Engineering and Technology Journal
Journal homepage: https://etj.uotechnology.edu.iq

Experimental investigation of applying multi-stage (PID-MRAC) controller for trajectory tracking of four mecanum wheeled mobile robot

Asaad D. Abdullsahib[a,b*], **Hassan M. Alwan**[b]

[a] Mechanical Engineering Dept., College of Engineering, University of Al-Qadisiyah, Al-Qadisiyah, Iraq.
[b] Mechanical Engineering Dept., University of Technology-Iraq, Alsina'a street, 10066 Baghdad, Iraq.
*Corresponding author Email: me.20.28@grad.uotechnology.edu.iq

HIGHLIGHTS

- A multi-stage controller combining MRAC and PID was designed to control the trajectory tracking of a 4-mecanum wheel robot
- A mobile robot with unique holders was manufactured to transport logistics items
- MATLAB/Simulink was programmed and connected to Arduino to send control commands and receive feedback
- The robot's performance was evaluated using the designed control algorithm

Keywords:

Four mecanum wheels mobile robot

Model reference adaptive control

PID

Trajectory tracking; Mobile robot control

ABSTRACT

This paper presents a practical investigation into the design and implementation of a multi-stage controller, which integrates a Model Reference Adaptive Controller (MRAC) with a Proportional-Integral-Derivative (PID) controller for trajectory tracking control in a Four Mecanum Wheels Mobile Robot (FMWMR). The MRAC adaptively updates the control parameters based on the difference between the actual and desired behaviors of the mobile robot. In contrast, the PID controller manages the angular velocities of the wheels. The mobile robot has been designed and constructed with two holders to facilitate the transportation of logistical items and constrain the weight to the center of the mobile robot to reduce the effect of the inertia force. The distinctive motion characteristics of the FMWMR are elucidated by explaining its kinematic and dynamic models. The control signals for the FMWMR motors are derived from the output of the multi-stage controller. The mobile robot operates through serial communication between MATLAB/Simulink and Arduino hardware. Experimental results demonstrating the control of the (FMWMR) by the proposed controller as it follows a Sine-Spiral trajectory are presented, highlighting the performance of the multi-stage controller monitoring the behavior of velocities and motor torques and illustrating the real-time adaptive variation of the MRAC control parameters.

1. Introduction

Mobile robot exploration, including both traditional and omnidirectional types, has experienced a significant rise in recent years, particularly in complex environments. Omnidirectional wheeled robots, namely those equipped with four Mecanum wheels, have attracted interest because of their unique benefits, including great mobility, the capacity to turn on the spot, and effective navigation in tight places [1, 2]. These robots are used in a wide range of fields, such as home automation, industrial environments, assistive equipment such as omniwheel chairs, manipulators, and healthcare solutions. Mecanum wheels have a distinctive structure with inclined rollers that enable sideways and curving motions while keeping the orientation centered [3]. Nevertheless, the management of movement becomes intricate as a result of the existence of four distinct motors, which introduces uncertainty into the system. Hence, the development of efficient control systems for quadruped Mecanum robots has emerged as a vital and relevant subject in the realm of robotics. It is imperative to examine the research that has investigated current advancements, with a specific focus on the control of Mecanum robots. The text explores tactics, algorithms, and techniques that improve the ability to move and control motion, such as using a (PID) controller for controlling medium-wheel mobile robots [4-7]. Many studies have been presented that apply the sliding mode control strategy to control trajectory tracking and path following by omnidirectional mobile robots [8-15]. A wide range of researchers implemented fuzzy logic. It combined it with other controllers for controlling mobile robots, aiming to make the control process adaptive by changing control parameters such as [16-23]. Moreover, some researchers decided to use neural networks to control the motion of mecanum-wheeled mobile robots while tracking trajectories [24, 25]. In the same field of controlling mobile robots that

http://doi.org/10.30684/etj.2025.155232.1852
Received 26 November 2024; Received in revised form 07 January 2025; Accepted 25 January 2025; Available online 23 March 2025
2412-0758/University of Technology-Iraq, Baghdad, Iraq
This is an open access article under the CC BY 4.0 license http://creativecommons.org/licenses/by/4.0

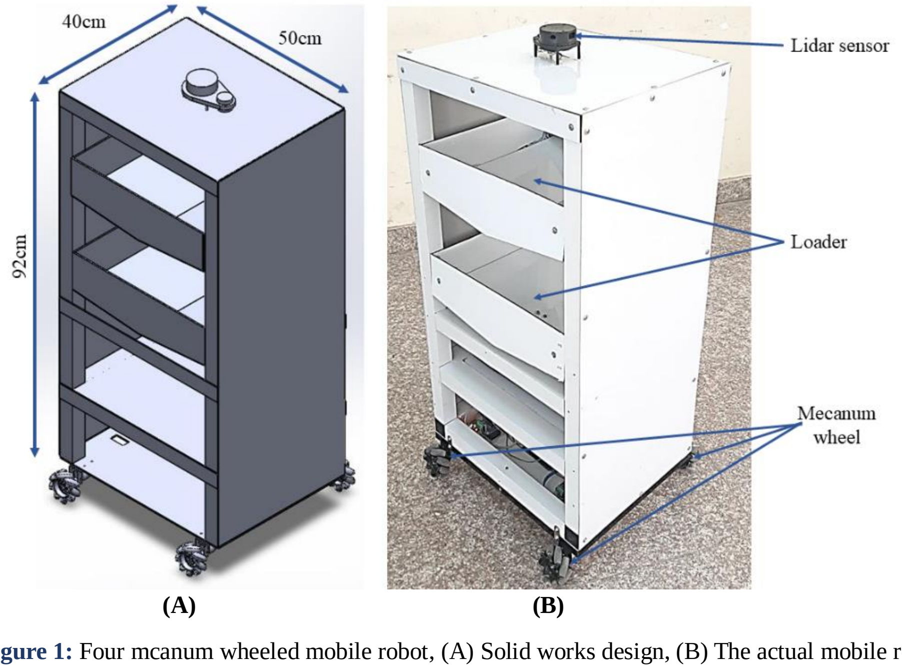
have mecanum wheels, some studies proposed control methodologies that combine different types of controllers or control with optimization techniques. A fuzzy PID control approach for an omnidirectional mobile robot's motion model and control features is developed in [26]. A wheeled mobile robot (WMR) is controlled for trajectory tracking by a fuzzy PID controller and optimized by a firefly optimization algorithm [27]. A controller for trajectory tracking of a three-wheeled omnidirectional mobile robot based on the Adaptive Neuro-Fuzzy Inference System (ANFIS) introduced in Allahyari et al., [28].

A study to control trajectory tracking by wheeled mobile robot used hybrid Bees-PSO to find neural controller gains values [29]. A trajectory-tracking nonlinear (PID NN) controller was introduced in this study for a differential wheeled mobile robot [30]. To improve heading precision, a PID setting control technique based on an RBF neural network model is applied to the Mecanum wheel's omnidirectional motion [31]. A linear active disturbance rejection controller (LADRC)-based attitude and position control system improves omnidirectional mobile robot disturbance rejection [32]. Other work introduced a robust controller for the four-wheeled omnidirectional mobile robot (FWOMR) that combines the Sliding Mode Control (SMC) algorithm with the Backstepping technique [33]. A robust MPC technique is proposed for trajectory tracking control of a four-wheeled FM-OMR under varied restrictions [34]. Tracking the trajectory control of a four-mecanum wheeled mobile robot (FMWMR) is implemented using Backstepping-Type 1 Fuzzy Logic-Particle Swarm Optimization (BSC-T1FLC-PSO) [35]. An adaptive robust controller based on the Udwadia-Kalaba Fundamental Equation (UKFE) is used to follow the trajectory of an omnidirectional wheeled mobile robot [36]. Other research finds the mobile robot position using a modified Kalman filter depending on the error between the predicted position measured by sensors and the reference position on a path [37]. A hybrid controller consisting of Computed Torque Control-Neural Network-Grey Wolf Optimization (CTC-NN-GWO) is proposed for controlling the Mecanum Wheeled Mobile Robot [38].

To the best of our knowledge, there is limited research on implementing (MRAC) with (PID) to control mobile robots, especially (FMWMR). In our opinion, most of the previous studies did not prove the actual changes in control parameters during the robot motion control process but rather showed the final control results of speeds. Also, some adaptive control methods were limited to a certain range of control parameters, such as using fuzzy control and Neural Networks. Also, previous research did not address the importance of changing the constraints on mobile robots when they carry objects. Our proposed controller aims to design a Multi-Stage controller to solve the problem of controlling three degrees of freedom and four wheels of the (FMWMR). It is designed and manufactured for logistic applications with two unique load holders. In this work, the (MRAC) aims to control the (FMWMR) based on the linearized dynamic model state-space. In contrast, the (PID) controller works on minimizing wheels' angular velocity error based on the kinematic model, then controlling mobile robot wheels by combining the two controllers to calculate the proper control signal.

2. Experimental setup of (FMWMR)

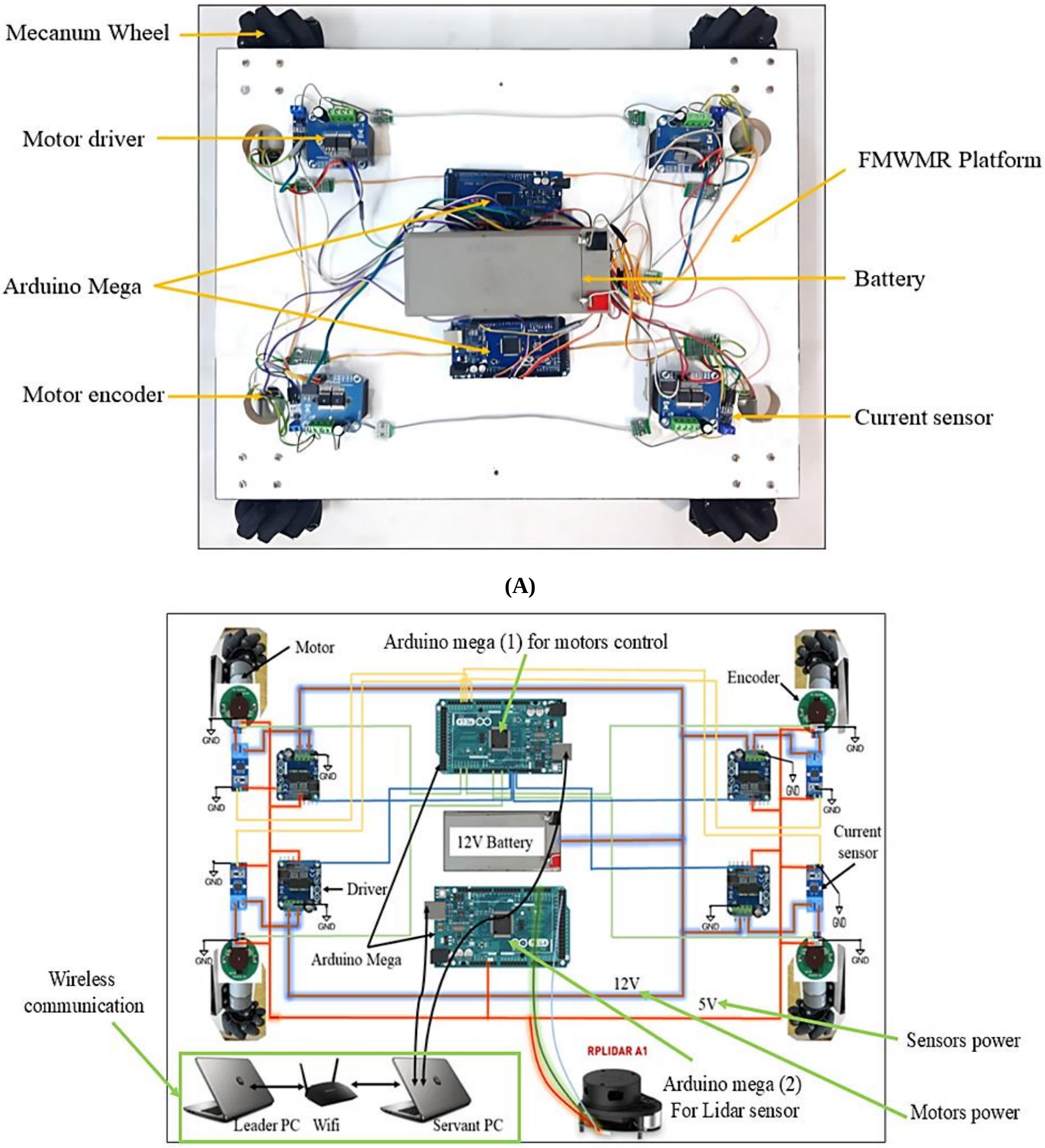
The FMWMR body was designed using Solid Works 2021a software as shown in Figure 1A. The dimensions of the mobile robot platform are (40×50) cm^2 and 92 cm height. The design has been taken into consideration to be suitable for logistics material handling applications. The mobile robot has two uniquely designed carriers, the bottom surface of which is inclined to the center of gravity to concentrate the load in the middle. Also, there is a place to carry the servant computer and mount the electronic parts away from the influence of the carried materials.



(A) (B)

Figure 1: Four mcanum wheeled mobile robot, (A) Solid works design, (B) The actual mobile robot

The electronic components of the robot include four basic parts, which are sensors, a control unit, actuators, and a servant computer. The manufactured (FMWMR) is shown in Figure 1B. Four encoders were used, and their basic function was to give feedback on the angular velocity of the wheels. In contrast, the (RPlidar A1) was adopted in order to detect the presence of obstacles in the environment. The control units involve two microcontrollers, the Arduino Mega (2560). Two loops of the controlling process are attached with MATLAB\Simulink. The first loop deals with controlling actuators by using motor drive H-bridge IBT2. The Arduino receives data from MATLAB\Simulink via serial communication between Arduino and computer to drive the actuators by generating PWMs and measuring RPMs and motor current from motor encoders and current sensors, respectively. To overcome the limitation of the low data transfer rate when using MATLAB/SIMULINK with Arduino for real-time control, a specific Arduino code is used to control motors and measure their speeds from encoders. These data are then sent and received between MATLAB and Arduino while SIMULINK is running. The second Arduino is used to receive obstacle detection data and send it to the computer. Figure 2A shows the connection of electronic components that mounted on the mobile robot platform and the connection block diagram is shown in Figure 2B.



(A)

(B)

Figure 2: Connection diagram: (A) Actual mobile robot platform, (B) Overall system connection diagram

2.1 Motors Specifications

The ChiHai Motor CHR-GM37-550 is a high-torque DC Motor with a 37 mm diameter and i90 gear ratio. Through the gear reduction mechanism, output shaft speed is reduced, and torque is increased. With compact size and high torque output, this motor offers exceptional performance and efficiency, making it a robust and versatile motor suitable for a wide range of robotics applications. Its compatibility with multiple voltage ranges, including 24V, 12V, and 6V, makes it adaptable to various power sources, providing flexibility in design and implementation. In this work, four CHR-GM37-550 motors with encoders have been used to drive the Mecanum wheel of each motor. The basic technical specifications are tabulated in Table 1.

Table 1: Technical specifications of the CHR-GM37-550 DC Motor

Parameter	Value
Ratio	1: 90
No-load current (mA)	≤ 1100
No-load speed (rpm)	180
Rated torque (Kg.cm)	27.0
Rated torque (m N.m)	2647.1
Rated torque (rpm)	120
Rated current (A)	≤ 6.0
Stall current (A)	20.0
gearbox (L)	24.0
Hall resolution	1530

2.2 Encoders

WMR applications utilize sensors to address localization challenges. A rotary encoder was utilized in this study, as depicted in Figure 3. The DC motor was fitted into each wheel. A rotary encoder is an electromechanical device that detects the rotational position of a wheel by sensing changes in the magnetic field and translates this information into electrical signals. This encoder has a resolution of 400 Pulses Per Revolution (PPR). The linear velocity of each wheel can be determined by Equation (1):

$$V_w = \frac{2\pi}{400} \times \text{ No.pulsespersecond } x \text{ wheel radius} \qquad (1)$$

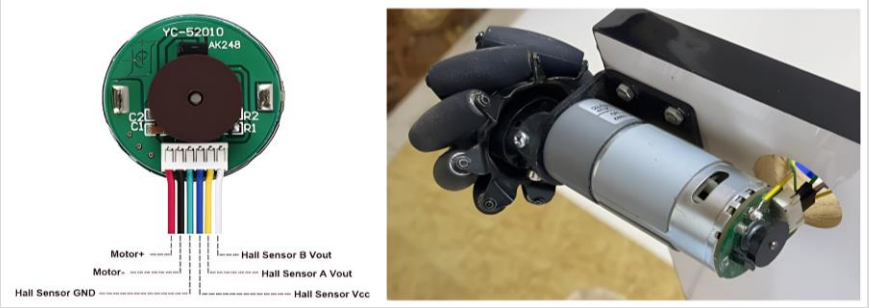
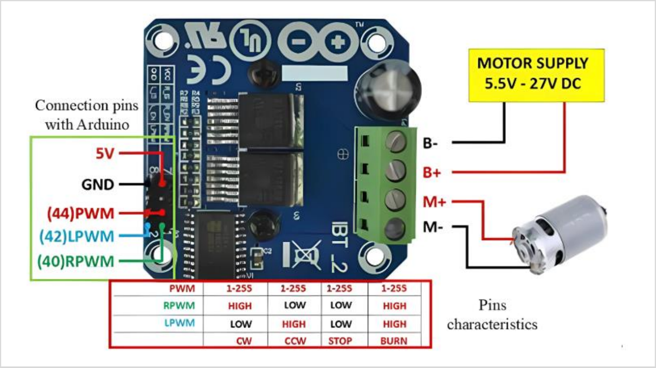


Figure 3: Rotary encoder connection with motor

2.3 IBT2 motor driver

The IBT2 is a comprehensive and self-contained module designed for efficiently controlling high currents in motor driving applications using an H bridge configuration. The inclusion of an integrated driver IC simplifies the process of connecting to a microcontroller. This IC has several capabilities, such as logic level inputs, current sense diagnosis, adjustable slew rate, dead time generation, and protection against overtemperature, overvoltage, undervoltage, overcurrent, and short circuits. The IBT2 offers a cost-effective solution for safeguarded high-current PWM motor drives while occupying minimal board space. The specifications of the IBT2 motor driver are: (1) Input Voltage: 6 ~ 27 Vdc. (2) Driver: Dual BTS7960 H Bridge Configuration. (3) Peak current: 43-Amp. (4) PWM capability of up to 25 kHz. (5) Control Input Level: 3.3~5 V. (6) Control Mode: PWM or level. (7) Working Duty Cycle: 0 ~100%. (8) Over-voltage Lock Out. (9) Under-voltage Shut Down. (10) Board Size (LxWxH): 50 mm x 50 mm x 43 mm. (11) Weight: ~66 g.

The power supply and control pins of the IBT2 motor driver are illustrated in Figure 4, where the pins (1, 2) are used to decide the direction of motor rotation, and the pins (4, 5) are used to control motor speed value. The pins (7, 8) are used for a 5V power supply, while on the other side, the pins (M+, M-) are connected to the motor, and pins (B+, B-) are the input 12V power.



PWM	1-255	1-255	1-255	1-255
RPWM	HIGH	LOW	LOW	HIGH
LPWM	LOW	HIGH	LOW	HIGH
	CW	CCW	STOP	BURN

Figure 4: Inputs/ outputs pins of IBT2 motor driver

3. FMWMR kinematic model

FMWMR has a motor that drives each wheel separately. Omni wheels with passive rollers have the ability to satisfy lateral and diagonal motion; this property makes the mobile robot suitable for many applications that need maneuverability [39]. Figure 5 offers insight into wheel arrangement.

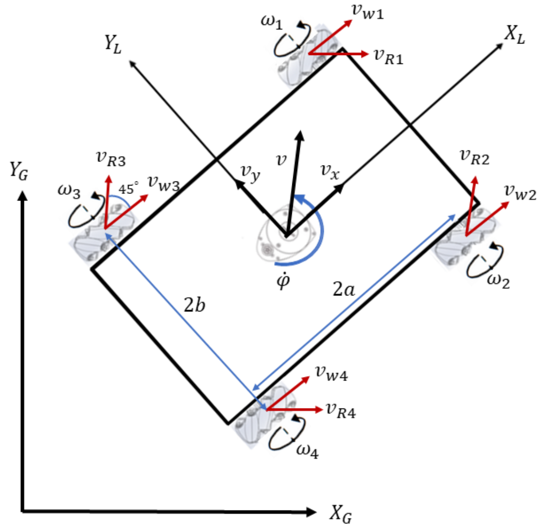


Figure 5: Schematic of four wheels mecanum mobile robot

The matrix-based inverse kinematics equations for the robot's wheel motion and linear and angular velocities are shown in Equation (2) [40]:

$$\begin{bmatrix}\omega_1\\\omega_2\\\omega_3\\\omega_4\end{bmatrix}=\frac{1}{\mathrm{rw}}\begin{bmatrix}1&-1&-(a+b)\\1&1&(a+b)\\1&1&-(a+b)\\1&-1&(a+b)\end{bmatrix}\begin{bmatrix}\dot{x}_L\\\dot{y}_L\\\dot{\varphi}_L\end{bmatrix}\tag{2}$$

A rotation matrix is used to convert Equation (2) from local to global coordinates, as shown in Equation (3):

$$\begin{bmatrix}\omega_1\\\omega_2\\\omega_3\\\omega_4\end{bmatrix}=\frac{1}{\mathrm{rw}}\begin{bmatrix}\cos(\varphi)&-\sin(\varphi)&0\\\sin(\varphi)&\cos(\varphi)&0\\0&0&1\end{bmatrix}\begin{bmatrix}1&-1&-(a+b)\\1&1&(a+b)\\1&1&-(a+b)\\1&-1&(a+b)\end{bmatrix}\begin{bmatrix}\dot{x}_L\\\dot{y}_L\\\dot{\varphi}_L\end{bmatrix}\tag{3}$$

$$\omega_i = J * \mathrm{R} * \dot{q}\tag{4}$$

where (R) is the rotation matrix, (J) is the inverse kinematics Jacobin matrix , and ($\dot{q}$) is the velocity vector. In the local coordinate system, the intended robot velocity and the inverse kinematic equation determine the forward kinematic equation. This connection can be expressed as (v) local = $J^{+} * \omega$, where J pseudo-inverse is (J^{+}), and calculated using $(J^{T}J)^{-1}(J^{T})$. The forward kinematics equation in local coordinates is derived (Equation(5)) [41]:

$$\begin{bmatrix}\dot{x}_L\\\dot{y}_L\\\dot{\varphi}_L\end{bmatrix}=\frac{\mathrm{rw}}{4}\begin{bmatrix}1&1&1&1\\-1&1&1&-1\\\frac{-1}{(a+b)}&\frac{1}{(a+b)}&\frac{-1}{(a+b)}&\frac{1}{(a+b)}\end{bmatrix}\begin{bmatrix}\omega_1\\\omega_2\\\omega_3\\\omega_4\end{bmatrix}\tag{5}$$

By converting local coordinates to global coordinates with the rotation matrix, we obtain in Equation (6):

$$\begin{bmatrix}\dot{x}_G\\\dot{y}_G\\\dot{\varphi}_G\end{bmatrix}=\frac{\mathrm{rw}}{4}\begin{bmatrix}\cos(\varphi)&-\sin(\varphi)&0\\\sin(\varphi)&\cos(\varphi)&0\\0&0&1\end{bmatrix}\begin{bmatrix}1&1&1&1\\-1&1&1&-1\\\frac{-1}{(a+b)}&\frac{1}{(a+b)}&\frac{-1}{(a+b)}&\frac{1}{(a+b)}\end{bmatrix}\begin{bmatrix}\omega_1\\\omega_2\\\omega_3\\\omega_4\end{bmatrix}\tag{6}$$

4. Dynamic model

This section assumes that the MWMR moves in a two-dimensional plane and ignores air resistance and the inertia of the little rollers on the mechanical wheels. The mobile robot's kinetic energy includes the rotational, translational kinetic energy of both the wheels and the mobile robot body. Derivatives of the Lagrangian (L) take into account generalized coordinates and their derivatives (q_i and $\dot{q}_i$). The right side of Equation (7) shows the system's external forces $F_{(i)}$:

$$\frac{d}{d_t}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = F_i \tag{7}$$

From Figure 1, if ($\varphi = 0$) the local coordinates (q_L) will be identical to global coordinates (q_G), So that the expression of velocities can be $\dot{q}_G = \dot{q}_L$ and lagrangian equation (Equation 8) by [42]:

$$L = \frac{1}{2} I_b \dot{\varphi_G}^2 + \frac{1}{2}(m_b + 4m_w)\left[\dot{x}_G{}^2 + \dot{y}_G{}^2\right] + \frac{1}{2} I_w(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \tag{8}$$

The rotational kinetic energy of each wheel is calculated in terms of linear and angular velocities ($\dot{x}_G, \dot{y}_G, \dot{\varphi_G}$) and physical characteristics such as wheel radius (rw), distance parameters (a, b), robot mass moment of inertia (I_b), and wheel moment of inertia (I_w) Equation (9):

$$L = \frac{1}{2} I_b \dot{\varphi_G}^2 + \frac{1}{2}(m_b + 4m_w)\left[\dot{x}_G{}^2 + \dot{y}_G{}^2\right] + \frac{1}{2}\frac{I_w}{r_w^2}[\dot{x}_G - \dot{y}_G - (a+b)\dot{\varphi_G}]^2 + \frac{1}{2}\frac{I_w}{r_w^2}[\dot{x}_G + \dot{y}_G + (a+b)\dot{\varphi_G}]^2 + \frac{1}{2}\frac{I_w}{r_w^2}[\dot{x}_G + \dot{y}_G - (a+b)\dot{\varphi_G}]^2 + \frac{1}{2}\frac{I_w}{r_w^2}[\dot{x}_G - \dot{y}_G + (a+b)\dot{\varphi_G}]^2 \tag{9}$$

The time derivatives of (L) with respect to the generalized coordinate velocities ($\dot{x}_G, \dot{y}_G, \dot{\varphi_G}$) expressing the accelerations as ($\ddot{x}_G, \ddot{y}_G, \ddot{\varphi}_G$).

$$\begin{bmatrix} \left(m_b + 4(m_w + \frac{I_w}{r_w^2})\right) & 0 & 0 \\ 0 & \left(m_b + 4(m_w + \frac{I_w}{r_w^2})\right) & 0 \\ 0 & 0 & (I_b + 4(m_w + \frac{I_w}{r_w^2})(a+b)^2) \end{bmatrix} \begin{bmatrix} \ddot{x}_G \\ \ddot{y}_G \\ \ddot{\varphi}_G \end{bmatrix} = M\ddot{q}_G \tag{10}$$

To convert global to local coordinates at (φ) [43].

$$\begin{bmatrix} \dot{x}_G \\ \dot{y}_G \\ \dot{\varphi_G} \end{bmatrix} = \begin{bmatrix} \cos(\varphi) & \sin(\varphi) & 0 \\ -\sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_L \\ \dot{y}_L \\ \dot{\varphi_L} \end{bmatrix} = \dot{q}_G = R_L^G \dot{q}_L \tag{11}$$

After the time differentiation of Equation (11):

$$\ddot{q}_G = R_L^G \ddot{q}_L + \dot{R}_L^G \dot{q}_L \tag{12}$$

$$\ddot{q}_G = \begin{bmatrix} \ddot{x}_G \\ \ddot{y}_G \\ \ddot{\varphi}_G \end{bmatrix} = \begin{bmatrix} \cos(\varphi) & \sin(\varphi) & 0 \\ -\sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \ddot{x}_L + \dot{\varphi}_L \dot{y}_L \\ \ddot{y}_L - \dot{x}_L \dot{x}_L \\ \ddot{\varphi}_L \end{bmatrix} \tag{13}$$

Also, force translation from local to global coordinates expression is ($F_G = R_L^G F_L$):

$$\begin{bmatrix} F_{x_G} \\ F_{y_G} \\ \tau_{\varphi_G} \end{bmatrix} = \begin{bmatrix} \cos(\varphi) & \sin(\varphi) & 0 \\ -\sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} F_{x_L} \\ F_{y_L} \\ \tau_{\varphi_L} \end{bmatrix} \tag{14}$$

Friction directly related to system velocity dissipation function is $F_f = \mu_{fq} \dot{F}_L$, where μ_{fq} is linear coefficients of friction in (x_L, y_L, φ_L) coordinates [44], therefore:

$$\begin{bmatrix} m_{11} & 0 & 0 \\ 0 & m_{22} & 0 \\ 0 & 0 & m_{33} \end{bmatrix} \begin{bmatrix} \ddot{x}_L + \dot{\varphi}_L \dot{y}_L \\ \ddot{y}_L - \dot{x}_L \dot{x}_L \\ \ddot{\varphi}_L \end{bmatrix} = \begin{bmatrix} F_{x_L} \\ F_{y_L} \\ \tau_{\varphi_L} \end{bmatrix} - \begin{bmatrix} \mu_{fx}\, \dot{x}_L \\ \mu_{fy}\, \dot{y}_L \\ \mu_{fz}\, \dot{\varphi}_L \end{bmatrix} \tag{15}$$

The general dynamic equation of a mobile robot is shown in Equation (16) [42]:

$$\mathrm{M(q)\ddot{q} + C\,(q, \dot{q})\dot{q} = \frac{1}{rw}\,\tau_i} \tag{16}$$

where ($\frac{1}{\mathrm{rw}}\,\tau_\mathrm{i} = \mathrm{F_i}$), after arranging Equation (15) and comparing it with the mobile robot's general dynamic equation, we got (see Equation (17)):

$$\begin{bmatrix} \mathrm{m_{11}} & 0 & 0 \\ 0 & \mathrm{m_{22}} & 0 \\ 0 & 0 & \mathrm{m_{33}} \end{bmatrix}\begin{bmatrix} \ddot{\mathrm{x}}_\mathrm{L} \\ \ddot{\mathrm{y}}_\mathrm{L} \\ \ddot{\varphi}_\mathrm{L} \end{bmatrix} + \begin{bmatrix} \mu_\mathrm{fx} & \mathrm{m_{11}}\,\dot{\varphi}_\mathrm{L} & 0 \\ -\mathrm{m_{22}}\dot{\varphi}_\mathrm{L} & \mu_\mathrm{fy} & 0 \\ 0 & 0 & \mu_\mathrm{fz} \end{bmatrix}\begin{bmatrix} \dot{\mathrm{x}}_\mathrm{L} \\ \dot{\mathrm{y}}_\mathrm{L} \\ \dot{\varphi}_\mathrm{L} \end{bmatrix} = \begin{bmatrix} \mathrm{F_{x_L}} \\ \mathrm{F_{y_L}} \\ \tau_{\varphi_\mathrm{L}} \end{bmatrix} \tag{17}$$

The torque of each wheel (τ_i) can be calculated by Equation (18):

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} = \begin{bmatrix} 1 & -1 & -(\mathrm{a}+\mathrm{b}) \\ 1 & 1 & (\mathrm{a}+\mathrm{b}) \\ 1 & 1 & -(\mathrm{a}+\mathrm{b}) \\ 1 & -1 & (\mathrm{a}+\mathrm{b}) \end{bmatrix}\begin{bmatrix} \mathrm{F_{x_L}} \\ \mathrm{F_{y_L}} \\ \mathrm{F_{\varphi_L}} \end{bmatrix} \tag{18}$$

5. Multi-stage controller design

The idea behind the design of a multi-stage controller is to control the (FMWMR) velocity while tracking a trajectory. The first stage uses (the PID) controller to estimate the control signal of each wheel of the mobile robot separately based on the error between the desired angular velocity that is calculated from the inverse kinematic equation and the actual measured angular velocity. Then, the motor control signal (voltage) from Equation 19 is calculated. The voltage is calculated based on the motor's electrical equation [45].

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} \frac{R_a}{Nk_t} & 0 & 0 & 0 \\ 0 & \frac{R_a}{Nk_t} & 0 & 0 \\ 0 & 0 & \frac{R_a}{Nk_t} & 0 \\ 0 & 0 & 0 & \frac{R_a}{Nk_t} \end{bmatrix}\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} + \frac{k_b N}{r_w}\begin{bmatrix} u_{pid1} \\ u_{pid2} \\ u_{pid3} \\ u_{pid4} \end{bmatrix} \tag{19}$$

where Ra: Resistance of armature, kt: The torque constant, and (N): gear ratio.

The control signal $[u_i]_{4\times1}$ is converted to the overall input control signal $[u_i]_{3\times1}$ by using the inverse of the Jacobian matrix (Equation 20):

$$\begin{bmatrix} \mathrm{u_x} \\ \mathrm{u_y} \\ \mathrm{u_\varphi} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ \frac{-1}{(\mathrm{a}+\mathrm{b})} & \frac{1}{(\mathrm{a}+\mathrm{b})} & \frac{-1}{(\mathrm{a}+\mathrm{b})} & \frac{1}{(\mathrm{a}+\mathrm{b})} \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \tag{20}$$

Let $[u]_n = [\mathrm{u_x} \quad \mathrm{u_y} \quad \mathrm{u_\varphi}]^T$ be the nominal control input for the second stage (MRAC) controller. The (MRAC) aims to minimize the overall system error, in other words, the error between the desired velocity $(v_{x_{desired}}, v_{y_{desired}}, \dot{\varphi}_{desired})$ and actual velocity $(v_{x_{actual}}, v_{y_{actual}}, \dot{\varphi}_{actual})$ of the (FMWMR). The MRAC controller updates the control parameters in real-time.

5.1 PID controller

PID is a configurable feedback controller. The PID controller structure has three parameters (gains): proportional (K_p), integral (K_i), and derivative (K_d). Below is the primary Equation for the time domain PID controller [46].

$$u_{pid_i}(t) = K_p e_i(t) + K_i \int e_i(\mathrm{t})d\mathrm{t} + K_d de_i(t)/\mathrm{dt} \tag{21}$$

where:

$$e_i = \omega_{i_{desired}} - \omega_{i_{actual}} \tag{22}$$

5.2 Model reference adaptive control

In control systems design, Model Reference Adaptive Control (MRAC) compares the real system performance with an ideal reference model to achieve the desired system performance. This comparison generates a system error signal that controls parameter change. This technique then modifies the controller parameters online to guide the uncertain dynamical system (Actual mobile robot) to follow the reference model and minimize the error [47]. The complexity of the conditions that face (FMWMR), especially in logistical environments with varying loads, led to the use of adaptive control methods, namely model reference adaptive control (MRAC). Start by defining the system state space. The state space of the dynamic model can be driven from Equation 16.

$$\begin{bmatrix} \ddot{x}_L \\ \ddot{y}_L \\ \ddot{\varphi}_L \end{bmatrix} = \begin{bmatrix} \frac{-\mu_{fx}}{m_{11}} & -\dot{\varphi}_L & 0 \\ \dot{\varphi}_L & \frac{-\mu_{fy}}{m_{22}} & 0 \\ 0 & 0 & \frac{-\mu_{fz}}{m_{33}} \end{bmatrix} \begin{bmatrix} \dot{x}_L \\ \dot{y}_L \\ \dot{\varphi}_L \end{bmatrix} + \begin{bmatrix} \frac{1}{rw\, m_{11}} & 0 & 0 \\ 0 & \frac{1}{rw\, m_{22}} & 0 \\ 0 & 0 & \frac{1}{rw\, m_{33}} \end{bmatrix} \begin{bmatrix} \tau_{x_L} \\ \tau_{y_L} \\ \tau_{\varphi_L} \end{bmatrix} \quad (23)$$

where $\tau_q = F_q \backslash rw$. By comparing Equation (23) with general state space form $\dot{x} = Ax + Bu$

where: $A = \begin{bmatrix} \frac{-\mu_{fx}}{m_{11}} & -\dot{\varphi}_L & 0 \\ \dot{\varphi}_L & \frac{-\mu_{fy}}{m_{22}} & 0 \\ 0 & 0 & \frac{-\mu_{fz}}{m_{33}} \end{bmatrix}$ and $B = \begin{bmatrix} \frac{1}{rw\, m_{11}} & 0 & 0 \\ 0 & \frac{1}{rw\, m_{22}} & 0 \\ 0 & 0 & \frac{1}{rw\, m_{33}} \end{bmatrix}$

To apply model reference adaptive control, the reference model should be time-invariant and linear. Because the term $\dot{\varphi}_L$ appear in the (A) matrix so that the system is nonlinear. This work presented a novel step to convert (FMWMR) state space from a nonlinear to linear by applying the Tylor series technique to linearize state space representation about equilibrium points ($\dot{x}_L = 0$, $\dot{y}_L = 0$, $\dot{\varphi}_L = 0$) [48]. From the linearization process, the new reference model is given in Equation (24):

$$\begin{bmatrix} \ddot{x}_L \\ \ddot{y}_L \\ \ddot{\varphi}_L \end{bmatrix} = \begin{bmatrix} \frac{-\mu_{fx}}{m_{11}} & 0 & 0 \\ 0 & \frac{-\mu_{fy}}{m_{22}} & 0 \\ 0 & 0 & \frac{-\mu_{fz}}{m_{33}} \end{bmatrix} \begin{bmatrix} \dot{x}_L \\ \dot{y}_L \\ \dot{\varphi}_L \end{bmatrix} + \begin{bmatrix} \frac{1}{rw\, m_{11}} & 0 & 0 \\ 0 & \frac{1}{rw\, m_{22}} & 0 \\ 0 & 0 & \frac{1}{rw\, m_{33}} \end{bmatrix} \begin{bmatrix} \tau_{x_L} \\ \tau_{y_L} \\ \tau_{\varphi_L} \end{bmatrix} \quad (24)$$

After linearizing the reference model, the term ($\dot{\varphi}_L$) disappears from the state matrix due to the omnidirectional motion of the FMWMR, as it maintains constant ($\dot{\varphi}_L = 0$) during motion. For this reason, the linear reference model is limited for omnidirectional motion scenarios, and the effect of ($\dot{\varphi}_L$) can be treated separately as a state. This simplification aims to facilitate the control process by reducing the risks of the automated system if there is a big change in orientation of FMWMR. In contrast, motion, the effect of ($\ddot{\varphi}_L$) should be taken into account in the state matrix. Still, the Mecanum wheels are designed specifically for omnidirectional motion without orientation changing. The resulting linearized state space shows that matrix (A) is known and Hurwitz and matrix (B) are known so that the reference model can be chosen as:

$$\dot{x}_m = A_m x_r + B_m r \quad (25)$$

where $r = [u]_n = [u_x \quad u_y \quad u_\varphi]^T$ which represents the nominal control signal calculated from Equation 20.

Let the uncertain plant model, whose objective is to follow the behavior of the reference model, be:

$$\dot{x}_p = A_p x_p + B_p \Lambda u \quad (26)$$

where Λ is a sign of B_p and both are known.

Firstly, define an ideal controller that perfectly cancels out the uncertainty and enables $x_p(t)$ to follow $x_m(t)$ as:

$$u^* = k_x^* x + k_r^* r(t) \quad (27)$$

where the superscript ∗ denotes ideal constant values that are unknown. Upon substituting into the plant model, we get the ideal closed-loop plant shown in Equation (28):

$$\dot{x}_p = \left(A_p + B_p k^*\right) x_p + B_p k_r^* r \quad (28)$$

Comparing the ideal closed-loop plant to the reference model, the ideal gains k_x^* and k_r^* can be determined by the following model-matching conditions:

$$\begin{gathered} A_p + B_p \Lambda k_x^* = A_m \\ B_p \Lambda k_r^* = B_m \end{gathered} \quad (29)$$

It turns out that the solutions for k_x^* and k_r^* always exist since there are two independent equations with two unknowns. The actual adaptive controller is an estimate of the ideal controller with a goal that, in the limit, the adaptive controller approaches the ideal controller.

Let:

$$u = k_x(t)x + k_r(t)r \tag{30}$$

be the adaptive controller, where $k_x(t)$ and $k_r(t)$ are the estimates of k_x^* and k_r^* respectively.

Let $\tilde{K}_x(t) = K_x(t) - K_x^*$ and $\tilde{K}_r(t) = K_r(t) - K_r^*$ be the estimation errors. Then, the closed-loop plant model is expressed as:

$$\dot{x}_p = (A_p + B\Lambda K_x^* + B\Lambda\widetilde{K}_x)x(t) + (B\Lambda K_r^* + B\Lambda\widetilde{K}_r)r(t) \tag{31}$$

The terms $A_p + B_p\Lambda k_x^*$ are equal to A_m and $B\Lambda K_r^*$ is equal to B_m , the closed-loop tracking error equation can now be formulated as shown in Equation (32) [49]:

$$\dot{e} = \dot{x}_m - \dot{x}_p = A_m e - B\Lambda\widetilde{K}_x x(t) - B\Lambda\widetilde{K}_r r(t) \tag{32}$$

Let $\widetilde{W}^T = [-\widetilde{K}_x \quad -\widetilde{K}_r]$ and $\sigma\big(x(t), c(t)\big) = [x(t) \;\; r(t)]^T$. Matrix $\widetilde{W}^T$ and the combined state vector $\sigma\,\big(x(t), c(t)\big)$ for the tracking error analysis. Then, the tracking error equation can be expressed as shown in Equation (33) [50]:

$$\dot{e} = A_m e - B\Lambda\widetilde{W}^T\sigma\big(x(t), r(t)\big) \tag{33}$$

Because A_m of the reference model is known and Hurwitz, there is one unique solution of the Lyapunov function [51]:

$$A_m^T P + PA_m = -R \tag{34}$$

If $R = I$, then P is a positive definite solution of the Lyapunov function: $P = \begin{bmatrix} \frac{m_{22}}{2\mu_{fx}} & 0 & 0 \\ 0 & \frac{m_{22}}{2\mu_{fy}} & 0 \\ 0 & 0 & \frac{m_{33}}{2\mu_{fz}} \end{bmatrix}$

Choosing the following Lyapunov candidate function to derive the adaptive laws:

$$V\big(e, \widetilde{W}^T\big) = e^T Pe + \Gamma^{-1}\text{trace}\big(\widetilde{W}^T\Lambda\,\widetilde{W}\big) \tag{35}$$

Let $\Gamma = \text{diag}(\gamma_x, \gamma_r)$, where γ_x and γ_r are called learning rates. The time Derivative of the Lyapunov Function is:

$$\dot{V}\big(e, \widetilde{W}^T\big) = 2e^T P\dot{e} + 2\Gamma^{-1}\text{trace}\left(\widetilde{W}^T\Lambda\,\dot{\widetilde{W}}\right) \tag{36}$$

By substation Equation (33) in (36) we get Equation (37):

$$\begin{aligned} \dot{V}\big(e, \widetilde{W}^T\big) &= 2e^T P(A_m e - B\Lambda\widetilde{W}^T\sigma\big(x(t), r(t)\big)) + 2\Gamma^{-1}\text{trace}\left(\widetilde{W}^T\Lambda\,\dot{\widetilde{W}}\right) \\ &= 2e^T PA_m e - 2e^T PB\Lambda\widetilde{W}^T\sigma\big(x(t), r(t)\big)) + 2\Gamma^{-1}\text{trace}\left(\widetilde{W}^T\Lambda\,\dot{\widetilde{W}}\right) \end{aligned} \tag{37}$$

For the first term of $\dot{V}\big(e, \widetilde{W}^T\big)$ to be negative definite, and the tracking error asymptotically goes to zero as P satisfies Equation 34. The second and third terms are identically equal to zero [50].

$$2\Gamma^{-1}\,\text{trace}\left(\widetilde{W}^T\Lambda\,\dot{\widetilde{W}}\right) = 2e^T PB\Lambda\widetilde{W}^T\sigma\big(x(t), r(t)\big)$$

Thus, the adaptive law is obtained as shown in Equation (38):

$$\dot{\widetilde{W}} = \Gamma\;\sigma\big(x(t), r(t)\big)e^T PB \tag{38}$$

But $\dot{K}_x$ and $\dot{K}_r$ are Implicitly in $\widetilde{W}^T$ and $\sigma\big(x(t), c(t)\big) = [x(t) \; r(t)]^T$, by breaking down Equation 38 matrixes to calculate $\dot{K}_x$ and $\dot{K}_r$ [52]:

$$\begin{aligned} \dot{K}_x^T &= \gamma_x x(t)e^T PB\Lambda \\ \dot{K}_r^T &= \gamma_r r(t)e^T PB\Lambda \end{aligned} \tag{39}$$

So, the adaptive control signal calculated by (MRAC) is:

$$[u]_{MRAC} = [\dot{K}_x][x] + [\dot{K}_r][u]_n \tag{40}$$

The overall motor control signal is:

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} \frac{R_a}{Nk_t} & 0 & 0 & 0 \\ 0 & \frac{R_a}{Nk_t} & 0 & 0 \\ 0 & 0 & \frac{R_a}{Nk_t} & 0 \\ 0 & 0 & 0 & \frac{R_a}{Nk_t} \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} + \frac{k_b N}{r_w} \begin{bmatrix} u_{pid1} \\ u_{pid2} \\ u_{pid3} \\ u_{pid4} \end{bmatrix} + \begin{bmatrix} 1 & -1 & -(\mathrm{a}+\mathrm{b}) \\ 1 & 1 & (\mathrm{a}+\mathrm{b}) \\ 1 & 1 & -(\mathrm{a}+\mathrm{b}) \\ 1 & -1 & (\mathrm{a}+\mathrm{b}) \end{bmatrix} \begin{bmatrix} u_{x_{MRAC}} \\ u_{y_{MRAC}} \\ u_{\varphi_{MRAC}} \end{bmatrix} \tag{41}$$

The control system block diagram is shown in Figure 6. MATLAB\SIMULINK block diagram that communicated with Arduino hardware is shown in Figure 7.

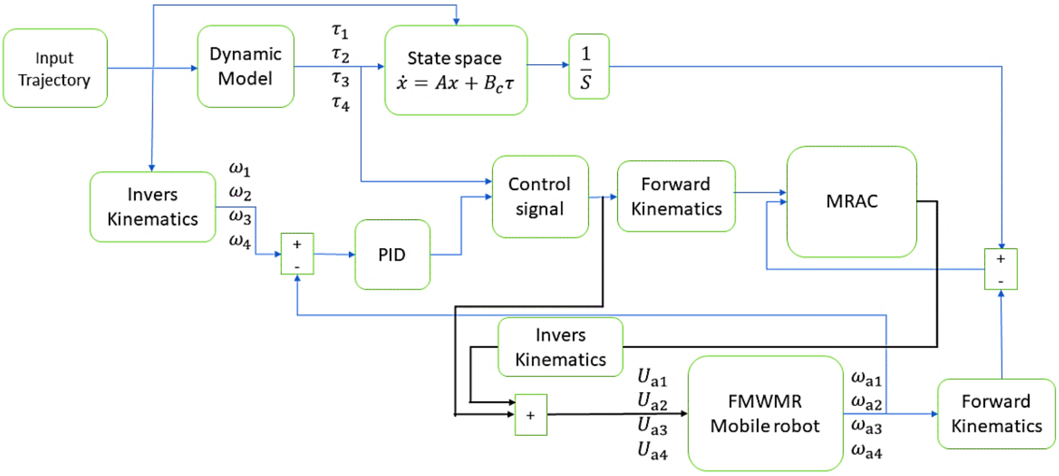


Figure 6: Control system block diagram

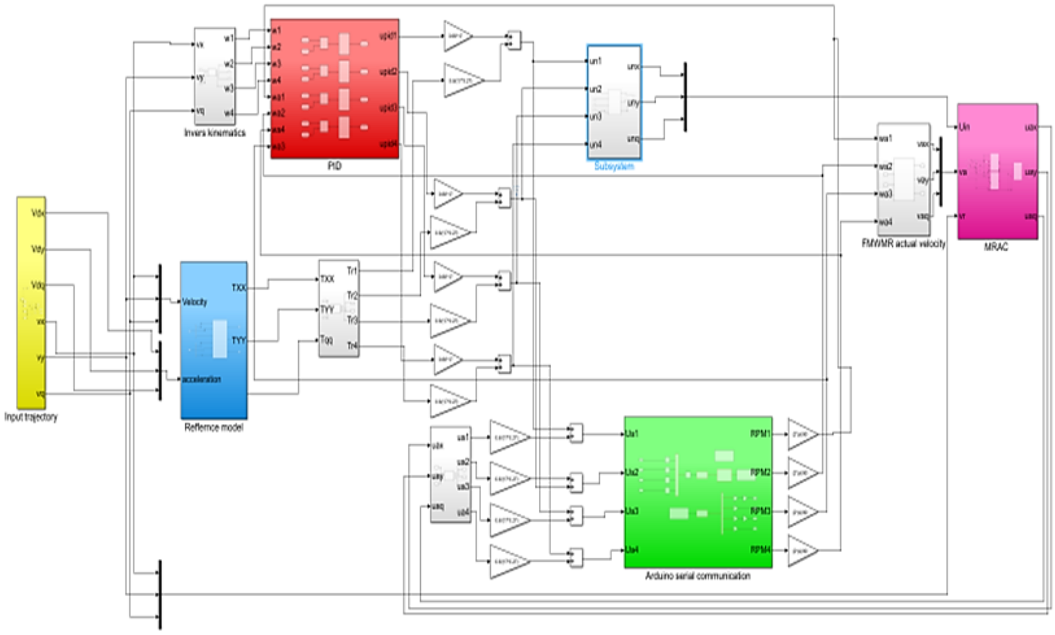


Figure 7: MATLAB\SIMULINK block diagram

6. Results and discussion

A Semi-Spiral trajectory shape has been chosen to test the performance of the designed control algorithm. The equation of the trajectory in terms of time is shown in Equation (42):

$$\begin{cases} \mathrm{x} = \mathrm{e}^{(\frac{0.2\,\mathrm{pi}\,\mathrm{t}}{45})}\,\sin(\frac{2\,\mathrm{pi}\,\mathrm{t}}{45}) \\ \mathrm{y} = -\mathrm{e}^{(\frac{0.2\,\mathrm{pi}\,\mathrm{t}}{45})}\,\cos(\frac{2\,\mathrm{pi}\,\mathrm{t}}{45}) \\ \varphi = 0 \end{cases} \tag{42}$$

The actual (FMWMR) parameters are shown in Table 2. The motors have been calibrated to estimate their parameters k_t and k_b and the (PID) controller parameters K_p, K_i and K_d have been found by using MATLAB tuner, which uses frequency-domain techniques. The control parameters of motors and (PID) are shown in Table 3.

Table 2: (FMWMR) parameters

Parameter	Symbol	Unit	Value
Mass of mobile robot	M	kg	22
Moment of inertia of mobile robot platform	I_b	kg.m ^{2}	0.1
Radius of wheel	rw	m	0.04
Mass of wheel	$\mathrm{m_w}$	kg	0.5
Wheels' moment of inertia	I_w	kg.m ^{2}	0.004
Length of platform	$2a$	m	0.4
Breadth of platform	$2b$	m	0.4
Coefficient of friction	$\mu_{fx} = \mu_{fy} = \mu_{fxy}$	-	0.27

Table 3: Motor and PID Controller parameters

Parameter	Parameter	Value
Motor Parameter	k_t	0.00936
	k_b	0.0822
(PID) parameters	K_p	0.81256
	K_i	1.87412
	K_d	0.005124

The initial position of the mobile robot is (0, 0, 0) and then it moves by velocity and acceleration inputs based on the trajectory equation. The actual performance of the (FMWMR) while tracking the desired trajectory is shown in Figure 8. The graph in the mentioned figure demonstrates that the actual trajectory represented by a solid red line matches the desired trajectory indicated by a dashed blue line.

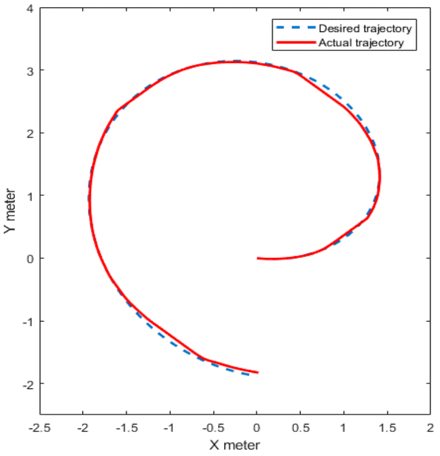
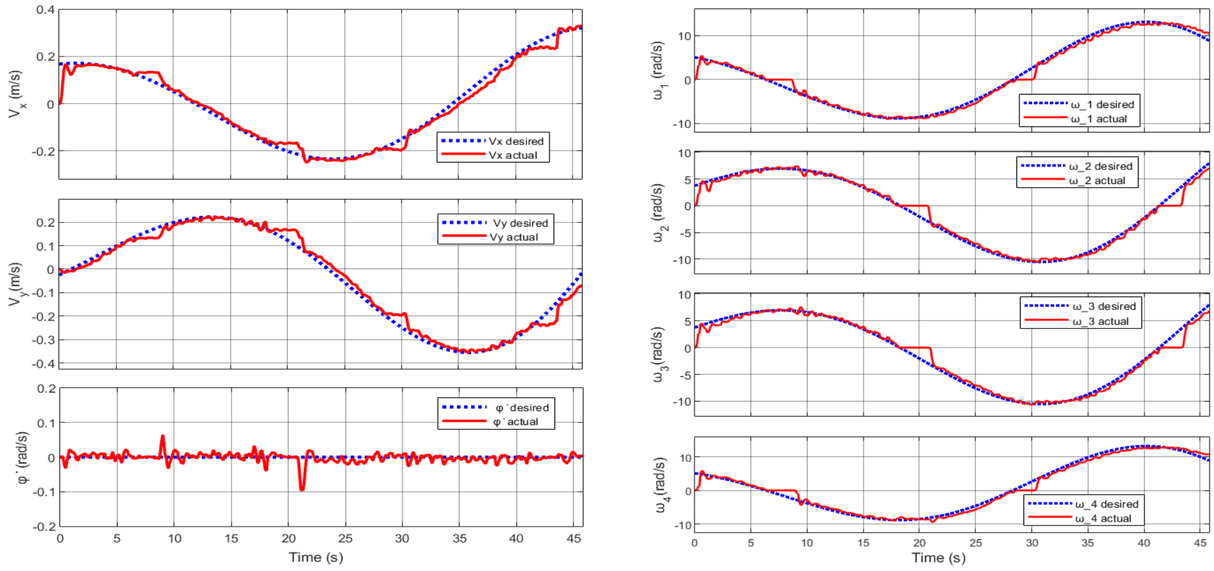


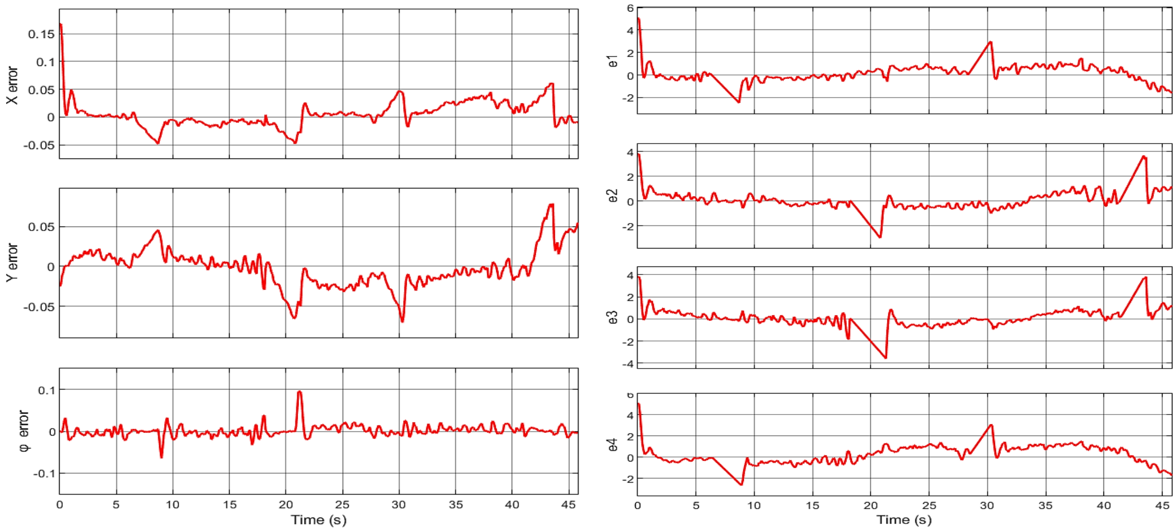
Figure 8: Actual (FMWMR) trajectory tracking

A comparison between the actual and desired velocity while the mobile robot is tracking the trajectory is illustrated in Figure 9 A. In contrast, Figure 9 B compares the desired and actual wheels' angular velocities. The desired velocity is represented by a dashed blue line and the actual one in a solid red line. The mobile robot reaches the initial velocity rapidly with a small oscillation appearing in velocity in (x-coordinate) in (1s) then the velocity follows the desired values in coordinates (x,y,φ). The velocity in (x-coordinate) start with (0.18 m/s) then decreases forward negative direction in second (12) until reaches (-0.23 m/s) after that the velocity increase to maximum value (0.3 m/s) while the velocity in (y-coordinate) started with (- 0.01 m/s) then increased in positive direction to (0.22 m/s) after that it decreases forward negative direction until it reaches the maximum value (-0.3 m/s). The angular velocity of the mobile robot oscillates about (0 rad/s) because the mobile robot can move without rotation by its mecanum wheels. The oscillations result from the behavior of wheel velocities when the controller Symbiotically guides the system to achieve the desired velocities, especially when the system velocity starts from rest condition. There are sudden deviations that appear between the actual and desired velocity in seconds (6, 18, 28, and 42) because of the changing of the wheels' rotation direction from positive to negative or vice versa. In second (6), wheels (1 and 4) changed their rotation direction from positive to negative, while in second (18), wheels (2 and 3) changed their rotation direction from positive to negative.



(A) **(B)**

Figure 9: FMWMR: (A) Actual and desired velocities profile. (B) wheels' actual angular velocity profile

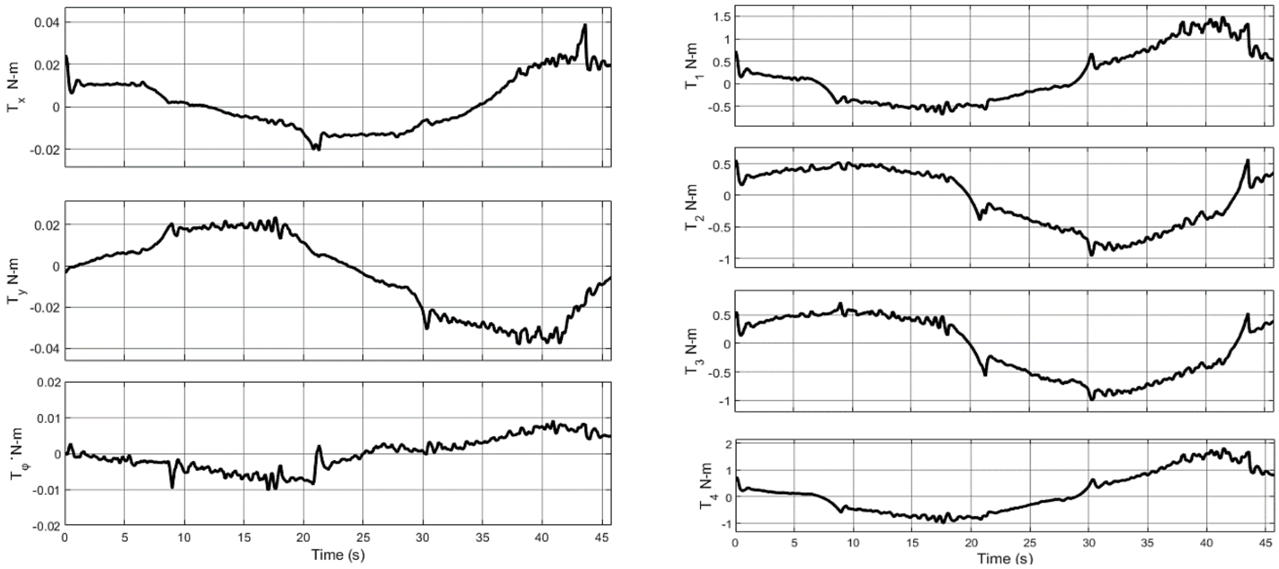


(A) **(B)**

Figure 10: Experimental (FMWMR): (A) Velocity tracking errors. (B) Wheels angular velocity error

This behavior was repeated when wheels (1 and 4) changed their rotation direction from negative to positive in the second (28). The deviation of the angular velocity of wheels (2 and 3) in second (42) appears because the rotation of these wheels is changed from a negative to a positive direction. During the tracking period, the mobile robot maintained its behavior by moving at the required speed and overcoming the deviations when they appeared. It can be observed from above that the (FMWMR) was successful without rotation because the wheel's rotation direction satisfied the omnidirectional motion in the desired angular velocities. The error between the desired and actual mobile robot velocity while trajectory tracking is shown in Figure 10 A and the wheel angular velocity error is illustrated in Figure 10 B. Initially, when the mobile robot starts its motion in (x-coordinate), there is a high error of about (0.17). It decreases immediately in (2) seconds close to (0), while the error of velocity in (y-coordinate) starts with (0), and the error of mobile robot angular velocity is (0) also. The value of error is kept at a small value throughout most of the motion period. Some peaks appear in seconds (6, 18, 28 and 42) these peaks are because of the deviations between the desired and actual mobile robot velocity. At the same seconds, the wheel's angular velocity error rises. As mentioned before, the deviations were caused by changes in the wheel rotation direction, so the error value increased. The designed Multi-Stage controller deals with both overall mobile robot velocity error by (MRAC) and wheels angular velocity error by (PID) so that the error returns to a small value instantly. It can be observed from the above that the controller efficiently trained the system to correct the error and restore the desired motion. The response of the controller indicates that the designed Multi-Stage controller has a high ability to cancel the effect of disturbances that resist mobile robot motion. Also, from the above scheme, it can be observed that the proposed controller is stable and effective in managing errors and disturbances.

The measured torques of the mobile robot in (x, y, φ) directions and wheels torque is shown in Figure 11. The mobile robot initially needs high torque to get started moving, as appears in Figure 11 A. The value of torque decreases after movement starts, which is related to the system dynamics because of the overcoming of static friction and inertia. The torque of the wheels, as shown in Figure 11 B, increases and decreases with variation in the wheels' angular velocity. Still, it has some overshoots when the wheel's rotation direction changes. The overshooting of torques results from the resistance of friction on wheels and the inertia when both mobile robots move and the wheels rotate. The behavior of torques indicates that the power of motors is under the control of the Multi-Stage controller. The increase and decrease of torque, especially when there is a need for sudden high torque, means that the controller has a high response to the dynamic behavior of the system. This is crucial for evaluating the performance of the (FMWMR) because the unique omnidirectional motion needs a real-time adaptive adjustment of controller outputs.



(A) **(B)**

Figure 11: FMWMR: (A) Actual torques action in (x, y, φ) (B) Actual wheel torque

The outputs from the multi-stage control methodology are shown in Figure 12 below. The control signal (U) is represented by (PWMs) and is converted to voltage to drive the motors. The values of the control signal highlight the dynamic behavior of the controller, where it demonstrates how the power adjusts and is delivered to the motors to manage the speed and torque. The range of (PWMs) varies from -255 to 255, which means that the voltage supply to the motor is between (0-12) Volt with motor rotation in a positive or negative direction. The overshoots of control signals of each wheel indicate that the controller estimates the proper power for each motor based on the dynamic behavior of the system. The smooth variation of the control signal along the traveling period means that the stability of the control process is high when the system is in a steady state. This demonstrates power management by the designed Multi-Stage controller when it accurately tracks the desired trajectory and showcases the quick response to the measured feedback.

Essentially, the (PID) deals with the error of the wheel separately and aims to keep each operating as accurately as possible. Still, it cannot reject unexpected disturbances or changes in the system dynamics, so the (MRAC) works to minimize the overall system error by adaptively generating a control signal to overcome any effect of disturbances. The output from (MRAC) is shown in Figure 13 below. As mentioned before, when the system starts operation, there are some restrictions, such as static friction force and inertia. Hence, the output from (MRAC) starts with a high value to guide the system for stable conditions. After that, when the system overcomes initial restrictions, the outputs from (MRAC) decrease because the error decreases. This adaptive behavior of the controller appears again when the mobile robot changes its motion direction because the wheels change their angular velocity and rotation direction, which generates disturbances that reset the motion. As shown in the mentioned figure the outputs from (MRAC) is varying in real time where U_{a_x} decrees and increase between second (20 and 23) and U_{a_y} increase between second (7 and 23). This is because the mobile robot's motion is more positive (y-coordinate) had the motion in (x-coordinate). After half of the period, the (MRAC) signal in the (x-coordinate) became negative the positive and in the (y-coordinate) negative most of the time because the mobile robot moved forward negative (y-coordinate) more than the x-coordinate) direction. Also, the values of the control signal increase when the overall mobile robot velocity increases. The signal U_{a_q} working to prevent rotation of the mobile robot while it is moving in omnidirectional motion so that it oscillates about 0.

The control parameters (k_x) and (k_r) of (MRAC) are adaptively modified in real-time based on the dynamic behavior of mobile robot d trajectory tracking. The continues updating of control parameters is illustrated in Figure 14. The control parameter (k_x) value depends on the actual mobile robot velocity and the overall system velocity error, while the parameter (k_r) is adjusted depending on the variation of the nominal control signal and error. From Figure 14 A and Figure 14 B, it can be observed that the values of (k_x) and (k_r) are increased when there is a need for a high signal from (MRAC). Then, the overall control signal of the system is calculated based on Equation 40, which is a combination of the nominal control signal that is calculated based on the system dynamic model and the output from the (PID) controller and the signal that comes from

(MRAC). In this way, the Multi-Stage controller adaptively controlled the mobile robot. The images of actual (FMWMR) trajectory tracking are shown in Figure 15.

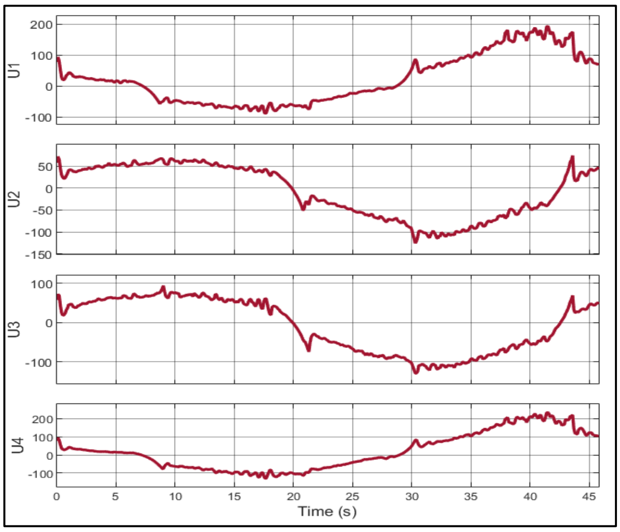


Figure 12: Overall multi-stage control methodology outputs

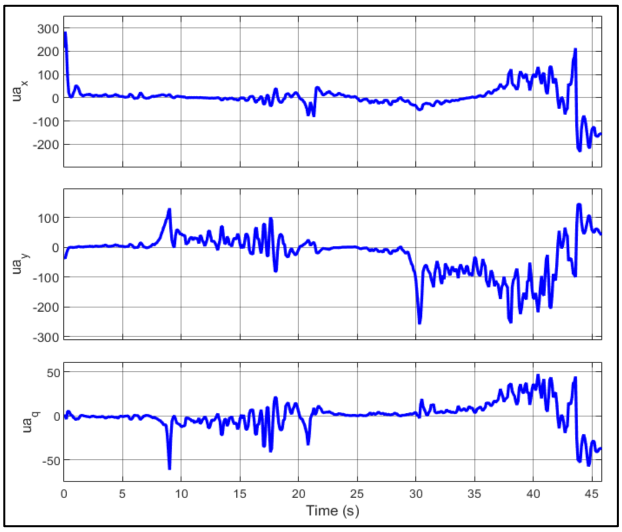
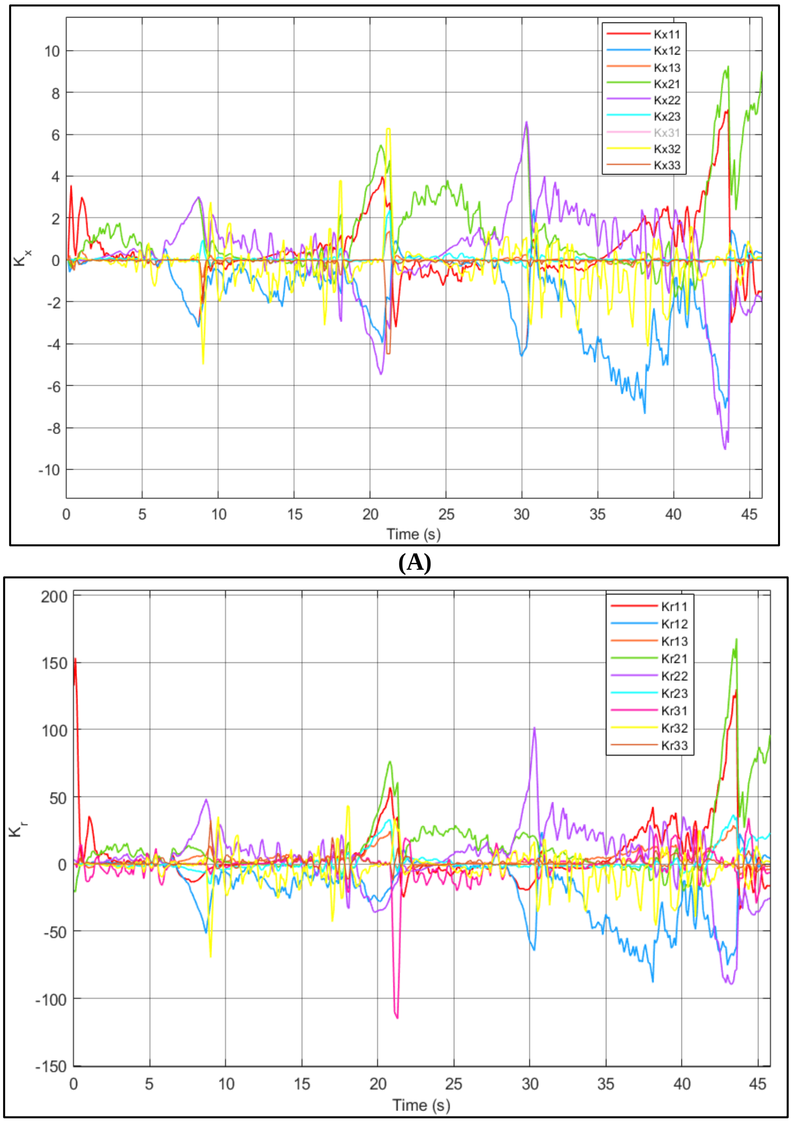


Figure 13: MRAC control output



(A)

(B)

Figure 14: MRAC control parameters (A) (k_x) (B) (k_r)

The experimental results of applying the multi-stage controller prove the efficiency and supervise ability to control the (FMWMR) while it is tracking the desired trajectory. The results of trajectory tracking control show a clear superiority for the Multi-Stage controller compared with other control Methodologies, such as the time-varying parameter PID controller used in [7], Adaptive Sliding Mode Controller proposed by Lu et al. [53], and nonlinear adaptive optimal H2 control method used in [54]. In summary, the proposed control methodology's promising results make it promising for controlling mobile robots, especially omnidirectional mobile robots, which are used to transport logistic items in various applications such as hospitals and industrial environments.

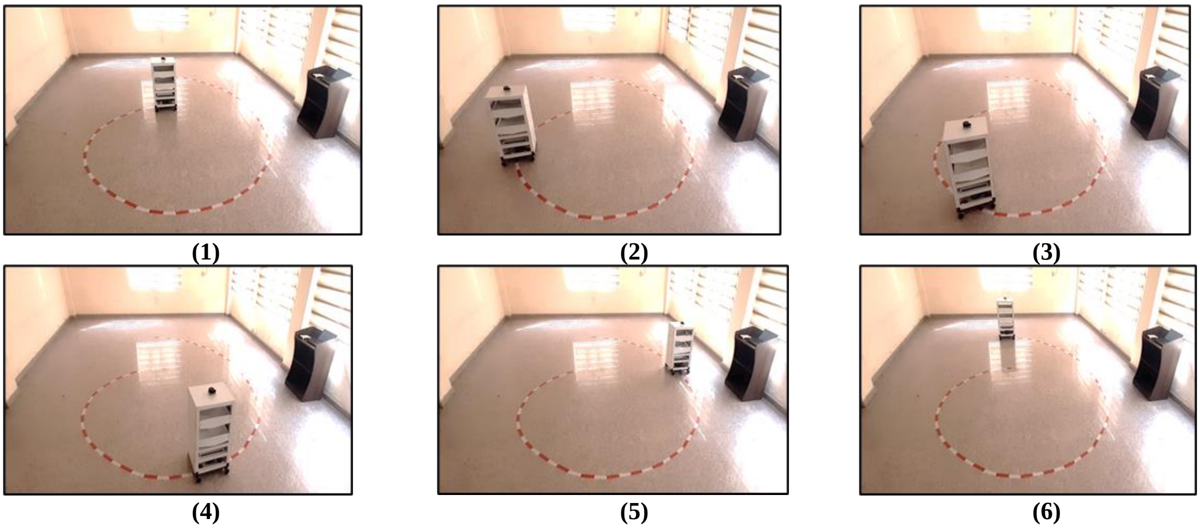


Figure 15: (FMWMR) trajectory tracking

7. Conclusion

This paper presented a multi-stage control methodology for trajectory tracking of the Four Mecanum Wheeled Mobile Robot (FMWMR). The proposed controller integrates Proportional-Integral-Derivative (PID) control with Model Reference Adaptive Control (MRAC), utilizing kinematic and dynamic models, respectively. The nominal control signal is derived from the dynamic characteristics of the mobile robot and subsequently adjusted by the MRAC outputs to counteract any disturbances affecting motion. The FMWMR has been specifically designed and constructed for logistics applications, with control implemented through MATLAB/SIMULINK, which interfaces with Arduino hardware to operate the mobile robot's motors. Experimental results demonstrate that the designed controller enables the mobile robot to follow the desired trajectory accurately. Measurements of the actual velocity of the mobile robot and the angular velocity of its wheels confirm that it operates at the intended speed while adhering to the principles of omnidirectional movement. Torque measurements indicate that the controller effectively supplies the wheels with the necessary power to achieve the desired motor velocities. Additionally, monitoring of the controller's outputs reveals that the control parameters are continuously updated throughout the mobile robot's travel period by the MRAC. Future work will focus on developing a path-planning and obstacle-avoidance algorithm for the FMWMR.

Author contributions

Conceptualization, **A. Abdullsahib**. and **H. Alwan**.; data curation, **A. Abdullsahib**.; formal analysis, **A. Abdullsahib**.; investigation, **A. Abdullsahib**; methodology, **H. Alwan**.; project administration, **H. Alwan**, resources, **A. Abdullsahib**.; software, A. Abdullsahib.; supervision, **H. Alwan**.; validation**, A. Abdullsahib**. and **H. Alwan**.; visualization, **A. Abdullsahib**. and **H. Alwan**; writing—original draft preparation, **A. Abdullsahib**.; writing—review and editing, **A. Abdullsahib**. and **H. Alwan.** All authors have read and agreed to the published version of the manuscript.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data availability statement

The data that support the findings of this study are available on request from the corresponding author.

Conflicts of interest

The authors declare that there is no conflict of interest.

References

[1] H. D. Quang, T. L. Tran, T. L. Manh, C. N. Manh, T. N. Nhu, N. B. Duy, Design a nonlinear MPC controller for autonomous mobile robot navigation system based on ROS, International Journal of Mechanical Engineering and Robotics Research, 11 (2022) 379-388. https://doi.org/10.18178/ijmerr.11.6.379-388

[2] H. A. Najim, I. S. Kareem, and W. E. Abdul-Lateef, Omnidirectional mobile robot with navigation using SLAM, Eng. Technol. J., 41 (2023) 196-202. https://doi.org/10.30684/etj.v41i1.2252

[3] Z. Yuan, Y. Tian, Y. Yin, S. Wang, J. Liu, L. Wu, Trajectory tracking control of a four mecanum wheeled mobile platform: an extended state observer-based sliding mode approach, IET Control Theory Appl., 14 (2020) 415-426. http://dx.doi.org/10.1049/iet-cta.2018.6127

[4] M. Alfiyan and R. D. Puriyanto, Mecanum 4 omni wheel directional robot design system using PID Method, J. Fuzzy Syst. Control, 1 (2023) 6-13. http://dx.doi.org/10.59247/jfsc.v1i1.27

[5] G. Cao, G. Cao, X. Zhao, C. Ye, S. Yu, B. Li and C. Jiang, Fuzzy adaptive PID control method for multi-mecanum-wheeled mobile robot, J. Mech. Sci. Technol., 36 (2022) 2019-2029. http://dx.doi.org/10.1007/s12206-022-0337-x

[6] R. Chotikunnan, P. Chotikunnan, N. Thongpance, T. Puttasakul, V. Pititheeraphab, M. Sangworasil, Application of PID control system in mecanum wheelchair, International Journal of Membrance Science Technology, 10, 2023, 3519-3529. http://dx.doi.org/10.15379/ijmst.v10i3.3395

[7] N. H. Thai, and T.T. K. Ly, Trajectory tracking control for mecanum wheel mobile robot by time-varying parameter PID controller, Bull. Electr. Eng. Inf., 11 (2022) 1902-1910. http://dx.doi.org/10.11591/eei.v11i4.3712

[8] S. Allahyari, H. Rahmanei, and S. A. A. Moosavian, Multi-aspect Optimal Sliding Mode Controller for a Mecanum Wheeled Robot, 11th RSI International Conference on Robotics and Mechatronics, 2023. http://dx.doi.org/10.1109/ICRoM60803.2023.10412554

[9] T. Guo, Trajectory tracking control of the Mecanum wheeled mobile robot based on the SMC methods, Proceedings, Sixth International Conference on Electromechanical Control Technology and Transportation, 12081, 2022. https://doi.org/10.1117/12.2624231

[10] Z. Li, Z. Li, Z. Sun, B. Chen, A Nonsingular Fast Terminal Sliding Mode Control Scheme for Mecanum-Wheels Omnidirectional Mobile Robots, 2023 5th International Conference on Robotics, Intelligent Control and Artificial Intelligence, 2023, 50-56. http://doi.org/10.1109/RICAI60863.2023.10489050

[11] J. Mowlaee, A. Sharghi, and R. A. Togh, Design of fixed-time terminal sliding mode control for robot with mecanum wheels, J. Nonlinear Sys. Electr. Eng., 8 (2022) 19-37.

[12] Z. Sun, S. Hu, D. He, W. Zhu, H. Xie, J. Zheng, Trajectory-tracking control of Mecanum-wheeled omnidirectional mobile robots using adaptive integral terminal sliding mode, Comput. Electr. Eng., 96 (2021) 107500. https://doi.org/10.1016/j.compeleceng.2021.107500

[13] Z. Sun, H. Xie, J. Zheng, Z. Man and D. He, Path-following control of Mecanum-wheels omnidirectional mobile robots using nonsingular terminal sliding mode, Mech. Syst. Signal Process., 147 (2021) 107128. https://doi.org/10.1016/j.ymssp.2020.107128

[14] D. Wang, Y. Gao, W. Wei, Q. Yu, Y. Wei, W. Li, Z. Fan, Sliding mode observer-based model predictive tracking control for Mecanum-wheeled mobile robot, ISA Trans., 151 (2024) 51-61. https://doi.org/10.1016/j.isatra.2024.05.050

[15] P. S. Yadav, V. Agrawal, J. C. Mohanta, M. D. Faiyaz Ahmed, A robust sliding mode control of mecanum wheel-chair for trajectory tracking, Mater. Today: Proc., 56 (2022) 623-630. http://dx.doi.org/10.1016/j.matpr.2021.12.398

[16] M. Crenganiș, R. E. Breaz, S. G. Racz, C. E. Girjob, C. M. Biris, et al., Fuzzy Logic-Based Driving Decision for an Omnidirectional Mobile Robot Using a Simulink dynamic model, Appl. Sci., 14 (2024) 3058. https://doi.org/10.3390/app14073058

[17] Felizardo C., Oscar C., Prometeo C.-A., Omnidirectional four wheel mobile robot control with a type-2 fuzzy logic behavior-based strategy, Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications, 49-62, 2020.

[18] H.-C. Huang and J.-J. Xu, Evolutionary machine learning for optimal polar-space fuzzy control of cyber-physical mecanum vehicles, Electronics, 9 (2020) 1945. https://doi.org/10.3390/electronics9111945

[19] D. N. Minh, H. D. Quang, N. D. Phuong, T. N. Manh, and D. Nam Bui, An Adaptive fuzzy dynamic surface control tracking algorithm for mecanum wheeled mobile robot, Int. J. Mech. Eng. Rob. Res., 12 (2023) 354-361. https://doi.org/10.18178/ijmerr

[20] B. Anil, M. Pandey, and S. Gajbhiye, Finite-Time Trajectory Tracking of a Four wheeled Mecanum Mobile Robot, arXiv preprint arXiv:2410.06762, (2024) 1-16. https://doi.org/10.48550/arXiv.2410.06762

[21] Z. Sun, S. Hu, N. Li, D. He, Trajectory-following control of mecanum-wheeled AGV using fuzzy nonsingular terminal sliding mode, 2020 4th CAA International Conference on Vehicular Control and Intelligence, 2020, 342-347. https://doi.org/10.1109/CVCI51460.2020.9338561

[22] T. T. Thuong, V. T. Ha, V. Q. Vinh, N. T. Hien, Design adaptive fuzzy dynamic surface controller combined with DWA algorithm in motion control for mecanum wheels omnidirectional mobile robots, International Conference on Engineering Research and Applications, 943, 2023. https://doi.org/10.1007/978-3-031-62238-0_36

[23] T. Zhao, X. Zou, and S. Dian, Fixed-time observer-based adaptive fuzzy tracking control for Mecanum-wheel mobile robots with guaranteed transient performance, Nonlinear Dyn., 107 (2022) 921-937. https://doi.org/10.1007/s11071-021-06985-0

[24] T. T. K. Ly, N. T. Thanh, H. Thien, T. Nguyen, A Neural network controller design for the mecanum wheel mobile robot, Eng. Technol. Appl. Sci. Res., 13 (2023) 10541-10547. https://doi.org/10.48084/etasr.5761

[25] M. Szeremeta and M. Szuster, Neural tracking control of a four-wheeled mobile robot with mecanum wheels, Appl. Sci., 12 (2022) 5322. https://doi.org/10.3390/app12115322

[26] Q. Jia, C. Chang, S. Liu, L. Zhang, S. Zhang, Motion control of omnidirectional mobile robot based on fuzzy PID, 2019 Chinese Control And Decision Conference, 2019, 5149-5154. https://doi.org/10.1109/CCDC.2019.8833047

[27] M. J. Mohamed, and M. Abbas, Design a fuzzy PID controller for trajectory tracking of mobile robot, Eng. Technol. J., 36 A1 (2018) 100-110. https://doi.org/10.30684/etj.2018.136785

[28] A. Alsharkawi, M. Al-Fetyani, E. M. Ijaabo, H. Khasawneh, Adaptive Neuro-Fuzzy Inference System for a Three-Wheeled Omnidirectional Mobile Robot, 2020 3rd International Conference on Applied Engineering, 2020, 1-6. http://doi.org/10.1109/ICAE50557.2020.9350379

[29] A. Al-Araji and N. Yousif, A Cognitive nonlinear trajectory tracking controller design for wheeled mobile robot based on hybrid bees-PSO algorithm, Eng. Technol. J., 35A (2017) 609-616. https://doi.org/10.30684/etj.2017.131978

[30] M. J. Mohamed, and M.K. Hamza, Design PID neural network controller for trajectory tracking of differential drive mobile robot based on PSO, Eng. Technol. J., 37 A (2019) 574-583. https://doi.org/10.30684/etj.37.12A.12

[31] N. Zijie, Z. peng, Y. Cui. Z. Jun, PID control of an omnidirectional mobile platform based on an RBF neural network controller, Ind. Robot. Int. J. Robot. Res. Appl., 49 (2022) 65-75. http://dx.doi.org/10.1108/IR-01-2021-0015

[32] H. Fu, Y. Li, Y. Wang, Z. Zhang, Omnidirectional mobile robot active disturbance rejection control2018 IEEE International Conference on Mechatronics and Automation, 2018, 227-232. https://doi.org/10.1109/ICMA.2018.8484414

[33] K. D. H. Thi, M. C. Nguyen, H. T. Vo, V. M. Tran, D. D. Nguyen, A. D. Bui, Trajectory tracking control for four-wheeled omnidirectional mobile robot using Backstepping technique aggregated with sliding mode control 2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics, 2019, 131-134. https://doi.org/10.1109/ICA-SYMP.2019.8646041

[34] D. Wang, W. Wei. Y. Yebah, Y. li, Y. Ga, A robust model predictive control strategy for trajectory tracking of omni-directional mobile robots, J. Intell. Rob. Sys., 98 (2020) 439-453. https://doi.org/10.1007/s10846-019-01083-1

[35] S. F. Hasana and H. M. Alwan, Modeling and Control of Wheeled Mobile Robot With Four Mecanum Wheels, Eng. Technol. J., 39 (2021) 779-789. https://doi.org/10.30684/etj.v39i5A.1926

[36] F. Dong, D. Jin, X. Zhao, J. Han, Adaptive robust constraint following control for omnidirectional mobile robot: An indirect approach, IEEE Access, 9 (2021) 8877-8887. https://doi.org/10.1109/ACCESS.2021.3049913

[37] A. Madhloom, F.A. Raheem, and A.R. Kareem, A modified Kalman filter-based mobile robot position measurement using an accelerometer and wheels encoder, Eng. Technol. J., 40 (2022) 267-274. https://doi.org/10.30684/etj.v40i1.2082

[38] R. J. Salman, H. M. Alwan, M. A. Yousif, A. M. Abdullah, Computed torque-NN-GWO dynamic hybrid control of manipulator robotic arm, Iraq. J. Comput. Commun. Control Sys. Eng., 24 (2024) 1-16. https://doi.org/10.33103/uot.ijccce.24.2.1

[39] E. Maulana, M. A. Muslim, and V. Hendrayawan, Inverse kinematic implementation of four-wheels mecanum drive mobile robot using stepper motors, in 2015 international seminar on intelligent technology and its applications, 2015. http://dx.doi.org/10.1109/ISITIA.2015.7219952

[40] Klancar, G., Andrej Z, Sašo B., Igor Š., Wheeled mobile robotics: from fundamentals towards autonomous systems, 2017.

[41] S. F. Hasan, and H. M. Alwan, Enhancing Tilt-Integral-Derivative Controller to Motion Control of Holonomic Wheeled Mobile Robot by Using New Hybrid Approach, IOP Conference Series: Materials Science and Engineering, 1094 (2021) 012097. http://dx.doi.org/10.1088/1757-899X/1094/1/012097

[42] I. Moreno-Caireta, E. Celaya, and L. Ros, Model predictive control for a Mecanum-wheeled robot navigating among obstacles, IFAC-Pap., 54 (2021) 119-125. http://dx.doi.org/10.1016/j.ifacol.2021.08.533

[43] V. Alakshendra, and S. S. Chiddarwar, A robust adaptive control of mecanum wheel mobile robot: simulation and experimental validation, 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2016, 5606-5611. http://dx.doi.org/10.1109/IROS.2016.7759824

[44] Tzafestas, S.G., Introduction to mobile robot control, Athens, Greece, 2013.

[45] Jagan, N., Control systems, BS Publications Hyderabad, 2008.

[46] Ogata, K., Modern control engineering, 680 E Colorado Bvld, Suite 180, Pasadena, CA 91101, 2020.

[47] Yucelen, T., Model Reference Adaptive Control, in Wiley Encyclopedia of Electrical and Electronics Engineering, John Wiley & Sons, In, 1-13, 2019.

[48] Socha, L., Linearization methods for stochastic dynamic systems, Springer Science & Business Media, Vol. 730. 2007.

[49] Eugene, L., Kevin, W. and Howe, D., Robust and adaptive control with aerospace applications, England: Springer-Verlag London, 2013.

[50] A. Albattat, B. Gruenwald, and T. Yucelen, Design and analysis of adaptive control systems over wireless networks, J. Dyn. Syst. Meas. Contr., 139 (2017) 074501.

[51] T. Yucelen, and W.M. Haddad, Low-frequency learning and fast adaptation in model reference adaptive control, IEEE Trans. Autom. Control, 58 (2012) 1080-1085. https://doi.org/10.1109/TAC.2012.2218667

[52] Nguyen, N.T. , Model-reference adaptive control, Springer, 2018.

[53] X. Lu,X. Zhang, G. Zhang, S. Jia, Design of adaptive sliding mode controller for four-Mecanum wheel mobile robot, 2018 37th Chinese Control Conference, 2018, 3983-3987. https://doi.org/10.23919/ChiCC.2018.8483388

[54] Y.-H., Chen, Nonlinear Adaptive Optimal control design and implementation for trajectory tracking of four-wheeled mecanum mobile robots, Mathematics, 12 (2024) 4013. https://doi.org/10.3390/math12244013