Fuzzy Logic Controller with Rplidar for Hybrid Three-Wheeled Omnidirectional Mobile Robot in Dynamic Environment

Ali A. Tuama * and Nabil H. Hadi

Mechanical, College of Engineering, University of Baghdad, Baghdad, Iraq
Email: Ali.Abdulhadi1403d@coeng.uobaghdad.edu.iq (A.A.T.); Dr.Nabil.Hassan@coeng.uobaghdad.edu.iq (N.H.H.)
*Corresponding author

Abstract**— This study presents the design and development of a fuzzy control system for a hybrid Three-Wheeled Omnidirectional Mobile Robot (3WOMR). As a holonomic robot, it can perform simultaneous translational and rotational motions. The proposed fuzzy controller enables efficient obstacle avoidance in environments with static and dynamic obstacles while minimizing structural vibration using rubber wheels for damping. An RPLIDAR sensor is employed to detect obstacle distances in three 45°-spaced sectors. The controller uses three input parameters distance from the head (DH), left (DL), and right (DR) to determine two output variables representing the angular velocities of the left and right wheels. The system operates through seventeen cognitive states and twenty-seven fuzzy rules implemented in Python using fuzzy logic libraries. Experimental tests were performed in a 3×4 m environment, both with and without obstacles. The robot successfully navigated the area, avoiding collisions and maintaining stability. MATLAB 2023b simulations confirmed the system's reliability and performance. The proposed fuzzy controller demonstrated improved accuracy and efficiency compared to existing methods, providing effective control of the robot's linear and angular motion for safe navigation in real-world conditions.**

Keywords**—Swedish wheel, fuzzy logic, autonomous navigation, avoidance obstacle, uncertainty, Python library**

I. Introduction

Mobile robots experience increasing interest because they serve numerous applications that operate in different regions including terrestrial [1], aerial [2] and aquatic environments [3]. The wheel-based mobile robots fall into two classification categories: holonomic and non-holonomic depending on their ability to steer. Non-holonomic robots consist of bicycle robots [4] and differential drive robots [5] and tricycles [6] together with car-like robots as described by Lima and Pereira [7]. Holonomic robots exhibit a condition where actuators match the available degrees of freedom in their systems. Omnidirectional mobile robots constitute holonomic systems since their design enables simultaneous operation between translational and rotational movements. The ability of holonomic robots to change direction without relying on their position makes them effective for personal help systems as described by Moreno *et al.* [8] alongside rehabilitation practice [9] and industrial manufacturing [10], and recreational sports activities [11].

Since Grabowiecki introduced the omnidirectional wheel in 1919 many diverse omnidirectional wheel mechanisms have emerged over the past few years. Numerous such examples were demonstrated starting with the ball castor introduced by Townsend [12], followed by the Mecanum wheel from Ilon [13] and the ball wheel developed by West and Asada [14] as well as the longitudinal orthogonal wheel from Mourioux *et al.* [15], the MY wheel designed by Ye and Ma [16], the MY wheel-II created by Ma *et al.* [17] and the ACROBAT drive system integrated by Inoue *et al.* [18]. These wheel mechanisms achieve active traction force in their primary motion path and passive movement in perpendicular directions through their free-rolling design.

Crenganiș [19] presented their works on Three-Wheeled Omnidirectional Mobile Robot (3WOMR) kinematic behaviour and dynamic properties.

The research investigated methods to control obstacle avoidance in the system. An obstacle avoidance system utilized a fuzzy controller which received comparison tests against documented methods from the available literature database. Multiple methods have been used to create control systems. The control methods of sliding-mode control and neural networks appeared in Alshorman *et al.* [20] and Li *et al.* [21] respectively.

At the same time Hadi [22] worked on this paper aimed at MR control modeling through PID fuzzy logic systems paired with sliding suppression algorithms development. This approach facilitates research into the dynamic characteristics of mobile robots with different configurations. The overall performance of the control system is evaluated, revealing several advantages of fuzzy logic controllers. The implementation of fuzzy logic proves simpler than traditional control theory because it omits complex mathematical modelling while using

Manuscript received April 9, 2025; revised April 23, 2025; accepted October 11, 2025; published January 23, 2026.

doi: 10.18178/ijmerr.15.1.65-79

control rules as its foundation. Additionally, fuzzy control is more resilient to changes in parameters. There are many control challenges in accelerators that involve phenomena difficult to model mathematically [22]. Al-Sahib and Jasim *et al.* [23] investigates the application of Fuzzy Logic Control (FLC) in the sonar systems of mobile robots. Fuzzy logic approaches provide substantial control to mobile robot navigation systems that function in partially known spaces critical for industrial and societal applications. Through fuzzy logic processing platforms, they solve the problem of handling multiple sonar sensor input data with their respective information capabilities.

Abiyev *et al.* [24] design and present a fuzzy controller system which will regulate omnidirectional robots. The paper thoroughly describes both mobile robot kinematic and dynamic principles as well as presenting a detailed explanation about its developed fuzzy controller algorithm. The algorithms have been developed for the purpose of monitoring both positional changes and rotational positioning in mobile robots.

Wondosen and Shieraw *et al.* [25] developed and simulated a mobile robot motion controller for navigating unknown dynamic outdoor spaces because this presents a non-linear system containing multiple control variables. A fuzzy logic control algorithm operates for both path planning and obstacle avoidance because it allows intermediate action calculation through rule averaging.

Key Features of the Project

- Using Arduino (slave) and Raspberry Pi 4 Model B as a master.
- Coding by python language and applied on raspberry Pi 4 as a master controller.
- Utilizing a Laser Range Scanner Sensor, commonly referred to as Lidar, for the purpose of obstacle detection.
- Design CAD and Operation Hybrid-Wheel Mobile Robot.
- Development of controller algorithms utilizing Fuzzy Logic Control (FLC) for the purpose of obstacle avoidance.

II. Material and Method of the 3WOMR

A. Mechanical System

1) Robot structure

The robot frame, crafted from stainless steel, is meticulously cut using a laser machine. The distance from center of robot to the omni wheel is 23 cm, as depicted in Fig. 1. The motor support for the robot frame and wheels is connected to the DC motors by Aluminum flanges as shown in Figs. 2 and 3. The. Finally, the assembled hybrid robot frames are shown in Fig. 4.

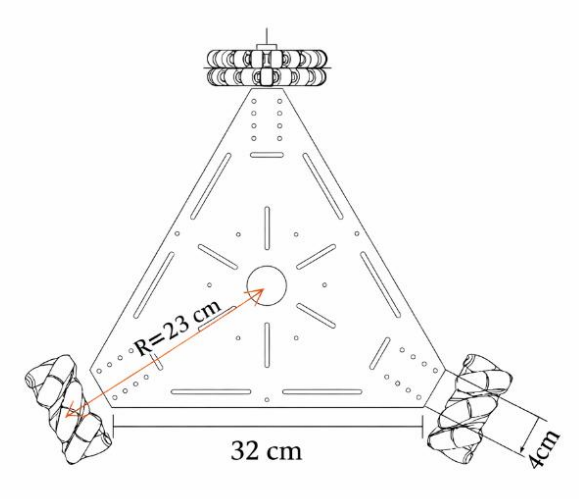


Fig. 1. Distance from robot center to wheel.

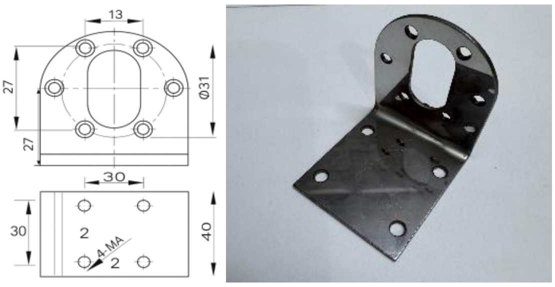


Fig. 2. Motor support for the robot's chassis.

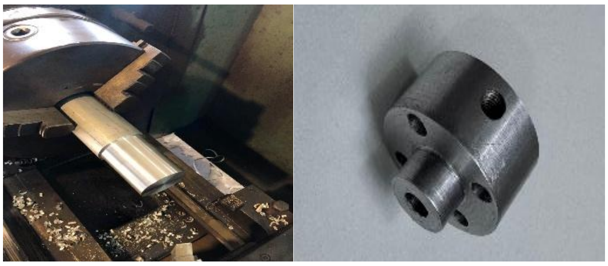


Fig. 3. Flange coupling Aluminium alloy.

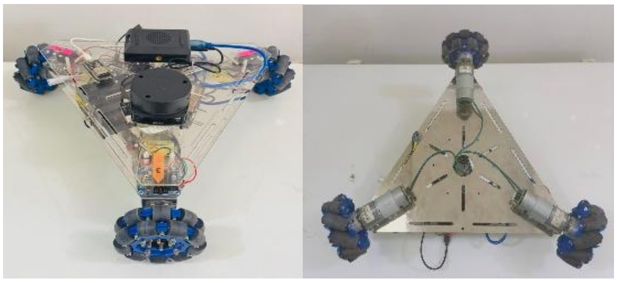
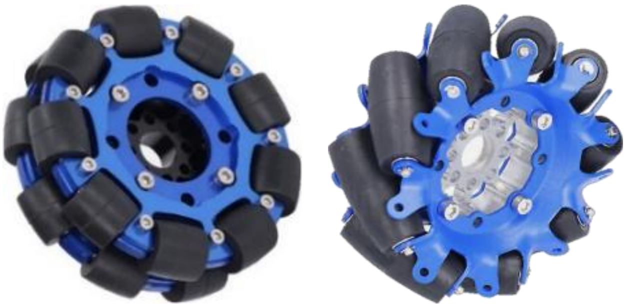


Fig. 4. Hybrid mobile robot viewed from above and below.

2) Omni and mecanum directional wheels

The omni wheel is distinguished by its capability to traverse independently in multiple directions [26]. This wheel typically rotates in a circular motion and can laterally navigate along its outer diameter through the application of a screw mechanism. The implementation of omnidirectional wheels facilitates the transition of robots from non-holonomic to holonomic configurations. In contrast to non-holonomic robots, which utilize conventional wheels and possess only two Degrees of Freedom (DOF), namely, forward and backward movement as well as rotation. holonomic omni wheels address this limitation by providing three DOF (x, y, θ). Consequently, A holonomic omni-directional robot can move freely in any direction while bypassing the need for wheel directional adjustment. These holonomic omni-directional wheels are capable of moving forward, backward, sliding laterally, and rotating while maintaining a fixed position. An illustration of the omni-directional wheels is presented in Fig. 5 and Table I parameters of wheel.



(a) (b)

Fig. 5. (a) Omni wheel (b) Mecanum wheel Configuration.

TABLE I. PARAMETER

Parameters	Description
Diameter	102 mm
Diameter	300 g, 297 g
width	39.5 mm, 39.75 mm
Body Material	Aluminium
Load Capacity single wheel	30 kg
Roller Material	Rubber
Number of Rollers per Wheel	9

3) DC motor JGB37-545B and encoder senser

A separate DC geared motor with encoder drives each wheel of the robot. The specifications for the 270:1 gearmotor, equipped with an encoder, operating at 12 V DC and having a rotational speed of 22 RPM, are provided in Table II. Fig. 6 shows the physical dimensions of the motor as it stands 76 mm long with a diameter of 37 mm.

TABLE II. TECHNICAL SPECIFICATIONS OF THE JGB37-545B DC MOTOR

Material	Description
Reduction Ratio	270:1
Rated Voltage	12 V
Voltage Rang	6–18 V
No-load speed	22 RPM
Load speed	19 RPM
Load Current	3 A
Torque	25 kg.cm
Weight	293 g

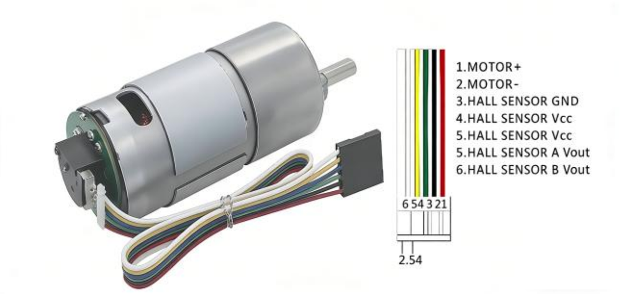


Fig. 6. Gearmotor with encoder DC 12V.

In WMR applications, sensors are used to tackle localization issues. This research incorporated a rotary encoder, shown in Fig. 7, with the DC motor on each wheel. A rotary encoder, an electromechanical device, detects the rotational position of a wheel by measuring changes in the magnetic field and converting these changes into electrical signals. It operates at a rate of 400 Pulses Per Revolution (PPR). The equation establishes how to calculate wheel velocity as follows:

$$V_w = \frac{2\pi}{400} \times \text{No.pulses persecond} \times \text{wheel raduis} \quad (1)$$

This encoder is an efficient and affordable option that provides two-channel and three-channel incremental magnetic encoding, making it suitable for challenging environments. It incorporates a magnetic grating and a magnetic-sensitive detection motor, generating two orthogonal square wave outputs with a 90° phase difference. Furthermore, the motor produces 16 pulse signals for each AB cycle.

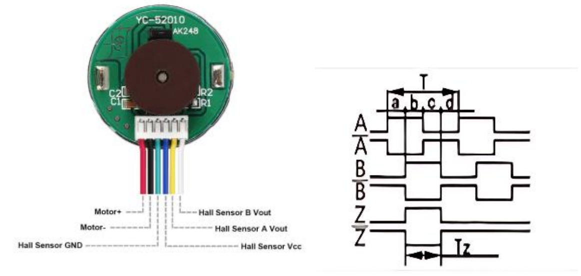


Fig. 7. Rotary encoder.

B. Electronic

1) Raspberry Pi 4

Research investigates the connection between Raspberry Pi 4 serving as the primary interface circuit and RP Lidar along with the Arduino Mega while using a 16 V power supply. The system operates with a 16 V power supply and engages in programming functions. The main controller operates with programming written in python language. The Raspberry Pi 4 includes a 40-pin GPIO port (General Purpose Input/Output) which enables control of several external devices through Fig. 8.

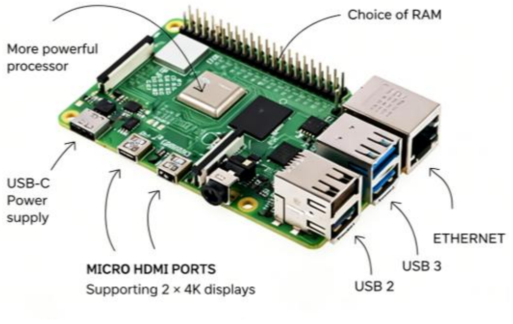


Fig. 8. Raspberry Pi 4 Model.

2) RPLiDAR A1M8-R6

Slamtec created the two-dimensional laser scanning device RPLiDAR A1M8-R6 with full 360-degree scanning ability reaching 12 m. The resulting 2D point cloud is versatile and can be used for mapping, localization, and modeling specific objects or environments. The device operates at a frequency of 5.5 Hz, enabling it to sample 360 points with each revolution. The laser measurement system employed in the RPLiDAR A1M8-R6 is based on triangulation and is suitable for use in environments without direct sunlight. The RP Lidar, integrated with a Raspberry Pi 4, enables environmental scans that provide valuable data for the Raspberry Pi. This functionality facilitates discussions aimed at overcoming obstacles in the surrounding environment as shown in Fig. 9.

Fig. 9. RPLiDAR A1M8-R6.

3) Arduino Mega 2560

In this research, Arduino functions as a sub-interface for the circuitry. It's an open-source electronic platform that enables programming using the C/CC+ language through its Integrated Development Environment (IDE). The device provides 54 digital I/O pins which include a PWM output on 15 pins and analog input through 16 pins and UART access through 4 pins. Users can access the 16 MHz crystal oscillator and USB connection and power jack and ICSP header and reset button functions on this device. This board serves as a crucial component in supporting control systems. In this study, the Arduino Mega 2560 microcontroller has been chosen, as illustrated in Fig. 10, for the implementation of the control algorithm associated with the 3WOMR, owing to its suitable performance characteristics Table III provides a comprehensive overview of the primary technical specifications associated with the Arduino Mega 2560 board.

TABLE III. SPECIFICATIONS OF THE ARDUINO MEGA 2560 MICROCONTROLLER BOARD

Microcontroller	ATmega2560
Operating voltage range	5 V
Recommended input voltage	7–12 V
Suggested input voltage	6–20 V
Number of Digital I/O pins	54
Number of analog input pins	16
DC current per I/O pin	40 mA
Direct current allowance for the 3.3 V pin	50 mA
EEPROM capacity	4 kB
CPU processing speed	16 MHz
Flash memory capacity	256 kB
Connection options	regular

Fig. 10. Arduino Mega 2560.

4) MPU9250 (HMC5 +MPU6050) 9Axis (attitude, gyro, magnetometer senser)

Fig. 11 illustrates a portion of the MPU-6050 sensor. The MPU-6050 is a Micro-Electromechanical System (MEMS) that ingeniously integrates both a gyroscope and an accelerometer. It generates output signals that can be conveniently read by a microcontroller. This sensor is conveniently configured using the I2C-Bus interface.

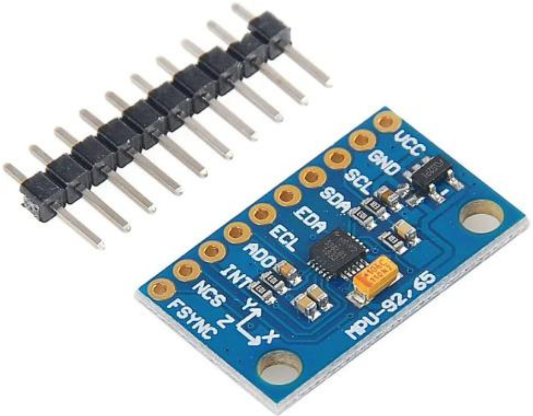


Fig. 11. MPU 9250.

III. ROBOT DESCRIPTION

The Three-Wheeled Omnidirectional Mobile Robot (3WOMR) is categorized as a holonomic mobile robot, characterized by its capability to execute simultaneous and independent movements in both translational and rotational dimensions. This robot is outfitted with three omni-wheels, which are symmetrically positioned at intervals of 120º along its circumference refer to Fig. 12. Each omnidirectional wheel and mecanum wheel are securely attached to its corresponding motor shaft, thereby establishing a shared rotational axis between the motor and the wheel. According to kinematics equation, the robot can achieve manoeuvres that are unattainable by conventional wheeled robots [2, 27–29].

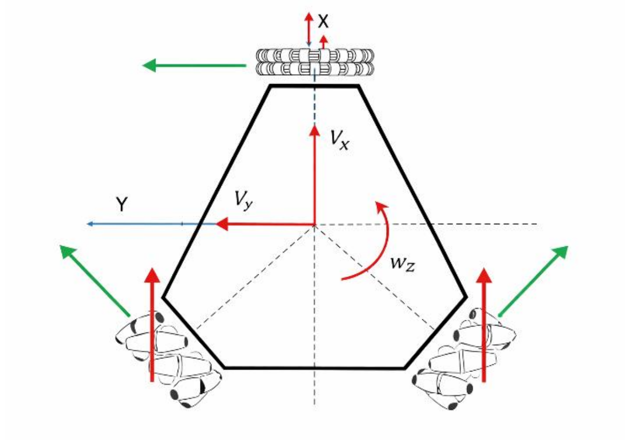
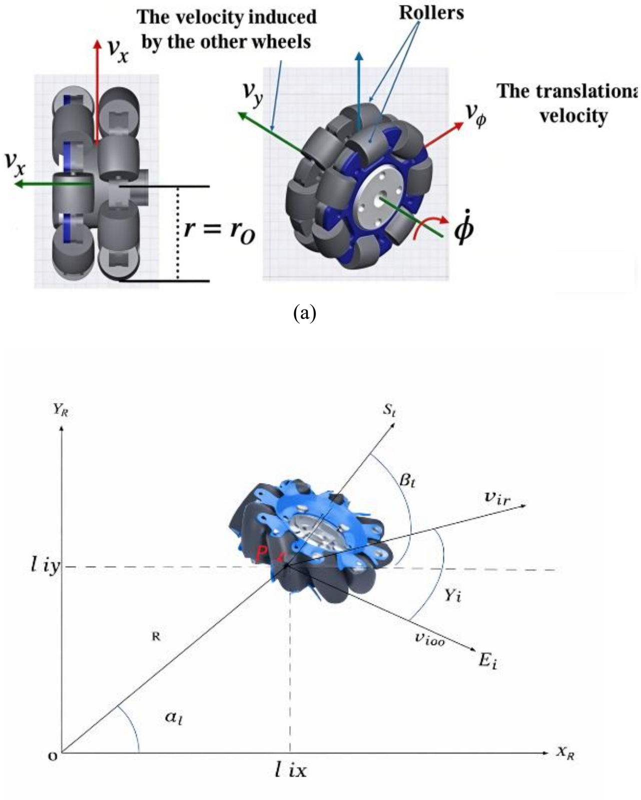


Fig. 12. Robot configuration.

As illustrated in Fig. 13, an omni-wheel and mecanum wheel is designed with multiple rollers, facilitating lateral movement that is perpendicular to the standard rolling direction. The functionality of an omnidirectional drive system necessitates the presence of a minimum of three omnidirectional wheels.



(a)

(b)

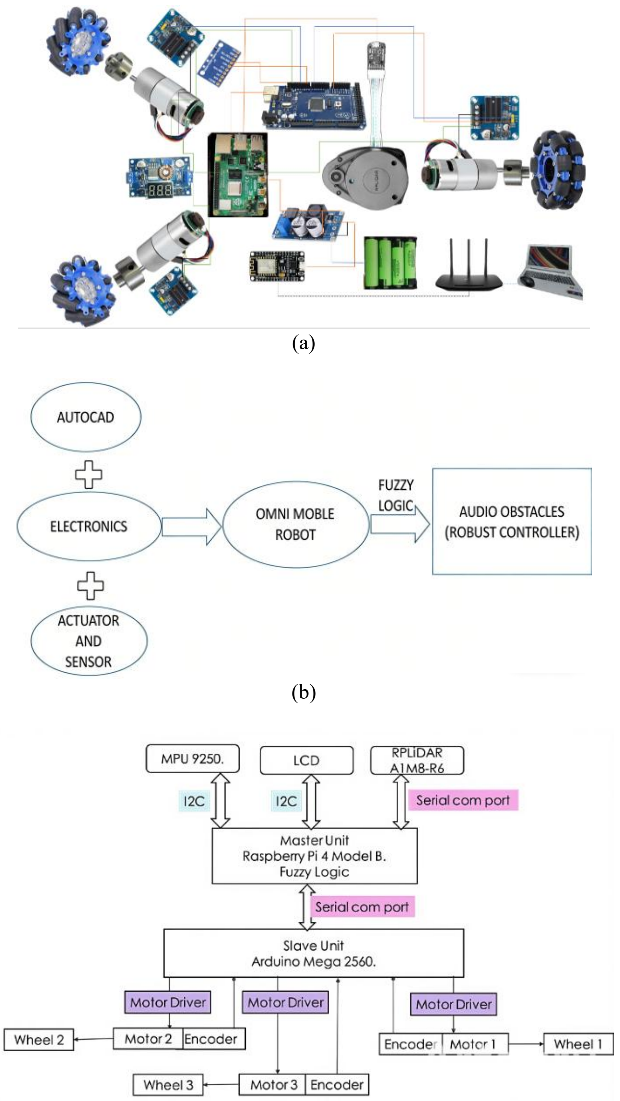
Fig. 13. Type wheel (a) Omni (b) mecanum wheel.

IV. Robot Assemble

In Figs. 14 and 15 illustrates the primary components of the robot along with their interconnections and schematic present work. Table IV enumerates all the components utilized in the assembly of this robot.

TABLE IV. List of Components Use on 3WOMR

Device	Quantity
Raspberry Pi 4 Model B	1
Omni wheel	1
Mecanum wheels	2
DC Driver 50 A	3
DC Motor JGB37 With encoder	3
Flange coupling	3
Arduino Mega 2560	1
RP Lidar A1M8-R6	1
Power Supple 16 V Step	4×4.2 ≈ 16 V
Step down 3 A	1
Step down 5 A	1
MPU-6050	1
Chassis From Stainless Steel	1



(a)

(b)

(c)

Fig. 14. Methodology mobile robot (a) Circuit diagram of a mechanical and electronic component connection of the purpose hybrid robot. (b) Schematic present work. (c) Block diagram for the proposed hybrid robot.

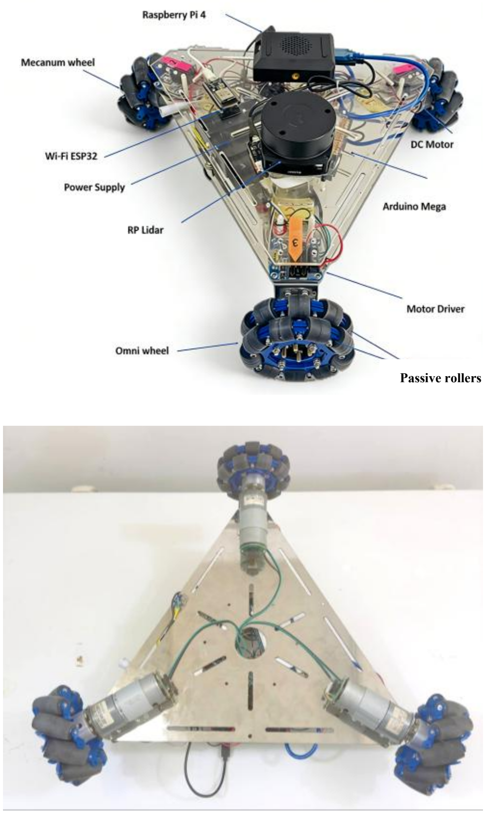


Fig. 15. Robot configuration with three omni wheels.

This study employs a Raspberry Pi 4 as the primary interface circuit, connected to an RP lidar sensor. The lidar sensor aids in identifying obstacles and guiding the robot's navigation. Furthermore, the Raspberry Pi communicates with an Arduino Mega via a cable, serving as a secondary interface. The Arduino receives data and commands from the main controller. Subsequently, the Arduino is connected to three DC drivers, each controlling one DC motor. The Arduino transmits information to the motors, enabling the robot to move to the desired position.

V. Forward and Inverse Velocity Jacobian Matrix

Fig. 16 shows arrangement of a robotic system equipped with three omnidirectional wheels. A robotic system utilizing two mecanum wheels and one omni wheel. Each wheel has three velocity components [30–34].

In Fig. 17, an omni-wheel and mecanum wheel is designed with multiple rollers, facilitating lateral movement that is perpendicular to the standard rolling direction. The functionality of an omnidirectional drive system necessitates the presence of a minimum of three omnidirectional wheels.

The system parameters and velocity specifications follow this definition:

(x, y): mobile robot's position

θ: orientation angle between X and X_R .

$X\,G\,Y$: global mobile robot's position

$X_R O Y_R$: the mobile robot's base frame refers to a Cartesian coordinate system that is associated to the movement of the robot's body canter.

$S_i P_i E_i$: coordinate of wheels of i th wheel and P_i represent wheel's canter point.

$\overrightarrow{OP_i}$: this vector represents the spatial distance between the canter of the robot and the canter of the i th wheel.

R : represent the distance between omnidirectional wheels and the base canter (centre of the robot O).

r_i: represent the distance from centre wheel's i to the roller center.

r_r: roller radius on the wheels.

α_i: the angle between $0P_i$ and X_R.

β_i: the angle between S_i and X_R.

γ_i: the angle between v_{ir} and E_i.

ω_i: wheels angular velocity $[\mathrm{rad/s}]$.

$v_{i\omega}$: represent the velocity of wheel revolutions $[\mathrm{m/s}]$.

v_{ir}: denotes the velocity of the passive roller of wheel i.

$v_x v_y[\mathrm{m/s}]$: Robot linear velocity.

$\omega_z[\mathrm{rad/s}]$: Robot angular velocity.

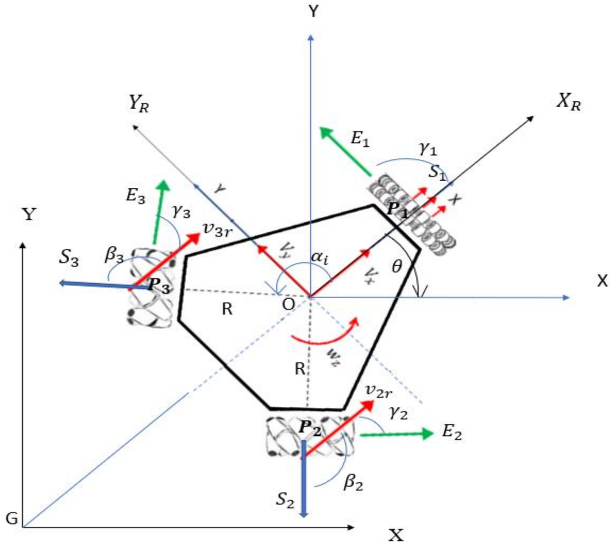
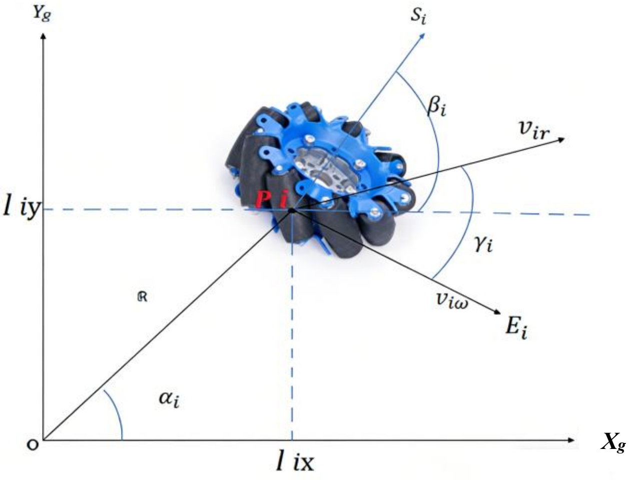
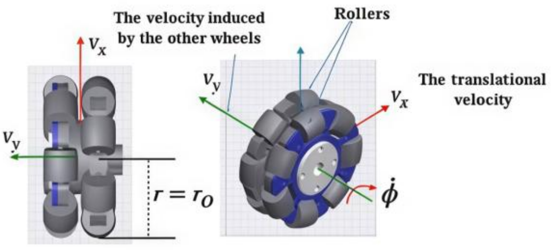


Fig. 16. Geometric structure of Velocity vector of wheel.



(a)


The velocity induced by the other wheels
Rollers
The translational velocity
v_x
v_y
v_y
v_x
$r = r_O$
$\dot{\phi}$

(b)

Fig. 17. Wheel in the robot coordinates (a) mecanum wheel (b) omni wheel.

The velocity of wheel i and tangential velocity of the free roller attached to wheel touch floor can be determined using Figs. 15 and 16.

$$v_{ir} = \frac{1}{cos\,45} r_r \omega_i \tag{2}$$

$$w_{Ei} = r_i \omega_i \text{ [4]}, i = 0, 1, 2, 3 \tag{3}$$

According to Fig. 18 and considering the Eq. (3), the velocity of the wheel i in the frame $S_i P_i E_i$, can be derived by:

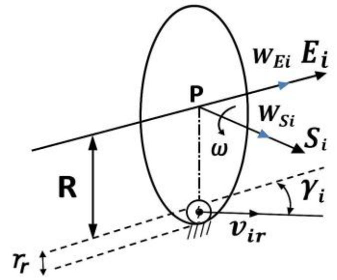


Fig. 18. wheel motion principle.

$$v_{S_i} = v_{ir}\, sin\, \gamma_i$$

$$v_{E_i} = \omega_i r_i + v_{ir}\, cos\, \gamma_i$$

$$\begin{bmatrix} v_{S_i} \\ v_{E_i} \end{bmatrix} = \begin{bmatrix} 0 & sin\,\gamma_i \\ r_i & cos\,\gamma_i \end{bmatrix} \begin{bmatrix} \omega_i \\ v_{ir} \end{bmatrix} = {}^{w_i}T_{P_i} \begin{bmatrix} \omega_i \\ v_{ir} \end{bmatrix} \tag{4}$$

The transformation of velocities from wheel i to its center point requires the following matrix:

$${}^{w_i}T_{P_i} = \begin{bmatrix} 0 & sin\ \gamma_i \\ r_i & cos\ \gamma_i \end{bmatrix} \tag{5}$$

The Eq. (6) provides the wheel center velocity measurement in the $X_R O Y_R$ coordinate system as depicted in Figs. 15 and 16.

$$\begin{bmatrix} v_{iX_R} \\ v_{iY_R} \end{bmatrix} = \begin{bmatrix} \cos\,\beta_i & -\sin\,\beta_i \\ \sin\,\beta_i & \cos\,\beta_i \end{bmatrix} \begin{bmatrix} v_{S_i} \\ v_{E_i} \end{bmatrix} = {}^{w_i}T_{P_i}\, {}^{P_i}T_R \begin{bmatrix} \omega_i \\ v_{ir} \end{bmatrix} \tag{6}$$

A transformation matrix connects the center of the i th wheel to the robot's coordinate system through utilization of the presented formulas in Eq. (7) [5].

$${}^{P_i}T_R = \begin{bmatrix} \cos\,\beta_i & -\sin\,\beta_i \\ \sin\,\beta_i & \cos\,\beta_i \end{bmatrix} \tag{7}$$

During planar motion the robot system contains:

$$\begin{bmatrix} v_{iX_R} \\ v_{iY_R} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -l_{iy} \\ 0 & 1 & l_{ix} \end{bmatrix} \begin{bmatrix} v_X \\ v_Y \\ \omega \end{bmatrix} = T' \begin{bmatrix} v_{X_R} \\ v_{Y_R} \\ \omega_R \end{bmatrix} \tag{8}$$

where:

$$T' = \begin{bmatrix} 1 & 0 & -l_{iy} \\ 0 & 1 & l_{ix} \end{bmatrix} \tag{9}$$

From Eqs. (5) and (7), the inverse kinematic model can be obtained:

$${}^{w_i}T_{P_i}\, {}^{P_i}T_R \begin{bmatrix} \omega_i \\ v_{ir} \end{bmatrix} = T' \begin{bmatrix} v_{X_R} \\ v_{Y_R} \\ \omega_z \end{bmatrix}, i = 0, 1, 2, 3 \tag{10}$$

The robot base velocity at point O can be derived from Eq. (11) using combined terms from Eqs. (6) and (8) under conditions of $r_i \neq 0, 0 < |\gamma_i| < \pi/2, \det\left({}^{P_i}T_R\right) \neq 0, \det\left({}^{w_i}T_{P_i}\right) \neq 0$

$$\begin{bmatrix} \omega_i \\ v_{ir} \end{bmatrix} = {}^{w_i}T_{P_i}{}^{-1} \cdot {}^{P_i}T_R{}^{-1} \cdot T' \begin{bmatrix} v_{X_R} \\ v_{Y_R} \\ \omega_z \end{bmatrix} \tag{11}$$

Eqs. (5) and (6) describe the relationship between the variables in the wheel frames of each robot and its center. By using inverse kinematics, the system's velocity can be determined by applying v_{ir} for the linear velocity and ω_i for the rotational speed of the i th wheel, as shown in Eq. (11), and conversely in Eq. (13).

$$\begin{bmatrix} v_{X_R} \\ v_{Y_R} \\ \omega_z \end{bmatrix} = T^+ \begin{bmatrix} \omega_i \\ v_{ir} \end{bmatrix} \tag{12}$$

$$\begin{bmatrix} \omega_i \\ v_{ir} \end{bmatrix} = T \begin{bmatrix} v_{X_R} \\ v_{Y_R} \\ \omega_R \end{bmatrix} \tag{13}$$

where:

$$T = {}^{w_i}T_{P_i}{}^{-1} \cdot {}^{P_i}T_R{}^{-1} \cdot T'$$

$$T^+ = (T^T T)^{-1} T^T$$

$$T = \begin{bmatrix} \cos\,\beta_i & -\sin\,\beta_i \\ \sin\,\beta_i & \cos\,\beta_i \end{bmatrix}^{-1} \cdot \begin{bmatrix} 0 & \sin\,\gamma_i \\ r_i & \cos\,\gamma_i \end{bmatrix}^{-1} \cdot \begin{bmatrix} 1 & 0 & -l_{iy} \\ 0 & 1 & l_{ix} \end{bmatrix}$$

Taking into account the relationships $l_{ix} = l_i \cos\ a_i$ and $l_{iy} = l_i \sin\ a_i$ and under the assumption that the wheels are of uniform size, the transformation matrix can be expressed as follows:

$$T := \frac{1}{-r} \begin{bmatrix} \frac{\cos\,(\beta_i - y_i)}{\sin\,(y_i)} & \frac{\sin\,(\beta_i - y_i)}{\sin\,(y_i)} & \frac{li\sin\,(-\alpha_i + \beta_i - y_i)}{\sin\,(y_i)} \\ -\frac{r\cos\,(\beta_i)}{\sin\,(y_i)} & -\frac{r\sin\,(\beta_i)}{\sin\,(y_i)} & -\frac{li\sin\,(-\alpha_i + \beta_i) r}{\sin\,(y_i)} \end{bmatrix} \tag{14}$$

$$T^{+} =$$

$$\frac{1}{{l_i}^2+1}\begin{bmatrix} -\frac{1}{2}\left({l_i}^2\sin\left(\beta_i\right) - {l_i}^2\sin\left(-\beta_i+2\alpha_i\right) + 2\sin\left(Bi\right)\right)r & \frac{1}{2}{l_i}^2\sin\left(y_i-\beta_i+2\alpha_i\right) - \frac{1}{2}\sin\left(-y_i+\beta_i\right){l_i}^2 - \sin\left(-y_i+\beta_i\right) \\ \frac{1}{2}r\left({l_i}^2\cos(\beta_i) - {l_i}^2\cos(-\beta_l+2\alpha_i) + 2\cos(\beta_i)\right) & -\frac{1}{2}{l_i}^2\cos\left(y_i-\beta_l+2\alpha_i\right) + \frac{1}{2}\cos\left(-y_i+\beta_l\right){l_i}^2 + \cos\left(-y_i+\beta_i\right) \\ \cos\left(\alpha_i-\beta_i\right)l_i r & \cos\left(\alpha_i-\beta_i+y_i\right)l_i \end{bmatrix} \quad (15)$$

The system's inverse kinematics can be obtained from Eq. (15), considering the relationship between the independent variables v_{ir} and ω_i for each joint, while assuming no wheel slipping occurs on the ground.

$$\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\\ \omega_4\end{bmatrix} =$$

$$\frac{-1}{r}\begin{bmatrix} \frac{\cos(\beta_1-\gamma_1)}{\sin\gamma_1} & \frac{\sin(\beta_1-\gamma_1)}{\sin\gamma_1} & \frac{l_1\sin\left(\beta_1-\gamma_1-\alpha_1\right)}{\sin\gamma_1} \\ \frac{\cos(\beta_2-\gamma_2)}{\sin\gamma_2} & \frac{\sin(\beta_2-\gamma_2)}{\sin\gamma_2} & \frac{l_2\sin\left(\beta_2-\gamma_2-\alpha_2\right)}{\sin\gamma_2} \\ \frac{\cos(\beta_3-\gamma_3)}{\sin\gamma_3} & \frac{\sin(\beta_3-\gamma_3)}{\sin\gamma_3} & \frac{l_3\sin\left(\beta_3-\gamma_3-\alpha_3\right)}{\sin\gamma_3} \\ \frac{\cos(\beta_4-\gamma_4)}{\sin\gamma_4} & \frac{\sin(\beta_4-\gamma_4)}{\sin\gamma_4} & \frac{l_4\sin\left(\beta_4-\gamma_4-\alpha_4\right)}{\sin\gamma_4} \end{bmatrix}\begin{bmatrix}v_X\\ v_Y\\ \omega_Z\end{bmatrix} \quad (16)$$

Eq. (17) shows the Jacobian matrix for the system's inverse kinematic:

$$T = \frac{-1}{r}\begin{bmatrix} \frac{\cos\left(\beta_1-\gamma_1\right)}{\sin\,\gamma_1} & \frac{\sin\left(\beta_1-\gamma_1\right)}{\sin\,\gamma_1} & \frac{l_1\sin\left(\beta_1-\gamma_1-\alpha_1\right)}{\sin\,\gamma_1} \\ \frac{\cos\left(\beta_2-\gamma_2\right)}{\sin\,\gamma_2} & \frac{\sin\left(\beta_2-\gamma_2\right)}{\sin\,\gamma_2} & \frac{l_2\sin\left(\beta_2-\gamma_2-\alpha_2\right)}{\sin\,\gamma_2} \\ \frac{\cos\left(\beta_3-\gamma_3\right)}{\sin\,\gamma_3} & \frac{\sin\left(\beta_3-\gamma_3\right)}{\sin\,\gamma_3} & \frac{l_3\sin\left(\beta_3-\gamma_3-\alpha_3\right)}{\sin\,\gamma_3} \\ \frac{\cos\left(\beta_4-\gamma_4\right)}{\sin\,\gamma_4} & \frac{\sin\left(\beta_4-\gamma_4\right)}{\sin\,\gamma_4} & \frac{l_4\sin\left(\beta_4-\gamma_4-\alpha_4\right)}{\sin\,\gamma_4} \end{bmatrix} \quad (17)$$

And for the forward kinematic according to the Eq. (12), we have:

$$\begin{bmatrix}v_X\\ v_Y\\ \omega_z\end{bmatrix} = T^{+}\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\end{bmatrix} \quad (18)$$

According to Figs. 19 and 20, ω_{Di} denotes the angular velocity of the wheel, while ω_{Mi} corresponds to the actual motion of the wheel.

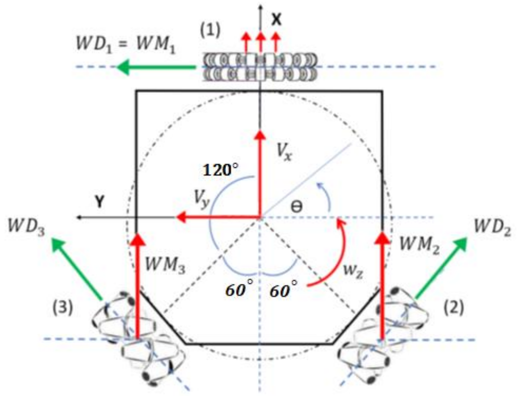


Fig. 19. Mathematical model.

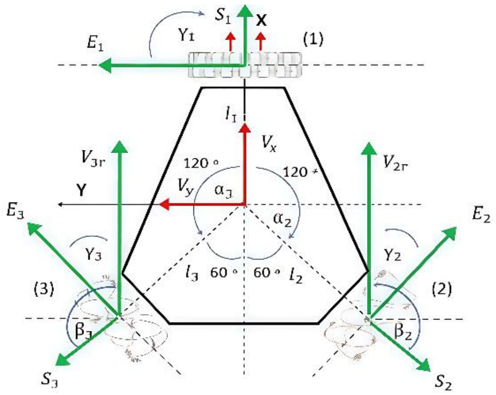


Fig. 20. Model of the robot.

Initially, the Jacobean matrix was reorganized for inverse kinematics utilizing the matrix presented in Eq. (17) for a platform equipped with two Mecanum wheels and a single omni wheel.

The variables $l_i = R$; i = 1;2;3 represent the distances from the robot's center to the center of the wheel as shown in Table V. Additionally, r denotes the radius measured in meters. The subsequent table presents the standard parameter values associated with the robot chassis.

TABLE V. COMMON PARAMETERS

Parameter	Value
R (cm)	23
R (cm)	5
M (kg)	3.260

ωD1, ωD2, and ωD3 represent the angular velocities of the wheels, measured in radians per second (rad/s). The subsequent Table VI presents the necessary parameters pertinent to both Mecanum and omni wheels [35].

TABLE VI. ROBOT PARAMETERS

Wheel	α_i	β_i	γ_i	R = l_i (cm)	r (cm)
1	0	0	−π/2	23	5
2	−2π/3	−2π/3	π/4	23	5
3	2π/3	2π/3	−π/4	23	5

All identified parameters demonstrate the correct positioning of wheels along with their attached secondary wheels. These parameters were then substituted into Eq. (12) to derive the inverse kinematics matrix.

$$\begin{bmatrix}\omega_{\text{D1}}\\ \omega_{D3}\\ \omega_{D3}\end{bmatrix} = \begin{bmatrix} 0 & 19.6 & 4.52 \\ 26.78 & 7.17 & 4.52 \\ -26.78 & 7.17 & 4.52 \end{bmatrix}\begin{bmatrix}v_x\\ v_y\\ \omega_z\end{bmatrix} \quad (19)$$

The derived matrix solves inverse kinematics equations, indicating how each wheel's velocity corresponds to the center's velocity command of the robot platform.

Specifically, wheel 1 is responsible for Y-directional motion, while wheels 2 and 3 contribute to X-directional front motion.

$$\omega_{D1} = 19.6 \times (v_y) + 4.52 \times (\omega_z) \quad (20)$$

$$\omega_{D2} = 26.78 \times (v_x) + 7.17 \times (v_y) + 4.52 \times (\omega_z) \quad (21)$$

$$\omega_{D3} = -26.78 \times (v_x) + 7.17 \times (v_y) + 4.52 \times (\omega_z) \quad (22)$$

The inverse kinematics model was employed to establish the wheel odometry of the robotic system. This model primarily facilitated the command of velocities to the robot platform through a FLC controller. The subsequent section presents the forward kinematics model for the system, derived by inverting the aforementioned matrix Eq. (19).

$$\begin{bmatrix} v_x \\ v_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} 0 & 0.01867 & 0.01867 \\ 0.08 & -0.04 & 0.04 \\ -0.12761 & 0.1744 & 0.1744 \end{bmatrix} \begin{bmatrix} \omega_{D1} \\ \omega_{D3} \\ \omega_{D3} \end{bmatrix} \quad (23)$$

VI. The Fuzzy Controller

A. General FLC

Fuzzy Logic Control (FLC), conceived by Zadeh in 1965 [36], is rooted in the foundational principles of Fuzzy Set Theory (FST). This theoretical framework effectively encapsulates the concept of uncertainty. A defining characteristic of FST is that the membership function (μ) assumes values within the interval of 0 to 1.

FLC uses linguistic variables instead of traditional mathematical equations to formulate control laws. These linguistic variables make it easier to define the operational characteristics of systems. Fig. 21 illustrates the structure of the FLC block diagram [37].

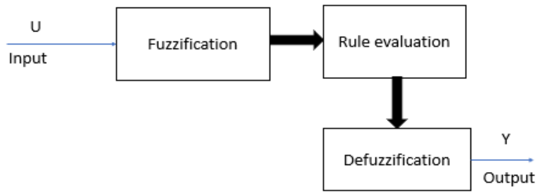


Fig. 21. Fuzzy logic layout controller [38].

Fuzzification involves the systematic conversion of inputs into corresponding fuzzy set memberships for a Fuzzy Logic Controller (FLC). This process is achieved through the application of Membership Functions (MFs), which operate within a range of 0 to 1. Various forms of MFs, such as triangular, bell-shaped, Gaussian, and trapezoidal, are illustrated in Fig. 22.

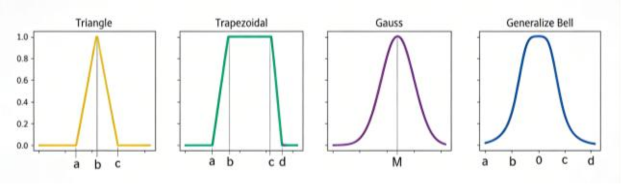


Fig. 22. Several types of Membership Functions (MFs) [37].

Within the framework of fuzzy logic controllers (FLC), two distinct types of fuzzy inference systems are utilized: the Mamdani and Sugeno models. These systems exhibit unique characteristics that influence the methodology employed for output computation. The rule structure inherent in the Mamdani fuzzy logic controller can be articulated as follows [37]:

When first input 1 represent *x* and second input 2 is *y* then the output represents *z*.

Sugeno FLC, the rules are of the form:

when first input 1 = *x* and second input 2 = *y*, then the output is $z = A_x + B_y + c$

Fig. 23 explains the Mamdani inference systems as below:

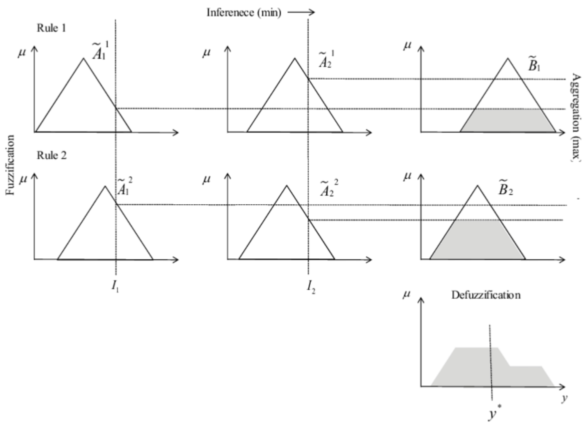


Fig. 23. Mamdani inference system [37].

The defuzzification process involves finding a numerical value from Fuzzy Set Theory (FST) that represents the set. It involves five different approaches. In this study, used the centre of gravity method. Its mathematical representation is as follows [38]:

$$COA = \frac{\int \mu_A(z) \cdot z d(z)}{\int \mu_A(z)\, d(z)} \quad (24)$$

B. Creating the FSI of the FLC

In the present work, the main purpose of using FLC for 3WOMR to obstacle avoidance in uncertainty environment with dynamic and static obstacle. The first step is to define a fl.mamfis object through the fuzzy lab library.

>>> fis = fl.mamfis()

The FLC controller received three input parameters including DH, DL as well as DR. A measurement of distance toward the head (DH) along with a measurement of distance towards the right (DR) and a measurement of distance to the left (DL) exists in Fig. 24 below.

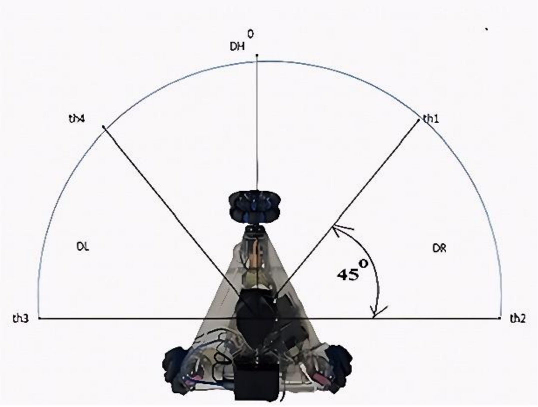


Fig. 24. Sets of sensor data readings.

The distances determined by the three respective groups are referred to as DH, DL, and DR.

$$\mathrm{DH} = \text{if}\ (th1 > angle \text{ or } angle > th4)$$

$$\mathrm{DL} = \text{if}\ (th4 > angle \geqslant th3)$$

$$\mathrm{DR} = \text{if}\ (th2 \geqslant angle > th1)$$

The controller uses measurements of distances DH, DL and DR to calculate angular velocity in the robot. To assess the spatial relationship between an object and the robot, it’s crucial to define the range within which the object’s position is expected to vary. The LiDAR sensor operates within a minimum range of 120 mm and a maximum range of 3000 mm. Subsequently, this data is utilized as input for the Fuzzy Inference System (FIS).

>>> $mind = 120$
>>> $maxd = 3000$
>>> $fis.addInput([minr, maxr], Name = '\ \ DH'\ \ )$

The membership functions associated with the fuzzy variable DH have been clearly defined. The research presented in reference [35] demonstrates the use of trapezoidal and triangular membership function pairs for setting both safe obstacle minimums and operational safety margins at higher speeds. The rules governing these functions were formulated through empirical trials, leading to the selection of those that yielded satisfactory movement in accordance with our evaluative criteria.

```
fis = addInput(fis,[0, 2500], Name='DH');
fis = addMF(fis,'DH','trapmf',[0,0,250,500],Name='N');% near
fis = addMF(fis,'DH','trimf',[250,750,1500],Name='M');% Mean
fis = addMF(fis,'DH','trapmf',[1000,1500,2500,2500],Name='F'); % Far
```

The terms N, M and F within the membership function correspond to the linguistic symbols near and medium and far. Fig. 25 shows the membership functions for the DH linguistic variable which was generated through plotmf function.
>>> plotmf(fis,'input',0)

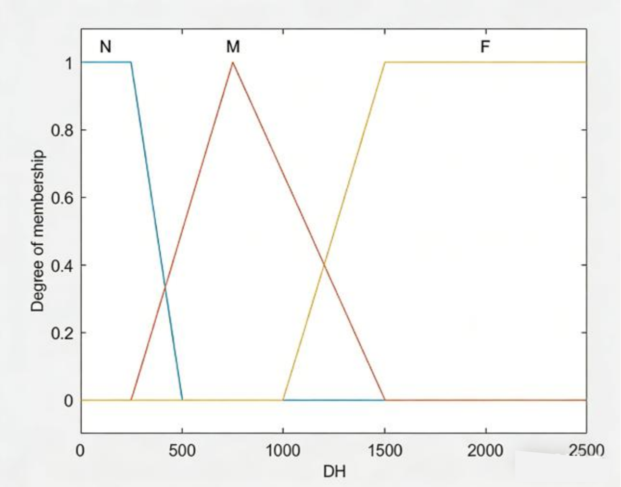


Fig. 25. The DH variable membership functions.

The elements DL and DR are included in a similar fashion as shown in Fig. 26 where their membership functions use equivalent parameters.

```
fis = addInput(fis,[0, 2500], Name='DL');
fis = addMF(fis,'DL','trapmf',[0,0,250,500],Name='N');% Near
fis = addMF(fis,'DL','trimf',[500,750,1000],Name='M');% Mean
fis = addMF(fis,'DL','trapmf',[750,1000,2500,2500],Name='F'); % Far

fis = addInput(fis,[0, 2500], Name='DR');
fis = addMF(fis,'DR','trapmf',[0,0,250,500],Name='N');% Near
fis = addMF(fis,'DR','trimf',[250,750,1500],Name='M');% Mean
fis = addMF(fis,'DR','trapmf',[500,1500,2500,2500],Name='F'); % Far
```

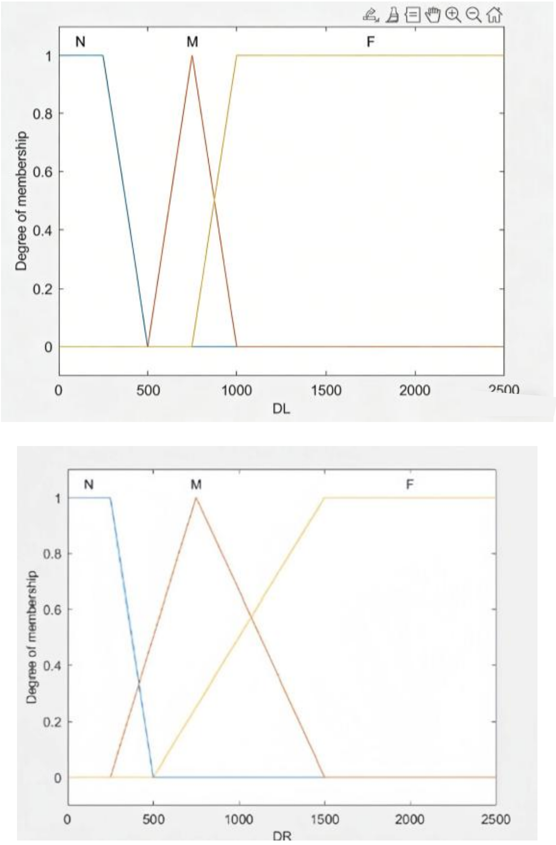
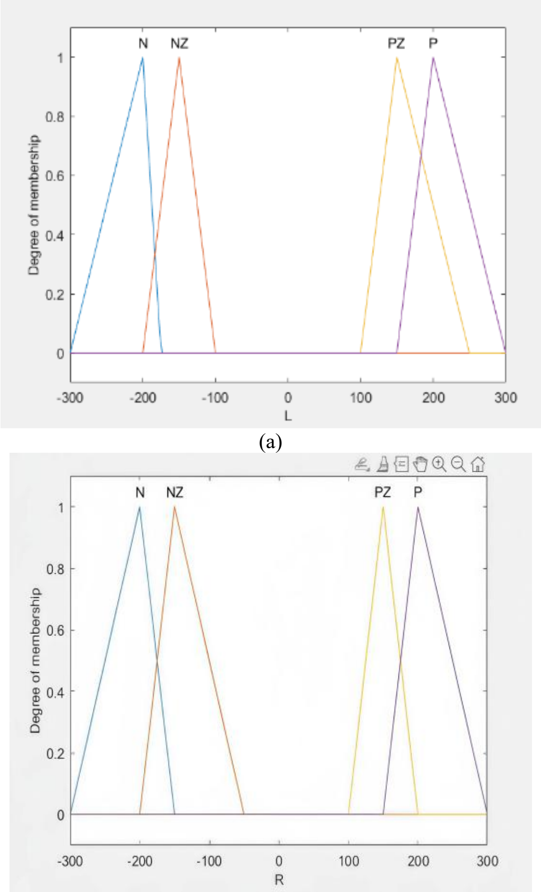


Fig. 26. Membership function for DL and DR variable.

To successfully execute the navigation task, it’s crucial that both the robotic system and the controlling mechanism are configured with specific parameters. For the robotic unit, adjustments were made to the angular velocity range, which is limited to $(-300 \lesssim \omega \lesssim 300)$. These parameters were derived from the instructional content of the 3WOMR Machine Learning course. Additionally, the Fuzzy Inference System (FIS) includes the angular velocity variable denoted as (Land R). The assessment of (Land R) results in classifications that are N (Negative Big), NZ (Negative), PZ (Positive), and P (Positive Big) as Fig. 27 demonstrates.

```
fis = addOutput(fis,[-300, 300], Name='L');
fis = addMF(fis,'L','trimf',[-300, -200, -175], Name='N');
fis = addMF(fis,'L','trimf',[-200, -150, -100], Name='NZ');
fis = addMF(fis,'L','trimf',[100, 150, 250], Name='PZ');
fis = addMF(fis,'L','trimf',[150, 200, 300], Name='P');

fis = addOutput(fis,[-300, 300], Name='R');
fis = addMF(fis,'R','trimf',[-300, -200, -150], Name='N');
fis = addMF(fis,'R','trimf',[-200, -150, -50], Name='NZ');
fis = addMF(fis,'R','trimf',[100, 150, 200], Name='PZ');
fis = addMF(fis,'R','trimf',[150, 200, 300], Name='P');
```



(a)

(b)

Fig. 27. Membership function: (a) represent (L) parameter and (b) represents (R) parameter.

The relation between FLC inputs and outputs becomes understandable through the fuzzy rules presented in Table VII below.

TABLE VII. (A), (B), (C) RULES LIST

(A)	(B)	(C)
1 1 1 1 1 1 1; % Rule 1	2 1 1 3 3 1 1; % Rule 1	3 1 1 3 3 1 1; % Rule 1
1 1 2 1 2 1 1; % Rule 2	2 1 2 4 3 1 1; % Rule 2	3 1 2 4 3 1 1; % Rule 2
1 1 3 1 2 1 1; % Rule 3	2 1 3 4 3 1 1; % Rule 3	3 1 3 4 3 1 1; % Rule 3
1 2 1 2 1 1 1; % Rule 4	2 2 1 3 4 1 1; % Rule 4	3 2 1 3 4 1 1; % Rule 4
1 2 2 1 1 1 1; % Rule 5	2 2 2 3 3 1 1; % Rule 5	3 2 2 4 4 1 1; % Rule 5
1 2 3 1 1 1 1; % Rule 6	2 2 3 3 3 1 1; % Rule 6	3 2 3 4 4 1 1; % Rule 6
1 3 1 2 1 1 1; % Rule 7	2 3 1 3 3 1 1; % Rule 7	3 3 1 3 4 1 1; % Rule 7
1 3 2 1 1 1 1; % Rule 8	2 3 2 3 3 1 1; % Rule 8	3 3 2 4 4 1 1; % Rule 8
1 3 3 1 1 1 1; % Rule 9	2 3 3 3 3 1 1; % Rule 9	3 3 3 4 4 1 1; % Rule 9

In Fig. 28, it’s evident that the formulation of rules for the Fuzzy Inference System (FIS) requires specialized knowledge. This expertise enables the precise determination of angular velocity based on various input values. The robot can encounter 27 distinct perceptual scenarios, categorized into three groups of inputs, each represented by three linguistic values. Each scenario corresponds to a specific response, articulated through 27 fundamental rules, as illustrated in Fig. 29 and detailed in Table VIII. Fig. 30 represents overview of the fuzzy model's surface.

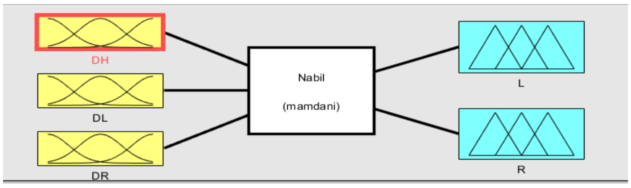
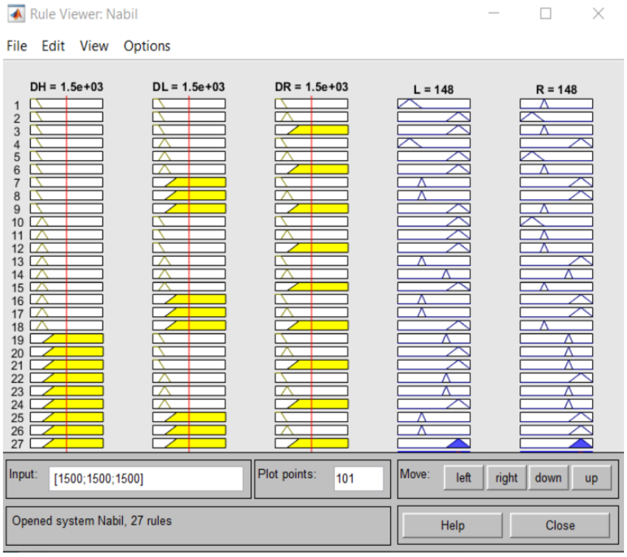


Fig. 28. Fuzzy Inference System (FIS) control.



Input: [1500;1500;1500]

Plot points: 101

Opened system Nabil, 27 rules

Fig. 29. Fuzzy rules of the FIS.

TABLE VIII. FUZZY RULES

Rules number	DH	DL	DR	L	R
1	N	N	N	N	N
2	N	N	M	PZ	NZ
3	N	N	F	PZ	N
4	N	M	N	NZ	PZ
5	N	M	M	P	NZ
6	N	M	F	P	N
7	N	F	N	N	PZ
8	N	F	M	NZ	P
9	N	F	F	PZ	N
10	M	N	N	PZ	PZ
11	M	N	M	PZ	PZ
12	M	N	F	P	PZ
13	M	M	N	PZ	PZ
14	M	M	M	PZ	PZ
15	M	M	F	PZ	PZ
16	M	F	N	PZ	P
17	M	F	M	PZ	PZ
18	M	F	F	P	P
19	F	N	N	PZ	PZ
20	F	N	M	PZ	PZ
21	F	N	F	PZ	PZ
22	F	M	N	PZ	PZ
23	F	M	M	PZ	PZ
24	F	M	F	PZ	PZ
25	F	F	N	P	PZ
26	F	F	M	PZ	PZ
27	F	F	F	P	P

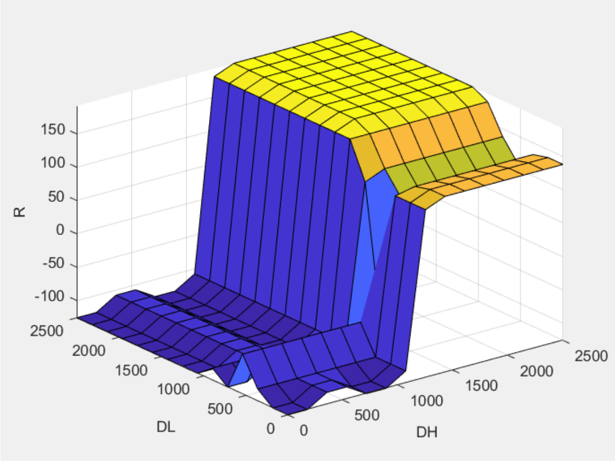


Fig. 30. Overview of the fuzzy model's surface.

The first three columns specify an input membership function that defines a particular input while the following two columns represent an output membership function linked to left and right wheel angular velocities. Each rule contains information about its weight in the penultimate column followed by an identification of the used fuzzy operator expressed as value "1" representing the "and" operator. The rules are incorporated into the Fuzzy Inference System (FIS) using the method addRule:

fis.addRule(ruleList).

Subsequently to the establishment of a Fuzzy Inference System (FIS) utilizing the fuzzy lab library, the eval fis function can be utilized to evaluate the variables DH, DL, and DR in accordance with the readings obtained from the sensors.

C. *Pose Update*

These equations are the pose update kinematics for a mobile robot. They take the robot's local forward/sideways velocities and rotation rate, and update its global position (x, y) and orientation θ. The Mean Square Error (MSE) refers to positional error remained below 0.06 m.

$$\dot{x}= v_x\cos\,\theta - v_y\sin\,\theta \tag{25}$$

$$\dot{y} = v_x\sin\,\theta + v_y\cos\,\theta \tag{26}$$

$$\dot{\theta} = \omega \tag{27}$$

Equations (simple time integral Δt) to update the global position (x, y, θ) from local velocities V_x, V_y and angular velocity ω:

$$x_{k+1} = x_k + \left(v_x\cos\,\theta_k - v_y\sin\,\theta_k\right)\Delta t \tag{28}$$

$$y_k = y_k + \left(v_x\sin\,\theta_k + v_y\cos\,\theta_k\right)\Delta t \tag{29}$$

$$\theta_{k+1} = \theta_k + \omega_k\Delta t \tag{30}$$

VII. RESUTS AND DISCUSSION

The Light Detection and Ranging (LiDAR) system as show in Fig. 31, mounted on the upper part of the robotic apparatus, functions as a two-dimensional scanning mechanism that collects environmental data within a designated scanning range. It's crucial to understand that this two-dimensional LiDAR is limited to scanning a single plane, making obstacles below its installation height invisible. Therefore, the height of the LiDAR installation should be carefully considered based on the specific operational environment.

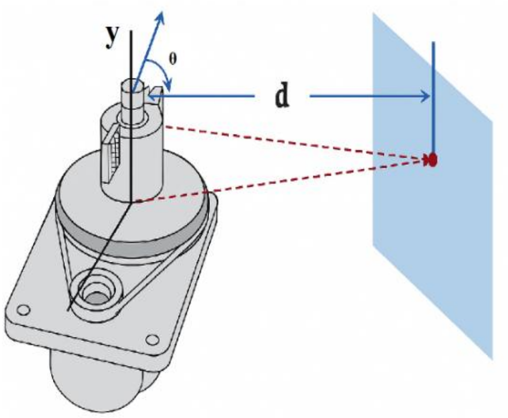


Fig. 31. LiDAR and obstacle.

During each scanning cycle, the LiDAR conducts a comprehensive 360° examination of its surroundings, generating Np data points, commonly known as a laser point cloud. These data points can be mathematically expressed as (dt_i , t_i), where dt_i and t_i, are centered on the robot, with the index i ranging from 1 to Np at a specific time t. Fig. 32 illustrates the original data acquired by the LiDAR in polar coordinates, with red dots indicating detected obstacles, R representing the scanning radius of the LiDAR, and the white triangle representing the robot's position.

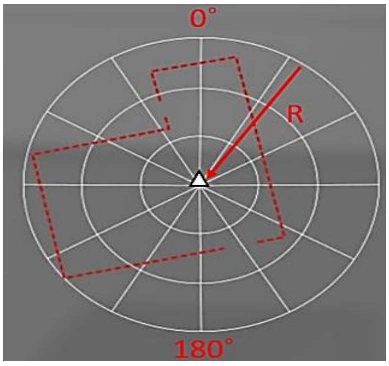


Fig. 32. Original data of LiDAR.

The operational space contained four obstacles with radius 0.15 m purposefully arranged around the robot. The four obstacles received specific placement positions near the edge of the robot platform, the case scenario requires avoidance of four stationary obstacles as shown in Fig. 33 and Table IX presents all the required parameters for the static obstacles and robot Start Point (SP) from (50,50) cm to End Point (EP) (50,450) cm. Under experiment conditions the hybrid mobile robot required 25.44 s to reach its target whereas the three omni wheel robot needed 48.5 s.

TABLE IX. Position and the Radius of the Static Obstacles

Obs. Number	Radius (cm)	Position (x, y) cm
1	15	50,150
2	15	150,150
3	15	150,250
4	15	50,450

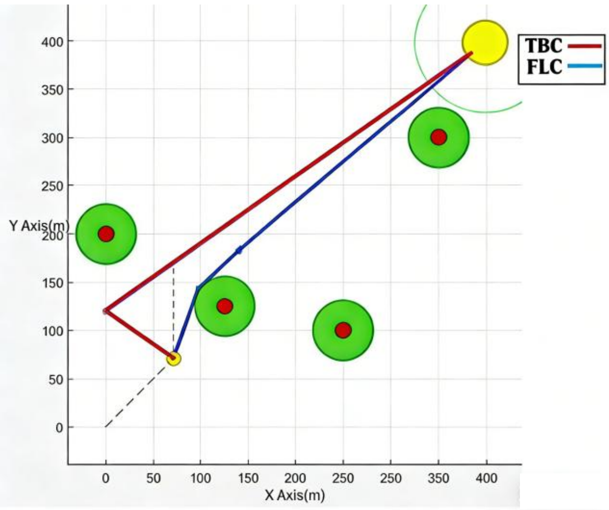


Fig. 33. Robot and Graphical User Interface (GUI) with four obstacles

In the specified scanning area of 3×4.5 m², as shown in Fig. 34, each obstacle can be represented of as a collection of distinct red points. The presence of obstacles leads to the creation of multiple point sets that effectively encompass these obstacles. In contrast, when obstacles are absent, no data points are generated. The direct representation of the environment using the original LiDAR data is impractical because the data points are discrete. A single point does not indicate the presence of an obstacle; rather, it is the collective aggregation of numerous data points that represent this information about obstacles location in environment. A substantial body of literature has focused on clustering techniques suitable for point cloud data. This study aims to compute collision-free trajectories for a mobile robot operating in environments with either dynamic or static obstacles. The comparison outcomes for ABC and FLC algorithms appear in Table X based on the performance Fig. 33 in MATLAB program.

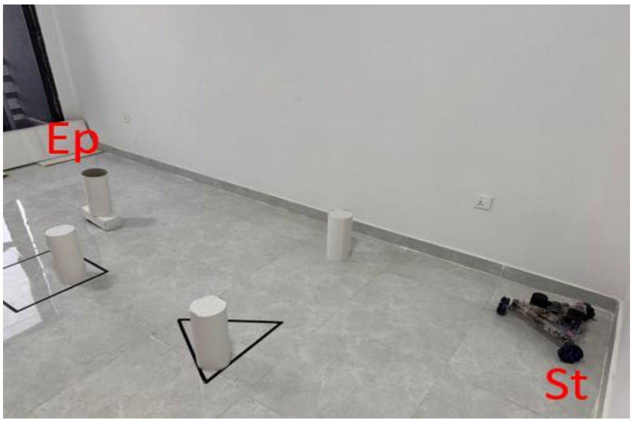


Fig. 34. Robot and real environment with four obstacles.

TABLE X. Comparison between ABC and FLC Algorithms

Obstacles Avoidance Algorithm	Path Length from SP to TP	Traveling time (s)	Via Point (x, y) cm	No. of turns

ABC (RED LINE)	574.7	33.634	P1(88,148)
			P2(175,21)
				8

FLC (GREEN LINE)	536.515	25.44	P1(88,148)
			P2(275,27)
				6

At the end, Table XI highlighted the Mean Square Error (MSE) for each state error component and Table XII Performance characteristics between FLC and PID as well as trajectory comparison plot as shown in Fig. 35.

$$(q_d - q) = (e_X, e_Y, e_\theta) \tag{31}$$

TABLE XI. MSE of for Two Controllers

Controller Methodology	MSE		
	e_x (m)	e_y (m)	e_θ (rad)
PID	8.055 e−02	7.782 e−02	4.904 e−03
FLC	3.344 e−02	6.798 e−02	3.041 e−03

TABLE XII. Performance Characteristics between FLC and PID

Controller Methodology	Rise Time	Setting Time (s)	Steady state error	Max.overshoot
PID	0.45	2.35	0.01	0.77
FLC	0.36	2.01	0.02	0.71

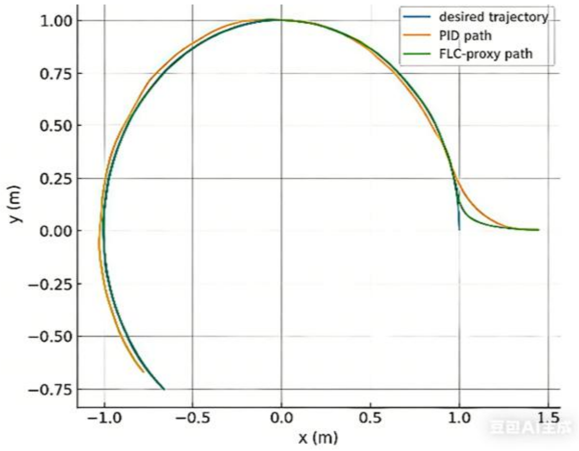


Fig. 35. Trajectory comparison plots (FLC vs. PID).

VIII. Conclusion and future Work

This study developed a fuzzy logic controller (FLC) for mobile robot obstacle avoidance using LiDAR sensing. The controller, designed with 27 fuzzy rules and three input distance parameters, demonstrated reliable navigation in both simple and cluttered environments. Simulation results revealed that the proposed fuzzy system achieved a 7% shorter path and a 9% faster travel time compared to the ABC method, confirming the effectiveness of fuzzy logic in managing nonlinear robotic systems and uncertainties in real-world navigation.

Building on these results, two practical configurations are proposed for future development:

1. FLC (Reactive) + PID (Inner-loop) for precision
   - The FLC acts as a decision-making planner that considers environmental constraints such as obstacle avoidance and high-level velocity references.
   - The PID controller, operating in the inner loop of speed and position, tracks the reference commands and compensates for disturbances caused by model inaccuracies.
   - This configuration is suitable when simplicity and low computational demand are prioritized.
2. FLC (Navigation) + MPC (Precision & Constraints)
   - Here, the FLC functions as a fast planner, providing a short planning horizon for the Model Predictive Control (MPC) module.
   - The MPC then solves a tightly constrained optimization problem, ensuring precise motion and robust constraint handling.
   - This structure is ideal for applications that demand high accuracy and optimal performance under complex conditions.

Future work may also include extending the controller for dynamic obstacle avoidance, integrating optimization algorithms to enhance fuzzy rule tuning, and incorporating velocity and trajectory control for improved real-time navigation in dynamic environments.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Ali A. Tuama. is the first author—methodology, correspondent, wring original draft, visualization, validation, configured. Nabil H. Hadi is review and editing, methodology. All authors had approved the final version. All authors had approved the final version

References

[1] G. M. Ramalho, S. R. Carvalho, E. C. Finardi, and U. F. Moreno, "Trajectory optimization using sequential convex programming with collision avoidance," *Journal of Control, Automation and Electrical Systems*, vol. 29, no. 3, pp. 318–327, 2018.

[2] Í. B. Viana, D. A. Santos, L. C. S. Góes, and I. A. A. Prado, "Distributed formation flight control of multirotor helicopters," *Journal of Control, Automation and Electrical Systems*, vol. 28, no. 4, pp. 502–515, 2017.

[3] S. Pezeshki, A. R. Ghiasi, M. A. Badamchizadeh, and K. Sabahi, "Adaptive robust control of autonomous underwater vehicle," *Journal of Control, Automation and Electrical Systems*, vol. 27, no. 3, pp. 250–262, 2016.

[4] B. S. Aleshin, V. N. Maksimov, V. V. Mikheev, and A. I. Chernomorskii, "Horizontal stabilization of the two-degree-of-freedom platform of a uniaxial wheeled module tracking a given trajectory over an underlying surface," *Journal of Computer and Systems Sciences International*, vol. 56, no. 3, pp. 471–482, 2017.

[5] A. Al-Mayyahi, W. Wang, and P. Birch, "Design of fractional order controller for trajectory tracking control of a non-holonomic autonomous ground vehicle," *Journal of Control, Automation and Electrical Systems*, vol. 27, no. 1, pp. 29–42, 2016.

[6] T. Taniguchi and M. Sugeno, "Trajectory tracking controller design for a tricycle robot using piecewise multi-linear models," in *Proc. International Multi Conference of Engineers and Computer Scientists (IMECS)*, Hong Kong, 2017.

[7] D. A. de Lima and G. A. S. Pereira, "Navigation of an autonomous car using vector fields and the dynamic window approach," *Journal of Control, Automation and Electrical Systems*, vol. 24, no. 1–2, pp. 106–116, 2013.

[8] J. Moreno *et al.*, "Design, implementation and validation of the three-wheel holonomic motion system of the Assistant Personal Robot (APR)," *Sensors*, vol. 16, no. 10, 1658, 2016.

[9] A. S. Kundua, O. Mazumdera, P. K. Lenkab, and S. Bhaumik, "Omnidirectional assistive wheelchair: Design and control with isometric myoelectric based intention classification," *Procedia Computer Science*, vol. 105, pp. 68–74, 2017.

[10] J. Qian, B. Zi, D. Wang, Y. Ma, and D. Zhang, "The design and development of an omni-directional mobile robot oriented to an intelligent manufacturing system," *Sensors*, vol. 17, no. 9, 2073, 2017.

[11] A. Aribowo, A. S. Putra, S. Lukas, and H. Tjahyadi, "Enhancing soccer robot movement accuracy using omnidirectional wheel," in *Proc. International Conference on Electrical Engineering and Informatics (ICELTICs)*, Banda Aceh, Indonesia, IEEE, 2017, pp. 119–123.

[12] C. M. Townsend, "Ball castors," Patent CH357524, Switzerland, Dec. 12, 1958.

[13] B. E. Ilon, "Wheels for a course stable self-propelling vehicle movable in any desired direction on the ground or some other base," U.S. Patent 3876255, Apr. 8, 1975.

[14] M. West and H. Asada, "Design of ball wheel mechanisms for omnidirectional vehicles with full mobility and invariant kinematics," *Journal of Mechanical Design*, vol. 119, no. 1, pp. 153–161, 1997.

[15] G. Mourioux, C. Novales, G. Poisson, and P. Vieyres, "Omnidirectional robot with spherical orthogonal wheels: Concepts and analyses," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Orlando, FL, USA, IEEE, 2006, pp. 3374–3379.

[16] C. Ye and S. Ma, "Development of an omnidirectional mobile platform," in *Proc. IEEE International Conference on Mechatronics and Automation (ICMA)*, Changchun, China, IEEE, 2009, pp. 1111–1115.

[17] S. Ma, C. Ren, and C. Ye, "An omnidirectional mobile robot: Concept and analysis," in *Proc. IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Guangzhou, China, IEEE, 2012, pp. 920–925.

[18] Y. Inoue, T. Hirama, and M. Wada, "Design of omnidirectional mobile robots with ACROBAT wheel mechanisms," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, IEEE, 2013, pp. 4852–4859.

[19] M. Crenganiș, R.-E. Breaz, and S.-G. Racz, "Fuzzy logic-based driving decision for an omnidirectional mobile robot using a Simulink dynamic model," *Applied Sciences*, vol. 14, no. 8, 3058, 2024.

[20] A. M. Alshorman *et al.*, "Sliding mode control for a class of nonlinear systems with application to a wheeled mobile robot," *Machines*, vol. 8, no. 3, 55, 2020.

[21] Z. Li, Y. Wang, X. Song, and Z. Liu, "Neural adaptive tracking control for wheeled mobile robots," in *Proc. International Forum on Mechanical, Control and Automation (FPM)*, Harbin, China, 2015.

[22] N. H. Hadi, "Fuzzy control of mobile robot in slippery environment," *Al-Khwarizmi Engineering Journal*, vol. 1, no. 2, pp. 41–51.

[23] N. K. A. Al-Sahib and A. R. Jasim, "Guiding mobile robot by applying fuzzy approach on sonar sensors," *Al-Khwarizmi Engineering Journal*, vol. 6, no. 3, pp. 36–44, 2010.

[24] R. H. Abiyev, I. S. Günsel, N. Akkaya, and E. Aytac, "Fuzzy control of omnidirectional robot," *Procedia Computer Science*, January 2017.

[25] A. Wondosen and D. Shieraw, "Fuzzy logic controller design for mobile robot outdoor," arXiv preprint, arXiv:2401.01756v3, 2024.

[26] M. K. S. H. Maldeniya *et al.*, "Remote controlled 4WD omnidirectional robot using mecanum wheels," in *Proc. International Research Symposium on Engineering Advancements (RSEA 2015)*, SAITM, Malabe, Sri Lanka, 2015.

[27] K. Watanabe *et al.*, "Feedback control of an omnidirectional autonomous platform for mobile service robots," *Journal of Intelligent and Robotic Systems*, vol. 22, no. 3, pp. 315–330, 1998.

[28] F. G. Pin and S. M. Killough, "A new family of omnidirectional and holonomic wheeled platforms for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 4, pp. 480–489, 1994.

[29] R. Rojas, "Omnidirectional control," Freie University Berlin, May 2005.

[30] C. Wang *et al.*, "Trajectory tracking of an omni-directional wheeled mobile robot using a model predictive control strategy," *Applied Sciences*, vol. 8, no. 2, 231, 2018.

[31] D. S. Alhanov and V. I. Rubtsov, "Modeling of a mobile robot on Mecanum wheels kinematics," *ITM Web of Conferences*, vol. 35, 04001, 2020.

[32] Y. Li, S. Dai, L. Zhao, X. Yan, and Y. Shi, "Topological design methods for Mecanum wheel configurations of an omnidirectional mobile robot," *Symmetry*, vol. 11, no. 10, 1268, 2019.

[33] H. M. Alwan, V. A. Nikolaevic, and S. F. Hasan, "Kinematic and dynamic modeling based on trajectory tracking control of mobile robot with Mecanum wheels," *FME Transactions*, vol. 51, no. 2, pp. 140–148, 2023.

[34] R. M. Hussein, "Design a new hybrid controller based on an improvement version of Grey Wolf Optimization for trajectory tracking of wheeled mobile robot," *FME Transactions*, vol. 51, pp. 140–148, 2023.

[35] M. Massoud, A. A. H. Ibrahim, and M. Atia, "Mechatronic design and path planning optimization for an omni wheeled mobile robot for indoor applications," in *Proc. 2021 31st International Conference on Computer Theory and Applications (ICCTA)*, December 2021.

[36] L. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

[37] N. Hacene and B. Mendil, "Motion analysis and control of three-wheeled omnidirectional mobile robot," *SBA*, 2019.

[38] T. L. Phuong and K. L. Khai, "Self-developed three wheels omni-directional for autonomous mobile robots," in *Proc. the 3rd International Conference on Machine Learning and Soft Computing*, 2019.