الجمهورية الجزائرية الديمقراطية الشعبية
People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research
المدرسة الوطنية العليا للتكنولوجيات المتقدمة
National Higher School of Advanced Technologies

Department of Industrial Engineering and Maintenance

Final Year Project to Obtain the Diploma of Engineering

-Field -
Electromechanics

-Speciality-
Mechatronics

- Subject -

MOBILE ROBOT CONTROL DESIGN BASED ON FUZZY LOGIC

Realized by:
SALHI Youcef

Members of The Jury:

MCA ENSTA	MEDDOUR Ikhlas	President
MAA ENSTA	BOUSSOUFA Ahmed	Examiner
PROFESSOR ENSTA	HAOUARI Fouad	Supervisor

Algiers, the 01/07/2024

Academic year **2023–2024**

Dedicated to

This thesis is dedicated to my parents, whose unwavering support, love, and encouragement have been the inspiration of all my achievements. Your sacrifices and guidance have been my driving force.

To my siblings, for always believing in me and cheering me on, no matter the distance.

And to my mentor, Dr. HAOUARI Fouad, for your invaluable guidance and inspiration.

This work is also dedicated to all aspiring researchers in the field of robotics and artificial intelligence. May you continue to push the boundaries of innovation and discovery.

SALHI Youcef.

Acknowledgments

I would like to express my deepest gratitude to my advisor, Professor HAOUARI Fouad, whose guidance, support, and encouragement have been invaluable throughout my research.

Special thanks to my family and friends for their unwavering support and understanding during the course of my study.

My sincere appreciation goes to my colleagues and friends, particularly [Names], for their companionship, intellectual discussions, and moral support throughout this journey. Your camaraderie made the challenging moments more manageable.

Summary

This thesis focuses on the study of mobile wheeled robots, more specifically on wheeled differential mobile robot(WDMR). this work focuses on elaborating a fuzzy logic controller for the mobile robot, which can be applied for target reach and obstacle avoidance. In the tests, the behavior of the robot was simulated in two(2) steps, the first test was done without taking in consideration the disturbances and uncertainties. For the second test, the disturbances and uncertainties were included. The task of detecting obstacles is done by sensors which provides the input of the fuzzy logic controller(FLC). This controller generates outputs to command the left and right wheels of the robot. The simulations are done using Matlab.

Key Words : Mobile robots, fuzzy logic command, obstacle avoidance, arget reach

---

Résumé

Cette thèse se concentre sur l'étude des robots mobiles à roues, plus précisément sur le robot mobile différentiel à roues (WDMR). Ce travail se concentre sur l'élaboration d'un contrôleur basé sur la logique floue pour le robot mobile, qui peut être appliqué pour atteindre des cibles et éviter les obstacles. Lors des tests, le comportement du robot a été simulé en deux étapes : le premier test a été réalisé sans prendre en compte les perturbations et les incertitudes. Pour le second test, les perturbations et les incertitudes ont été incluses. La tâche de détection des obstacles est effectuée par des capteurs qui fournissent les entrées du contrôleur logique flou (FLC). Ce contrôleur génère des sorties pour commander les roues gauche et droite du robot. Les simulations sont effectuées en utilisant Matlab.

Mots clés : Robots mobiles, commande par logique floue, évitement d'obstacles, atteinte de la cible.

ملخص

تركز هذه الأطروحة على دراسة الروبوتات المتنقلة ذات العجلات، وبالتحديد على الروبوت المتنقل التفاضلي ذو العجلات . يركز هذا العمل على تطوير جهاز تحكم يعتمد على المنطق الضبابي للروبوت المتنقل، والذي يمكن تطبيقه للوصول إلى الهدف وتجنب العقبات. في الاختبارات، تم محاكاة سلوك الروبوت في خطوتين: تم إجراء الاختبار الأول دون أخذ الاضطرابات والشكوك في الاعتبار. في الاختبار الثاني، تم تضمين الاضطرابات والشكوك. تتم مهمة اكتشاف العقبات بواسطة المستشعرات التي توفر المدخلات لجهاز التحكم بالمنطق الضبابي . يقوم هذا الجهاز بتوليد مخرجات للتحكم في العجلات اليسرى واليمنى للروبوت. تم إجراء المحاكاة باستخدام ﭙتلب

الكلمات المفتاحية

الروبوتات المتنقلة، الأمر بالمنطق الضبابي، تجنب العوائق، الوصول إلى الهدف.

List of Figures

List of Tables

Contents

Acronyms

L : Distance between the two wheels

r : Radius of the wheels

V_R: Linear speed of the right wheel

V_L : Linear speed of the left wheel

V : Linear speed of the robot

ω : Angular speed of the robot around its Center of Instantaneous Rotation (CIR)

x, y : Coordinates of the robot

P : Position on the robot

F_X : Force applied on the x-axis

F_Y: Force applied on the y-axis

m : Mass of the robot

F_r: Force applied on the right wheel

F_l : Force applied on the left wheel

τ_r : Torque applied on the right wheel

τ_l : Torque applied on the left wheel

V(t) : Applied voltage

i(t) : Armature current

k_e : Back EMF constant

$\mu_A(x)$: Membership function

T(x) : The set of linguistic values of x

Dist : Distance to goal

θ_E: Orientation error

U : The universe of discourse associated with the base value

Global Introduction

0.1 Context and Problem

Beyond traditional industrial applications, robots are increasingly present in our daily lives, with significant areas of application such as medicine, agriculture, security, and home assistance. These robots are also becoming more mobile, capable of operating in aerial, maritime, and terrestrial environments. Terrestrial mobile robotics holds a historically important place.

The term "mobile robot" typically denotes a vehicle equipped with perception, decision-making, and action capabilities, enabling autonomous or semi-autonomous operation within various environments. They are capable of executing programmed tasks without human intervention or with minimal human involvement.

Notably, wheeled mobile robots, which use a particularly efficient rolling locomotion, are already used in various fields such as industrial logistics, agriculture with the automation of tractors, space exploration, planetary exploration, security tasks like area surveillance, and missions to search for and rescue victims in the event of natural or industrial disasters.

Here are two main operating modes for a mobile robot: tele-operated and autonomous. In tele-operated mode, a person controls the robot remotely. Commands are transmitted to the robot via a control interface, such as a joystick or keyboard, and are sent through a communication link, such as the internet or a satellite. Thus, the robot must execute the operator's commands, who perceives the environment around the robot through various means such as cameras or radars, in order to provide instructions tailored. In contrast, in autonomous mode, the robot must make its own decisions. This means it must be able to perceive its environment accurately and know how to react accordingly, depending on the level of autonomy. It is up to the robot to plan its path and determine the movements necessary to reach its goal. Mobile robots attracted a lot of attention for research and development, many search papers were published about the modeling and developing control systems of mobile robots [7], [17], [19].

Mobile robots face many challenges standing against reliable and efficient operation. This challenges reaches a variety of fields including hardware, software and environmental interaction. some of them are: Navigation and Path Planning, Sensor Integration and Accuracy, Control Systems

Control system are crucial for the operation of the mobile robot, it is challenging to design a control algorithm assures navigation throw dynamic environments, avoid obstacles, and that the robot perform tasks efficiently [18], [27].

Sensor Integration in mobile robot represent a way to interact with the surrounding environment, and this pose a challenge in assuring an accurate and reliable operation

0.2 Objectives and Contribution

This thesis has purpose of building a mathematical model for a mobile robot and build a fuzzy logic controller which empower the robot to avoid obstacle under the absence and presence of disturbances and uncertainties.

0.3 Thesis Plan

In this thesis, the work is divided in three(3) chapters. the first chapter is an overview on mobile robotics, a second chapter dedicated on fuzzy logic and the design of a robot controller based on it, the third and final chapter will be about simulating a mobile robot by a fuzzy logic controller and discuss the results.

Chapter 1

OVERVIEW ON MOBILE ROBOTS

1.1 Introduction

Mobile robots are autonomous or semi-autonomous machines capable of moving through and interacting with their environment. These systems are studied because of their relative simplicity compared to other types of mobile robots,They are extensively utilized for researching autonomous systems. This chapter provides the major information about mobile robots, including their types, localization methods applied in robotics and mathematical modeling of a robot.

1.2 Concept of holonomy

Holonomy, in the context of mobile robots, refers to the ability of a robot to move in all directions instantaneously and independently, without the need to change its orientation. A robot is said to be holonomic if it can move omnidirectionally, meaning in all possible directions and rotations, without any movement restrictions .

This property is crucial for the maneuverability and flexibility of mobile robots, as it allows a holonomic robot to move efficiently in complex and cluttered environments without needing to perform additional maneuvers to adjust its orientation. Omnidirectional robots are an example of holonomic robots, as they can move instantaneously in all directions without requiring prior orientation changes [13] [21] [16].

1.3 Concept of non holonomy

In the context of mobile robots, refers to the condition where a robot cannot move instantaneously in all directions and rotations without changing its orientation. Unlike holonomic robots, which have omnidirectional movement capabilities, non-holonomic robots are restricted in their ability to maneuver and may require specific movements or adjustments to change direction.

Non-holonomic robots often have constraints on their movement due to their mechanical design or control systems. For example, a car-like robot with differential drive may need to perform a series of forward and backward motions combined with turning actions to change its direction effectively. This lack of instant omnidirectional movement can pose challenges in navigating complex environments but is often compensated for with careful planning and control strategies [13], [21], [16].

1.4 Classes of Wheeled Mobile Robots:

To move a mobile robot on a surface, at least two degrees of freedom are required, which means at least two motors. The combination of the choice of wheels and their arrangement also gives a robot its specific mode of locomotion [12], [18].

1. **Uni-Cycle Robot**: This robot is driven by two independent wheels and may have a number of caster wheels to ensure stability. The two main wheels provide the necessary movement, while the caster wheels support balance and maneuverability.

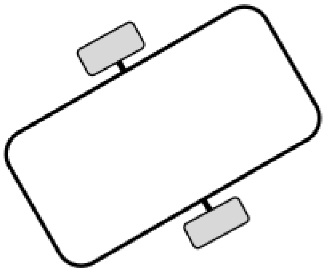


Figure 1.1: unicycle mobile robot

2. **Tricycle Robot**: Composed of two fixed wheels on the same axis and a centrally located steerable wheel positioned along the robot's longitudinal axis. The robot moves through two actions: longitudinal speed and the orientation of the steerable wheel. This setup allows for straightforward movement similar to that of a tricycle or car.

3. **Omnidirectional Robot**: A mobile robot is considered omnidirectional if it can independently control its velocities along the x and y axes and its rotational velocity around the z axis.

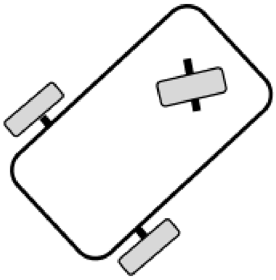


Figure 1.2: tricycled mobile robot

From a kinematic perspective, this is not possible with fixed or centrally steerable wheels. However, an omnidirectional robot can be achieved using a set of three offset steerable wheels or three Swedish wheels arranged at the vertices of an equilateral triangle. This configuration allows for movement in any direction without needing to reorient the robot.

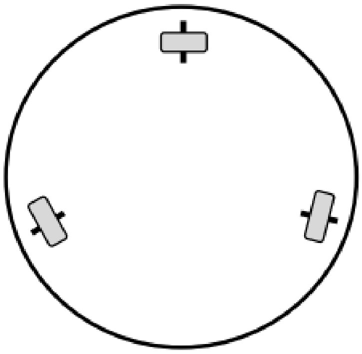


Figure 1.3: Omnidirection mobile robot

4. **car-type mobile robot**: A car-type mobile robot is similar to a tricycle. It consists of two fixed wheels placed on the same axle and two centered, steerable wheels also placed on the same axle. However, the car-type mobile robot is more stable since it has an additional point of support. All other properties of the car robot are identical to the tricycle robot. The tricycle robot can be reduced to the car robot by replacing the two front wheels with a single wheel placed at the center of the axle, while leaving the center of rotation unchanged.

In summary, the car-type mobile robot is a variation of the tricycle robot design that has an additional wheel for increased stability, but maintains the same core functionality.

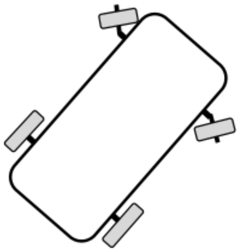


Figure 1.4: car-type mobile robot

1.5 Comparaison of the different types of wheelded mobile robots

robot types	**Advantages**	**Inconvenients**
Unicycle	stable self-rotation week mecanical complexity	non-holonome
Tricycle	medium mecanical complexity	non holonome low stability no self rotation
Omnidirectionnel	holonome stableself rotation	important mecanical complexity
vehicule-type	stable medium mecanical complexity	non holonome not self-rotating

Table 1.1: The different types of wheelded mobile robots

1.6 Localization in Mobile Robotics

Regardless of the robot's type, its crucial to know its location and position over time to know where it's heading. To know a robot localisation we need three (3) coordinates; two Cartesian coordinates for the position and an angle for the orientation. In a more formal manner, the task of localization involves calculating the transformation that maps the coordinate frame attached to the robot to a coordinate frame attached to the environment.

The localization system encompasses the sensors and data processing software that the vehicle uses to autonomously estimate its movement or position in space [26]. In which we can obtain two different sets of data:

- Relative Localization: Allows the vehicle to navigate by dead reckoning, using only the measurements of its own movements provided by proprioceptive sensors.
- Absolute Localization: Utilizes measurements from exteroceptive sensors to estimate the vehicle's position in a coordinate frame attached to the environment.

Sensors are fundamental elemnts for the localizations systm, there for we can classify them into two (2) types:

- **proprioceptive sensors**
- **exteroceptive sensors**

1.6.1 Exteroceptive sensors:

They are essential for helping robots understand and interact with their surroundings. These sensors gather information about the environment, recognize objects, model the surroundings, and detect interactions like position and force. Here are some common types of exteroceptive sensors:

- Cameras: Think of cameras as the robot's eyes. They capture images and videos, allowing the robot to see and identify objects, helping it navigate by visual cues just like we do.

- LIDAR (Light Detection and Ranging): LIDAR works like a super-precise radar. It sends out laser beams and measures how long they take to bounce back, creating a detailed 3D map of the surroundings. This helps the robot understand the shape and distance of objects around it.
- Ultrasonic Sensors: These sensors are like a bat's echolocation system. They emit sound waves that bounce off nearby objects, helping the robot detect obstacles and avoid bumping into things.
- Infrared Sensors: Infrared sensors use light to measure distances and detect objects close to the robot. They're great for spotting things nearby and are often used to prevent collisions and identify edges.

Each sensor type has its own pros and cons. The right sensor depends on what the robot needs to do and where it operates. For instance, while GPS is fantastic for outdoor navigation, it's not useful indoors. On the other hand, ultrasonic sensors are excellent for avoiding close-up obstacles but aren't suitable for mapping larger areas. By choosing the right mix of sensors, robots can better navigate and interact with their environments autonomously and efficiently.
In figure 1.5 illustrate below we can see the most common sensors and their characteristics

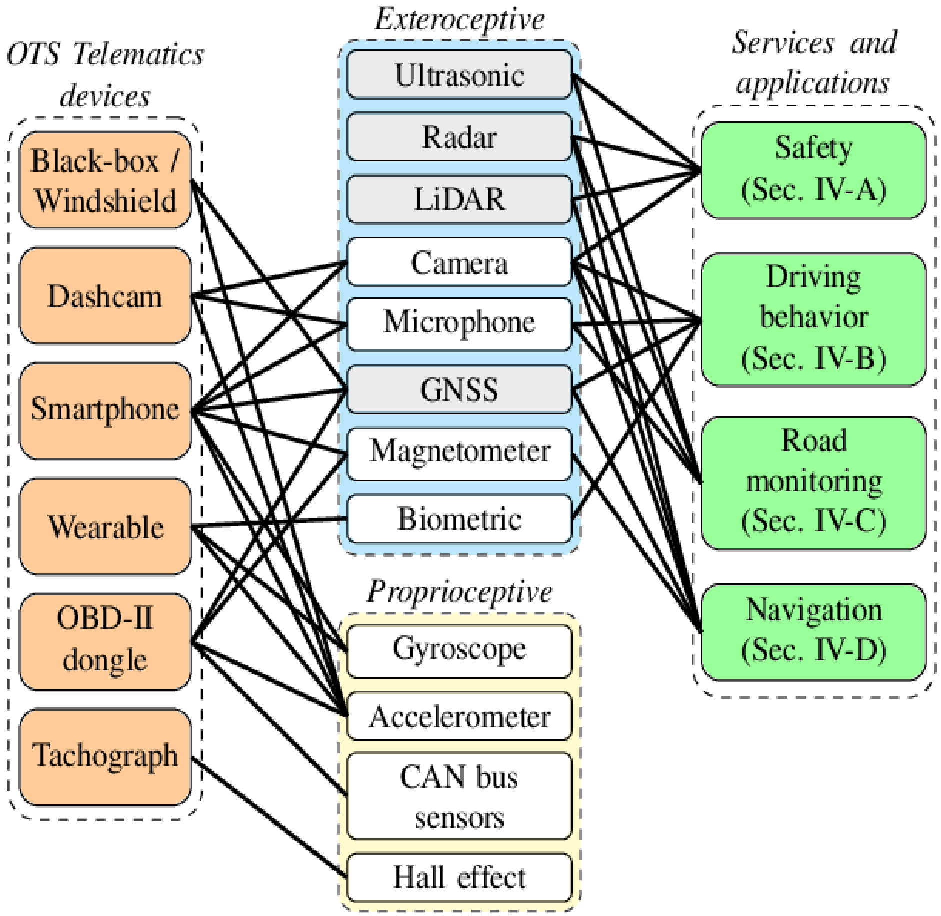


Figure 1.5: Exteroceptive Sensors

1.6.2 Proprioceptive sensor:

They are like the robot's internal senses, giving it feedback about its own state. For a mobile robot, these sensors can include those that measure its speed and acceleration, helping it understand how fast it's moving and how quickly it's changing direction. for exemple *the odometry sensor* which tracks the robot's position by measuring the rotation of its wheels or other movement mechanisms. This helps the robot keep track of how far it has traveled and its current orientation.

1.6.2.1 odometry

It's one of the most popular methods for relative localization in wheeled robots. It works by incrementally calculating the robot's position based on the speed and movement of its wheels. However, odometry has its challenges. It requires precise knowledge of the robot's geometry, such

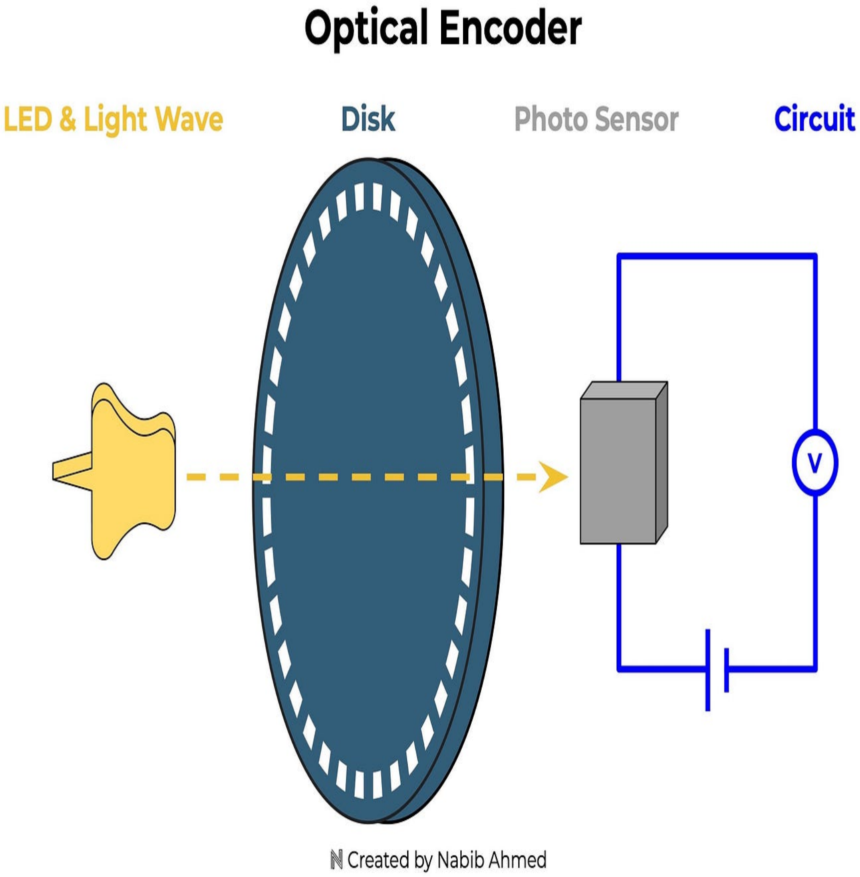


Figure 1.6: Wheel Odometry Model for Differential Drive Robot

as the exact diameter of the wheels, the distance between the wheels and the contact points with the ground. These parameters are often difficult to measure accurately and can vary depending on the type of surface the robot is moving on [2].

Moreover, issues like wheel slippage can drastically affect the accuracy of the position estimate, and these problems are usually hard to detect directly. On certain terrains, these factors can make odometry almost useless as a localization method. Despite these challenges, odometry is used to determine the position (x, y) and orientation (θ) of a mobile robot navigating on a flat surface, relative to its initial starting position. This technique relies on integrating the incremental movements of the wheels, measured by incremental encoders. By knowing the wheel radius (r) and the number of pulses (n) from the encoder with a resolution of (α) during a specific time interval

(Δt), we can calculate the distance (Δd) the wheel has traveled:

$$\Delta d = r \cdot n \cdot \alpha$$

So odometry helps the robot figure out how far it has moved and in which direction by keeping track of how much the wheels have turned. Figure 1.6 is a simple illustration of odometry.

1.7 wheelded differentiel mobile robot

A wheeled differential mobile robot is a type of robot that moves on wheels using a differential drive mechanism for motion control.

In this type of robot, usually, two or more wheels are situated from end to end on one side and from end to end on the opposite side of the robot body, and each wheel can be independently driven to move at a different speed and direction. The robot can turn, move forward, or backward easily by simply changing the velocity of the wheels on each side of the body [3], [5], [12], [24].

The differential drive allows relatively straightforward control as well as good maneuverability, which has made it a popular choice for many wheeled mobile robots, including robotic platforms used for research and education, and in exploration, surveillance, and logistics applications.

Now, before proceeding further, the following assumptions are made:

- The system under consideration is rigid (i.e., the body does not deform).
- The mass of the mobile body is constant
- The study take place on the XY plan(takes place in a two-dimensional space defined by the XY coordinates)
- The friction between the wheels and the surface is ignored
- The wheels do not slide

1.7.1 The Instantaneous Center of Rotation (ICR)

Refers to the point around which the robot is momentarily rotating. This concept is important to understand and control the robot's movement, it is well illustrated in figure 1.7.

The location of the ICR can be found using the wheel velocities and the distance between the wheels. let's take v_r and v_l as the velocities of the right and left wheels respectively, and L is the distance between the wheels, the ICR coordinates are calculated as follows:

$$ICR_x = \frac{L}{2} \cdot \frac{v_r + v_l}{v_r - v_l}$$

1.8 Kinematic Modeling

Kinematics is the fundamental study of the behavior of mechanical systems. In mobile robotics, it allows to comprehend and predict how a robot will move, enabling the design of robots that can perform specific tasks effectively and the development of precise control strategies [22], [15].

We denote $R = \left(\vec{O}, \vec{X}, \vec{Y}, \vec{Z}\right)$ as an arbitrary fixed frame, where the $\vec{Z}$ axis is vertical, and $R^{'} = \left(\vec{O'}, \vec{X'}, \vec{Y'}, \vec{Z'}\right)$ as a mobile frame attached to the robot The point $O^{'}$ is the center of the axis of the drive wheels.

The position or often referred to as the posture of the robot is defined as the vector:

$$P = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{1.1}$$

Where x and y are respectively the abscissa and ordinate of point $O^{'}$ in $\Re$, and θ is the angle $(\vec{x}, \vec{x^{'}})$. The posture of the robot is thus defined in a space M of dimension $m = 3$. The configuration of a mechanical system is known when the position of all its points in a given frame is known.

The linear velocity of each driving wheel is determined from the average of the angular velocities of both wheels [28]:

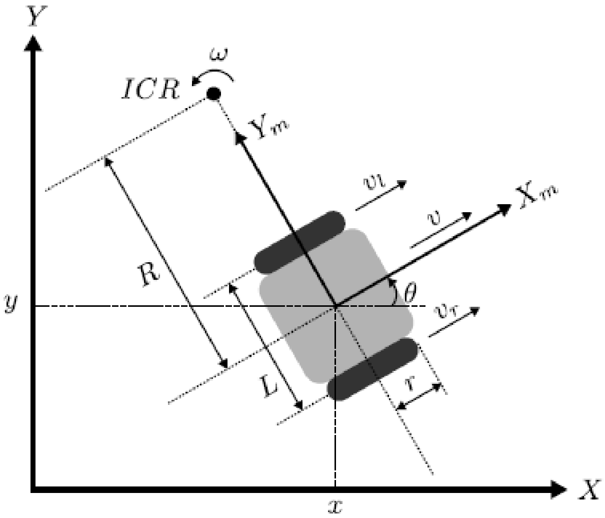


Figure 1.7: Differentiel Mobile Robot

$$V = \frac{V_r + V_l}{2} \tag{1.2}$$

The linear velocities in the x and y directions are given by:

$$\dot{x} = \frac{dx}{dt} = V\cos(\theta) \tag{2}$$

$$\dot{y} = \frac{dy}{dt} = V\sin(\theta) \tag{3}$$

The angular velocity θ of the Differential Wheeled Mobile Robot (DWMR) is given by:

$$\dot{\theta} = \frac{d\theta}{dt} = \omega = \frac{V_r - V_l}{L} \tag{4}$$

Substituting the linear velocity V into Equations (2) and (3) yields:

$$\dot{x} = \frac{V_r + V_l}{2}\cos(\theta) \tag{5}$$

$$\dot{y} = \frac{V_r + V_l}{2}\sin(\theta) \tag{6}$$

The individual velocities of the right and left wheels can be calculated using:

$$V_r = V + \frac{L}{2}\omega \tag{7}$$

$$V_l = V - \frac{L}{2}\omega \tag{1.3}$$

we can set the kinematic equation model by:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{bmatrix} \cos(\theta) & \cos(\theta) \\ \sin(\theta) & \sin(\theta) \\ \frac{1}{L} & -\frac{1}{L} \end{bmatrix} \begin{pmatrix} V_l \\ V_r \end{pmatrix} \tag{1.4}$$

1.9 Dynamic Modeling

The dynamic modeling of a wheeled differential mobile robot involves describing the forces and torques that cause the robot's motion, in addition to the kinematic relationships. This includes the robot's mass, inertia, and the effects of friction and other forces acting on the robot.

The relevant equations for the dynamic modeling are as follows:

Equations of Motion:

Translational Dynamics:

$$\sum F_x = m\ddot{x} \tag{1.5}$$

$$\sum F_y = m\ddot{y} \tag{1.6}$$

Rotational Dynamics:

$$\sum \tau = I\ddot{\theta} \tag{1.7}$$

Wheel Dynamics:

Wheel Torque:

$$F_r = \frac{\tau_r}{r} \tag{1.8}$$

$$F_l = \frac{\tau_l}{r} \tag{1.9}$$

Where τ_r and τ_l are the torques applied to the right and left wheels, respectively, and r is the radius of the wheels.

The resultant forces in the x and y directions and the resultant torque about the center of mass are:

$$F_x = (F_r + F_l)\cos(\theta) \tag{1.10}$$

$$F_y = (F_r + F_l)\sin(\theta) \tag{1.11}$$

$$\tau = (F_r - F_l)\frac{L}{2} \tag{1.12}$$

Where L is the distance between the two wheels (wheelbase).

1.9.1 DC motor

A Differential wheeled Mobile Robot (DWMB) uses two(2) independent driven wheels placed on the both sides of the robot, and the control of the entire system can be reduced to the control of the motors (in this case DC Motors) [25]. Figure 1.8 provides a better understanding on the mechanical and electrical characteristic of the DC motor

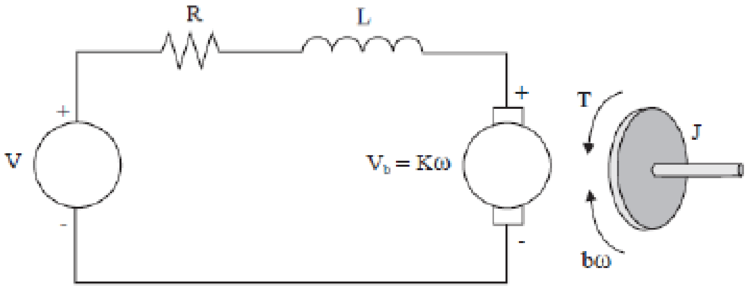


Figure 1.8: DC- Motor representation

electrical characteristic: The electrical dynamics of the motor are given by:

$$V(t) = L\frac{di(t)}{dt} + Ri(t) + E(t) \tag{1.13}$$

where:

- $V(t)$: Applied voltage
- $i(t)$: Armature current

- R: Armature resistance
- L: Armature inductance
- $E(t)$: Back EMF
- $E(t) = k_e\omega(t)$
- k_e: Back EMF constant
- $\omega(t)$: Angular velocity

Mechanical characteristic: The mechanical dynamics of the motor are given by:

$$J\frac{d\omega(t)}{dt} + B\omega(t) = T(t) - T_L(t) \tag{1.14}$$

where:

- J: Moment of inertia
- B: Damping coefficient (friction)
- $T(t)$: Torque produced by the motor
- $T(t) = k_t i(t)$
- k_t: Torque constant
- $T_L(t)$: Load torque

the combinaison of both the mechanical and electrical parts will give us the following model:

$$V(t) = L\frac{di(t)}{dt} + Ri(t) + k_e\omega(t) \tag{1.15}$$

$$J\frac{d\omega(t)}{dt} + B\omega(t) = k_t i(t) - T_L(t) \tag{1.16}$$

1.10 Motion generation

Motion generation is the process of deciding and acting on a sequence of movements for a mobile robot that will take it from a current position to a goal location without hitting obstacles. we can break this process to the following steps:

- path planning
- trajectoary generation
- Motion Control
- Sensor Integration

1.10.1 Path planning:

That means it involves the generation of a feasible and efficient path from the robot's current location to the goal location while avoiding the obstacles placed in its way [20], [14]. In general, it's devided into two types: Global Path Planning and Local Path Planning.

1.10.1.1 Global path planning

Global path planning considers the entire environment and aims to come up with a collision-free path from the start to the goal position. There are many algorithm used for this purpose in which we mention

- Probabilistic Roadmaps (PRM): building a map of the available space through random sampling and linking nearby points. Once the map is established, it connects the starting and goal points to this map and searches for the shortest path between them.
- A* (A-Star) Algorithm: calculates the shortest path from the start point to the desired point with the use of heuristic estimation
- Dijkstra's Algorithm: calculates the shortest path from the start point to the desired point without the use of heuristic estimation

- Rapidly-exploring Random Trees (RRT): the algorithm creates a roadmap of the free space by randomly sampling points within the configuration space and connecting nearby points. In the second phase, the query phase, the algorithm connects the start and goal points to this roadmap and then searches for the shortest path between them.

1.10.1.2 Local Path Planning

On the other hand, has a specific focus on real-time obstacle avoidance and short-term navigation, where considerations are made regarding the immediate environment of the robot. It supplements global path planning, providing a way of making adjustments to a pre-planned path on the fly concerning sensor feedback. some of the famous algorithms used for this purpose are :

- Dynamic Window Approach (DWA)
- Vector Field Histogram (VFH)
- Artificial Potential Fields (APF)

There are also some hybrid algorithms combining both of the previous types such as the Hybrid A*/Dijkstra algorithm

1.10.2 Trajectory Generation

Transform the way-point of the path into a trajectory sequence of movements as a function of time as seen in figure 1.9.

several methods are used for the purpose of generating the smoothest movement from one position to another, in [6] the following methods were used:

- Polynomial Trajectory Generation: with the use of polynomial functions its possible to create the path of a mobile robot
- Spline-Based Methods: series of polynomial segments form polynomial functions to generate trajectories
- Time-Optimal Trajectory Generation:create paths that minimize the time to move from a start to a goal position.

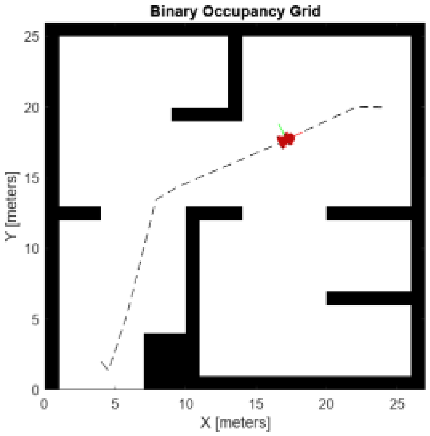


Figure 1.9: Path Generation Example

1.10.3 Motion Control

The motion control of a mobile robot is done the by the control of the its actuators [23], it can be complex or simple which results into two levels of control:

1.10.3.1 Low Level Control

Low-level control refers to controlling the robot actuators to follow the trajectory, usually using PID controllers .

1.10.3.2 High Level Control

High-level control makes tasks like deciding when to re-plan the path, handle unexpected obstacles,

1.10.4 Sensor Integration

It is necessary for motion generation to use and integrate data from sensors mentioned in earlier, for example the use of an ultrasonic sensor to detect obstacles and objects in front of the robot as shown in figure 1.10.

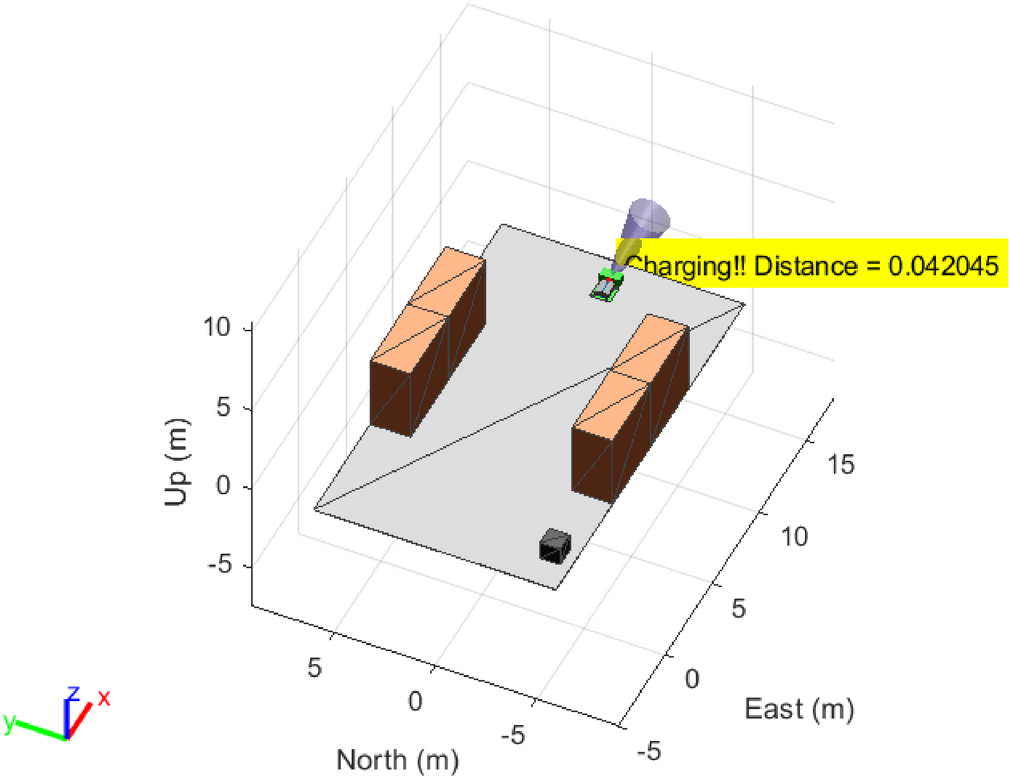


Figure 1.10: Sensor Integration Example

1.11 Global look on the robot

This figure 1.11 presents the major composing elements of a mobile robot, where in the next chapter we will focus on the architeture of the controller

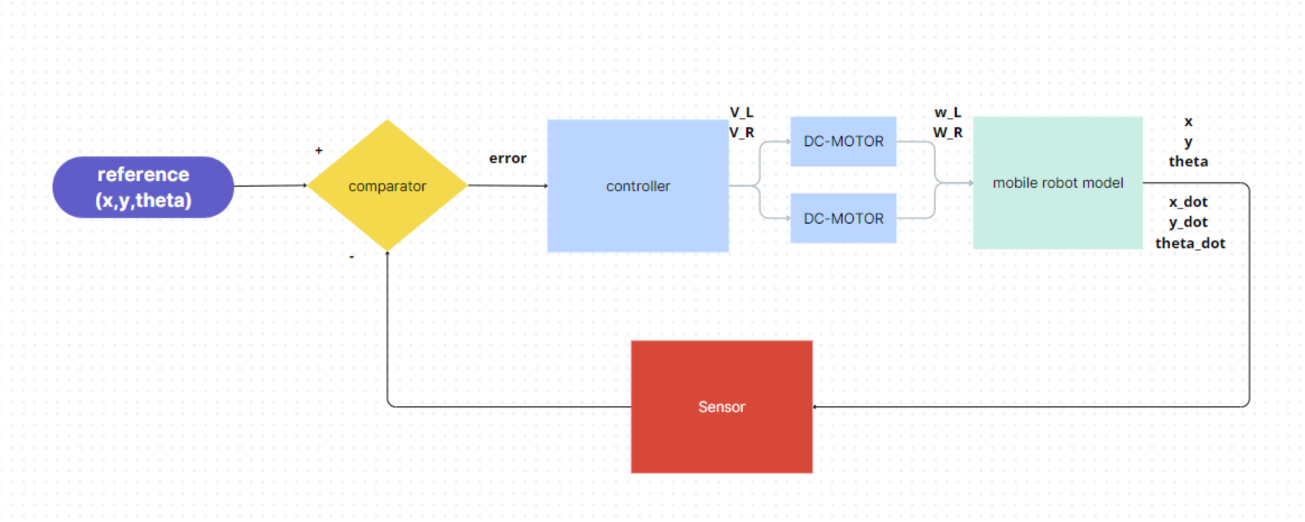


Figure 1.11: general system schema

1.12 Conclusion

A theoretical study is presented in this chapter related to wheeled mobile robots specifically wheeled differential mobile robot, which provides better insights about the kinematic modeling and motion generation of a robot. This work sets the foundation to understand and further the scope of mobile robotics and to further exploited in the next chapter to make a controller system.

Chapter 2

FUZZY LOGIC CONTROL

2.1 Introduction

Fuzzy logic provides a framework for dealing with uncertainty and imprecision in decision-making and control systems. Unlike traditional binary logic, which relies on crisp, yes-or-no values (0 or 1), fuzzy logic allows for the representation of vague or ambiguous concepts [8].

For example, a binary rule might dictate that the robot should stop if it detects an obstacle directly in front of it. However, in a crowded environment, the robot may encounter situations where the obstacle is not directly in its path but still poses a potential collision risk.

Fuzzy logic on the other hand, allows the robot to make more nuanced decisions based on fuzzy rules that consider factors like the proximity and velocity of nearby objects. For instance, the robot could have fuzzy rules that prioritize slowing down if it detects objects in its vicinity, with the degree of slowing determined by the perceived proximity of the obstacles [1], [33], [10].

By incorporating fuzzy logic into its control system, the mobile robot can navigate through complex and dynamic environments more effectively, making decisions that are better suited to the uncertain and ambiguous nature of real-world scenarios [31], [11].

The complexity of the controller depends on number of rules, in [4] and eight(8) rule Fuzzy logic controller was designed for for path planning and obstacle avoidance for mobile robots operating in hostile environments. the controller can also depends oh the behavior of the robot where in [9] a hierarchical behavior based control strategy was made, in which four different reactive behaviors are combined by means of a fuzzy supervisor. In [30] it was shown that the number of input membership function effects the performance quality of the controller.

For Obstacle avoidance, the use of sensor is crucial because it provides the necessary information about the obstacle in which they will become the inputs of the controller as seen in [29].

2.2 Theory of fuzzy logic

2.2.1 Definition of a fuzzy set

If U is a collection of objects or values denoted by x, then a fuzzy set A in U is defined by the set of ordered pairs, as follows:

$$A = \{(x, \mu_A(x)) \mid x \in U\}$$

where:

x is an element from the universe of discourse U,

$\mu_A(x)$ is the membership function of x in A,

$\mu_A(x_i)$ maps each element x_i to a membership value in the interval $[0, 1]$.

This membership value $\mu_A(x)$ represents the degree to which x belongs to the fuzzy set A.

2.2.2 Membership Function

A certain fact will have a membership function equal to 1 for the considered operating point, while an uncertain fact will have a membership function less than or equal to 1. When the certain fact corresponds to the statement of the value of a variable $x = x_0$, we have $\mu_{x_0}(x_0) = 1$ for $x = x_0$ and $\mu_{x_0}(x_0) = 0$ for $x \neq x_0$, resulting in a singleton .

An uncertain fact such as x approximately equal to x_0 is expressed with a mathematical formula as membership function, it's parameterized according to the its complexity, we can find for example:

2.2.2.1 Triangular Membership Function

: It is defined by three parameters: a, b, and c, where a < b < c.

$$\mu_A(x) = \begin{cases} 0 & \text{if } x \leq a \\ \frac{x-a}{b-a} & \text{if } a < x \leq b \\ \frac{c-x}{c-b} & \text{if } b < x \leq c \\ 0 & \text{if } x > c \end{cases} \tag{2.1}$$

where

- $\mu_A(x)$ membership value of x in the fuzzy set A
- and a, b, and c are the parameters defining the triangular shape of the membership function.

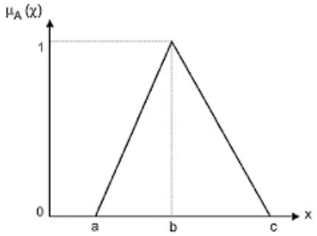


Figure 2.1: triangular membership function

In figure 2.1 an illustration about this function is made.

2.2.2.2 Trapezoidal Membership Function

A trapezoidal MF is specified by four parameters a, b, c, d as follows:

$$\mu_A(x) = \begin{cases} 0 & \text{if } x \leq a \\ \frac{x-a}{b-a} & \text{if } a < x \leq b \\ 1 & \text{if } b < x \leq c \\ \frac{d-x}{d-c} & \text{if } c < x \leq d \\ 0 & \text{if } x > d \end{cases} \tag{2.2}$$

Where:

- $\mu_A(x)$ is the membership value of x in the fuzzy set A.
- a, b, c, and d are the parameters that define the trapezoidal shape of the membership function.

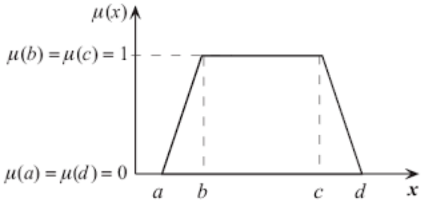


Figure 2.2: Trapezoidal Membership Function

In figure 2.2 an illustration about this function is made.

2.2.2.3 Gaussian Membership Function

A Gaussian MF is specified by two parameters a, b as follows:

$$\mu_A(x) = e^{-\frac{(x-a)^2}{2b^2}} \tag{2.3}$$

Where:

- $\mu_A(x)$ is the membership value of x in the fuzzy set A.
- a is the center of the Gaussian distribution.
- b is the standard deviation, controlling the width of the Gaussian curve.

In figure 2.3 an illustration about this function is made.

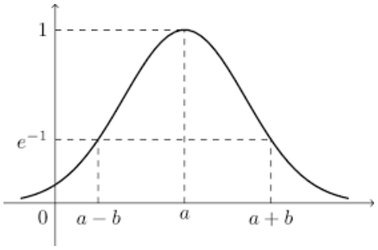


Figure 2.3: Gaussian Membership Function

2.2.3 Operations in fuzzy logic

The generalization of the negation, intersection, and union operators from ordinary set theory .

- **Negation Operator (NOT):**

$$\mu_{\neg A}(x) = 1 - \mu_A(x)$$

- **Intersection Operator (AND):**

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$$

- **Union Operator (OR):**

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$$

2.2.4 Interference rules

Inference rules in fuzzy logic are the set of rules that link the input linguistic variables of a system to the output linguistic variables using fuzzy operators. The syntax is similar to that of binary logic: if *condition* then *consequence.* The condition can be a simple or compound statement.

In general, the rules can be expressed in the following form:

If "antecedent 1" AND/OR "consequence 1"
"antecedent 2" AND/OR "consequence 2"
. . .
"antecedent n" Then "consequence n"

- Antecedents: The "conditions" of the rule.
- Consequences: The "actions" that result from the conditions.

The antecedents correspond to the fuzzy inputs (degrees of membership) determined during the fuzzification process.

2.2.5 Linguistic Variable

The description of a certain situation, phenomenon, or process generally includes fuzzy qualifiers such as: few, many, a lot, rarely, frequently, often cold, warm, hot, small, medium, large, etc. . These words or phrases in natural or formal language form linguistic variables. Since words are generally less precise than numbers, the concept of a linguistic variable provides a way to describe these phenomena approximately. Therefore, linguistic variables represent a state in the system to be controlled or a control variable in a fuzzy controller.

A linguistic variable is a triplet $(x, U, T(x))$ where:

- x: A variable defined on the set U;
- U: The universe of discourse associated with the base value;
- $T(x)$: The set of linguistic values that the variable x can take within a normalized universe of discourse U.

2.3 Architecture of the fuzzy logic control

To apply fuzzy rules and develop a control signal, the Mamdani inference method for designing a fuzzy controller is presented in this section. Figure 2.4 illustrates the basic configuration of a fuzzy controller:

- **Fuzzy Input:** This step involves representing all crisp inputs as a state or conditions in a fuzzy system. Fuzzy inputs are linguistic variables characterized by fuzzy sets composed of membership functions defining their degree of membership.
- **Fuzzification Strategy and Fuzzy Set:** It is the process of assigning a set of membership degrees to each linguistic term on a given input variable. This procedure converts crisp input values into corresponding fuzzy sets. Implemented functions can be of any membership type, such as triangular, trapezoidal, Gaussian, or sigmoidal.
- **Knowledge Base and Fuzzy Inference:** The knowledge base consists of fuzzy rules defining the relationship between a fuzzy input and fuzzy output. Each rule usually takes

the form "IF-THEN", where the antecedent (IF) part contains combinations of input fuzzy sets, and the consequent (THEN) part defines the output fuzzy set to produce. Fuzzy inference synthesizes the fuzzy rules to establish a mapping presented by the input fuzzy sets to yield the overall output fuzzy set based on the exits in input fuzzy sets and the activation of each rule.

- **Defuzzification:** This process combines the consequences of different rules firing for the same output variable to form a single combined output fuzzy set. Defuzzification determines a single crisp value from the combined output fuzzy set using a method like centroid, maxima, or weighted average.

- **Fuzzy output:**The crisp output value is achieved through the process of defuzzification, representing the control action or decision taken by the fuzzy logic system based on the input fuzzy sets, the fuzzy rules in the knowledge base, and the defuzzification process. Accordingly, this crisp output value finally represents the conclusions of the fuzzy logic system that guide the action or decision determinations.

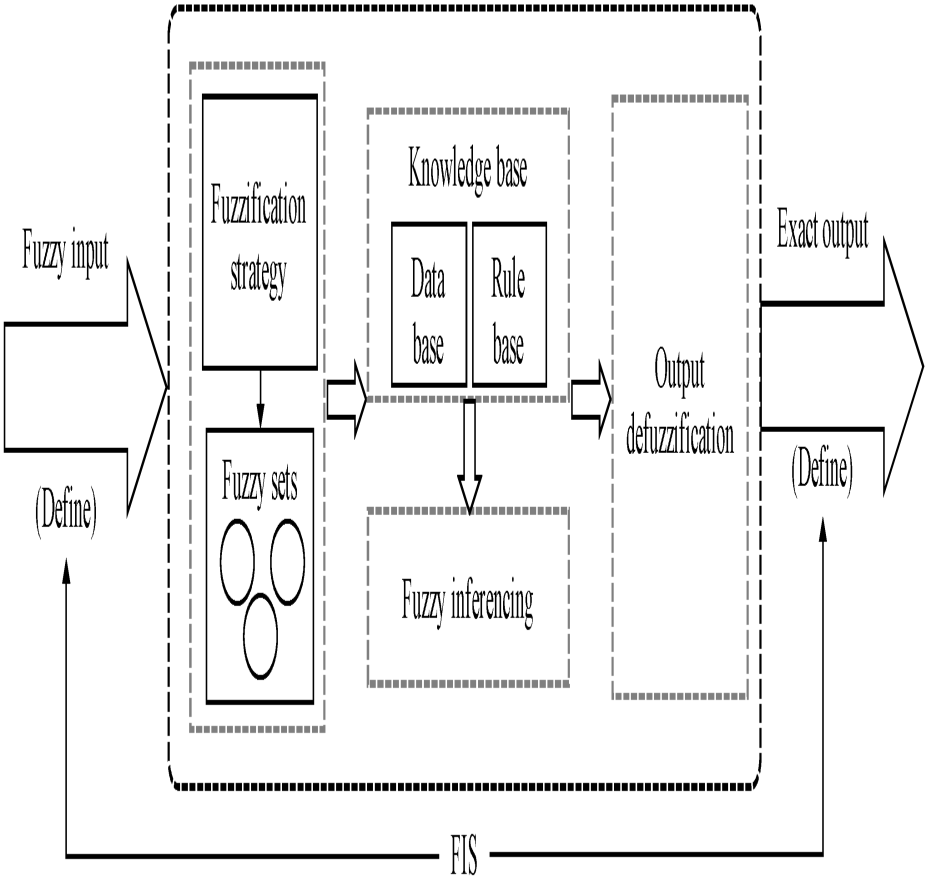


Figure 2.4: fuzzy logic architecture

2.4 Controller Design

There are different types of control systems currently used in various application areas. The most commonly used control system today in the industry is the PID controller. But in this research our focus is on fuzzy control system in which it will solve mobile robot navigation problems. In fuzzy controller design, the first step is determining the inputs to the controller and the outputs to be controlled.

2.4.1 Selection of input variables

While the main goal of a mobile robot is to reach destination, this its necessary to stay updated with the position of the mobile robot, therefore to know the distance and the required orientation to the desired position. As result we can have two(2) inputs : *the distance to goal* "**Dist**" and *the orientation* "θ_E". this two inputs are represented by the following equations:

$$Dist = \sqrt{(X_T - X)^2 + (Y_T - Y)^2} \tag{2.4}$$

$$\theta_E = \theta_D - \theta \tag{2.5}$$

where (X_D, Y_D) and θ_D are the desired position and initial orientation, (X, Y) and θ are the current position and current orientation

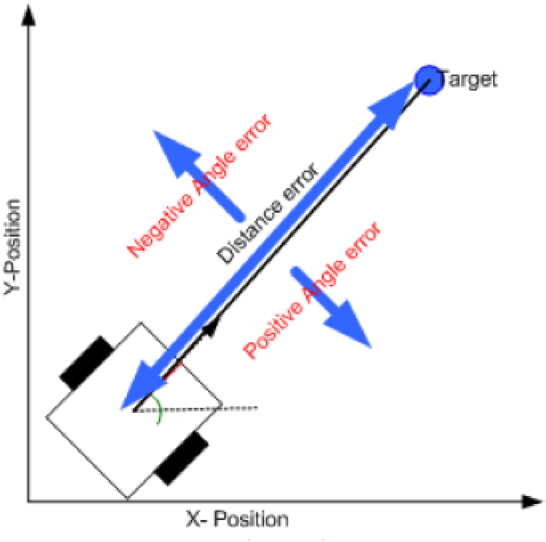


Figure 2.5: input of the controller

It's also necessary to be able to avoid obstacles along the way to the goal destination, and that can be achieved using ultrasonic sensors, they will be mounted in the front of the robot, center, left and right side of it. This sensors will provide feedback about the distance to obstacle, and it is a very important information for the robot motion control. From the above we can recognize two major behavior for the robot

1. Reach Target
2. Obstacle avoidance

Reach Target(RT) is the behavior of the robot in which it reaches the desired position. Obstacle avoidance(OA) behavior take place when the sensors of the robot detects an obstacle along the path, so the robot must change direction in order to avoid collision with the obstacle

2.4.2 Selection of output variables

As seen in chapter (1) the differential mobile robot has two(2) wheels which can run with different speeds, therefore we can adjust the wheels speed to obtain the required speed and orientation. As result we set the output of the controller to be "*the left wheel speed*" and "*the right wheel speed*". The final goal is simply avoid obstacles in the absence and presence of disturbances and uncertainties [32].

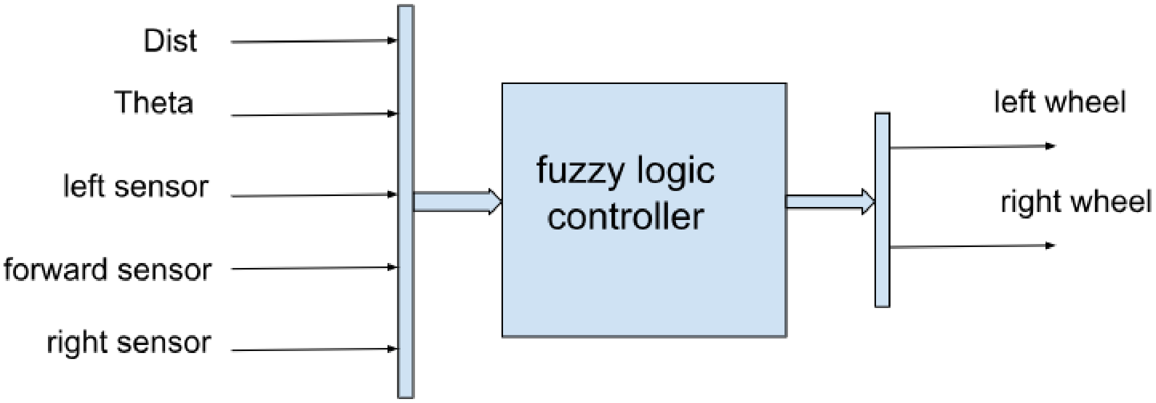


Figure 2.6: Inputs and Outputs of the controller

From figure 2.6 we can now start setting our fuzzy logic controller(FLC).

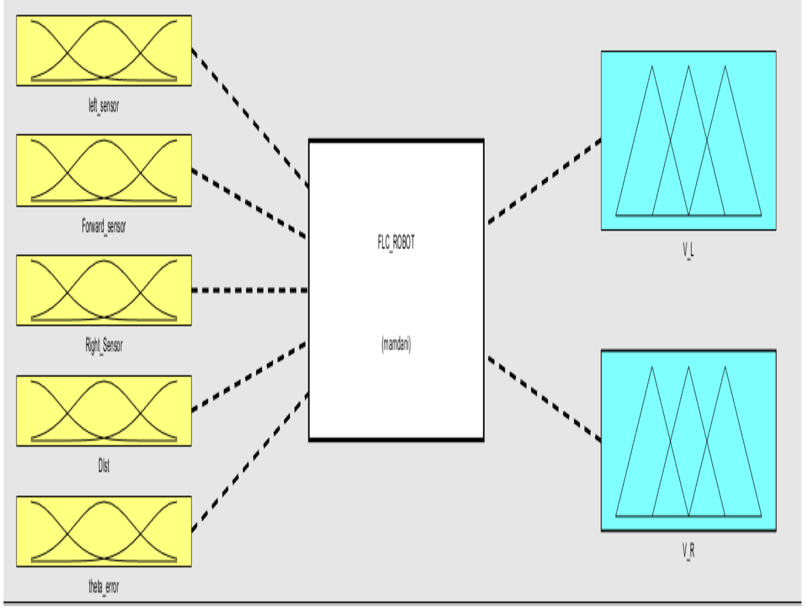


Figure 2.7: Fuzzy logic design

The membership function are divided into **Input membership Function** and **Output membership functions** as shown in tables 2.1 and 2.2 below:

fuzzy set	membership function	parameters
leftsensor	Triangular	Near, Medium, Far
forwardsensor	Triangular	Near, Medium, Far
rightsensor	Triangular	Near, Medium, Far
Dist	Triangular	Close, Medium, Far
theta error	Triangular	right, forward, left

Table 2.1: Input membership functions

fuzzy set	membership function	parameters
V_R	Triangular	NH , N , P , PH
V_L	Triangular	NH , N , P , PH

Table 2.2: Output membership functions

In figures (2.11, 2.9, 2.8, 2.12, 2.10, 2.13, 2.14), the Membership Functions for the inputs and outputs of the controller are illustrated

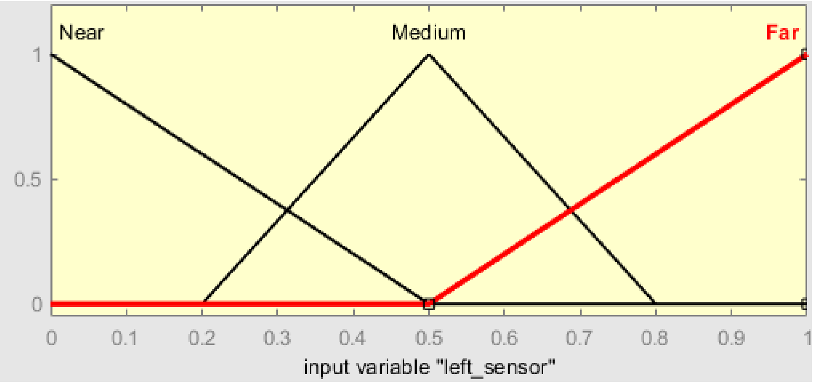


Figure 2.8: left sensor membership function

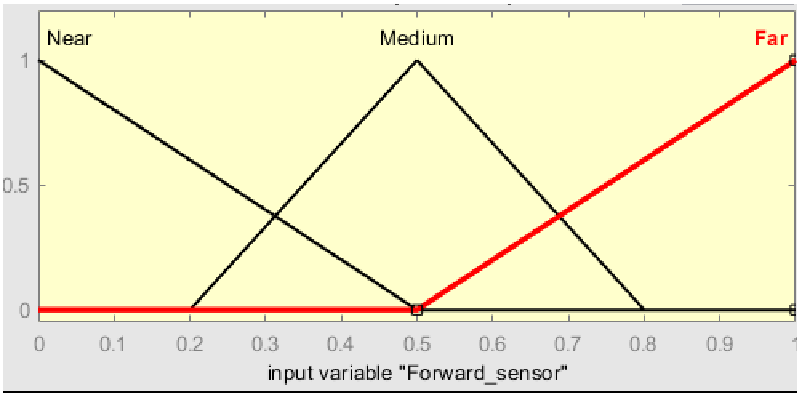


Figure 2.9: forward sensor membership function

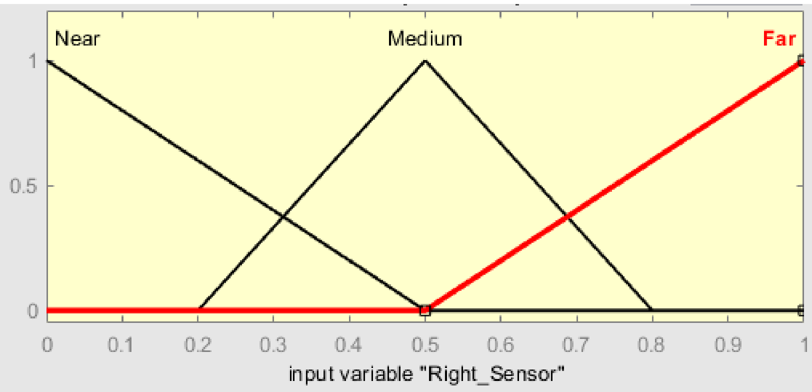


Figure 2.10: right sensor membership function

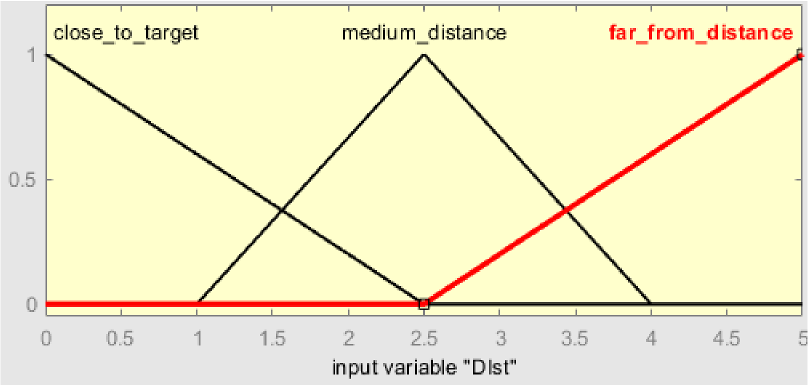


Figure 2.11: distance to goal membership function

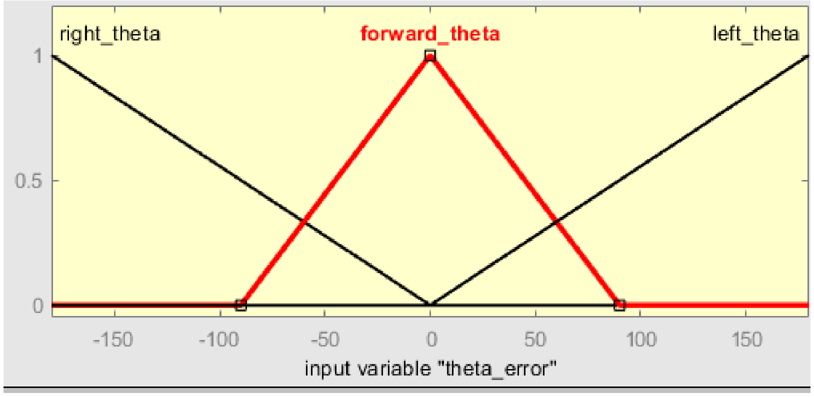


Figure 2.12: orientation error membership function

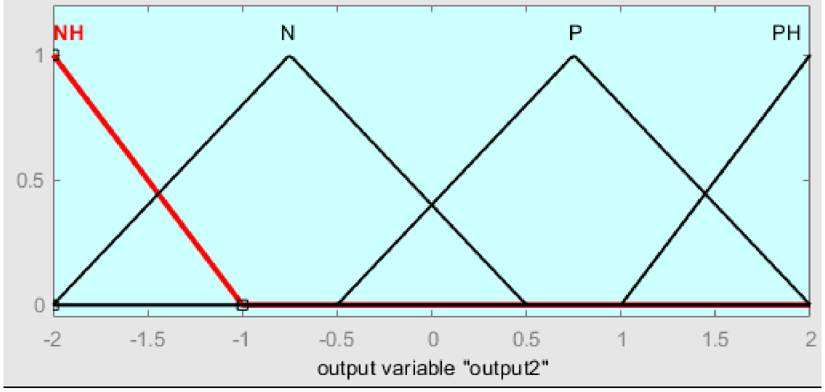


Figure 2.13: right wheel speed membership function

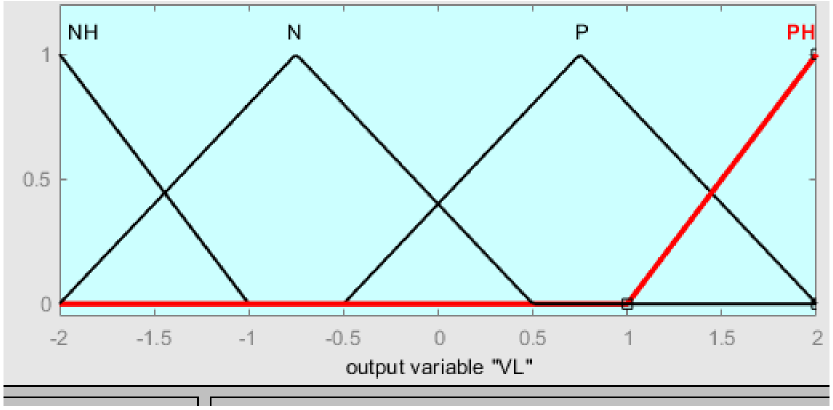


Figure 2.14: left wheel speed membership function

2.4.2.1 obstacle avoidance rules:

In table 2.4 and 2.3 below, the necessary rules to avoid obstacle are made based on the information generated by three (03) ultrasonic sensor. This sensors are mounted in the left, front, and right side of the robot to provide an awareness for the robot surroundings.

Number	Left sonsr	Forward sensor	Right sensor	Left wheel speed	Right wheel speed
19	far	near	near	NH	PH
20	far	near	medium	NH	PH
21	far	near	far	PH	NH
22	far	medium	near	N	PH
23	far	medium	medium	N	P
24	far	medium	far	N	P
25	far	far	near	N	PH
26	far	far	medium	P	PH
27	far	far	far	P	P

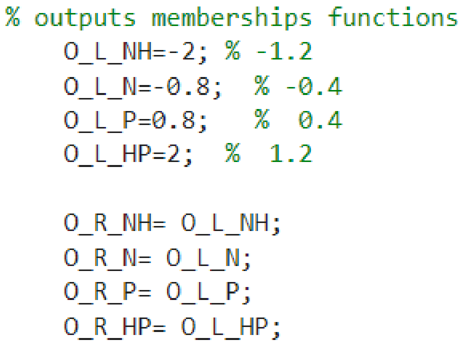
Table 2.3: Fuzzy rules for Target Reach (Part 2)

Number	Left sensor	Forward sensor	Left sensor	Left wheel speed	Right wheel speed
1	Near	Near	Near	NH	NH
2	Near	Near	Medium	PH	NH
3	Near	Near	Far	HP	NH
4	Near	Medium	near	NH	NH
5	Near	Medium	medium	PH	N
6	Near	Medium	far	PH	N
7	Near	Far	near	NH	NH
8	Near	Far	medium	P	N
9	Near	Far	far	P	N
10	Medium	near	near	NH	PH
11	Medium	near	medium	NH	PH
12	Medium	near	far	PH	NH
13	Medium	medium	near	N	PH
14	Medium	medium	medium	NH	NH
15	Medium	medium	far	PH	N
16	Medium	far	near	N	PH
17	Medium	far	medium	P	P
18	Medium	far	far	PH	P

Table 2.4: Fuzzy rules for Target Reach (Part 1)

As shown in figure 2.15 the fuzzy outputs represent the following meanings:

- NH: Negative High, (maximum wheel's speed reaches 2 m/s backwards)
- N: Negative, (wheel's speed reaches around 0.8 m/s backward)
- P: Positive, (wheel's speed reaches around 0.8 m/s forward)
- PH: Positive High, (maximum wheel's speed reaches 2 m/s forwards)



```
% outputs memberships functions
    O_L_NH=-2; % -1.2
    O_L_N=-0.8;  % -0.4
    O_L_P=0.8;   %  0.4
    O_L_HP=2;  %  1.2

    O_R_NH= O_L_NH;
    O_R_N= O_L_N;
    O_R_P= O_L_P;
    O_R_HP= O_L_HP;
```

Figure 2.15: Outputs of the fuzzy logic controller

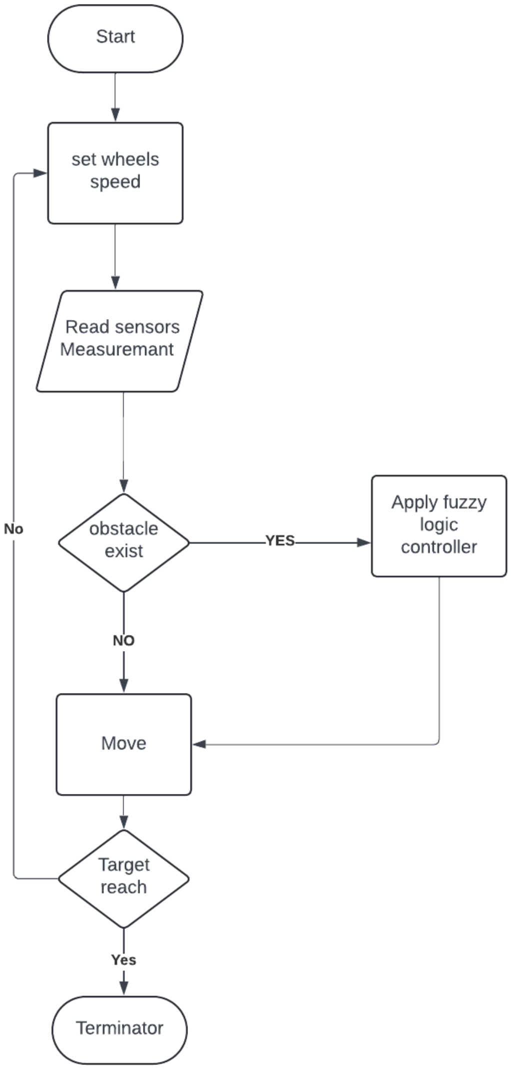


Figure 2.16: obstacle avoidance flowchart

The figure 2.16 represent the process behind obstacle avoidance base on the fuzzy logic controller

2.5 conclusion

In this chapter, we explored the concepts of fuzzy logic, including the definition of fuzzy inputs and outputs, the fuzzification strategy, and defuzzification. We then applied these concepts to mobile robotics, designing a controller that utilizes sensor feedback as inputs and generates the desired wheel speeds as outputs, thereby meeting the functional requirements of a differential wheeled mobile robot.

To validate the effectiveness of this controller, we conducted a series of simulations using MATLAB. The results of these simulations, along with a detailed discussion of the controller's performance, will be presented in the next chapter.

Chapter 3

Simulation and Results

3.1 Introduction

In this chapter, the kinematic model for a differential mobile robot has been built based on t the mathematical modeling seen in chapter (01). After that, the use of a fuzzy logic controller (FLC) for obstacle avoidance has been established using Matlab2023, in order to test the robustness of the designed controller. In this section, we categorize the tests based on the robot's behavior into two types: the first test evaluates the robot's performance under ideal conditions, without any disturbances or uncertainties. The second test assesses the robot's performance when disturbances and uncertainties are introduced, providing a more comprehensive evaluation of the controller's robustness and adaptability. It is noted that any system can be affected by a disturbances whether its internal like a sensor noise, actuator faults or an external disturbances such as uneven terrain, or external force. Therefore, it necessary to put it in consideration while making this simulation

3.2 Parameters values

All the tests have been done during a time simulation of $t = 200s$ under this conditions:

R(radius of wheels in meters)	0.035
L(distance between wheels in meters)	0.28
(x_0, y_0)	(0 , 3.6)

Table 3.1: values of robot parameters

3.3 Membership function

These functions were selected following the evaluation of different types of membership functions, including trapezoidal, triangular, and Gaussian functions. Triangular functions were favored because they offer the ability to efficiently handle situations both in the absence and presence of disturbances and uncertainties in the data. This selection aims to provide the user with robust and reliable tools capable of overcoming obstacles related to the complex environments in which these functions will be applied.

3.4 Obstacle avoidance test

The obstacle to avoid in this simulation consists of two circular walls represented by concentric circles with respective radius of 3 meters and 4.2 meters. The coordinates of the walls are generated and plotted to visualize the simulation environment.

3.5 Results without disturbances and uncertainties

Figure 3.1 represents the robot behavior for obstacle avoidance,start moving from the initial position (x0, y0) and whenever the sensors detects an obstacle ahead, the (FLC) gives input control signals to adjust the wheels speed and make hard turns to avoid the obstacle.

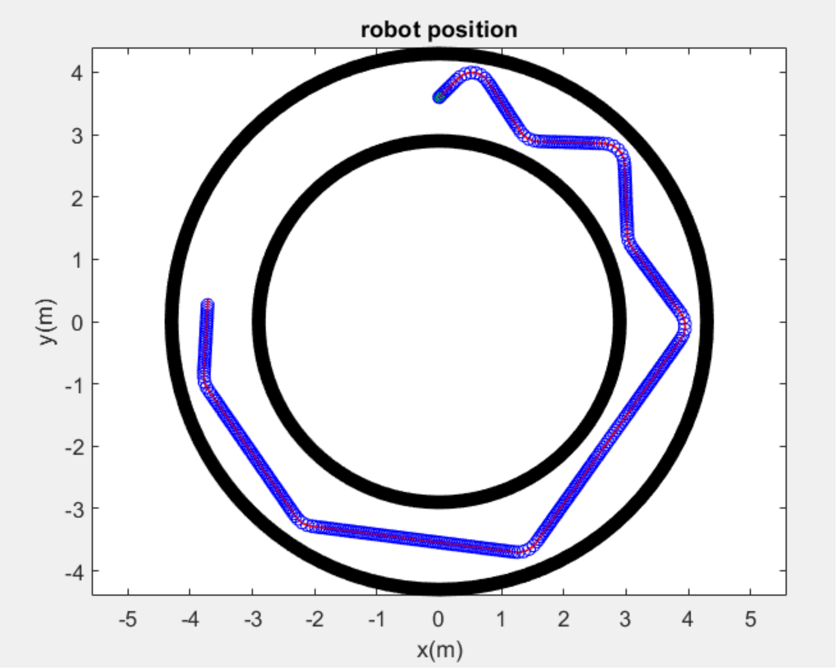


Figure 3.1: robot mouvement without disturbance

Figure 3.2 and Figure 3.3 represent the variation of x and y values of the robot during this simulation

Starting from the initial position of (0 , 3.6) converging to the final value of (x_f, y_f) = (-3.75 , -0.25). This simulation is regardless to any sort of disturbance

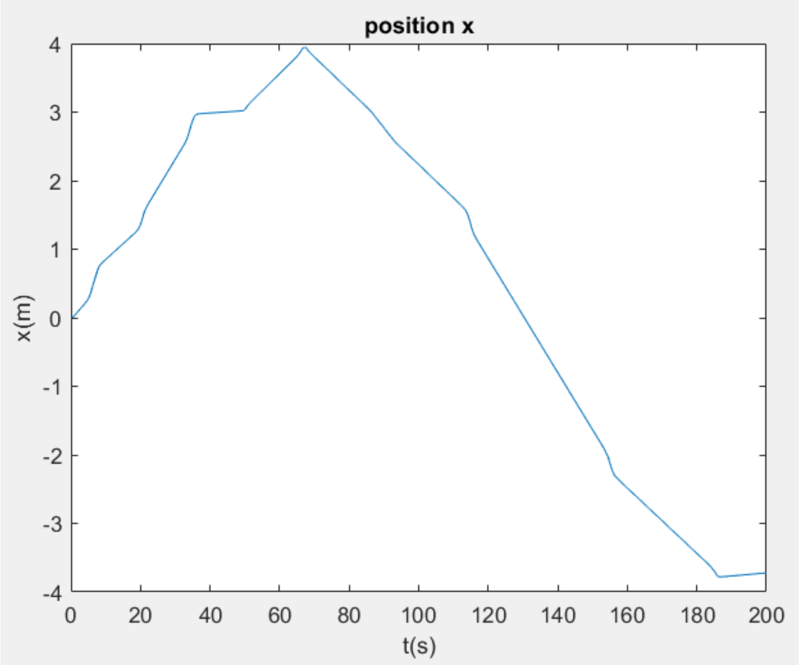


Figure 3.2: x values without disturbance

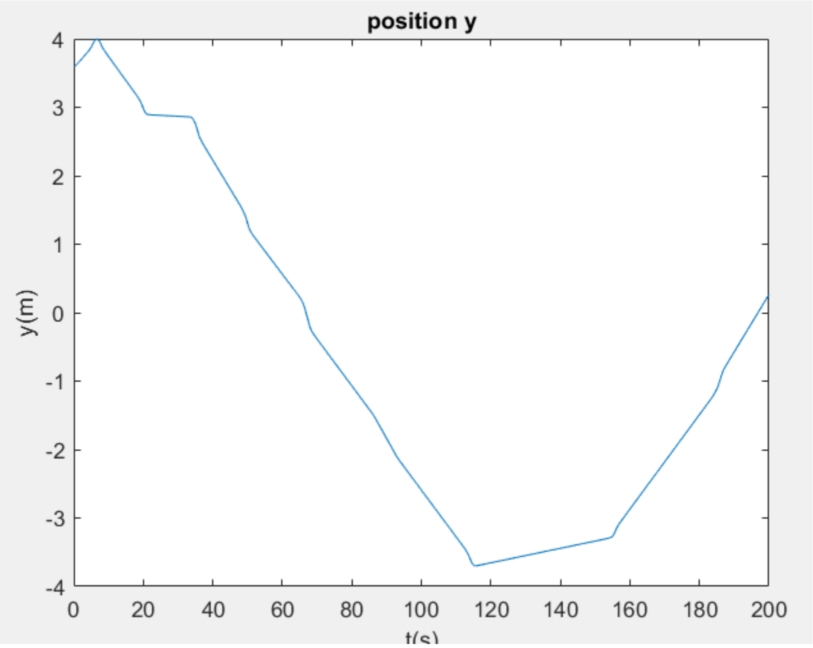


Figure 3.3: y values without disturbance

In figure 3.4 the movement of the robot is characterized by marked decelerations as it approaches obstacles, followed by accelerations once these are circumvented. This illustrates the robot's obstacle avoidance strategy.

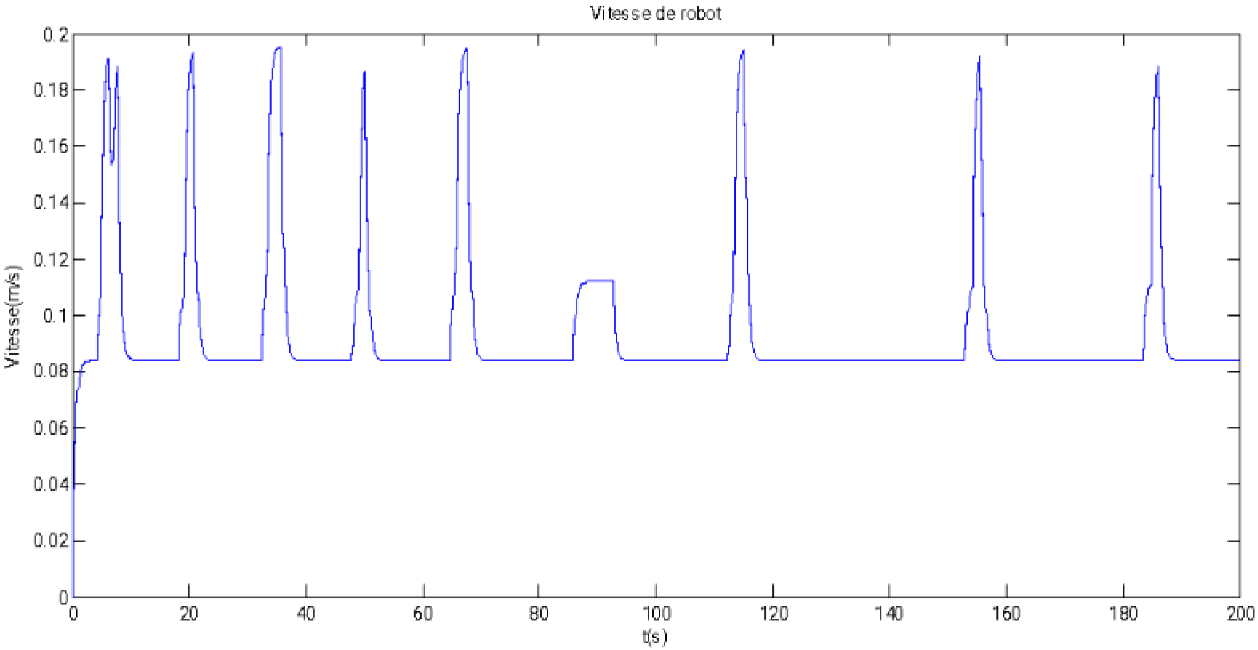


Figure 3.4: Speed without disturbances and uncertainties

3.6 Results with disturbances and uncertainties

In figure 3.5, from the first sight its noticed that the robot have the same behavior as in figure 3.1, detect an obstacle and the (FLC) generates the outputs to adjust the wheels of the robot. But only this time, a factor of disturbance is added to see how it will effect this system. the major difference between the two movement is the end position, where the difference can clearly be seen between the two(2) figures 3.1 and 3.5.

From analysing figure 3.6 and figure (3.7) and while it looks like they have the same variation as in figure 3.2 and figure 3.3, but when comparing the final position is different, in this case (x_f,y_f)=(-0.35 , 3.4) and comparing it to the previous test results, a difference is noted. Therefore the existence of disturbances and uncertainties encountered during the simulation can affect the final position reached by the system, owing to the duration of the simulation. As the simulation progresses over time, small disturbances or variations in the environment can accumulate and lead to the system ending up in a slightly different location than originally planned. However, it is important to note that these perturbations do not fundamentally impact the main objective of the system, which is to successfully navigate and avoid obstacles. The core functionality of obstacle avoidance remains unaffected, even if the final position deviates somewhat from the ideal trajectory due to the cumulative effects of disturbances over the course of the simulation.

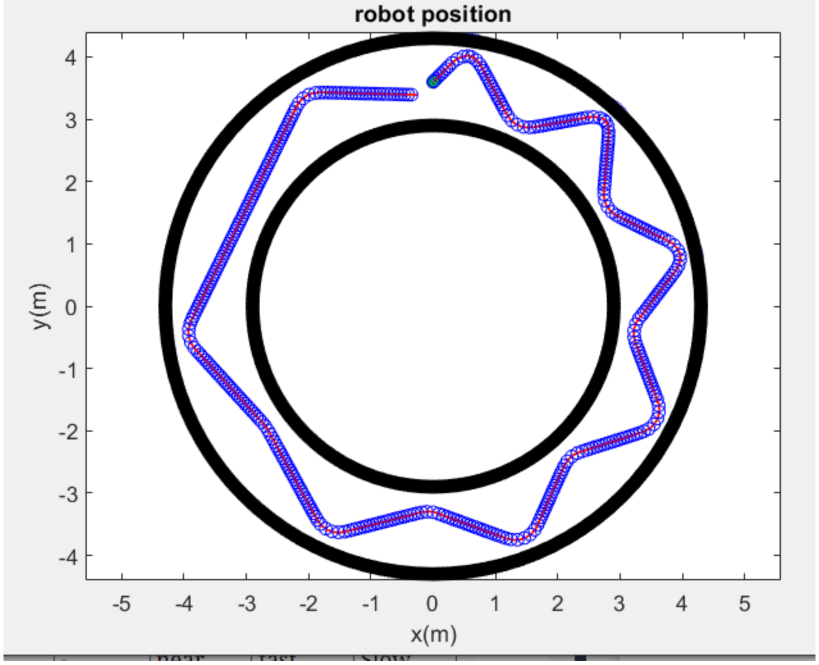


Figure 3.5: robot movement with disturbance

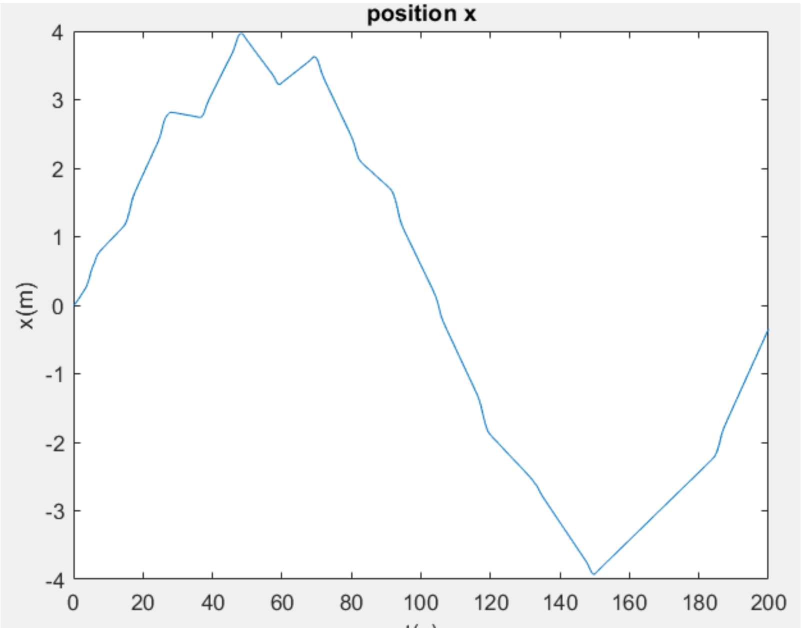


Figure 3.6: x values with disturbance

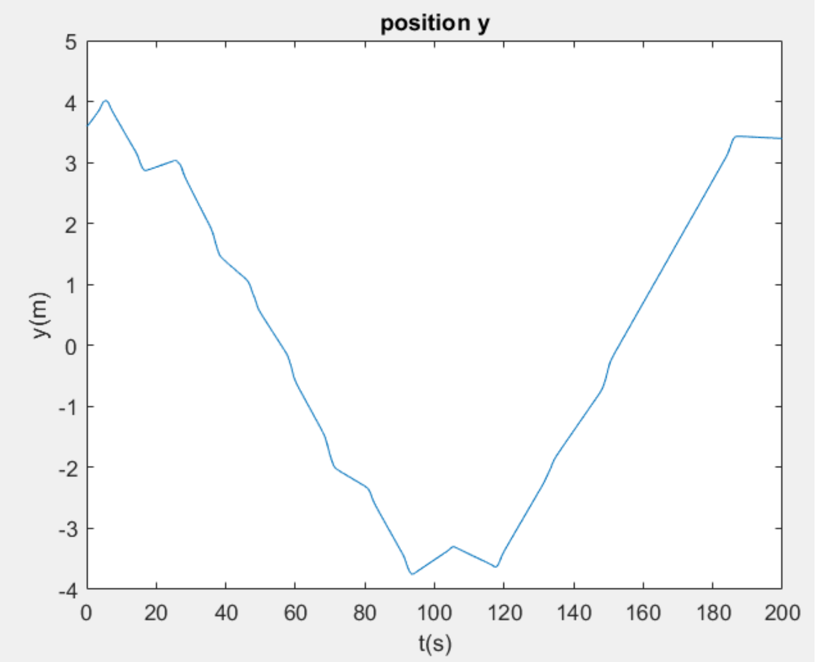


Figure 3.7: y values with disturbance

In figure 3.8 even in the presence of disturbances and uncertainties, the movement of the robot is characterized by marked decelerations as it approaches obstacles, followed by accelerations once these are circumvented. This obstacle avoidance strategy is maintained, although fluctuations in

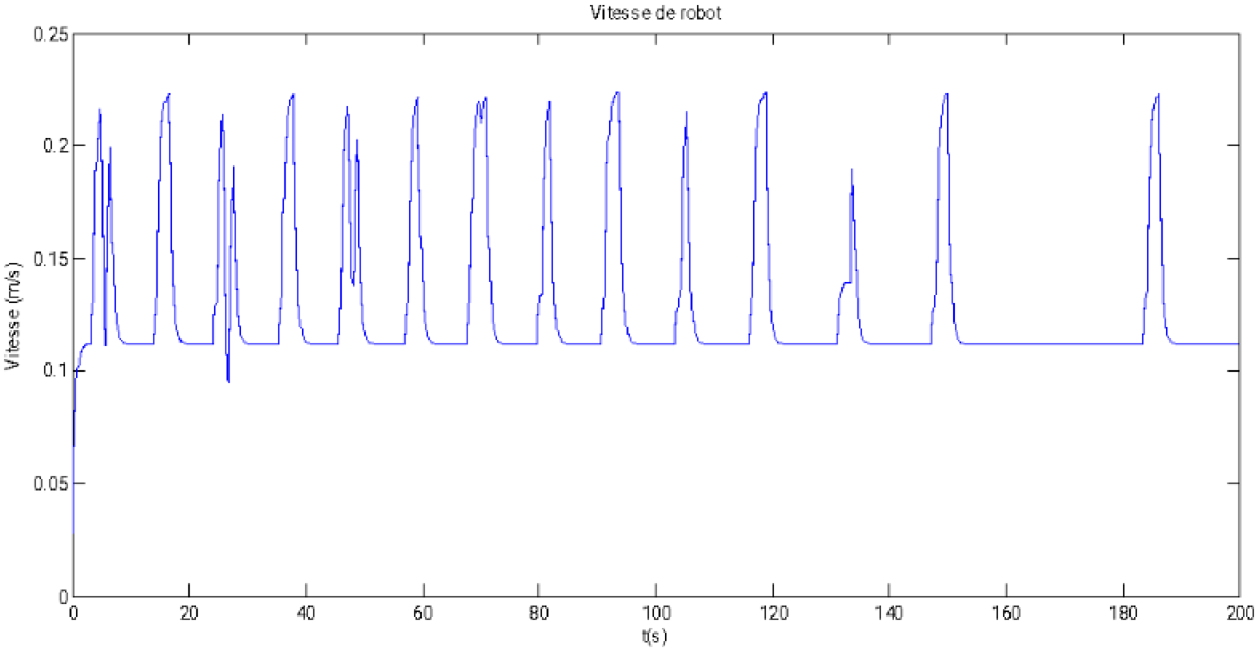


Figure 3.8: Speed with disturbances and uncertainties

speed compared to the first test are observed, reflecting the impact of disturbances on the robot's behavior.

3.6.1 Analyse and discussion

This robot starts moving in straightforward direction, which means both the left and right wheels have an equal speed. By the time one of the three sensors mounted in front of the robot detects an obstacle, and the fuzzy logic controller(FLC) receives those data,it generated an Input control signals for avoiding the obstacle by the variation of the wheels speed. For example, in figure 3.5 in the beginning of the trajectory, the robot moves forward till it almost reaches the outside wall and then it changes direction and turn to the right, this action took place because of the (FLC).

3.7 Conclusion

The circular environment provides a simple and efficient testing map for obstacle avoidance based on the fuzzy logic controller, this simulations are good enough to prove the robustness of this controller for obstacle avoidance. The sensors provides the input data of the controller, therefore their precision effects the outputs of the controller and the performance of the robot in general. The disturbances and uncertainties represents imprecise or incomplete information about the environment caused by limitations in sensor accuracy, unknown environmental conditions, and their existence may influence the final position, they do not affect the primary task, which is obstacle avoidance. The system is designed to prioritize avoiding obstacles, ensuring safety and reliability even when external factors cause deviations in the end position. This robust approach allows for effective navigation in the considered environments without compromising the core objective.

General Conclusion and Perspectives

This work provides an overview on mobile robots more specifically on differential wheeled mobile robot. Furthermore, fuzzy logic method was elaborated, and how to design controller for mobile robot uses. By putting these subjects together, a deeper insight about how fuzzy logic can make mobile robots work better and more flexibly in dynamic

This thesis is devided into three(3) chapters. In the first chapter, the fundamental concepts of mobile robotics were introduced. the various types of mobile robots were discussed, their key components, and the challenges they face in real-world applications. Mobile robots are essential in numerous fields, including industrial automation.

The second chapter delved into fuzzy logic control, detailing its principles, components, and advantages. Fuzzy logic offers a robust framework for handling imprecise and uncertain information, making it suitable for mobile robots. The structure of a fuzzy logic controller, including fuzzification, the inference engine, the rule base, and defuzzification, highlighting how these components work together to make decisions based on approximate reasoning.

In the final chapter, the application of a fuzzy logic controller on a mobile robot was made. The simulation results showed the FLC's effectiveness in managing uncertainties and disturbances, validating its potential for real-world deployment. considering the analysis of mobile robots, fuzzy logic control, and simulation presented in this thesis promising avenues for future research and development emerge to address current limitations for the developing of autonomous robotics fields. The advancements in AI and Machine Learning, Sensing and Perception, and Enhanced Simulation Techniques will overcome current limitation and upgrade mobile robots.

To conclude, this work has demonstrated the impact that fuzzy logic control can have on mobile robots. By connecting big ideas and hands-on use, we've shown that FLC is a strong and good way to make mobile robots more reliable.

Appendix A

obstacle avoidance without disturbance

```
close all
clear all;
clc;

% robot parameters
R = 0.035; L = 0.28; shape_r=0.1;
min_range=0.1; max_range=0.4;

% simulation parameters
sim_delta = 0.1;
t=0;
t_final=200;

% var for initialization
x_init=0; y_init=3.6; theta_init=45*pi/180;
xp_pos=x_init ;yp_pos=y_init; thetap=theta_init;

% vectors for visualization
vect_t=[];
```

```
vect_x=[];
vect_y=[];
vect_theta=[];

vect_L_dis=[];
vect_R_dis=[];
vect_F_dis=[];



% vector of the enivronment points
[ wall_u wall_d] = environment_type2();

while(1)

    % returns the distances measured and detect(0 and 1 if there is obs )
    [L_dis, R_dis, F_dis, detect]
    %= sensor_value_obs(xp_pos, yp_pos, thetap,min_range,max_range, wall_u, wall_d)

    % vectors used for the animation
    vect_L_dis=[vect_L_dis L_dis];
    vect_R_dis=[vect_R_dis R_dis];
    vect_F_dis=[vect_F_dis F_dis];

    if (detect == 0)
        w_lc=0.6; w_rc=0.6;
    else
        [w_lc, w_rc]= obstacle_avoidance_FLC_corr(L_dis, R_dis, F_dis);
    end
```

```
    % robot cmd input
    [ w_l,w_r] = Diff_Robot_Model(w_lc, w_rc, sim_delta);

    % odometry
    [xn_pos,yn_pos,thetan]= Odometry(w_l, w_r, sim_delta,xp_pos,yp_pos, thetap,[R L]);
    xp_pos=xn_pos ;yp_pos=yn_pos; thetap=thetan;

    % filling the plot vectors
    vect_t=[vect_t t];
    vect_x=[vect_x xp_pos];
    vect_y=[vect_y yp_pos];
    vect_theta=[vect_theta thetap];

    t=t+sim_delta;
    if(t>t_final)
        break
    end
end

figure(1)
draw_PFE2(vect_x, vect_y,vect_theta,vect_t,x_init,y_init);
$title('robot position'); xlabel('x(m)'); ylabel('y(m)');
figure(2)
plot(vect_t,vect_x); title('position x'); xlabel('t(s)'); ylabel('x(m)');
figure(3)
plot(vect_t,vect_y); title('position y'); xlabel('t(s)'); ylabel('y(m)');
```

Appendix B

fuzzy logic controller

```
% Obstacle Avoidance
% L_sensor = {N, M, F}
% R_sensor = {N, M, F}
% F_sensor = {N, M, F}

% L = {NH, N, P, HP}
% R = {NH, N, P, HP}


function [Lw, Rw]= obstacle_avoidance_FLC_corr(L_sensor, R_sensor, F_sensor)

% fuzzyfication trianglar_fct means
$max(min((X-(a))/((b)-(a)),((c)-X)/((c)-(b))), 0) __a/b\c___
    L_N= trianglar_fct( 0+0.1, 0.1+0.1 ,0 , L_sensor, 'start');
    L_M= trianglar_fct( 0.05+0.1,0.15+0.1 ,0.25+0.1 , L_sensor, 'tria');
    L_F= trianglar_fct( 0.25+0.1, 0.3+0.1 ,0 , L_sensor, 'end');

    R_N= trianglar_fct( 0+0.1, 0.1+0.1 ,0 , R_sensor, 'start');
    R_M= trianglar_fct( 0.05+0.1,0.15+0.1 ,0.25+0.1 , R_sensor, 'tria');
```

```
    R_F= trianglar_fct( 0.25+0.1, 0.3+0.1 ,0 , R_sensor, 'end');



    F_N= trianglar_fct( 0+0.1, 0.1+0.1 ,0 , F_sensor, 'start');
    F_M= trianglar_fct( 0.05+0.1,0.15+0.1 ,0.25+0.1 , F_sensor, 'tria');
    F_F= trianglar_fct( 0.25+0.1, 0.3+0.1 ,0 , F_sensor, 'end');

% activation degree
    Ad=zeros(27,1);
    Ad(1)=min([L_N F_N R_N]);
    Ad(2)=min([L_N F_N R_M]);
    Ad(3)=min([L_N F_N R_F]);
    Ad(4)=min([L_N F_M R_N]);
    Ad(5)=min([L_N F_M R_M]);
    Ad(6)=min([L_N F_M R_F]);
    Ad(7)=min([L_N F_F R_N]);
    Ad(8)=min([L_N F_F R_M]);
    Ad(9)=min([L_N F_F R_F]);

    Ad(10)=min([L_M F_N R_N]);
    Ad(11)=min([L_M F_N R_M]);
    Ad(12)=min([L_M F_N R_F]);
    Ad(13)=min([L_M F_M R_N]);
    Ad(14)=min([L_M F_M R_M]);
    Ad(15)=min([L_M F_M R_F]);
    Ad(16)=min([L_M F_F R_N]);
    Ad(17)=min([L_M F_F R_M]);
    Ad(18)=min([L_M F_F R_F]);
```

```
    Ad(19)=min([L_F F_N R_N]);
    Ad(20)=min([L_F F_N R_M]);
    Ad(21)=min([L_F F_N R_F]);
    Ad(22)=min([L_F F_M R_N]);
    Ad(23)=min([L_F F_M R_M]);
    Ad(24)=min([L_F F_M R_F]);
    Ad(25)=min([L_F F_F R_N]);
    Ad(26)=min([L_F F_F R_M]);
    Ad(27)=min([L_F F_F R_F]);

% rules activation
    % L = {NH, N, P, HP}
    A_L_NH= max([Ad(1) Ad(4) Ad(7) Ad(10) Ad(11) Ad(14) Ad(19) Ad(20)]);
    A_L_N=  max([Ad(13) Ad(16) Ad(22) Ad(23) Ad(24) Ad(25)]);
    A_L_P= max([Ad(8) Ad(9) Ad(17) Ad(26) Ad(27)]);
    A_L_HP= max([Ad(2) Ad(3) Ad(5) Ad(6) Ad(12) Ad(15) Ad(18) Ad(21)]);

    % R = {NH, N, P, HP}
    A_R_NH= max([Ad(1) Ad(2) Ad(3) Ad(4) Ad(7) Ad(12) Ad(14) Ad(21)]);
    A_R_N = max([Ad(5) Ad(6) Ad(8) Ad(9) Ad(15)]);
    A_R_P = max([ Ad(17) Ad(18) Ad(23) Ad(24) Ad(27)]);
    A_R_HP = max([Ad(10) Ad(11) Ad(13) Ad(16) Ad(19) Ad(20) Ad(22) Ad(25) Ad(26)]);

% outputs memberships functions
    O_L_NH=-2; % -1.2
    O_L_N=-0.8;  % -0.4
    O_L_P=0.8;   %  0.4
    O_L_HP=2;  %  1.2
```

```
    O_R_NH= O_L_NH;
    O_R_N= O_L_N;
    O_R_P= O_L_P;
    O_R_HP= O_L_HP;

% defuzzification with  COG (center of gravity)
Lw=(O_L_NH*A_L_NH + O_L_N*A_L_N + O_L_P*A_L_P + O_L_HP*A_L_HP )/
$(A_L_NH + A_L_N + A_L_P + A_L_HP );
Rw=(O_R_NH*A_R_NH + O_R_N*A_R_N + O_R_P*A_R_P + O_R_HP*A_R_HP)/
$(A_R_NH + A_R_N + A_R_P + A_R_HP );

% MeOM (mean of maxima)
%Lw=(O_L_NH*A_L_NH + O_L_N*A_L_N + O_L_P*A_L_P + O_L_HP*A_L_HP )/3;
%Rw=(O_R_NH*A_R_NH + O_R_N*A_R_N + O_R_P*A_R_P + O_R_HP*A_R_HP)/3;

end
```

Bibliography

[1] D. N. M. Abadi and M. H. Khooban. Design of optimal mamdani-type fuzzy controller for nonholonomic wheeled mobile robots. *Journal of King Saud University-Engineering Sciences*, 27(1):92–100, 2015.

[2] M. Al Zamel. *Estimation techniques, to the odometric problem of an autonomous vehicle (BAT-MOB).* PhD thesis, Politecnico di Torino, 2021.

[3] H. M. Y. BABIKER. Mobile robot path following and obstacles avoidance using fuzzy logic control.

[4] H. Batti, C. B. Jabeur, H. Fourati, and H. Seddik. Fuzzy logic based control for autonomous mobile robot navigation and obstacles avoidance. In *2022 IEEE Information Technologies & Smart Industrial Systems (ITSIS)*, pages 1–6. IEEE, 2022.

[5] B. Bayle. Robotique mobile. *Ecole Nationale Superieure de Physique de StrasbourgUniversite Louis Pasteur*, 2007, 2008.

[6] T. A. de Araujo Costa, A. M. Ferreira, and M. S. Dutra. Parametric trajectory generation for mobile robots. In *ABCM Symposium Series in Mechatronics*, volume 3, pages 300–307, 2008.

[7] A. De Luca, G. Oriolo, and M. Vendittelli. Control of wheeled mobile robots: An experimental overview. *RAMSETE: articulated and mobile robotics for services and technologies*, pages 181–226, 2002.

[8] T. Edward, L. Tanya, and J. Mo. Introduction to fuzzy logic control with application to mobile robotics. *NASA Center for Autonomous Control Engineering Department of Electrical and Computer Engineering University of New Mexico*, 2009.

[9] S. A.-L. El-Teleity, Z. B. Nossair, H. M. A.-K. Mansour, and A. TagElDein. Fuzzy logic control of an autonomous mobile robot. In *2011 16th International Conference on Methods & Models in Automation & Robotics*, pages 188–193. IEEE, 2011.

[10] M. Elsayed, M. Sokar, S. Abd Rabbo, and M. El-Arabi. Wheeled mobile robot control using advanced fuzzy control. In *Proceedings of the 17th Int. AMME Conference*, volume 19, page 21, 2016.

[11] M. Faisal, R. Hedjar, M. Al Sulaiman, and K. Al-Mutib. Fuzzy logic navigation and obstacle avoidance by a mobile robot in an unknown dynamic environment. *International Journal of Advanced Robotic Systems*, 10(1):37, 2013.

[12] B. A. Francis, M. Maggiore, B. A. Francis, and M. Maggiore. Models of mobile robots in the plane. *Flocking and rendezvous in distributed robotics*, pages 7–23, 2016.

[13] Z. Ghania. *Commande Robuste d'un Robot Mobile à Roues*. PhD thesis, Université Mustapha Ben Boulaid Batna 2, Département d'électronique, 2017.

[14] I. A. Hassan, I. A. Abed, and W. A. Al-Hussaibi. Path planning and trajectory tracking control for two-wheel mobile robot. *Journal of Robotics and Control (JRC)*, 5(1):1–15, 2024.

[15] B. D. Hirpo and W. Zhongmin. Design and control for differential drive mobile robot. *International Journal of Engineering Research & Technology (IJERT)*, 6(10):327–329, 2017.

[16] Z. Houcine. *Conception, réalisation et commande floue d'un robot mobile*. PhD thesis, Université Mohamed Boudiaf des sciences et de la technologi, 2013.

[17] C. Iurian, F. Ikhouane, J. Rodellar Benedé, and R. Griñó Cubero. Identification of a system with dry friction. 2005.

[18] K. Kanjanawanishkul. Motion control of a wheeled mobile robot using model predictive control: A survey. *KKU Research Journal*, 17(5):811–837, 2012.

[19] C. W. Liew. *Obstacle avoidance robot applying fuzzy control system*. PhD thesis, Universiti Tun Hussein Onn Malaysia, 2014.

[20] L. Liu, X. Wang, X. Yang, H. Liu, J. Li, and P. Wang. Path planning techniques for mobile robots: Review and prospect. *Expert Systems with Applications*, page 120254, 2023.

[21] B. Y. Merbah Amira. *commande des mouvemount d'un robot mobile par logique floue*. PhD thesis, Université saad dahlab, Département d'électronique, 2017.

[22] A. S. Mohammed, B. Alkali, I. Suleiman, O. Adedipe, B. Salihu, O. Agonga, et al. Simulink modelling and simulation of mobile agricultural robots for farm monitoring. 2021.

[23] P. Morin and C. Samson. Motion control of wheeled mobile robots. *Springer handbook of robotics*, 1:799–826, 2008.

[24] M. Ni et al. Modelling and control for a class of mobile robots. *IFAC Proceedings Volumes*, 40(18):115–120, 2007.

[25] B. Nouri. Modelling and control of mobile robots. *ICMSAO/05, American University of Sharjah, UAE*, 2005.

[26] P. K. Panigrahi and S. K. Bisoy. Localization strategies for autonomous mobile robots: A review. *Journal of King Saud University-Computer and Information Sciences*, 34(8):6019–6039, 2022.

[27] J. J. Park and B. Kuipers. A smooth control law for graceful motion of differential wheeled mobile robots in 2d environment. In *2011 IEEE International Conference on Robotics and Automation*, pages 4896–4902. IEEE, 2011.

[28] E. Petrovic, V. Nikolic, I. Ciric, M. Simonovic, S. Pavlovic, M. Mancic, and B. Rajkovic. Kinematic model and control of mobile robot for trajectorytracking. *Annals of the Faculty of Engineering Hunedoara*, 14(2):161, 2016.

[29] R. D. Puriyanto and A. K. Mustofa. Design and implementation of fuzzy logic for obstacle avoidance in differential drive mobile robot. *Journal of Robotics and Control (JRC)*, 5(1):132–141, 2024.

[30] R. Rashid, I. Elamvazuthi, M. Begam, and M. Arrofiq. Fuzzy-based navigation and control of a non-holonomic mobile robot. *arXiv preprint arXiv:1003.4081*, 2010.

[31] R. Singh and T. Bera. Obstacle avoidance of mobile robot using fuzzy logic and hybrid obstacle avoidance algorithm. In *IOP Conference Series: Materials Science and Engineering*, volume 517, page 012009. IOP Publishing, 2019.

[32] A. Wondosen and D. Shiferaw. Fuzzy logic controller design for mobile robot outdoor navigation. *arXiv preprint arXiv:2401.01756*, 2024.

[33] P. Yanik, G. Ford, and W. McDaniel. An introduction and literature review of fuzzy logic applications for robot motion planning. In *Proceedings of ASEE Southeast Section Conference*, pages 1–10, 2010.