

Experimental Evaluation of a Nav2 Regulated Pure Pursuit-PID Architecture for Trajectory Tracking of a Four-Wheel Mecanum Mobile Robot

Quang Vi Ngo,¹ Van Vu Duong,² Guanqiu Qi,³ and Zhiqin Zhu⁴

¹Department of Robotic and Intelligent Systems Engineering (RISE), Faculty of Electrical & Electronics Engineering, Thuyloi University, Hanoi, Vietnam

²Faculty of Electrical & Electronics Engineering, Thuyloi University, Hanoi, Vietnam

³Computer Information Systems Department, State University of New York at Buffalo State, Buffalo, NY, USA

⁴College of Automation, Chongqing University of Posts and Telecommunications, Chongqing, China

(Received 10 July 2026; Revised 25 August 2026; Accepted 07 September 2026; Published online 05 October 2026)

Abstract: This paper reports an experimental evaluation of a Nav2 Regulated Pure Pursuit (RPP)-PID architecture for trajectory tracking of a four-wheel Mecanum mobile robot. The high-level RPP controller runs on an NVIDIA Jetson Nano at 50 Hz, while four proportional-integral-derivative (PID) loops on an STM32 microcontroller regulate individual wheel speeds at 100 Hz. PP and RPP were evaluated using the same robot hardware, map, PID gains, start pose, goal pose, and reference path. Each quantitative tracking condition was repeated five times. RPP reduced the mean RMSE from 0.12 ± 0.03 m to 0.08 ± 0.01 m for curved-path tracking and from 0.10 ± 0.02 m to 0.05 ± 0.02 m for goal reaching. Obstacle-avoidance experiments were also conducted with both controllers under matched conditions and are discussed as system-level navigation tests because planning, localization, perception, and control remained active. The controller uses nominal Mecanum kinematics without explicit slip or delay compensation. The results demonstrate improved geometric tracking accuracy for RPP under the tested indoor conditions and confirm stable online operation of the distributed RPP-PID architecture.

Keywords: kinematic model; Mecanum mobile robot; path tracking; PID control; Pure Pursuit; Regulated Pure Pursuit; ROS 2 Navigation2

I. INTRODUCTION

Mobile robots play a vital role in modern automation and have been widely deployed in applications such as industrial logistics, service robotics, automated inspection, and intelligent transportation systems [1,2]. Among the various mobile robot architectures, omnidirectional platforms equipped with Mecanum wheels have received significant attention because of their holonomic motion capability. Unlike conventional wheeled robots, Mecanum-wheeled platforms can simultaneously perform planar translation and rotation, enabling high maneuverability in confined and cluttered environments [3].

Despite their kinematic advantages, achieving accurate path tracking and robust motion control for Mecanum-wheeled mobile robots remains a significant challenge. The unique wheel–roller configuration introduces complex wheel–ground interactions, resulting in lateral slip and deviations from the ideal kinematic model [4,5]. These nonlinear effects become increasingly pronounced at higher operating speeds and along high-curvature trajectories, leading to degraded path-tracking performance and reduced control stability.

Numerous control methods have been proposed to address the path-tracking problem in mobile robots. Among them, model predictive control (MPC) has gained considerable attention because it

optimizes control inputs over a finite prediction horizon while explicitly incorporating system dynamics and operational constraints [6]. Nevertheless, the high computational complexity associated with solving an optimization problem at each sampling instant poses challenges for real-time implementation, particularly on resource-constrained embedded platforms.

Intelligent control methods, including fuzzy logic control (FLC), have also been widely used to address the nonlinearities and uncertainties inherent in robotic systems [7,8]. Owing to its model-free nature, FLC offers enhanced adaptability under uncertain operating conditions. However, its control performance is highly dependent on the design and tuning of expert-defined fuzzy rules and membership functions, which can limit its robustness and generalization across diverse operating environments.

Geometric path-tracking methods, particularly Pure Pursuit (PP), have been widely used because of their simple implementation and low computational cost [9]. The PP method determines a lookahead point along the reference trajectory and generates a steering command that guides the robot toward the selected target point. Despite its computational efficiency, the tracking performance of PP is highly sensitive to the choice of lookahead distance. Consequently, its accuracy may degrade when navigating high-curvature trajectories or operating under dynamically changing conditions.

To alleviate this sensitivity, Adaptive Pure Pursuit (APP) methods dynamically adjust the lookahead distance based on the robot's velocity or operating conditions [8]. Although this adaptive

Corresponding author: Zhiqin Zhu (e-mail: zhuzq@cqupt.edu.cn).

strategy improves tracking performance under varying conditions, APP does not explicitly consider safety constraints or the dynamic limitations of the robotic platform, which may compromise performance during aggressive maneuvers or in complex environments.

Sliding mode control (SMC) is a robust nonlinear control method that is typically designed based on Lyapunov stability theory to drive system states toward a predefined sliding surface aiming to overcome external disturbances and modeling uncertainties [10,11]. For Mecanum-wheeled mobile robots, SMC effectively compensates for lateral slip and kinematic uncertainties, thereby improving path-tracking accuracy and robustness [12]. However, its practical implementation is often hindered by the chattering phenomenon and the need for careful parameter tuning, which can increase implementation complexity and limit real-time performance on resource-constrained embedded platforms.

More recently, Regulated Pure Pursuit (RPP) has been proposed as an enhanced variant of the conventional PP method by incorporating velocity-scaled lookahead selection, curvature- and obstacle-aware linear velocity regulation, and collision checking [9]. By adaptively adjusting the robot's linear velocity according to path geometry and environmental constraints, RPP achieves an effective balance among path-tracking accuracy, motion stability, and computational efficiency. Furthermore, RPP is implemented as a controller plugin within the ROS 2 Navigation2 (Nav2) framework, enabling straightforward integration and deployment on real-world robotic platforms.

However, existing studies on RPP have predominantly focused on nonholonomic mobile robots or simulation-based environments. Comprehensive experimental investigations involving Mecanum-wheeled mobile robots remain scarce, particularly under real-world operating conditions where lateral slip and wheel-ground interactions significantly influence motion behavior and path-tracking performance.

At the low-level control layer, proportional-integral-derivative (PID) controllers are widely used for individual wheel-speed regulation because of their simple structure, ease of implementation, and reliable control performance [14,15].

Recent advances in embedded computing platforms, including the NVIDIA Jetson Nano and STM32 microcontrollers, have enabled online implementation of increasingly sophisticated control algorithms [10]. In the proposed multi-rate architecture, the high-level Nav2 RPP [22] controller executes at 50 Hz on the Jetson Nano, while the low-level wheel-speed PID loops execute at 100 Hz on the STM32. This separation allows the navigation layer to generate body-velocity commands and the faster embedded layer to regulate individual wheel speeds.

Motivated by these challenges, this paper presents an end-to-end experimental evaluation of an RPP-PID architecture on a physical four-wheel Mecanum robot. The high-level Nav2 RPP controller regulates body-frame velocity commands according to path curvature, obstacle proximity, and remaining goal distance, while the STM32-based low-level layer maps these commands to wheel-speed references and applies independent PID control. The nominal Mecanum kinematic model provides the body-to-wheel velocity mapping, and the physical experiments assess the complete closed-loop response under practical wheel-ground interaction, localization uncertainty, actuator dynamics, and communication effects.

The main contributions of this paper are summarized as follows:

A distributed embedded implementation that combines a Nav2-based RPP configuration on an NVIDIA Jetson Nano with four STM32-based wheel-speed PID loops is presented.

A physical-platform evaluation compares PP and RPP in curved-path tracking, goal-reaching, and obstacle-avoidance scenarios using the same robot hardware, map, PID gains, start pose, goal pose, and reference path.

Five repeated trials were conducted for each quantitative PP-RPP tracking condition, and the RMSE results are reported as mean $\pm$ standard deviation together with maximum tracking and lateral-error values.

The experimental analysis treats wheel-ground interaction, localization uncertainty, actuator dynamics, and communication effects as coupled properties of the physical closed-loop system. The reported results therefore characterize end-to-end geometric tracking performance without attributing the measured error reduction to a single disturbance mechanism.

Accordingly, this study evaluates whether the implemented RPP-PID architecture improves geometric tracking accuracy over conventional PP under matched indoor conditions and whether the 50 Hz high-level and 100 Hz low-level control structure supports stable online operation on the selected Mecanum platform.

II. COORDINATE SYSTEM AND MATHEMATICAL MODEL OF THE FOUR-WHEEL MECANUM ROBOT

A. COORDINATE SYSTEM DEFINITION FOR THE MECANUM MOBILE ROBOT

To describe the robot's motion, we establish the coordinate axes as shown in Figs. 1–3.

The global coordinate frame $\{X_A O_A Y_A\}$ is attached to the fixed working environment.

The local coordinate frame $\{X_B O_B Y_B\}$ is fixed at the center of the robot.

x and y denote the current position of the robot, while θ represents the robot heading angle, defined as the angle between the X_A and X_B axes.

α_j denotes the angle between the vector $O_B Q_j$ and the X_B -axis, which defines the position of the j -th wheel with respect to the robot center.

$H_j Q_j Z_j$ denotes the coordinate frame of the j -th wheel ($j = 1, 2, 3, 4$), whose origin is located at the wheel center Q_j .

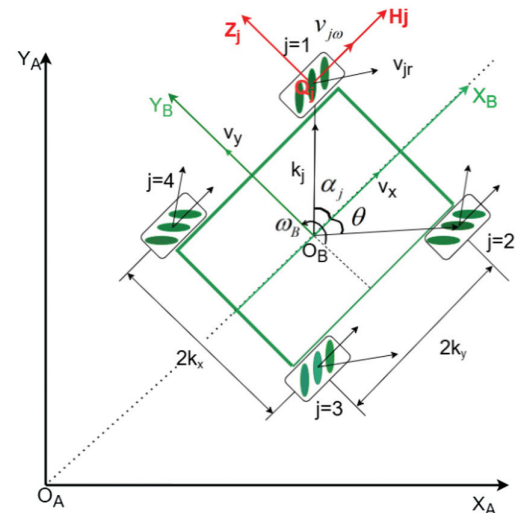


Fig. 1. Coordinate system for the mobile robot.

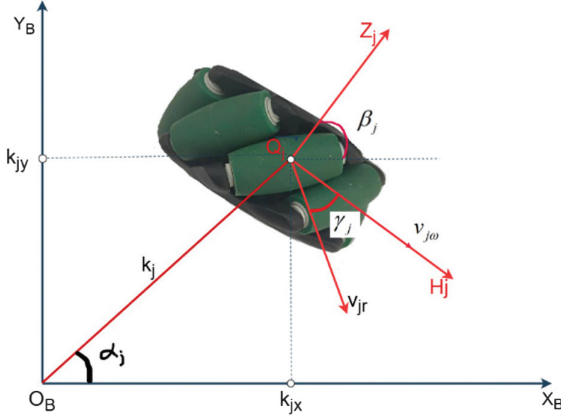


Fig. 2. Coordinate frame of the j -th wheel.

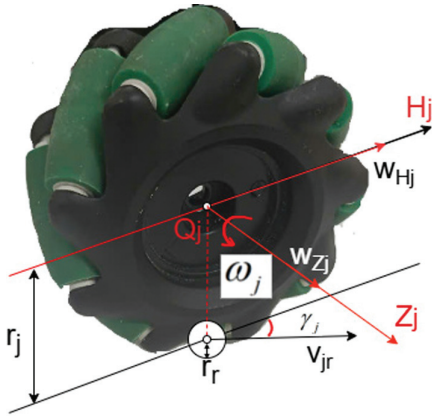


Fig. 3. Kinematic principle of the j -th Mecanum wheel.

$2k_x$ and $2k_y$ represents the wheelbase and track width of the robot, that is, the distance between the centers of the left and right wheels and the distance between the centers of the front and rear wheels.

k_j denotes the distance from the center of the j -th wheel to the robot center O_B .

r_j denotes the radius of the j -th wheel.

ω_j denotes the angular velocity of the j -th wheel (rad/s).

$v_{j\omega}$ denotes the linear velocity of the wheel (m/s).

v_x and v_y denote the translational velocities of the mobile robot (m/s).

ω_B denotes the angular velocity of the mobile robot.

r_r denotes the radius of the roller of the j -th wheel, that is, the distance between the wheel center and the roller center.

β_j denotes the angle between the X_B -axis and the Z_j -axis, representing the wheel orientation angle.

γ_j denotes the angle between the roller velocity vector v_{jr} and the H_j axis, commonly referred to as the roller inclination angle.

Figure 1 shows the configuration of a four-wheel omnidirectional mobile robot with Mecanum wheels mounted in an X-shaped arrangement.

B. KINEMATIC MODELING OF THE FOUR-WHEEL MECANUM MOBILE ROBOT

As shown in Fig. 3, the angular velocity of the j -th wheel and the tangential velocity of the passive roller in contact with the ground can be calculated as follows:

$$v_{jr} = \frac{1}{\cos 45^\circ} r_r \omega_j \quad (1)$$

And:

$$w_{H_j} = r_j \omega_j \quad (2)$$

With $j = 1, 2, 3, 4$

Based on Fig. 3 and Eqs. (1) and (2), the velocity of the j -th wheel in the wheel coordinate frame is given by:

$$\begin{cases} v_{Z_j} = v_{jr} \sin \gamma_j \\ v_{H_j} = \omega_j r_j + v_{jr} \cos \gamma_j \end{cases}$$

$$\begin{bmatrix} v_{Z_j} \\ v_{H_j} \end{bmatrix} = \begin{bmatrix} 0 & \sin \gamma_j \\ r_j & \cos \gamma_j \end{bmatrix} \begin{bmatrix} \omega_j \\ v_{jr} \end{bmatrix} = {}^{w_j}T_{Q_j} \begin{bmatrix} \omega_j \\ v_{jr} \end{bmatrix} \quad (3)$$

Transformation matrix relating the velocity of the j -th wheel to the velocity at its center:

$$\text{Where : } T_F = {}^{w_j}T_{Q_j} = \begin{bmatrix} 0 & \sin \gamma_j \\ r_j & \cos \gamma_j \end{bmatrix} \quad (4)$$

Based on Figs. 2 and 3, the velocity of the wheel center in the $X_B O_B Y_B$ coordinate frame can be expressed as Eq. (5):

$$\begin{bmatrix} v_{jX_B} \\ v_{jY_B} \end{bmatrix} = \begin{bmatrix} \cos \beta_j & -\sin \beta_j \\ \sin \beta_j & \cos \beta_j \end{bmatrix} \begin{bmatrix} v_{Z_j} \\ v_{H_j} \end{bmatrix}$$

$$= \begin{bmatrix} \cos \beta_j & -\sin \beta_j \\ \sin \beta_j & \cos \beta_j \end{bmatrix} \begin{bmatrix} 0 & \sin \gamma_j \\ r_j & \cos \gamma_j \end{bmatrix} \begin{bmatrix} \omega_j \\ v_{jr} \end{bmatrix}$$

$$= {}^{w_j}T_{Q_j} T_R \begin{bmatrix} \omega_j \\ v_{jr} \end{bmatrix} = T_E T_F \begin{bmatrix} \omega_j \\ v_{jr} \end{bmatrix} \quad (5)$$

The transformation matrix relating the j -th wheel center to the robot coordinate frame can be derived from Eq. (6).

Where

$$T_E = Q_j T_R = \begin{bmatrix} \cos \beta_j & -\sin \beta_j \\ \sin \beta_j & \cos \beta_j \end{bmatrix} \quad (6)$$

Since the robot operates in a two-dimensional plane, the following relationship holds

$$\begin{bmatrix} v_{jX_B} \\ v_{jY_B} \end{bmatrix} = \begin{bmatrix} 1 & 0 & k_{jx} \\ 0 & 1 & k_{jy} \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} = T_M \begin{bmatrix} v_{X_B} \\ v_{Y_B} \\ \omega_B \end{bmatrix} \quad (7)$$

Where

$$T_N = \begin{bmatrix} 1 & 0 & k_{jx} \\ 0 & 1 & k_{jy} \end{bmatrix} \quad (8)$$

Based on Eqs. (4) and (6), the inverse kinematics of the robot can be expressed as:

$$T_E T_F \begin{bmatrix} \omega_j \\ v_{jr} \end{bmatrix} = T_M \begin{bmatrix} v_{X_B} \\ v_{Y_B} \\ \omega_B \end{bmatrix} \quad (9)$$

From $r_j \neq 0, 0 < |\gamma_j| < \frac{\pi}{2}, \det(T_E) \neq 0, \det(T_F) \neq 0$

By combining Eqs. (5) and (7), the relationship between the robot body velocity at O_B and the angular velocity of the j -th wheel can be obtained, as expressed in Eq. (10):

$$\begin{bmatrix} \omega_j \\ v_{jr} \end{bmatrix} = T_E^{-1} T_F^{-1} T_M \begin{bmatrix} v_{X_B} \\ v_{Y_B} \\ \omega_B \end{bmatrix}, j = 1, 2, 3, 4 \quad (10)$$

According to Eqs. (4) and (6), the relationship between the wheel velocities and the robot body velocity can be established. In the inverse kinematic model, the robot body velocity is determined from the wheel motions, where v_j and ω_j represent the linear and angular velocities of the j -th wheel, respectively, as given in Eq. (11). Conversely, the wheel velocities can be calculated from the robot body velocity using the forward kinematic model expressed in Eq. (12):

$$\begin{bmatrix} v_{X_B} \\ v_{Y_B} \\ \omega_B \end{bmatrix} = T^+ \begin{bmatrix} \omega_j \\ v_{jr} \end{bmatrix} \quad (11)$$

$$\begin{bmatrix} \omega_j \\ v_{jr} \end{bmatrix} = T \begin{bmatrix} v_{X_B} \\ v_{Y_B} \\ \omega_B \end{bmatrix} \quad (12)$$

where $T = T_E^{-1} \cdot T_F^{-1} \cdot T_M$, $T^+ = (T^T T)^{-1} T^T$

The corresponding equation is given as follows:

$$T = \begin{bmatrix} \cos \beta_j & -\sin \beta_j \\ \sin \beta_j & \cos \beta_j \end{bmatrix}^{-1} \begin{bmatrix} 0 & \sin \gamma_j \\ r_j & \cos \gamma_j \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & k_{jx} \\ 0 & 1 & k_{jy} \end{bmatrix} \quad (13)$$

By considering $k_{jx} = k_j \cos \alpha_j$ and $k_{jy} = k_j \sin \alpha_j$, and assuming identical wheel dimensions, the transformation matrix can be derived by multiplying the three matrices, yielding:

$$T := -\frac{1}{r} \begin{bmatrix} \frac{\cos(\beta_j - \gamma_j)}{\sin(\gamma_j)} & \frac{\sin(\beta_j - \gamma_j)}{\sin(\gamma_j)} & \frac{k_j \sin(-\alpha_j + \beta_j - \gamma_j)}{\sin(\gamma_j)} \\ -\frac{r \cos(\beta_j)}{\sin(\gamma_j)} & -\frac{r \sin(\beta_j)}{\sin(\gamma_j)} & -\frac{k_j \sin(-\alpha_j + \beta_j) r}{\sin(\gamma_j)} \end{bmatrix} \quad (14)$$

$$T^+ = \frac{1}{k_j^2 + 1} \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \\ T_{31} & T_{32} \end{bmatrix} \quad (15)$$

where

$$T_{11} = -\frac{1}{2}(k_j^2 \sin(\beta_j) - k_j^2 \sin(-\beta_j + 2\alpha_j) + 2 \sin(\beta_j))r$$

$$T_{12} = \frac{1}{2}k_j^2 \sin(\gamma_j - \beta_j + 2\alpha_j) - \frac{1}{2} \sin(-\gamma_j + \beta_j)k_j^2 - \sin(-\gamma_j + \beta_j)$$

$$T_{21} = \frac{1}{2}r(k_j^2 \cos(\beta_j) - k_j^2 \cos(-\beta_j + 2\alpha_j) + 2 \cos(\beta_j))$$

$$T_{22} = -\frac{1}{2}k_j^2 \cos(\gamma_j - \beta_j + 2\alpha_j) + \frac{1}{2} \cos(-\gamma_j + \beta_j)k_j^2 + \cos(-\gamma_j + \beta_j)$$

$$T_{31} = \cos(\alpha_j - \beta_j)k$$

$$T_{32} = \cos(\alpha_j - \beta_j + \gamma_j)k_j$$

Since a relationship exists between the independent variables v_{jr} and ω_j at each wheel assembly and the linear and angular velocities of the robot body, and under the assumption of pure rolling without wheel-ground slip, the inverse kinematic model of the system can be derived as given in Eq. (16).

Nominal kinematic assumption. The pure-rolling model provides the body-to-wheel velocity mapping used by the controller in Eqs. (16–26). On the physical platform, wheel-ground interaction, localization uncertainty, actuator dynamics, and communication effects jointly influence the resulting closed-loop trajectory:

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = -\frac{1}{r} \begin{bmatrix} \frac{\cos(\beta_1 - \gamma_1)}{\sin(\gamma_1)} & \frac{\sin(\beta_1 - \gamma_1)}{\sin(\gamma_1)} & \frac{k_1 \sin(-\alpha_1 + \beta_1 - \gamma_1)}{\sin(\gamma_1)} \\ \frac{\cos(\beta_2 - \gamma_2)}{\sin(\gamma_2)} & \frac{\sin(\beta_2 - \gamma_2)}{\sin(\gamma_2)} & \frac{k_2 \sin(-\alpha_2 + \beta_2 - \gamma_2)}{\sin(\gamma_2)} \\ \frac{\cos(\beta_3 - \gamma_3)}{\sin(\gamma_3)} & \frac{\sin(\beta_3 - \gamma_3)}{\sin(\gamma_3)} & \frac{k_3 \sin(-\alpha_3 + \beta_3 - \gamma_3)}{\sin(\gamma_3)} \\ \frac{\cos(\beta_4 - \gamma_4)}{\sin(\gamma_4)} & \frac{\sin(\beta_4 - \gamma_4)}{\sin(\gamma_4)} & \frac{k_4 \sin(-\alpha_4 + \beta_4 - \gamma_4)}{\sin(\gamma_4)} \end{bmatrix} \begin{bmatrix} v_{X_B} \\ v_{Y_B} \\ \omega_B \end{bmatrix} \quad (16)$$

The Jacobian matrix associated with the inverse kinematics of the system is given by Eq. (17) [5]:

$$T = -\frac{1}{r} \begin{bmatrix} \frac{\cos(\beta_1 - \gamma_1)}{\sin(\gamma_1)} & \frac{\sin(\beta_1 - \gamma_1)}{\sin(\gamma_1)} & \frac{k_1 \sin(-\alpha_1 + \beta_1 - \gamma_1)}{\sin(\gamma_1)} \\ \frac{\cos(\beta_2 - \gamma_2)}{\sin(\gamma_2)} & \frac{\sin(\beta_2 - \gamma_2)}{\sin(\gamma_2)} & \frac{k_2 \sin(-\alpha_2 + \beta_2 - \gamma_2)}{\sin(\gamma_2)} \\ \frac{\cos(\beta_3 - \gamma_3)}{\sin(\gamma_3)} & \frac{\sin(\beta_3 - \gamma_3)}{\sin(\gamma_3)} & \frac{k_3 \sin(-\alpha_3 + \beta_3 - \gamma_3)}{\sin(\gamma_3)} \\ \frac{\cos(\beta_4 - \gamma_4)}{\sin(\gamma_4)} & \frac{\sin(\beta_4 - \gamma_4)}{\sin(\gamma_4)} & \frac{k_4 \sin(-\alpha_4 + \beta_4 - \gamma_4)}{\sin(\gamma_4)} \end{bmatrix} \quad (17)$$

According to Eq. (11), the forward kinematic model of the system can be expressed as:

$$\begin{bmatrix} v_{X_B} \\ v_{Y_B} \\ \omega_B \end{bmatrix} = T^+ \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} \quad (18)$$

C. RELATIONSHIP BETWEEN MOTION AND THE TRANSFORMATION MATRIX

According to the inverse kinematic model, there is a relationship between the velocity of each wheel assembly and the velocity of the robot's body center. Therefore, the robot's body motion can be determined from the individual wheel velocities.

According to robot kinematics theory, the inverse kinematic problem can be solved when the Jacobian matrix satisfies the required rank condition, ensuring a unique mapping between the wheel velocities and the robot body motion. Consequently, the effective degrees of freedom of the wheel motions are mapped to the robot's translational and rotational motions.

A typical four-wheel Mecanum mobile robot configuration is illustrated in Fig. 1. The geometric parameters of this configuration are summarized in Table I, following refs. [10,16–18].

By substituting the geometric parameters listed in Table I into the Jacobian matrix given in Eq. (17), the following expression can be obtained:

$$T = \frac{1}{r} \begin{bmatrix} 1 & -1 & -(k_x + k_y) \\ 1 & 1 & (k_x + k_y) \\ 1 & 1 & -(k_x + k_y) \\ 1 & -1 & (k_x + k_y) \end{bmatrix} \quad (19)$$

$$T^+ = \frac{r}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -\frac{1}{k_x + k_y} & \frac{1}{k_x + k_y} & -\frac{1}{k_x + k_y} & \frac{1}{k_x + k_y} \end{bmatrix} \quad (20)$$

Based on Eqs. (11) and (12), the forward and inverse kinematic models of the four-wheel Mecanum robot can be expressed as follows:

Table I. Geometric configuration parameters of the four-wheel Mecanum mobile robot

j	Wheels	α_j	β_j	γ_j	k_j	k_{jx}	k_{jy}
1	1sw	$\frac{\pi}{4}$	$\frac{\pi}{2}$	$-\frac{\pi}{4}$	k	k_x	k_y
2	2sw	$-\frac{\pi}{4}$	$-\frac{\pi}{2}$	$\frac{\pi}{4}$	k	k_x	k_y
3	3sw	$\frac{3\pi}{4}$	$\frac{\pi}{2}$	$\frac{\pi}{4}$	k	k_x	k_y
4	4sw	$-\frac{3\pi}{4}$	$-\frac{\pi}{2}$	$-\frac{\pi}{4}$	k	k_x	k_y

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 & -1 & -(k_x + k_y) \\ 1 & 1 & (k_x + k_y) \\ 1 & 1 & -(k_x + k_y) \\ 1 & -1 & (k_x + k_y) \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega_B \end{bmatrix} \quad (21)$$

$$\begin{cases} \omega_1 = \frac{1}{r}(v_x + v_y + (k_x + k_y)\omega) \\ \omega_2 = \frac{1}{r}(v_x + v_y + (k_x + k_y)\omega) \\ \omega_3 = \frac{1}{r}(v_x + v_y - (k_x + k_y)\omega) \\ \omega_4 = \frac{1}{r}(v_x - v_y + (k_x + k_y)\omega) \end{cases} \quad (22)$$

Therefore,

$$\begin{bmatrix} v_x \\ v_y \\ \omega_B \end{bmatrix} = \frac{r}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -\frac{1}{k_x + k_y} & \frac{1}{k_x + k_y} & -\frac{1}{k_x + k_y} & \frac{1}{k_x + k_y} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} \quad (23)$$

Longitudinal velocity of the four-wheel Mecanum robot:

$$v_x(t) = \frac{r}{4}(\omega_1 + \omega_2 + \omega_3 + \omega_4) \quad (24)$$

Lateral velocity of the four-wheel Mecanum robot:

$$v_y(t) = \frac{r}{4}(-\omega_1 + \omega_2 + \omega_3 - \omega_4) \quad (25)$$

Angular velocity of the four-wheel Mecanum robot:

$$\omega_B(t) = \frac{r}{4(k_x + k_y)}(-\omega_1 + \omega_2 - \omega_3 + \omega_4) \quad (26)$$

III. RPP CONTROLLER

A. CONVENTIONAL PURE PURSUIT ALGORITHM

PP [9] is a geometric path-tracking algorithm that has been extensively applied in mobile robot navigation due to its simplicity, computational efficiency, and ease of implementation. The fundamental principle of the algorithm is to continuously guide the robot toward a lookahead point located on the reference trajectory.

Let the reference trajectory (P) be represented as an ordered sequence of waypoints according to refs. [19,20]:

$$P = \{p_0, p_1, \dots, p_n\} \quad (27)$$

where each waypoint $p_j = (x_j, y_j) \in \mathbb{R}^2$ is defined in the map coordinate frame.

The local planner is defined as a function f that computes the linear and angular velocity commands required for tracking the reference point P_t at time t , as expressed in Eq. (28):

$$(v_t, \omega_t) = f(P_t) \quad (28)$$

The algorithm first determines the closest point p_r on the reference trajectory relative to the current robot position. A lookahead point p_L is then selected based on a predefined lookahead distance L_a . Specifically, p_L is chosen as the first point along the trajectory for which the accumulated path distance from p_r reaches or exceeds L_a , as expressed in Eq. (29):

$$\text{dist}(p_j) = \sqrt{(x_r - x_j)^2 + (y_r - y_j)^2} \quad (29)$$

$$p_j = \begin{cases} \text{dist}(p_{j-1}) < L_a \\ \text{dist}(p_j) \geq L_a \end{cases} \quad (30)$$

To ensure that the center of the robot follows the circular arc toward the target point PPP (where P coincides with the lookahead point p_L), the following conditions must be satisfied:

$$\angle O_B H P = \pi - 2\angle P O_B N = \pi - 2\left(\frac{\pi}{2} - \alpha\right) \quad (31)$$

Applying the law of sines to $\triangle P H O_B$, the following relationship can be obtained:

$$\frac{P O_B}{\sin 2\alpha} = \frac{H O_B}{\sin\left(\frac{\pi}{2} - \alpha\right)} \quad (32)$$

Eq. (32) can be rewritten as follows:

$$\frac{l}{\sin 2\alpha} = \frac{R}{\sin\left(\frac{\pi}{2} - \alpha\right)} \quad (33)$$

Eq. (33) can be simplified as follows:

$$R = \frac{l}{2 \sin \alpha} \quad (34)$$

where l represents the distance between the current robot position and the lookahead point p_L or P. After determining the lookahead point, the curvature of the reference trajectory can be computed based on geometric relationships. Assuming that the robot follows a circular arc of radius (R), the trajectory curvature is defined as ($k = \frac{1}{R}$). Therefore, the curvature can be expressed as given in Eq. (34):

$$k = \frac{1}{R} = \frac{2 \sin \alpha}{l} \quad (35)$$

where k represents the trajectory curvature and serves as the steering command for the four-wheel Mecanum mobile robot. To provide a clearer interpretation of the control law, the lateral error e_l is defined as the perpendicular distance between the robot heading direction and the target point. Based on this definition, the following equation can be derived:

$$\sin \alpha = \frac{e_l}{l} \quad (36)$$

By substituting Eq. (36) into Eq. (35), the trajectory curvature k can be expressed as:

$$k = \frac{2}{l^2} e_l \quad (37)$$

The above formulation reveals that the PP algorithm is essentially equivalent to a proportional steering controller. The steering command is generated from the lateral tracking error measured at the lookahead point, where the effective control gain is given by $\frac{2}{l^2}$.

In practice, this control gain is often adjusted independently to maintain stable trajectory-tracking performance under different operating velocities. Consequently, the lookahead distance l is typically defined as a function of the robot velocity.

In conventional PP implementations, the commanded linear velocity v_c is generally assumed to be constant and represents the nominal operating velocity of the robot. Accordingly, the maximum linear velocity of the robot, denoted by $v_{\max}$, can be expressed as follows according to ref. [21]:

$$v_c = v_{\max} \quad (38)$$

If the robot moves along a trajectory with curvature k at a linear velocity v_c , the following relationship holds:

$$\omega_c = k * v_c \quad (39)$$

Despite its simplicity and widespread adoption, the conventional PP algorithm exhibits several inherent limitations:

The lookahead distance L_a is difficult to tune and significantly affects the tracking performance;

The commanded linear velocity v_c is typically maintained as a constant value;

Velocity and acceleration constraints are not explicitly considered in the control law.

Consequently, the tracking performance and stability of conventional PP may degrade under dynamic operating conditions and on sharp-curvature paths. RPP addresses part of this limitation by retaining the geometric PP steering law while regulating translational speed according to path and environmental constraints.

B. REGULATED PURE PURSUIT

The RPP controller evaluated in this study follows the established Nav2 RPP formulation. It retains the geometric steering law of PP and regulates linear velocity using path curvature, obstacle proximity, and approach-to-goal distance [9,13]. Accordingly, this study adopts the established term RPP. The high-level controller regulates the commanded speed, while the nominal Mecanum kinematic mapping and low-level PID loops execute the resulting body-frame commands.

Curvature of the reference trajectory;

Distance to nearby obstacles;

Remaining distance to the target goal.

1). CURVATURE-BASED HEURISTIC. The curvature-based heuristic controls the robot's linear velocity based on the curvature of the reference trajectory. When the robot follows a path segment with a small curvature radius, the commanded velocity is reduced to improve tracking stability and minimize overshoot in high-curvature regions. The curvature-regulated velocity v'_{curv} is computed from the radius of curvature R of the circular arc between the robot position and the lookahead point $p_L \equiv P$, as expressed below:

$$v'_{curv} = \begin{cases} v \cdot \frac{R}{R_{min}} & \text{if } R \leq R_{min} \\ v & \text{if } R > R_{min} \end{cases} \quad (40)$$

where

R_{min} is the minimum curvature radius threshold.

R denotes the radius of curvature.

v'_{curv} denotes the curvature-regulated linear velocity of the robot.

When $R > R_{min}$, the trajectory corresponds to a straight path or a gentle curve, and the robot maintains its nominal linear velocity. In contrast, when $R \leq R_{min}$, the robot is required to negotiate a sharp turn, and the linear velocity is reduced accordingly.

2). OBSTACLE-PROXIMITY HEURISTIC. The obstacle-proximity heuristic adjusts the robot's linear velocity according to the distance to nearby obstacles. As the robot approaches an obstacle, the commanded velocity is reduced to improve navigation safety and collision avoidance performance.

Let d_0 represent the minimum distance between the robot and the surrounding obstacles. The proximity-regulated velocity v'_{prox} can be expressed as:

$$v'_{prox} = \begin{cases} v \cdot g_d \cdot \frac{d_0}{d_{prox}} & \text{if } d_0 \leq d_{prox} \\ v & \text{if } d_0 > d_{prox} \end{cases} \quad (41)$$

where

d_0 is the distance to the nearest obstacle

d_{prox} is the safety distance threshold;

$g_d \in [0,1]$ is the velocity regulation gain.

3). GOAL-DISTANCE HEURISTIC. The goal-distance heuristic decreases the commanded linear velocity when the robot approaches the goal location. This mechanism improves stopping accuracy and prevents abrupt deceleration near the target.

The corresponding goal-regulated velocity is defined as:

$$v'_{goal} = \begin{cases} v \cdot \frac{d_g}{d_{goal}} & \text{if } d_G \leq d_{goal} \\ v & \text{if } d_G > d_{goal} \end{cases} \quad (42)$$

where

- d_G is the remaining distance to the goal;

- d_{goal} is the goal-distance threshold for velocity reduction.

4). INTEGRATION OF HEURISTICS IN Nav2. Within the Nav2 framework, the commanded linear velocity v_c is first regulated by the curvature and obstacle-proximity heuristics. As a result, the intermediate velocity command can be expressed as follows [20]:

$$v'_{curv,prox} = \min(v'_{curv}, v'_{prox}, v'_{min}) \quad (43)$$

where

- + v'_{min} denotes the minimum allowable velocity after regulation.

Subsequently, the goal-distance heuristic is applied to $v'_{curv,prox}$, and the final commanded velocity is determined as follows:

$$v' = \min(v'_{goal}, v'_{curv,prox}) \quad (44)$$

where v'_{goal} denotes the minimum allowable linear velocity near the goal position.

When the final regulated velocity v' is lower than the nominal velocity command v_c , the controller substitutes v_c with v' , thereby constraining the robot's translational speed.

As a result, the proposed regulation strategy:

- Mitigates overshoot during sharp turning maneuvers;
- Improves navigation safety in environments containing nearby obstacles or human operators.

These heuristics regulate the commanded speed while retaining the nominal Mecanum kinematic mapping. The resulting command is executed through the low-level PID loops, so the measured trajectory reflects the integrated response of the complete RPP-PID architecture.

Table II summarizes the specifications of the DC motors used in the four-wheel Mecanum mobile robot.

IV. PID CONTROLLER

The PID controller has a simple structure, stable performance, and ease of implementation, making it one of the most widely used control strategies in industrial and robotic applications. According to refs. [23–25], the PID controller consists of three main components:

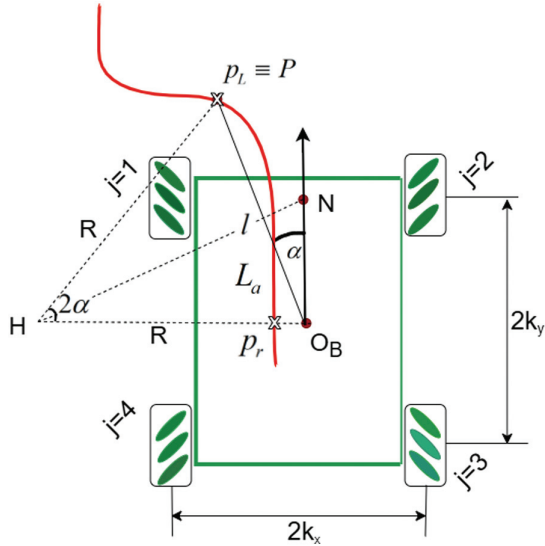
- Proportional term;
- Integral term;
- Derivative term.

The overall PID control structure is illustrated in Fig. 4.

The PID control law is expressed as follows according to ref. [25]:

Table II. Technical specifications of the motors used in the four-wheel Mecanum mobile robot

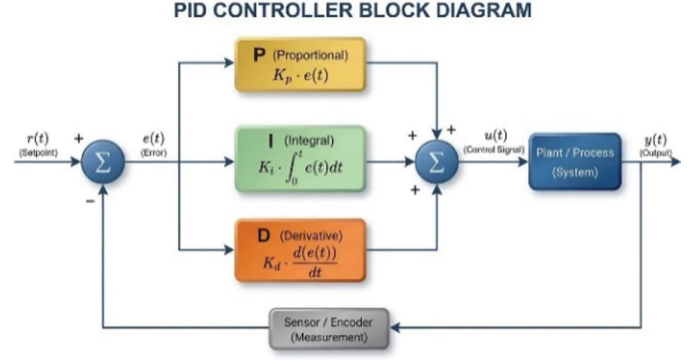
JGB37-520 Motor	
Parameter	Value
Rated voltage	12VDC
No-load speed	333RPM
Loaded speed	~250RPM
No-load current	~120mA
Rated load current	~1A
Gear ratio	30:1
Rated torque	3.5 kg-cm
Maximum stall torque	5 kg-cm
Motor body diameter	37 mm
Motor length	~47 mm
Shaft diameter	6 mm
Gearbox length	22 mm
Encoder	Hall sensor, dual-channel
Encoder pulses per main shaft revolution	$11 \times 30 = 330$ pulses/channel
Encoder supply voltage	3.3–5VDC

**Fig. 4.** Geometric relationship between the robot position, the reference trajectory, and the lookahead point.

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (45)$$

V. EXPERIMENTAL RESULTS AND DISCUSSION

The experiments report conventional PP, Nav2 RPP, and the nominal response of the low-level wheel-speed PID loops. Three navigation scenarios were considered: curved-path tracking, goal reaching, and obstacle avoidance. Both PP and RPP were evaluated in all three navigation scenarios. A separate wheel-speed experiment summarizes the nominal transient response of the PID layer (Fig. 5).

**Fig. 5.** PID controller for the four DC motors of the mobile robot.

- Curvature-based trajectory-tracking experiments;
- Autonomous obstacle-avoidance and navigation experiment in the operating environment;
- Goal-tracking and target convergence experiments;
- Velocity response evaluation of the PID controller for the wheel-drive system.

During the experiments, the robot operated in one bounded indoor workspace, $X = 0\text{--}5.5$ m and $Y = -1.5\text{--}1.5$ m, at moderate operating speeds.

A. EXPERIMENTAL PROTOCOL AND EVALUATION METRICS

All PP-RPP comparisons were conducted on the same four-wheel Mecanum platform using the same map, PID gains, start pose, goal pose, and reference path. Each quantitative tracking condition was repeated five times. This matched design holds the physical platform and path geometry constant and identifies the high-level controller configuration as the principal comparison variable.

The reported metrics are RMSE, maximum tracking error, and maximum lateral error. For each quantitative scenario, RMSE was calculated for every trial and summarized as mean $\pm$ standard deviation over $n = 5$ repeated trials; the maximum-error columns report the corresponding extrema observed across the repeated tests. The quantitative comparison focuses on geometric tracking accuracy, while the obstacle-avoidance experiments are discussed separately as system-level navigation tests.

B. Experimental evaluation of the PP algorithm

+ Curvature-Tracking Experiment:

Figure 6 illustrates the comparison between the reference trajectory (dashed line) and the actual trajectory of the robot (solid line).

Figure 6 shows that the robot closely follows the reference trajectory along the curved path. Tracking errors mainly occur in regions of high curvature; however, they remain relatively small, indicating stable control performance.

Figure 7 shows the lateral tracking error during the curved-path test. Larger deviations occur near the high-curvature portion of the path, where path geometry, localization uncertainty, actuator dynamics, and wheel-ground interaction jointly influence the robot motion. The tracking error remained bounded across the repeated tests, demonstrating stable trajectory tracking under this experimental condition.

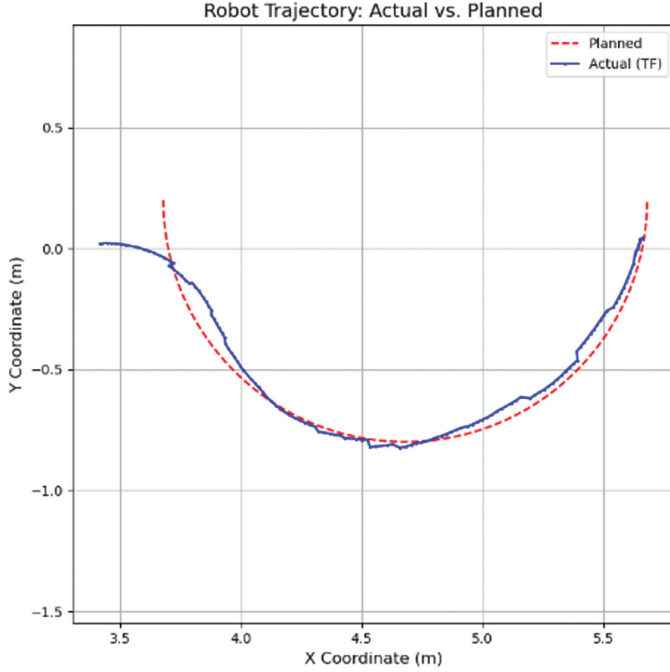


Fig. 6. Experimental curved-path tracking using the PP algorithm.

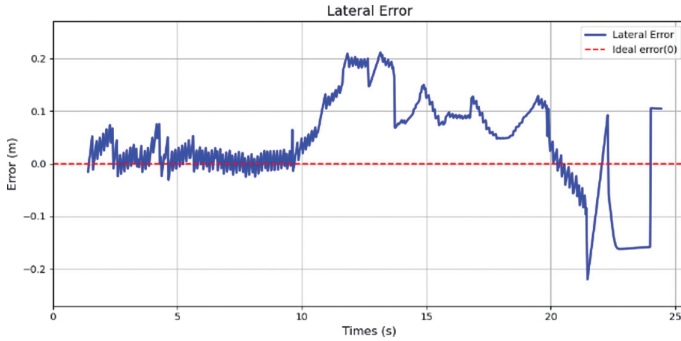


Fig. 7. Lateral error of the mobile robot during curved-path tracking using the PP algorithm.

Figure 8 presents the real-world experimental results using the Nav2 framework integrated with SLAM. The robot is capable of accurate localization, navigation, and obstacle avoidance in the experimental environment while still maintaining stable trajectory-tracking performance.

+ Obstacle-Avoidance Experiment:

The conventional PP controller was also evaluated in the obstacle-avoidance scenario using the same robot hardware,

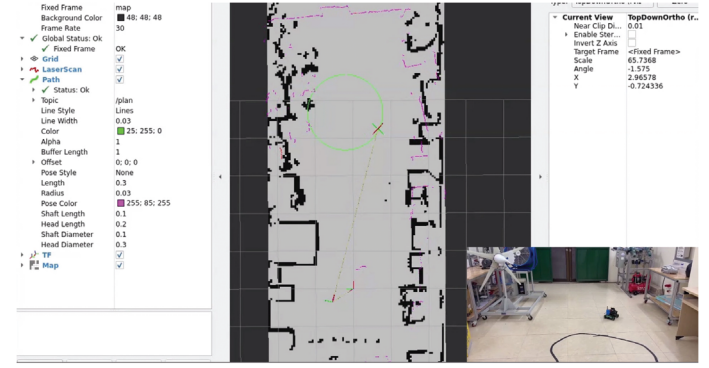


Fig. 8. Real-world experimental results of the mobile robot during curved-path tracking using the PP algorithm.

map, PID gains, start-goal pair, and reference path as the RPP controller. Obstacle avoidance was evaluated as an end-to-end system task using the complete Nav2-SLAM-planning-controller pipeline. Its results are therefore presented separately from the geometric tracking metrics in Table III.

+ Goal-Tracking Experiment of the Mobile Robot:

Figure 9 presents the planned and actual trajectories of the mobile robot during the goal-tracking experiment. As observed, the actual trajectory closely follows the reference path throughout the motion. Minor deviations are only noticeable during the initial stage and near the target position. These results indicate that the PP algorithm performs effectively in goal-tracking tasks involving relatively simple trajectories and can successfully guide the robot toward the desired destination.

Figure 10 illustrates the lateral tracking error over time. The error oscillates around the ideal value ($e = 0$) with a relatively small magnitude throughout most of the experiment. Several local fluctuations and peak errors can be observed during the motion, reflecting the combined effects of localization uncertainty, sensor noise, actuator dynamics, and wheel-ground interaction. The error remains bounded throughout the experiment, indicating stable goal-tracking behavior under the tested condition.

Figure 11 shows the real-world experimental results obtained using the Nav2 framework integrated with SLAM. The robot is capable of accurately localizing itself, planning a feasible path, and navigating toward the target position in a stable manner. The experimental results demonstrate that the PP algorithm can effectively accomplish goal-tracking tasks in practical environments while maintaining reliable localization and trajectory-following performance.

C. EXPERIMENTAL EVALUATION OF THE RPP Configuration

+ Curvature-Tracking Experiment:

Table III. Quantitative performance comparison between PP and RPP under matched conditions (mean $\pm$ standard deviation over $n = 5$ trials)

Alg.	Test scenario	RMSE (m)	Max error (m)	Max lateral error (m)
PP	Curvature tracking	0.12 ± 0.03	0.28	0.08
PP	Goal tracking	0.10 ± 0.02	0.22	0.06
RPP	Curvature tracking	0.08 ± 0.01	0.18	0.05
RPP	Goal tracking	0.05 ± 0.02	0.11	0.02

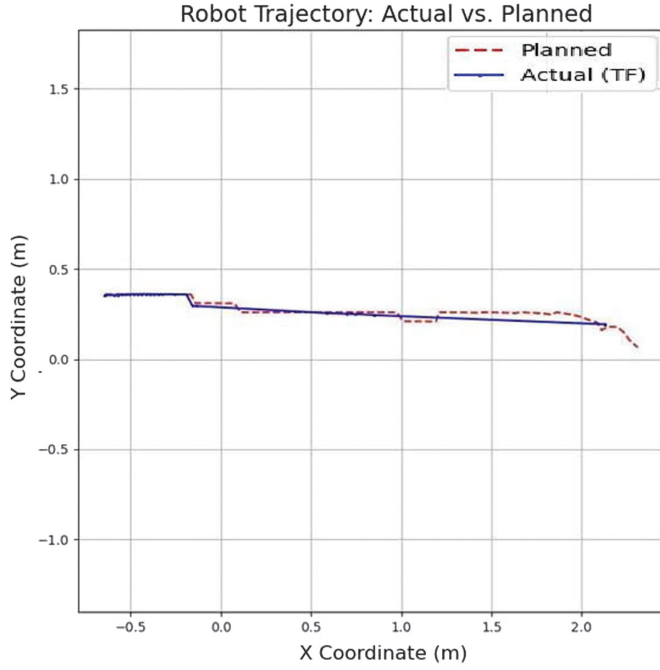


Fig. 9. Experimental trajectory of the mobile robot during goal-tracking using the PP algorithm.

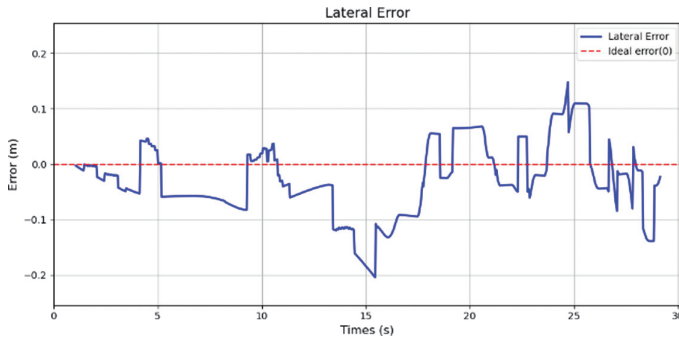


Fig. 10. Lateral error of the mobile robot during goal-tracking using the PP algorithm.

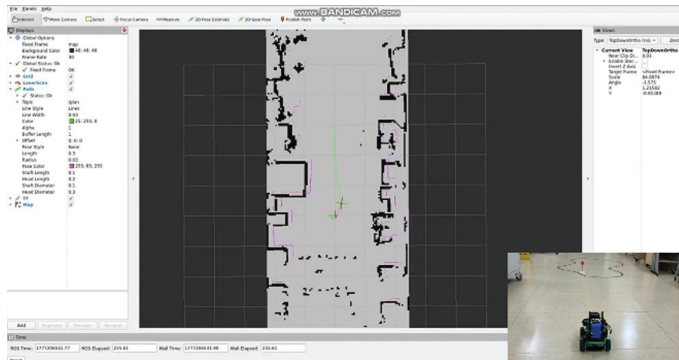


Fig. 11. Real-world experimental results of the mobile robot during goal-tracking using the PP algorithm.

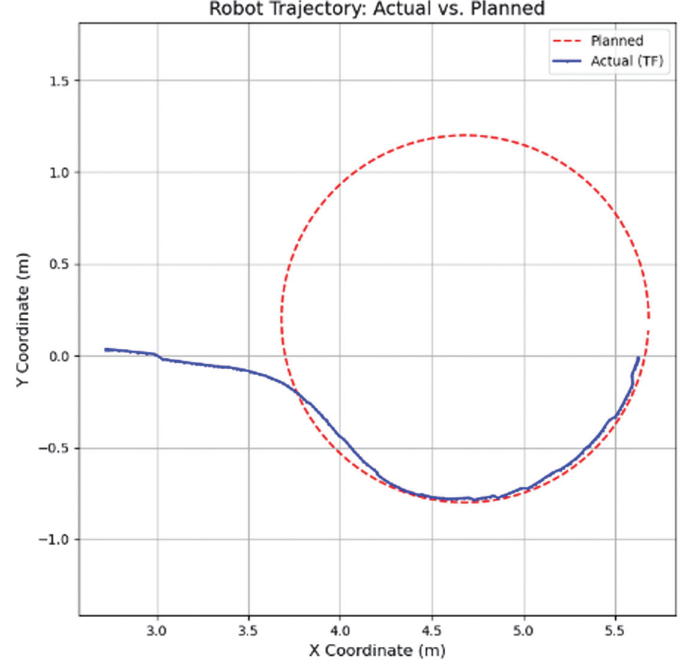


Fig. 12. Experimental curved-path tracking of the mobile robot using the RPP controller.

Figure 12 shows that the robot demonstrates accurate trajectory-tracking capability along the reference trajectory on the curved arc-shaped path. The actual trajectory generally remains close to the planned trajectory, though small tracking errors persist in regions of high curvature during turning maneuvers.

Figure 13 shows the robot's lateral error over time. The error oscillates around the ideal value and rapidly converges to a stable state after trajectory adjustments, indicating that the control algorithm maintains good tracking stability.

Figure 14 presents the real-world experimental results using the Nav2 framework integrated with SLAM. The robot demonstrates stable localization, navigation, and obstacle-avoidance capabilities in the experimental environment while maintaining effective trajectory-tracking performance.

Across five matched curved-path trials, RPP reduced the RMSE from 0.12 ± 0.03 m to 0.08 ± 0.01 m. The consistent reduction demonstrates the practical benefit of curvature-aware velocity regulation on the tested high-curvature trajectory: when

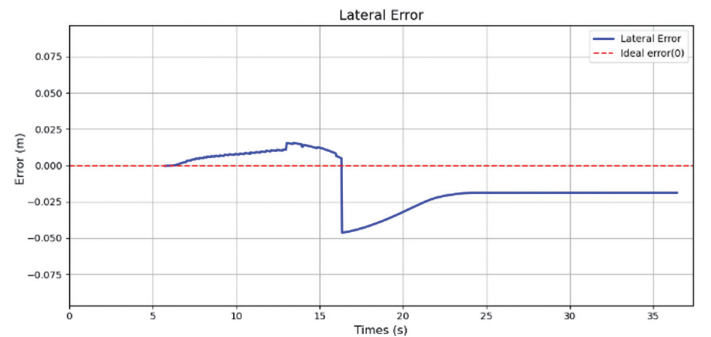


Fig. 13. Lateral error of the mobile robot during curved-path tracking using the RPP controller.

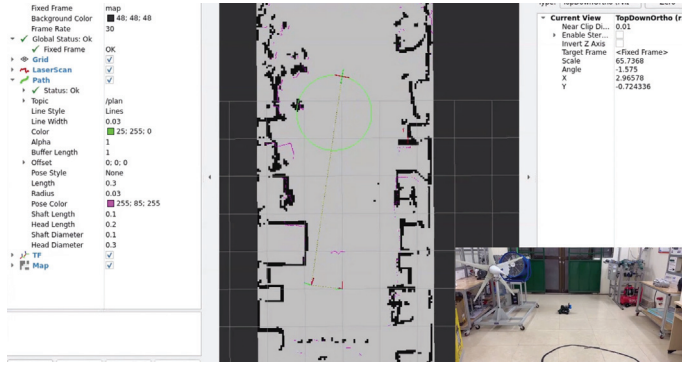


Fig. 14. Real-world experimental results of the mobile robot during curved-path tracking using the RPP controller.

the estimated turning radius decreases, RPP lowers the commanded linear speed and thereby improves end-to-end geometric tracking under the matched operating condition.

+ Obstacle-Avoidance Experiment:

Figures 15–17 present the RPP obstacle-avoidance experiment. A corresponding PP experiment was conducted under the

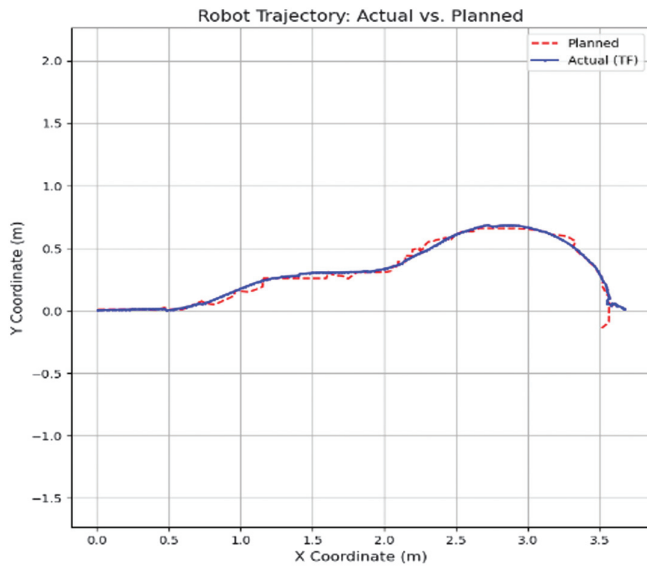


Fig. 15. Experimental obstacle-avoidance scenario of the mobile robot using the RPP controller.

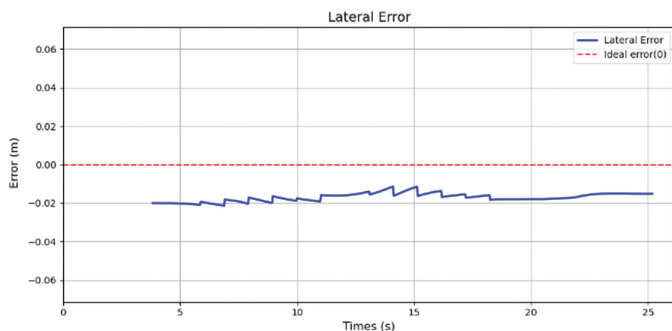


Fig. 16. Lateral error of the mobile robot during obstacle avoidance using the RPP controller.

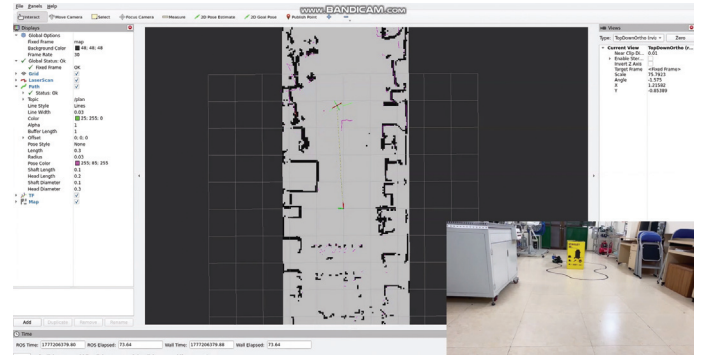


Fig. 17. Real-world experimental results of the mobile robot during obstacle avoidance using the RPP controller.

matched hardware, map, PID, start-goal, and reference-path conditions described in Section 5.1. In both cases, the complete Nav2-SLAM-controller pipeline localized the robot, updated the plan, and generated velocity commands during obstacle traversal. This system-level task characterizes the end-to-end navigation behavior of the two controller configurations.

Within this pipeline, RPP applies obstacle-proximity velocity regulation to reduce the commanded speed near inflated obstacle costs. The obstacle experiments establish executable navigation for both controllers under the matched setup, while the quantitative controller comparison in Table III remains focused on curved-path and goal-reaching tracking errors.

+ Goal-Tracking Experiment of the Mobile Robot:

Figure 18 shows that the robot closely follows the reference trajectory along the straight path, with the actual trajectory nearly overlapping the planned trajectory. Figure 19 shows a very small, stable lateral error around the ideal value, demonstrating the control system's accurate trajectory-tracking capability. Figure 20 presents the real-world experimental results, showing that the robot operates stably in the experimental environment while maintaining reliable localization and navigation performance with the Nav2 framework integrated with SLAM.

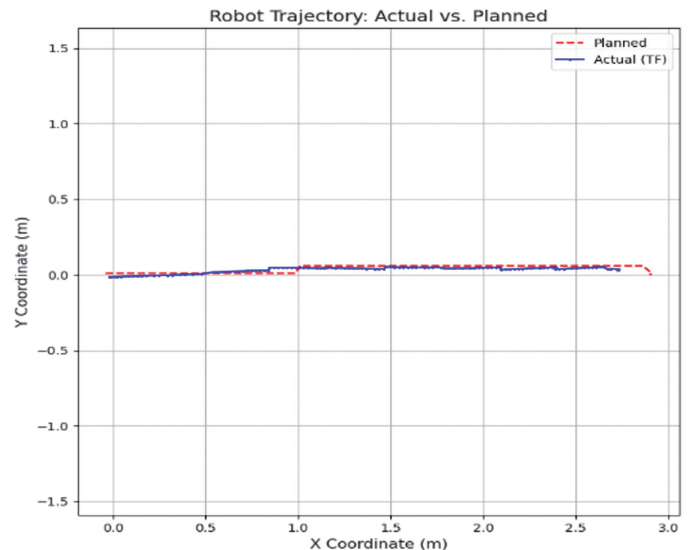


Fig. 18. Experimental goal-tracking scenario of the mobile robot using the RPP controller.

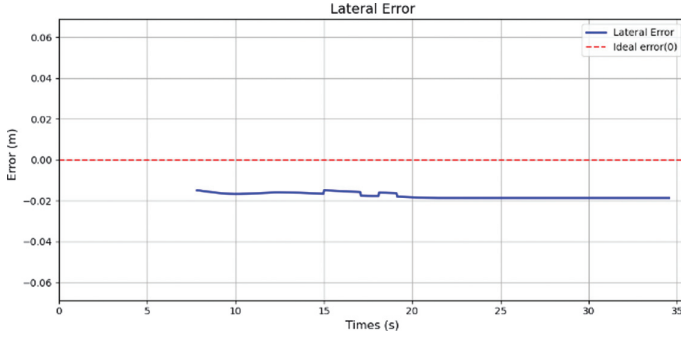


Fig. 19. Lateral error of the mobile robot during goal tracking using the RPP controller.

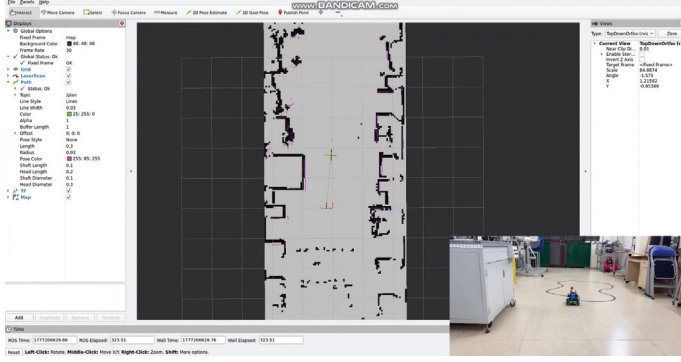


Fig. 20. Real-world experimental results of the mobile robot during goal tracking using the RPP controller.

The goal-distance heuristic enables smooth deceleration near the target position, thereby reducing oscillation and improving stopping accuracy during final convergence.

Table III summarizes five matched trials for each quantitative scenario. RPP reduced the mean RMSE from 0.12 ± 0.03 m to 0.08 ± 0.01 m in curved-path tracking and from 0.10 ± 0.02 m to 0.05 ± 0.02 m in goal tracking. The corresponding maximum tracking and lateral errors were also lower for RPP. Obstacle avoidance was evaluated with both controllers under matched conditions and is discussed separately because the complete planning, localization, perception, and control pipeline remained active.

D. NOMINAL WHEEL-SPEED PID Response

The blue curve represents the reference signal, while the orange, purple, pink, and red curves correspond to Motors 1, 2, 3, and 4, respectively, associated with the Mecanum wheels of the mobile robot.

Figures 21 and 22 and Table IV characterize the nominal dynamic response of the four wheel-speed control loops using fixed gains $K_p = 1.55$, $K_i = 6.55$, and $K_d = 0.045$. The STM32 PID loops operated with a 10 ms sampling period, corresponding to 100 Hz. The measured wheel-speed traces converge toward the reference command with the reported rise time, settling time, overshoot, and steady-state error, supporting the coordinated low-level wheel-speed regulation required by the high-level navigation layer.

Table IV summarizes the nominal transient characteristics and the multi-rate controller settings. The low-level STM32 wheel-speed PID loop uses a 10 ms sampling period and therefore operates at 100 Hz, whereas the high-level Nav2 RPP controller

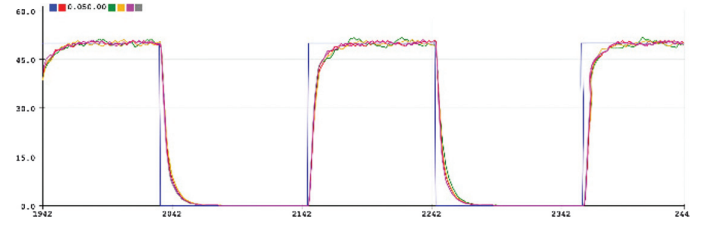


Fig. 21. PID controller of the four-wheel Mecanum mobile robot.

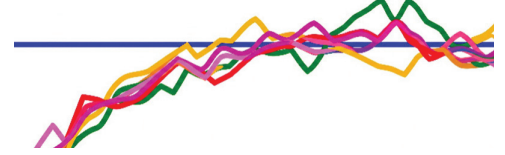


Fig. 22. Magnified view of the PID controller response during the time interval from 1942 ms to 2042 ms.

Table IV. Nominal dynamic-response characteristics of the PID wheel-speed controller

Control parameter	Value
Proportional gain (K_p)	1.55
Integral gain (K_i)	6.55
Derivative gain (K_d)	0.045
Rise time	0.18 s
Settling time	0.42 s
Maximum overshoot	3.8 %
Steady-state error	<2%
Encoder resolution	330 pulses/rev
Low-level PID sampling period	10 ms
Low-level PID control frequency	100 Hz
High-level Nav2 RPP control frequency	50 Hz

operates at 50 Hz. These values refer to different layers of the distributed control architecture.

During the physical experiments, the distributed control stack executed online with the Nav2 RPP controller running at 50 Hz on the Jetson Nano and the wheel-speed PID loops running at 100 Hz on the STM32. These coordinated execution rates enabled continuous high-level path tracking and low-level wheel-speed regulation.

VI. CONCLUSION

This paper presents an experimental evaluation of an RPP-PID architecture on a four-wheel Mecanum mobile robot. The high-level RPP controller was implemented as an RPP configuration in Nav2 on an NVIDIA Jetson Nano, while an STM32 microcontroller executed four wheel-speed PID loops.

Across five matched curved-path and goal-reaching trials, RPP produced lower geometric tracking-error values than PP. The PP and RPP experiments used the same hardware, map, PID gains, start and goal poses, and reference paths. Obstacle avoidance was also performed with both controllers under matched conditions and demonstrates end-to-end operation of the complete Nav2-SLAM-controller pipeline.

The lower tracking errors represent the integrated response of curvature-aware velocity regulation, nominal Mecanum kinematics, and low-level wheel-speed control under practical wheel-ground interaction, localization uncertainty, actuator dynamics, and communication effects.

The results demonstrate the feasibility of onboard closed-loop operation and show that the tested Nav2 RPP configuration achieves lower geometric tracking errors than conventional PP under the matched indoor conditions.

The present evaluation focuses on matched PP-RPP tracking trials, system-level obstacle avoidance, and nominal PID wheel-speed response in one indoor workspace at moderate operating speeds. Broader validation will examine multiple surfaces and payloads, controlled speed-curvature combinations, disturbance-oriented PID conditions, additional tracking controllers, and detailed timing and communication profiles.

ACKNOWLEDGMENT

This research is funded by the Thuyloi University Science, Technology and Innovation Development Fund under Project No. CS2026-13, and jointly sponsored by Central Government Program for Guiding Local Regional Innovation System Development Project: 202502ZYDF029, Science and Technology Innovation Key R&D Program of Chongqing: CSTB2023TIAD-STX0019. The authors gratefully acknowledge Thuyloi University for providing financial support for this research project.

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no conflicts of interest to report regarding the present study.

REFERENCES

- [1] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, 2nd ed. in Intelligent Robotics and Autonomous Agents series. Cambridge, MA, USA: MIT Press, 2011.
- [2] G. Cook and F. Zhang, *Mobile Robots: Navigation, Control and Sensing, Surface Robots and AUVs*, 1st ed. Hoboken, NJ: Wiley, 2019, DOI: [10.1002/9781119534839](https://doi.org/10.1002/9781119534839).
- [3] "2D Rotations | Articulated Robotics." [Online]. Available: <https://articulatedrobotics.xyz/tutorials/coordinate-transforms/rotation-matrices-2d/>, Accessed on: Mar. 20, 2026
- [4] J. C. O. Hernández and D. I. R. Almeida, "Kinematic control in a four-wheeled Mecanum mobile robot for trajectory tracking," *J. Eng.*, vol. 2024, no. 9, p. e70006, Sep. 2024, DOI: [10.1049/tje2.70006](https://doi.org/10.1049/tje2.70006).
- [5] "(PDF) Kinematic Model of a Four Mecanum Wheeled Mobile Robot," *ResearchGate*, DOI: [10.5120/19804-1586](https://doi.org/10.5120/19804-1586).
- [6] M. T. Ferrari *et al.*, "A comparative study of linear MPC and feedback linearization for an omnidirectional robot," in *2025 IEEE International Conference on Advanced Robotics (ICAR)*, Oct. 2025, pp. 360–365, DOI: [10.1109/ICAR65334.2025.11338625](https://doi.org/10.1109/ICAR65334.2025.11338625).
- [7] C.-L. Hwang and Y.-H. Li, "Fuzzy finite-time pose tracking control with nonlinear output scale factor for a four mecanum-wheel vehicle in smooth indoor and complex outdoor environments," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 56, no. 1, pp. 148–163, Jan. 2026, DOI: [10.1109/TSMC.2025.3622961](https://doi.org/10.1109/TSMC.2025.3622961).
- [8] C. M. Kumile and N. S. Tlale, "Intelligent distributed fuzzy logic control system (IDFLCS) of a mecanum wheeled autonomous guided vehicle," in *IEEE International Conference Mechatronics and Automation*, 2005, Jul. 2005, pp. 131–137, vol. 1, DOI: [10.1109/ICMA.2005.1626535](https://doi.org/10.1109/ICMA.2005.1626535).
- [9] S. Macenski *et al.*, "Regulated pure pursuit for robot path tracking," May 2023, arXiv: [arXiv:2305.20026](https://arxiv.org/abs/2305.20026). DOI: [10.48550/arXiv.2305.20026](https://doi.org/10.48550/arXiv.2305.20026).
- [10] Z. Yuan *et al.*, "Trajectory tracking control of a four mecanum wheeled mobile platform: an extended state observer-based sliding mode approach," *IET Control Theory Appl.*, vol. 14, no. 3, pp. 415–426, 2020, DOI: [10.1049/iet-cta.2018.6127](https://doi.org/10.1049/iet-cta.2018.6127).
- [11] D. Wang *et al.*, "Sliding mode observer-based model predictive tracking control for mecanum-wheeled mobile robot," *ISA Trans.*, vol. 151, pp. 51–61, Aug. 2024, DOI: [10.1016/j.isatra.2024.05.050](https://doi.org/10.1016/j.isatra.2024.05.050).
- [12] N. Banik, J. Ghommam, and M. H. Rahman, "Finite-time pure pursuit guidance control of a four mecanum wheeled mobile robot with active disturbance rejection," *IEEE Access*, vol. 13, pp. 39214–39234, 2025, DOI: [10.1109/ACCESS.2025.3546864](https://doi.org/10.1109/ACCESS.2025.3546864).
- [13] "Nav2 - Nav2 1.0.0 documentation." [Online]. Available: <http://nav2.org>, Accessed on: Mar. 30, 2026
- [14] E. Malayjerdi, H. Kalani, and M. Malayjerdi, "Self-tuning fuzzy PID control of a four-mecanum wheel omni-directional mobile platform," in *Iranian Conference on Electrical Engineering (ICEE)*, May 2018, pp. 816–820. doi: [10.1109/ICEE.2018.8472568](https://doi.org/10.1109/ICEE.2018.8472568).
- [15] N. H. Thai, T. T. K. Ly, and L. Q. Dzong, "Trajectory tracking control for mecanum wheel mobile robot by time-varying parameter PID controller," *Bull. Electr. Eng. Inform.*, vol. 11, no. 4, pp. 1902–1910, Aug. 2022, DOI: [10.11591/eei.v11i4.3712](https://doi.org/10.11591/eei.v11i4.3712).
- [16] K.-L. Han, H. Kim, and J. S. Lee, "The sources of position errors of omni-directional mobile robot with Mecanum wheel," in *2010 IEEE International Conference on Systems, Man and Cybernetics*, Oct. 2010, pp. 581–586, DOI: [10.1109/ICSMC.2010.5642009](https://doi.org/10.1109/ICSMC.2010.5642009).
- [17] H. Li *et al.*, "Design and implementation of omnidirectional mobile robot for materials handling among multiple workstations in manufacturing factories," *Electronics*, vol. 12, no. 22, 4693, Nov. 2023, DOI: [10.3390/electronics12224693](https://doi.org/10.3390/electronics12224693).
- [18] J. Li *et al.*, "Autonomous tracking control for four-wheel independent steering robot based on improved pure pursuit," vol. 29, no. 4, pp. 466–473.
- [19] S. Zhao *et al.*, "Biomimetic adaptive pure pursuit control for robot path tracking inspired by natural motion constraints," *Biomimetics*, vol. 9, no. 1, p. 41, Jan. 2024, DOI: [10.3390/biomimetics9010041](https://doi.org/10.3390/biomimetics9010041).
- [20] "DWPP: Dynamic Window Pure Pursuit Considering Velocity and Acceleration Constraints." [Online]. Available: <https://arxiv.org/html/2601.15006v1#bib.bib14>, Accessed on: Apr. 24, 2026
- [21] "DWPP: Dynamic Window Pure Pursuit Considering Velocity and Acceleration Constraints." [Online]. Available: <https://arxiv.org/html/2601.15006v1#bib.bib14>, Accessed on: Apr. 23, 2026
- [22] "DWPP: Dynamic Window Pure Pursuit Considering Velocity and Acceleration Constraints." [Online]. Available: <https://arxiv.org/html/2601.15006v1#bib.bib14>, Accessed on: Apr. 24, 2026
- [23] Y. Huang *et al.*, "Path tracking based on improved pure pursuit model and PID," in *2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology (ICCSIT)*, Oct. 2020, pp. 359–364, DOI: [10.1109/ICCSIT50869.2020.9368694](https://doi.org/10.1109/ICCSIT50869.2020.9368694).
- [24] A. M. Díaz and A. Rodríguez-Angeles, "A PI2D feedback control type for second order systems," in *Advances in PID Control*. V. D. Yurkevich, Ed. Rijeka, Croatia: IntechOpen, 2011, pp. 65–84, DOI: [10.5772/10660](https://doi.org/10.5772/10660).
- [25] H. Mikulas *et al.*, "Constrained Series PI, PID and PIDA Controller Design Inspired by Ziegler–Nichols," *Power Electron. Drives*, vol. 9, no. 1, Wrocław University of Science and Technology, 2024, pp. 331–346. DOI: [10.2478/pead-2024-0021](https://doi.org/10.2478/pead-2024-0021).