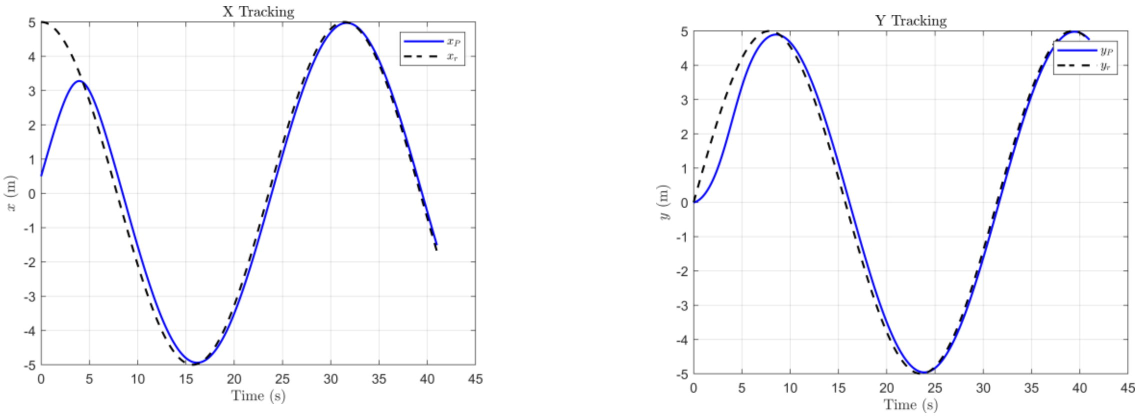


Figure 3.2: Actual vs. reference $xp\ (t)$ and $yp\ (t)$ for the circular trajectory.

The evolution of the control inputs used to follow the circular path is plotted in Figure 3.3 The linear velocity *v* reaches its upper bound briefly during the initial acceleration but quickly stabilizes. The angular velocity *ω* follows a periodic trend corresponding to the constant curvature of the circular path. Both control signals exhibit smooth profiles with no erratic variations, confirming the stability of the closed-loop system and the feasibility of execute constraints.

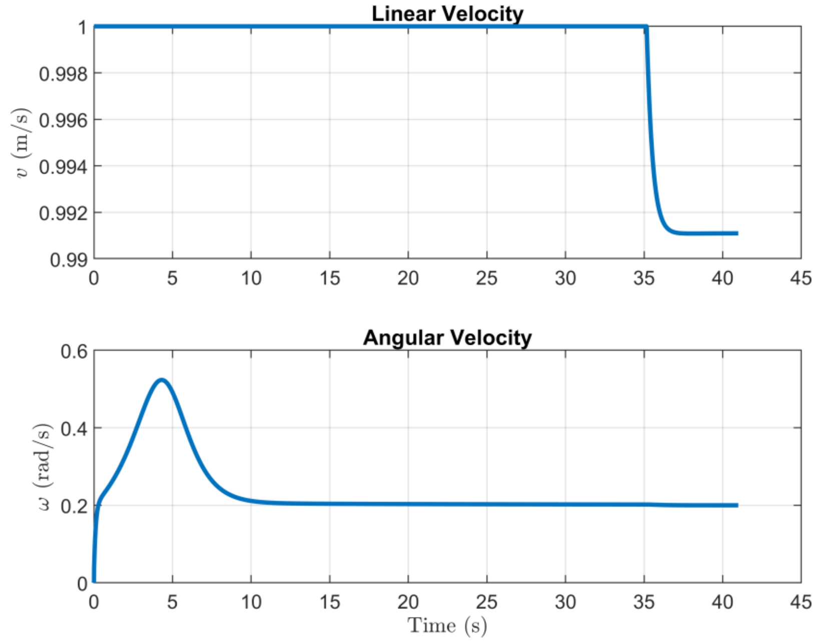


Figure 3.3: Linear velocity *v* and angular velocity *ω* for the circular path.

III.4 Figure (8) Trajectory Results

To evaluate the controller under more challenging conditions, a figure-eight trajectory is used. This path introduces varying curvature and sharp turns, particularly at the center crossing. Figure 3.4 illustrates the resulting path followed by point P compared to the reference. Despite the trajectory's complexity, the robot succeeds in maintaining close adherence throughout the simulation. The deviations near the sharpest turns are minor and temporary, and the path is rejoined smoothly. This result confirms that the controller generalizes well to time-varying and nonlinear reference geometries.

Figures 3.5 and 3.6 present the individual tracking results in $xp\ (t)\ and\ yp\ (t)$, respec- tively. The signals remain smooth and closely follow the reference in both axes. The x component exhibits more pronounced oscillations due to the sinusoidal nature of the figure-eight in the horizontal direction, while the y component shows larger amplitude variation around the center of the figure. In both plots, the error remains bounded and converges rapidly after any disturbance caused by the directional switching in the path. The corresponding control signals for the figure-eight tracking are plotted in Figure 3.7. Compared to the circular path, the control inputs show higher variations, especially in the angular velocity ω. This is expected, as the robot must execute sharp turns to reverse direction and maintain path convergence. Nevertheless, the control remains smooth and within acceptable limits. The linear velocity adapts to reduce speed during curvature peaks and increases during straight sections, demonstrating proper coordination between the two control inputs.

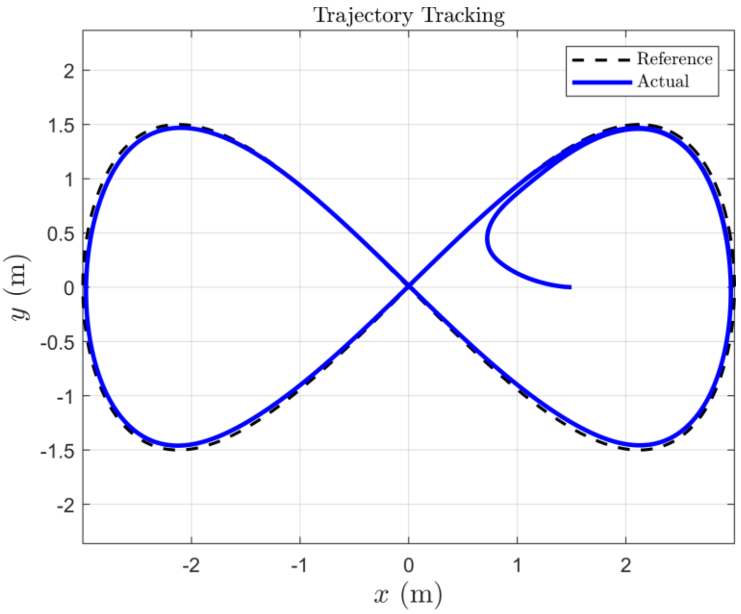


Figure3.4: Reference vs. actual path of point P for figure-8 trajectory.

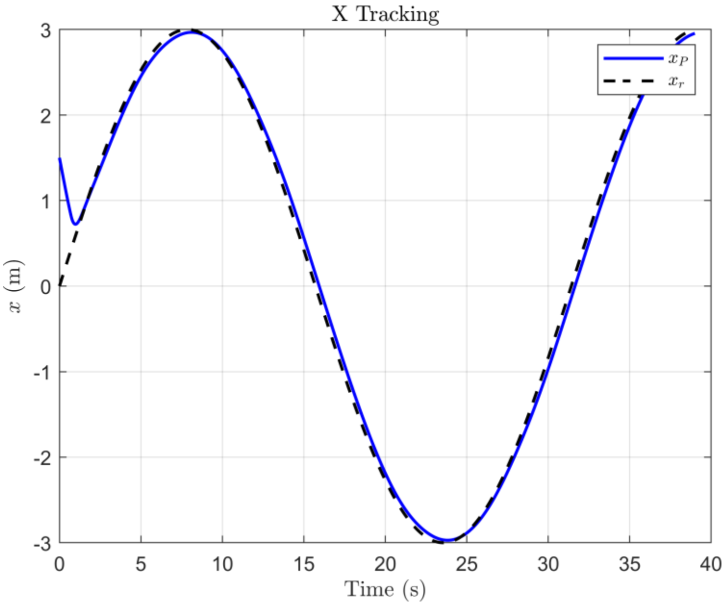


Figure 3.5: Tracking performance in $xp\ (t)$ for the figure-8 path.

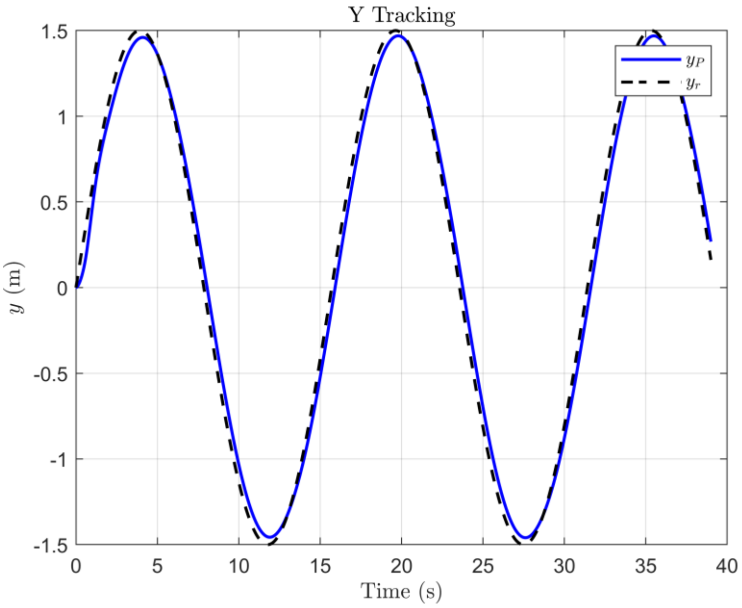


Figure 3.5: Tracking performance in $xp\ (t)$ for the figure-8 path.

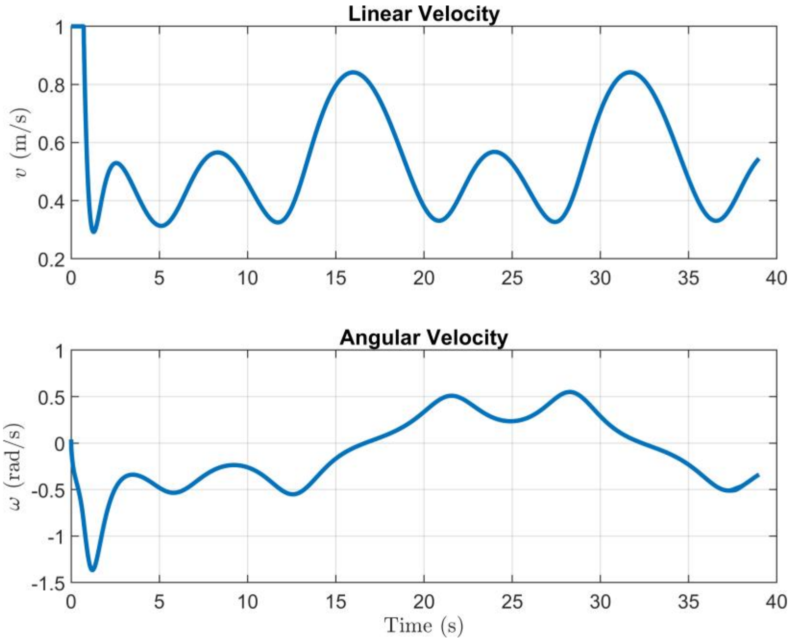


Figure 2.7: Linear and angular control signals for figure-8 trajectory.

III.5.Conclusion

This chapter presented a detailed simulation-based evaluation of a PID controller designed to regulate a constrained forward tracking point on a unicycle mobile robot. The results for both the circular and figure-eight trajectories demonstrate that the controller can accurately and smoothly track different types of paths while preserving system stability and respecting kinematic constraints. For the circular path, the system exhibits steady and uniform behavior with minimal deviation. For the figure-eight path, the controller handles high curvature and directional changes without instability or performance degra- dation. Control inputs remain within saturation bounds, and their profiles are smooth throughout the simulation. These results confirm the viability of classical PID techniques for non-holonomic path tracking when designed in polar coordinates and tuned appro- priately. The proposed approach provides a reliable foundation for further developments, including integration with dynamic models, adaptive gain tuning, or obstacle-aware nav- igation.

General Conclusion

General Conclusion :

The present work focused on the trajectory tracking control of unicycle-type mobile robots using classical control techniques, particularly the PID controller. The study was motivated by the challenges posed by the non-holonomic nature of such robots, which limits their motion and complicates trajectory planning and control.

The research was structured into several stages. It began with a general overview of robotics, followed by a detailed modeling of the unicycle robot's kinematics and the formulation of a control strategy targeting a constrained point located in front of the robot. This point represents a more realistic tracking objective in practical scenarios, such as sensor placement or task-specific tooling.

A Proportional-Integral-Derivative (PID) control law was developed in polar coordinates, allowing for intuitive regulation of the distance and orientation errors. The stability of the closed-loop system was analyzed using Lyapunov theory, establishing the conditions under which the proposed control strategy ensures convergence. Furthermore, to overcome the limitations of manual tuning, the PID gains were optimized using a Genetic Algorithm (GA), which provided a systematic and performance-driven approach to gain selection.

Simulation results were presented for two reference trajectories—a circular path and a figure-eight path. These scenarios allowed us to validate the robustness and flexibility of the controller under different curvature and motion profiles. The results confirmed that the proposed controller achieves high tracking accuracy, smooth control inputs, and system stability across varying conditions.

In conclusion, this work demonstrates that classical control methods, when carefully adapted and combined with optimization techniques, remain effective and relevant for solving complex control problems in mobile robotics. The approach provides a solid foundation for future extensions, including dynamic modeling, disturbance rejection, obstacle avoidance, and integration with modern AI-based planning algorithms.

Bibliography

Bibliography :

Chapter 1 :

[1] KHESRANI Samia , (Contribution à la modélisation et à la commande des systèmes robotiques).Sétif

[2] Spyros G. Tzafestas, “Introduction to mobile robot control, ” Elsevier edition, 2014.

[3] Benoit belebo mevo, "contribution a la commande adaptative et robuste d’un robot mobile de type unicycle avec modèle non-linéaire", Université du Québec en Abitibi Témiscamingue, Mai 2019

[4] Mr. MAZIOUA Toufik Mr.ZEGROUR Abdelhakim, “Etude et réalisation d’un robot a base mobile de type unicycle” Université A. MIRA-BEJAIA, 2020.

[5] Site web :www.Roboticbeast.com

[6] **Tagliavini, L., Colucci, G., Botta, A., Cavallone, P., Baglieri, L., & Quaglia, G. (2022).** Wheeled Mobile Robots: State of the Art Overview and Kinematic Comparison Among Three Omnidirectional Locomotion Strategies. Journal of Intelligent & Robotic Systems, 106(3), Article 57.

[7] S. Grzonka, D. Zhou, and A. Zell, “Aerial Robotics,” ResearchGate, 2011.

[8] Wikipédia. (2024). *Robot sous-marin autonome*. Wikipédia

[9] BPI France. (n.d.). *Innovations en robotique médicale : révolutionner les soins de santé*. Big média – BPI France.

Chapter 2 and Chapter 3:

[1] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, Introduction to Autonomous Mobile Robots. MIT Press, 2nd ed., 2011

[2] A. Kelly, Mobile Robotics: Mathematics, Models, and Methods. Cambridge University Press, 2013

[3] C. Canudas-de Wit, B. Siciliano, and G. Bastin, Theory of Robot Control. Springer,1996