Experimental Validation of an Adaptive Controller for a Mecanum-Wheel Robot with Unknown Center-of-Gravity Offset and Slope Inclination

Chawannat Chaichumporn[1], Supaluk Prapan[2], Nghia Thi Mai[3],
Md Abdus Samad Kamal[4], Iwanori Murakami[5], Kou Yamada[6]
Faculty of Engineering, Thai-Nichi Institute of Technology, Bangkok, 10250, Thailand[1,2]
Graduate School of Science and Technology, Gunma University, Kiryu, 376-8515, Japan[1,4,5,6]
Faculty of Electronics Engineering 1, Posts and Telecommunications Institute of Technology, Hanoi, Vietnam[3]

***Abstract*—High-precision path tracking for a Four-Mecanum-Wheel Mobile Robot (FMWMR) is challenged by real-world factors such as payload-induced shifts in the center-of-gravity (CoG) and operation on inclined surfaces. These uncertainties introduce complex, coupled dynamic forces that degrade the performance of conventional controllers. This study addresses this problem with a Model Reference Adaptive Controller (MRAC), which learns and compensates for these unpredictable dynamic effects in real time. To ensure effective operation on physical hardware, the controller incorporates practical solutions for motor friction and control signal stability. The proposed approach is validated through implementation of the MRAC on a Rosmaster X3 robot. A performance comparison is made against a well-tuned Proportional-Integral-Derivative (PID) controller is conducted across twelve distinct scenarios. The results show that the adaptive controller reduced the position Root Mean Square Error (RMSE) by an average of 52.7% and the Integral of Time-weighted Absolute Error (ITAE) by 61.5%. This work validates the MRAC as a powerful and robust solution for robots operating in unpredictable environments.**

***Keywords*—Model reference adaptive control; Mecanum Wheel Robot; center of gravity offset**

I. Introduction

Omnidirectional mobile robots equipped with Mecanum wheels are integral to modern logistics, manufacturing and service applications that require high maneuverability in constrained environments [1], [2], [3]. However, achieving precise trajectory tracking on these platforms is hindered by the dynamic uncertainties encountered in practice. Two particularly consequential factors are unknown shifts in the center of gravity (CoG) due to variable payloads and gravitational components with coupled disturbance torques that arise during operation on inclined terrain. [4], [5], [6]. When these conditions are present simultaneously, the performance and robustness of conventional fixed-gain controllers can be severely compromised.

Conventional approaches to this tracking problem often employ linear controllers, such as the Proportional-Integral-Derivative (PID) controller, due to their simplicity and effectiveness under nominal conditions [7]. However, the performance of these fixed-gain controllers is substantially degraded when the robot dynamic parameters deviate from the nominal model [8]. To address bounded uncertainties, robust control strategies like sliding mode control (SMC) have been proposed [9]. Although effective at rejecting bounded disturbances, SMC often induces high-frequency chattering in the control signal, leading to increased energy consumption and mechanical wear. Mitigating this chattering typically requires conservative gain tuning, which in turn sacrifices tracking performance.

Although more advanced research has focused on adaptive and predictive strategies, these methods often address dynamic uncertainties in isolation. For example, some adaptive methods successfully compensate for payload variations, but their effectiveness is often limited to demonstrations on flat terrain [10]. In contrast, other studies have developed predictive strategies for navigation on sloped surfaces but do so under the critical assumption of a known and fixed CoG [11], [12]. A key limitation in the existing literature is the failure to explicitly address the significant nonlinear coupling torques that arise when a robot with an off-center CoG operates on an inclined plane. This unresolved coupling motivates the present work, as a unified framework that handles both uncertainties simultaneously remains an open challenge.

This research builds on the theoretical framework established in [13], where an MRAC was proposed to ensure the closed-loop stability of a dynamic model incorporating CoG and slope effects. However, a significant gap often exists between theoretical stability and practical performance due to unmodeled real-world dynamics. To bridge this sim-to-real gap, this study presents a rigorous experimental validation of the controller on an FMWMR. Implemented on a Rosmaster X3 platform, the controller's performance under simultaneous CoG and slope uncertainties is quantitatively assessed against a carefully tuned PID benchmark using the root mean square error (RMSE) of the position, heading error, and the integral of time-weighted absolute error (ITAE) as metrics [14], [15], [16]. The implementation is augmented with practical measures to address hardware non-idealities, including deadzone compensation, command-rate limiting, and an adaptation soft start [17], [18], [19]. To ensure that all experiments are dynamically feasible and repeatable, smooth reference trajectories are generated using quintic splines to reduce jerk transients [20], [21].

This study addresses the identified sim-to-real gap through the experimental implementation and validation of the proposed MRAC on an FMWMR. The contributions are threefold:

- The experimental validation of the model framework

from [13] on a physical robotic platform. The controller's practical effectiveness is demonstrated under challenging conditions that involve simultaneous coupled dynamic uncertainties.

- The presentation of a methodology for practical implementation. This includes essential compensation techniques for real-world hardware non-idealities, such as motor stiction and actuator saturation, which are critical for bridging the sim-to-real gap.
- The quantitative validation of the adaptive controller against a well-tuned PID benchmark. The results provide definitive proof of substantial improvements in tracking accuracy and disturbance rejection, particularly when the system is challenged by the complex coupling of inertial and gravitational forces.

This study is organized as follows: The system modeling and adaptive controller design are summarized in Section II. The experimental setup and implementation details are described in Section III. The experimental procedure and evaluation metrics are defined in Section IV. The results and discussion are reported in Section V. Conclusions and directions for future work are provided in Section VI.

II. System Modeling and Adaptive Controller Design

This section summarizes the controller design based on the comprehensive dynamic model of the FMWMR proposed in [13]. The model captures the effects of an offset CoG and the gravitational forces on an inclined plane.

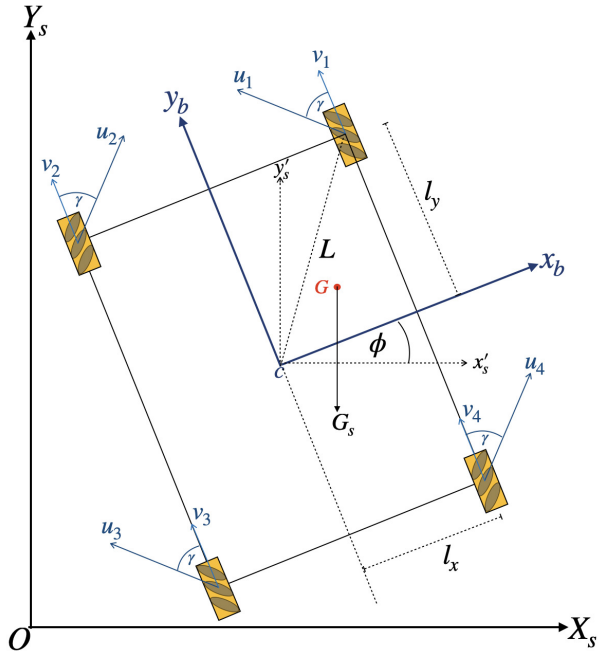

Fig. 1. A schematic of the FMWMR. The Geometric center (c), Center of gravity (G), and Wheel arrangement are indicated.

A. Kinematic Model

Consider a schematic of the FMWMR in Fig. 1. The relationship between the global frame velocity $\dot{q}$ and the body-frame velocity v_b is defined by Eq. (1):

$$\dot{q} = R(\phi)v_b, \tag{1}$$

where, $\dot{q}$ is the time derivative of the robot's pose vector in the global frame $\{X_s, O, Y_s\}$ and the rotation matrix $R(\phi)$. The robot's pose is given by the vector $q = [x, y, \phi]^{\mathrm{T}}$, where (x, y) are the Cartesian coordinates of the robot's geometric center and ϕ is the yaw angle. The body-frame velocity vector is $v_b = [v_{bx}, v_{by}, \omega_b]^{\mathrm{T}}$, which consists of the forward velocity v_{bx}, lateral velocity v_{by}, and angular velocity ω_b relative to the robot's local frame $\{x_b, c, y_b\}$. The matrix $R(\phi)$ is the rotation matrix that transforms body-frame velocities to the global frame, defined as Eq. (2):

$$R(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2}$$

The forward kinematic model of FMWMR defines the relationship between the angular velocities of the individual wheels and the resulting linear velocities of the robot's body frame. This mapping from wheel rotational speeds ω_{wh} to body-frame velocities v_b is given by Eq. (3):

$$v_b = J\omega_{wh}, \tag{3}$$

where, $\omega_{wh} = [\omega_1, \omega_2, \omega_3, \omega_4]^{\mathrm{T}}$ is the vector of angular velocities of the four wheels and J is the Jacobian matrix given by:

$$J = \frac{R_w}{4}\begin{bmatrix} -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 \\ \frac{1}{l_x+l_y} & -\frac{1}{l_x+l_y} & -\frac{1}{l_x+l_y} & \frac{1}{l_x+l_y} \end{bmatrix},$$

R_w is the radius of the Mecanum wheels, while l_x and l_y represent the longitudinal and lateral distances from the geometric center of the robot to the center of each wheel, respectively.

B. Dynamic Model

The dynamic model of the FMWMR is derived using a Newton-Euler formulation based on two key assumptions: the robot is a rigid body, and the wheels maintain a pure rolling contact without slip. Other effects, such as actuator dynamics and viscous friction, are treated as unmodeled disturbances to be compensated for by the controller's adaptive and feedback mechanisms. The resulting dynamic model in the robot's body frame is expressed as Eq. (4):

$$H\dot{v}_b + Cv_b + G_b = F_t, \tag{4}$$

where, H is the inertia matrix that explicitly includes the CoG offset written as Eq. (5):

$$H = \begin{bmatrix} m & 0 & -my_g \\ 0 & m & mx_g \\ -my_g & mx_g & I_z \end{bmatrix}, \tag{5}$$

C is the Coriolis and centrifugal matrix and is written as Eq. (6):

$$C = \begin{bmatrix} 0 & -m\omega_b & -m\omega_b y_g \\ m\omega_b & 0 & m\omega_b x_g \\ m\omega_b y_g & -m\omega_b x_g & 0 \end{bmatrix}, \tag{6}$$

G_b is the gravitational vector and is given by Eq. (7):

$$G_b = mg\sin\alpha \begin{bmatrix} \sin\phi \\ \cos\phi \\ x_g\cos\phi - y_g\sin\phi \end{bmatrix}, \tag{7}$$

F_t is the vector of control forces and torque applied by the wheels, m is the total mass of the robot, I_z is the moment of inertia about the vertical axis passing through the CoG, g is the acceleration due to gravity, α is slope angle of the surface and (x_g, y_g) represent the coordinates of the robot's CoG within the body frame, measured relative to the geometric center.

C. Adaptive Controller Design

The control objective is to achieve asymptotic tracking of a desired trajectory, q_d, with the robot's pose, q, in the presence of parametric uncertainties within the dynamic model. The MRAC scheme is designed to meet this objective. The core of this approach is to establish a stable, reference error dynamic, which is accomplished through the formulation of a composite filtered error signal. The first step in this design is to mathematically define this error and its dynamics.

1) Error dynamics formulation: A key challenge in controlling dynamic systems is that the control inputs F_t directly influence accelerations $\dot{v}_b$ rather than the pose q. This structure creates a second-order tracking problem. To simplify this, a filtered tracking error s is introduced, which effectively transforms the second-order problem into a first-order stabilization problem. This composite signal is defined by Eq. (8):

$$s = v_b - v_r, \tag{8}$$

where, v_b is the robot's actual body-frame velocity and v_r is the reference velocity vector is strategically designed as Eq. (9):

$$v_r = R(\phi)^{\mathrm{T}}\left(\dot{q}_d - \Lambda e\right), \tag{9}$$

$e = q - q_d$ is the pose error and Λ is a positive-definite gain matrix. This construction is critical because it ensures that if the controller drives $s \to 0$, the pose error, e, is guaranteed to converge exponentially according to the stable dynamic $\dot{e} = -\Lambda e$.

The error dynamics for control design are derived by differentiating Eq. (8) and substituting the system's dynamic model, yielding Eq. (10):

$$H\dot{s} + Cs = F_t - (H\dot{v}_r + Cv_r + G_b). \tag{10}$$

This equation provides a clear structure for the control design. However, the term $(H\dot{v}_r + Cv_r + G_b)$ contains the unknown physical parameters. The next step, therefore, is to restructure this term to systematically handle these uncertainties.

2) System and control law: To address the parametric uncertainty in the system dynamics, a linear parameterization is performed. This crucial step reformulates the dynamic terms containing unknown physical properties into a structure that is linear in the parameters. This separates the dynamics into two distinct components: a vector θ containing all the unknown physical parameters and a known regressor matrix Y constructed from measurable signals of the reference trajectory. This reformulation is expressed as Eq. (11):

$$H\dot{v}_r + Cv_r + G_b = Y\theta, \tag{11}$$

where, the vector of unknown parameters θ, and known regressor Y and defined as Eq. (12):

$$\theta = [m, I_z, mx_g, my_g, m\sin\alpha, mx_g\sin\alpha, my_g\sin\alpha]^{\mathrm{T}}, \tag{12}$$

and

$$Y = \begin{bmatrix} y_{11} & y_{12} & 0 \\ 0 & 0 & \dot{\omega}_{r\phi} \\ 0 & y_{23} & y_{33} \\ y_{14} & 0 & y_{34} \\ g\sin\phi & g\cos\phi & 0 \\ 0 & 0 & g\cos\phi \\ 0 & 0 & -g\sin\phi \end{bmatrix}^{\mathrm{T}},$$

where, y_{ij} are functions of the reference trajectory and its derivatives are defined as:

$$\begin{aligned} y_{11} &= \dot{v}_{rx} - v_{ry}\omega_{r\phi}, & y_{12} &= \dot{v}_{ry} - v_{rx}\omega_{r\phi}, \\ y_{23} &= \dot{\omega}_{r\phi} + \omega_{r\phi}^2, & y_{33} &= \dot{v}_{ry} + v_{ry}\omega_{r\phi}, \\ y_{14} &= -(\dot{\omega}_{r\phi} + \omega_{r\phi}^2), & y_{34} &= -(\dot{v}_{rx} + v_{rx}\omega_{r\phi}). \end{aligned}$$

Based on this equation, the control law is formulated as Eq. (13):

$$F_t = Y\hat{\theta} - Ks, \tag{13}$$

where, $\hat{\theta}$ is the estimation of θ and K is a positive-definite feedback gain matrix. This law consists of a feed-forward term, $Y\hat{\theta}$, to cancel the estimated system dynamics, and a stabilizing feedback term -Ks. The parameter estimates are updated using the following adaptation law Eq. (14):

$$\dot{\hat{\theta}} = -\Gamma Y^{\mathrm{T}} s, \tag{14}$$

where, Γ is a positive definite matrix of adaptation gains.

With the complete control and adaptation laws now fully defined, it is necessary to formally prove that this closed-loop system is stable and that the tracking objective is indeed achieved.

3) Stability analysis: The stability of the closed-loop system is demonstrated using Lyapunov's direct method. Consider the Lyapunov candidate function, Eq. (15):

$$V = \frac{1}{2}s^{\mathrm{T}}Hs + \frac{1}{2}\tilde{\theta}^{\mathrm{T}}\Gamma^{-1}\tilde{\theta}, \tag{15}$$

where, $\tilde{\theta} = \theta - \hat{\theta}$ is the parameter estimation error. The function V is positive definite as it is a quadratic form of the tracking and estimation errors. The time derivative of V, after substituting the error dynamics and using the skew-symmetric property of the matrix $(\dot{H} - 2C)$, simplifies as Eq. (16):

$$\dot{V} = s^{\mathrm{T}}(F_t - Y\theta) + \tilde{\theta}^{\mathrm{T}}\Gamma^{-1}\dot{\tilde{\theta}}. \tag{16}$$

Substituting the control law in Eq. (13) and adaptation law in Eq. (14), and noting that $\dot{\tilde{\theta}} = -\dot{\hat{\theta}}$, yields the final expression for the derivative as Eq. (17):

$$\dot{V} = -s^{\mathrm{T}}Ks. \tag{17}$$

Since K is positive-definite, $\dot{V} \leq 0$, proving that V is nonincreasing and that s and $\tilde{\theta}$ are bounded. By Barbalat's Lemma, it can be further shown that $s \to 0$ as $t \to \infty$. As established in the error dynamics formulation, this ensures the asymptotic convergence of the pose tracking error, thus rigorously satisfying the overall control objective.

D. Control Architecture Overview

The control architecture, illustrated in Fig. 2, operates through two nested loops. First, an outer loop determines the pose error e between the desired trajectory and the robot's actual pose. This error is processed by the Reference Velocity according to Eq. (9) to produce a reference velocity designed to guide the robot. Next, an inner loop computes the filtered error signal, as defined by Eq. (8), by comparing the robot's actual body velocity to this reference velocity. The filtered error signal is central to the adaptive mechanism, driving both the Adaptation Law to update parameter estimates $\hat{\theta}$ and the control law to calculate the final control force F_t. This closed-loop structure allows the controller to drive the tracking error to zero while simultaneously learning and compensating for the unknown dynamic forces in real-time.

III. Experimental Setup

This section details the experimental framework used to validate the performance and robustness of the MRAC. The hardware platform and state estimation methods are designed to facilitate rigorous testing under significant variations in the robot's payload distribution.

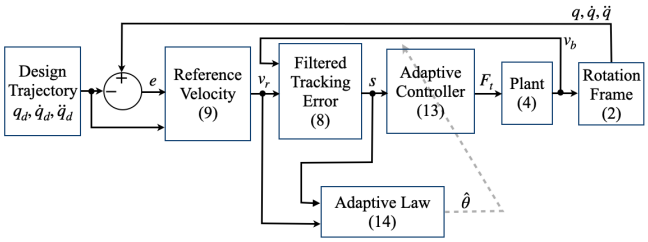

Fig. 2. Block diagram of the proposed MRAC framework.

A. Hardware Platform

The experiments are conducted on a Rosmaster X3, the FMWMR, as depicted in Fig. 3.

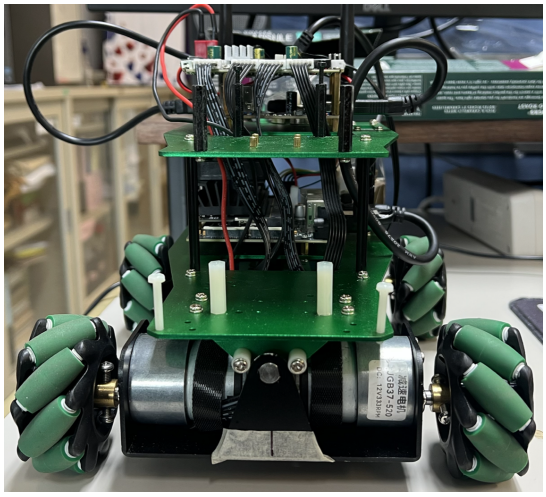

Fig. 3. The Rosmaster X3 experimental platform.

The platform's key physical parameters are summarized in Table I.

TABLE I. Rosmaster X3 Robot Parameters

Parameter	Symbol	Value
Robot Mass (Base)	m_r	2.0 [kg]
Added Mass	m_b	0.4 [kg]
Half Wheelbase	l_x	0.085 [m]
Half Track Width	l_y	0.080 [m]
Mecanum Wheel Radius	R_w	0.0325 [m]
Motor Encoder Resolution	N_{enc}	1320 [PPR]

The robot is equipped with two primary sensor systems for state estimation: four high-resolution quadrature wheel encoders fitted to the DC motors and an onboard IMU.

These sensors provide the raw data necessary for estimating the robot's pose and velocity, a process which is detailed next.

B. State Estimation and Signal Processing

Accurate estimation of the robot's state, comprising its world-frame pose and body-frame velocities, is essential for feedback control. This state is derived by processing and fusing data from the onboard sensors.

Estimation of body-frame velocities is a two-part process. First, the linear components are derived from the motor encoders. The angular velocity of each individual wheel is calculated from its respective motor encoder as in Eq. (18):

$$\omega_i = \frac{2\pi \Delta N_i}{N_{enc} \Delta t}, \tag{18}$$

where, ΔN_i is the encoder pulse count over the sampling interval Δt. These individual wheel velocities are then assembled into a vector and transformed into the robot's body-frame linear velocity v_b using the inverse kinematic model from Eq. (3). To ensure accuracy and mitigate noise, the body-frame angular velocity, ω_b, is derived by numerically differentiating the filtered yaw measurement from the IMU.

Next, the world-frame pose is estimated. The robot's orientation, ϕ, is obtained directly from the IMU's yaw measurement, with an initial offset recorded to define the world frame. The world-frame position (x, y) is then calculated by rotating the body-frame velocity estimate, v_b, into the world frame and performing first-order Euler integration at each time step. This process constitutes the robot's odometry.

Finally, to mitigate measurement noise before use in the control loop, the estimated state derivatives are processed with a first-order digital low-pass filter [19]. The filter is implemented by Eq. (19):

$$y_k = y_{k-1} + \beta(x_k - y_{k-1}), \tag{19}$$

where, y_k is the filtered output at the current time step, x_k is the raw input signal and β is the filter coefficient and calculated from the filter time constant τ as Eq. (20):

$$\beta = \frac{\Delta t}{\tau + \Delta t}. \tag{20}$$

The selection of the time constant, τ, presents a critical trade-off between noise rejection and phase lag, which can impact control stability. Based on empirical tuning, a time constant of $\tau = 0.04$ [s] is selected for the relatively clean IMU signal to prioritize a fast response. A slightly larger constant of $\tau = 0.05$ [s] is applied to the noisier, encoder-derived linear velocities to provide more aggressive filtering.

With a reliable state estimate available for feedback, the controller requires a physically achievable reference trajectory to follow. The generation of these smooth trajectories is a critical prerequisite for high-performance control.

C. Generation of Smooth Reference Trajectories

The performance of the controller is fundamentally limited by the quality of the reference trajectory. It is commanded to follow. For a physical system such as an FMWMR, commanding instantaneous changes in velocity or acceleration is impossible and leads to detrimental effects such as wheel slip, actuator saturation, and mechanical vibrations. Therefore, ensuring that the trajectory is sufficiently smooth is not an optimization, but a critical requirement to achieve high-fidelity motion.

1) Reference trajectory design: High-fidelity motion control requires reference trajectories that are physically realizable. Trajectories with discontinuous acceleration degrade tracking performance by inducing wheel slip and actuator saturation. Therefore, all reference signals are generated using quintic polynomial splines to ensure smoothness.

Unlike simpler cubic splines, quintic splines provide C^2-continuity, which guarantees a continuous acceleration profile. This is achieved by defining six boundary conditions: initial and final position (p_i, p_f), velocity (v_i, v_f), and acceleration (a_i, a_f). By setting the initial and final accelerations to zero, the jerk is eliminated, ensuring smooth rest-to-rest maneuvers as illustrated in Fig. 4. Here, Fig. 4(a) details the smooth S-curve position profile, Fig. 4(b) the corresponding bell-shaped velocity profile, and Fig. 4(c) the C^2-continuous acceleration. The acceleration begins and ends at zero, a condition which eliminates jerk and ensures the smoothness of the entire trajectory.

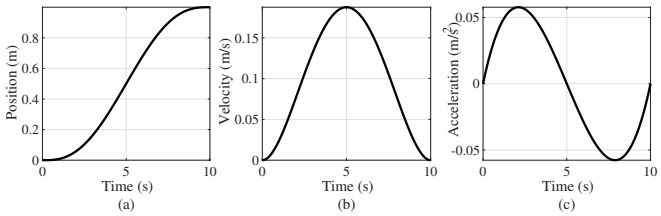

Fig. 4. Kinematic profiles of a 1 [m] maneuver using a quintic spline. (a) The smooth S-curve position profile, (b) The corresponding bell-shaped velocity profile, (c) The C^2-continuous acceleration.

The position profile $p(t)$ for any single degree of freedom is defined by the quintic polynomial as Eq. (21):

$$p(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3 + c_4 t^4 + c_5 t^5, \tag{21}$$

where, the six coefficients $c = [c_0, \ldots, c_5]^{\mathrm{T}}$ are uniquely determined by Eq. (22):

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & T & T^2 & T^3 & T^4 & T^5 \\ 0 & 1 & 2T & 3T^2 & 4T^3 & 5T^4 \\ 0 & 0 & 2 & 6T & 12T^2 & 20T^3 \end{bmatrix}^{-1} \begin{bmatrix} p_i \\ v_i \\ a_i \\ p_f \\ v_f \\ a_f \end{bmatrix}, \tag{22}$$

where, T is total trajectory time.

Solving this system yields the unique set of coefficients required to generate the smooth trajectory for each experimental path. This method is used to generate separate, smooth trajectories for each experimental path. While this method produces ideal reference signals, deploying the theoretical controller on the physical hardware requires additional practical considerations to compensate for real-world non-idealities.

D. Practical Implementation and Non-Ideality Compensation

To ensure robust performance on the physical hardware, the theoretical MRAC framework is augmented with three enhancements to compensate for non-ideal dynamics: friction, command saturation, and aggressive initial adaptation.

1) Feed-forward compensation: Nonlinear friction, particularly stiction, introduces a deadzone that degrades low-velocity tracking performance. To counteract this, a continuous, model-based feed-forward compensator is implemented using the hyperbolic tangent function to avoid control chattering as Eq. (23):

$$u_{ff} = \begin{bmatrix} V_{dz,x} \tanh\left(\frac{v_{rx}}{\epsilon_v}\right) \\ V_{dz,y} \tanh\left(\frac{v_{ry}}{\epsilon_v}\right) \\ V_{dz,\omega} \tanh\left(\frac{\omega_{r\phi}}{\epsilon_\omega}\right) \end{bmatrix}, \tag{23}$$

In this formulation, the critical parameter V_{dz} represents the breakaway voltage, which corresponds to the minimum voltage required to overcome stiction. The terms $V_{dz,x}$, $V_{dz,y}$, and $V_{dz,\omega}$ are the specific breakaway voltages for the longitudinal, lateral, and rotational axes. The parameters ϵ_v and ϵ_ω are smoothing factors that dictate the sharpness of the compensation signal's transition. All parameters are empirically tuned to improve low-velocity response and minimize steady-state error without introducing oscillations.

In addition to static friction, another practical limitation is the physical saturation of motor commands, which can degrade tracking performance, particularly during aggressive maneuvers.

2) Goal-proximity-based rate limiting: Fixed acceleration limits impose an inherent trade-off between a robot's transit speed and its final positioning accuracy. High acceleration values reduce travel time but increase the risk of overshooting the target, while low values ensure precision at the cost of efficiency. To resolve this, a state-dependent rate-limiting strategy is implemented, which dynamically modulates the acceleration limit, a_{lim}, based on the robot's proximity to its target d_{goal}. The modulation is performed in a two-step process. First, a unitless scaling factor, k, is calculated based on the robot's position within a pre-defined deceleration zone, as in Eq. (24):

$$k = \text{clip}\left(\frac{d_{goal} - d_{min}}{d_{max} - d_{min}}, 0, 1\right), \tag{24}$$

where, d_{max} and d_{min} define the outer and inner boundaries of the deceleration zone. As the robot travels through this zone, the value of k transitions linearly from 1 to 0, with the clip function, $\text{clip}(X, lower, upper)$ clips the values in array X to the range $[lower, upper]$ by replacing values less than lower with lower and values greater than upper with upper, that ensures it remains bounded.

This scaling factor is then used to linearly interpolate the final acceleration limit a_{lim} between two prescribed bounds, as in Eq. (25):

$$a_{lim} = a_{min} + k \cdot (a_{base} - a_{min}), \tag{25}$$

where, a_{base} represents the high baseline acceleration for rapid transit (applied when $k \approx 1$), while a_{min} is the conservative minimum acceleration for precise final movements (applied when $k \approx 0$). This method ensures a smooth transition from aggressive to conservative motion, optimizing both speed and accuracy.

Beyond compensating for these physical non-linearities, the controller's transient behavior during startup must also be managed to ensure stability.

3) Adaptation law for soft start: A significant initial tracking error can induce large, abrupt updates to the adaptive parameters $\hat{\theta}$, potentially causing control signal saturation or transient instability. To mitigate this, a soft-start mechanism is implemented by scheduling the adaptation gain matrix Γ. The gain is linearly ramped from an initial value of zero to its nominal steady-state value according to the following rule Eq. (26):

$$\Gamma(t) = \Gamma_0 \cdot \min\left(1.0, \frac{t}{T_{ramp}}\right), \tag{26}$$

where, $\Gamma(t)$ is the time-varying adaptation gain matrix, Γ_0 is its nominal value and T_{ramp} is the total duration of the ramp-up period.

The ramp duration T_{ramp} is determined empirically, selected as the minimum time required to suppress undesirable transients in the initial control signal and ensure a smooth initialization of the adaptation process.

For reproducibility, the empirically determined values for these implementation parameters are provided in Table II.

TABLE II. EMPIRICALLY TUNED IMPLEMENTATION PARAMETERS

Category	Parameter	Value
Friction Compensation		
	Deadzone Voltage $V_{dz,x}, V_{dz,y}, V_{dz,\omega}$	0.1 V
	Smoothing Factor $\epsilon_v, \epsilon_\omega$	0.1
Command Rate Limiting		
	Base Acceleration a_{base}	2.0 m/s^2
	Minimum Acceleration a_{min}	0.5 m/s^2
	Max Deceleration Distance d_{max}	0.4 m
	Min Deceleration Distance d_{min}	0.1 m
Adaptation Soft Start		
	Adaptation Ramp Time T_{ramp}	0.1 s

IV. EXPERIMENTAL PROCEDURE AND EVALUATION

A. Experimental Scenarios

To rigorously evaluate the controller's performance and robustness, experiments are conducted on a custom-built testbed, as shown in Fig. 5. The experimental design systematically varies three key factors: path type, surface condition, and payload distribution. These factors and their corresponding conditions are summarized in Table III, and the path waypoints are detailed in Table IV.

To ensure statistical reliability, each of the twelve resulting experimental scenarios is repeated three times for both the proposed MRAC and a baseline PID controller. The performance metrics presented in the results are the arithmetic mean of these three trials.

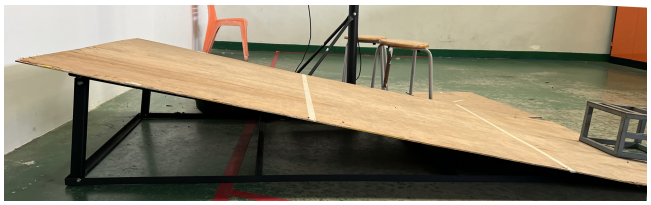

Fig. 5. The experimental testbed (2.4 m × 1.2 m), shown in its inclined configuration.

B. Center of Gravity Manipulation

To isolate the dynamic effects of an off-center payload, two CoG conditions are established without altering the robot's total mass.

- Centered CoG (Nominal): The stock robot's CoG is not at its geometric center. A baseline condition is established by adding a balancing mass, with its position empirically adjusted to ensure equal weight distribution across all four wheels.
- Off-center CoG: For this condition, the same balancing mass is relocated to a fixed position to induce a significant, known CoG shift, as specified in Table III.

This methodology ensures that any observed performance differences between the two payload conditions are attributable solely to the change in dynamic coupling caused by the CoG shift, not by a change in total mass.

TABLE III. EXPERIMENTAL FACTORS AND CONDITIONS

Factor	Conditions
Path Type	Straight, Diagonal, Square
Surface Condition	1. Flat Surface 2. Inclined Surface ($\pi/18$ rad)
Payload Distribution	1. Centered CoG (Nominal) 2. Off-center CoG $(x_g, y_g) = (-4.5, -26.7)$ mm

TABLE IV. REFERENCE TRAJECTORY DEFINITIONS

Path Name	Description	Waypoints (x, y) [m]
Straight	1 m translation along y-axis	(0,0) → (0,1)
Diagonal	1 m translation in x and y	(0,0) → (1,1)
Square	4 m perimeter path with sharp corners	(0,0) → (0,1) → (1,1) → (1,0) → (0,0)

C. Benchmark Controller for Comparison

For performance comparison, a fixed-gain PID controller is implemented as a benchmark. The controller's gains are carefully tuned on the physical hardware. Initial estimates are obtained using the Ziegler-Nichols method and subsequently optimized using MATLAB's PID Tuner with experimental step response data. The final, optimized gains for the translational and rotational axes are presented in Table V.

TABLE V. OPTIMIZED PID CONTROLLER GAINS

Control Axis	k_p	k_i	k_d
Translational (x, y)	2.80	0.06	0.15
Rotational (ϕ)	2.20	0.04	0.12

D. Controller Gain Tuning Methodology

The performance of MRAC is determined by three gain matrices of the error convergence matrix Λ, the feedback matrix K and the adaptation matrix Γ. A systematic two-phase empirical tuning procedure is conducted on the physical hardware to achieve a balance between responsiveness and stability. The final gains are listed in Table VI.

1) Phase 1: Tuning baseline feedback gains Λ and K: First, a stable, non-adaptive baseline controller is established by setting the adaptation gain $\Gamma = 0$ and initializing the parameter vector $\hat{\theta}$ with nominal estimates. The gains in $\Lambda = \text{diag}(\lambda_p, \lambda_p, \lambda_\phi)$ are tuned to set the desired error convergence rate without causing oscillation. Subsequently, the feedback gains $K = \text{diag}(k_p, k_p, k_\phi))$ are tuned to ensure a critically damped response without amplifying sensor noise or causing jerky commands.

2) Phase 2: Tuning adaptation gains Γ: Second, the adaptation gains in the diagonal matrix $\Gamma = \text{diag}(\gamma_1, \ldots, \gamma_7)$ are tuned to set the parameter learning rates. The selection of these gains presents a critical trade-off: Low values result in slow adaptation, while high values can cause noisy parameter estimates and instability.

The gains are tuned incrementally, starting with small values. They are increased to the highest level that provided rapid error convergence without introducing oscillations into the control signal. This process is performed sequentially using maneuvers designed to excite specific dynamics, for example, tuning mass/inertia gains during linear accelerations and CoG/slope gains during tests with an off-center payload on the inclined surface.

TABLE VI. MRAC GAIN PARAMETERS

Gain Parameter	Value
Error Convergence Gains (Λ)	
Translational Gain λ_p	2.5
Rotational Gain λ_ϕ	2.5
Control Gain (K)	
Translational Gain k_p	0.5
Rotational Gain k_ϕ	0.5
Adaptation Gains (Γ)	
γ_1 for m	0.0400
γ_2 for I_z	0.0180
γ_3 for mx_g	0.0120
γ_4 for my_g	0.0010
γ_5 for $m \sin\alpha$	0.0050
γ_6 for $mx_g \sin\alpha$	0.0006
γ_7 for $my_g \sin\alpha$	0.0006

E. Performance Metrics

To provide a quantitative assessment of controller performance, two standard metrics are chosen. One is the RMSE to evaluate tracking accuracy and the other is the ITAE to evaluate

settling performance. The specific equations are defined as Eq. (27), Eq. (28) and Eq. (29):

$$\text{RMSE}_p = \sqrt{\frac{1}{N}\sum_{i=1}^{N} e_p(t_i)^2}, \tag{27}$$

$$\text{RMSE}_h = \sqrt{\frac{1}{N}\sum_{i=1}^{N} e_h(t_i)^2} \tag{28}$$

and

$$\text{ITAE}_p = \int_0^T t \cdot |e_p(t)|\, dt, \tag{29}$$

where, $e_p(t) = p(t) - p_d(t)$ is the error between actual position $p(t)$, desired position $p_d(t)$ and $e_h(t) = \phi(t) - \phi_d(t)$ and N is total number of data points.

V. Results and Discussion

A. Experimental Results

This section presents the experimental results that compare the proposed MRAC with the baseline PID controller in all 12 test scenarios. To highlight the controllers' distinct behaviors, a detailed graphical analysis is presented for two representative scenarios: the nominal case (straight-line motion, flat surface, centered payload) and the most challenging case (square-path motion, inclined surface, off-center payload). A comprehensive summary of the performance metrics for all experiments is provided at the end of this section.

1) Performance in nominal conditions: The experimental results for a straight line trajectory 1 [m] under nominal conditions are shown in Fig. 6. Performance comparison between MRAC and PID controllers is shown in Fig. 6(a) to Fig. 6(d), where the ideal trajectory is represented by a solid black line, the PID response by a blue dashed line, and the MRAC response by a red dotted line. Fig. 6(a) shows the robot's lateral position, where both responses remain close to the ideal zero position. Fig. 6(b) shows the primary translation, which follows the smooth S-shaped reference trajectory. The robot's orientation is shown in Fig. 6(c), where the desired zero-radian heading is successfully maintained. A top-down view of the path is provided in Fig. 6(d). Fig. 6(e) shows the RMSE of position over time. The final RMSE for the MRAC is approximately 1.5 [cm], a 25% reduction compared to the PID's 2.0 [cm] error. Finally, Fig. 6(f) confirms that the heading error for both controllers remains minimal and stable throughout the maneuver.

2) Performance under challenging conditions: The experimental results for a square path trajectory under challenging conditions (an inclined surface with an off-center CoG) are presented in Fig. 7. The robot's position along the x- and y-axes is shown in Fig. 7(a) and Fig. 7(b), where the MRAC's red dotted line more closely follows the sharp transitions of the reference compared to the PID's response. The robot's orientation is shown in Fig. 7(c). A top-down view of the path is provided in Fig. 7(d), which illustrates that the PID controller exhibits significant corner overshoot, while the MRAC maintains sharp, accurate tracking. The RMSE of position in Fig. 7(e) highlights the performance difference; the PID's error accumulates, reaching a final RMSE of approximately 2.5 [cm], whereas the MRAC maintains a stable and bounded error of just 1.0 [cm]. Finally, Fig. 7(f) confirms that the MRAC's heading error profile is visibly smoother than the PID's.

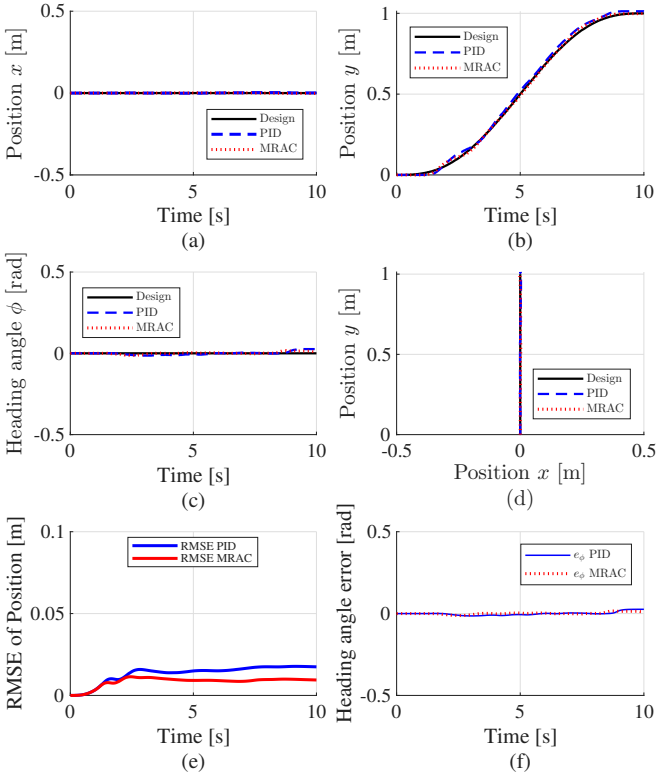

Fig. 6. Straight path tracking results for the nominal case comparing PID and MRAC controllers: (a) Position x over time. (b) Position y over time. (c) Heading angle ϕ over time. (d) Robot's trajectory in the $x-y$ plane. (e) RMSE of position over time. (f) Heading angle error over time.

3) Quantitative performance summary: The complete quantitative results from all twelve experimental scenarios are compiled in Table VII. For each path type, surface condition, and CoG configuration, the table lists the position RMSE, heading RMSE, and position ITAE for both the PID and MRAC controllers. The best-performing result in each case is highlighted in bold.

B. Discussions

The experimental results presented in Section V-A consistently demonstrate the superior performance and robustness of the MRAC framework compared to a well-tuned PID controller. The quantitative data in Table VII reveals that, on average, the MRAC reduced the position RMSE by 52.7% and the position ITAE by 61.5%. This substantial improvement can be directly attributed to the controller's adaptive nature.

The key strength of the MRAC lies in its ability to estimate and compensate for dynamic uncertainties in real time, a capability that fixed-gain controllers lack. This is particularly evident in the most challenging scenarios. For instance, in the

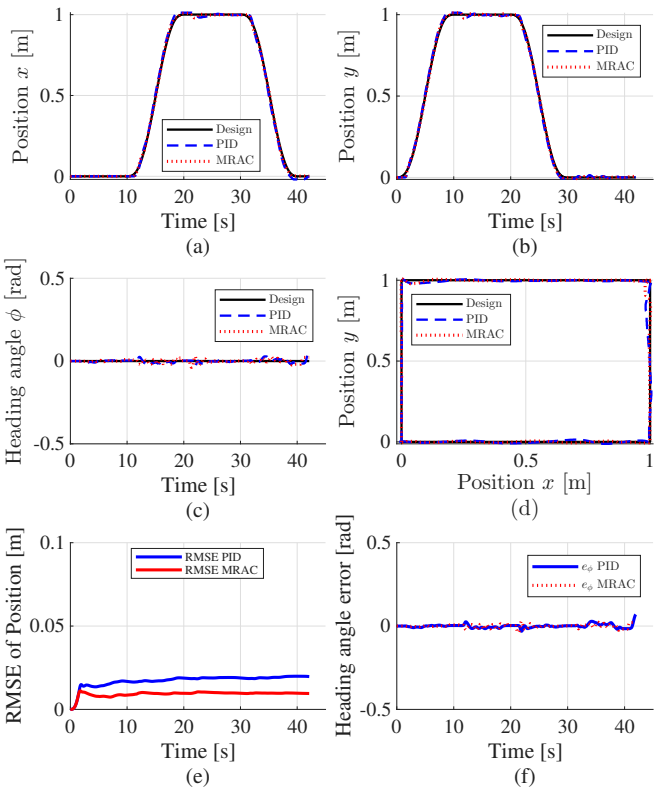

Fig. 7. Square path tracking results under challenging conditions comparing PID and MRAC controllers: (a) Position x over time. (b) Position y over time. (c) Heading angle ϕ over time. (d) Robot's trajectory in the $x-y$ plane. (e) RMSE of position over time. (f) Heading angle error over time.

square path test with an off-center CoG on an inclined surface, the PID controller's performance degraded significantly, as shown by the large corner overshoots in Fig. 7(d). This occurs because fixed PID gains cannot counteract complex coupled disturbance torques that arise from the combination of CoG offset and gravitational forces. In contrast, the MRAC uses the regressor matrix Y and the adaptation law in Eq. (14) to continuously update its internal model, which effectively cancels these disturbances and maintains high-precision tracking.

Furthermore, the MRAC's robustness to internal dynamic coupling was highlighted during the diagonal path tests. The PID controller struggled to manage the simultaneous translational and rotational motion, resulting in a heading RMSE of 0.0947 [rad]. In contrast, the MRAC achieved a heading RMSE of just 0.0131 [rad], an 86% reduction. This result demonstrates the effectiveness of the adaptation mechanism in decoupling the system dynamics, which is a critical feature for holonomic platforms like the FMWMR.

The controller's results are highly repeatable, as confirmed by running each experiment three times. The statistical variability is very low, with the standard deviation of the error typically remaining below 5% of the mean value. A sensitivity analysis further confirmed the controller's robustness to parameter tuning. Moderate changes (approximately $\pm 20\%$) to the controller gains resulted in graceful performance degradation rather than instability, proving that its effectiveness is an inherent feature of the adaptive design, not the result of fragile,

TABLE VII. COMPARATIVE PERFORMANCE METRICS FOR ALL PATH TYPES

Path	Condition Surface/CoG	Controller	RMSE_p [m]	RMSE_h [rad]	ITAE_p [m s]
Straight	Flat/Center	PID	0.0170	0.0189	23.9470
		MRAC	**0.0094**	**0.0105**	**9.4366**
	Flat/Offset	PID	0.0191	0.0299	21.0450
		MRAC	**0.0100**	**0.0259**	**9.7486**
	Slope/Center	PID	0.0164	0.0369	20.0144
		MRAC	**0.0061**	**0.0367**	**5.8353**
	Slope/Offset	PID	0.0163	0.0346	20.4149
		MRAC	**0.0091**	**0.0305**	**11.9156**
Diagonal	Flat/Center	PID	0.0236	0.0947	37.6089
		MRAC	**0.0106**	**0.0131**	**12.0401**
	Flat/Offset	PID	0.0232	0.0573	37.4503
		MRAC	**0.0111**	**0.0477**	**13.2413**
	Slope/Center	PID	0.0234	0.0949	35.7405
		MRAC	**0.0075**	**0.0348**	**7.8987**
	Slope/Offset	PID	0.0222	0.0863	37.7239
		MRAC	**0.0117**	**0.0719**	**13.4543**
Square	Flat/Center	PID	0.0191	0.0387	396.1914
		MRAC	**0.0092**	**0.0165**	**158.7575**
	Flat/Offset	PID	0.0192	0.0391	406.1247
		MRAC	**0.0085**	**0.0168**	**148.7291**
	Slope/Center	PID	0.0191	0.0300	396.3069
		MRAC	**0.0093**	**0.0268**	**187.6281**
	Slope/Offset	PID	0.0197	0.0182	417.5931
		MRAC	**0.0095**	**0.0150**	**167.6022**

hyperspecific tuning.

To address the practical trade-offs between the two controllers, a comparison of their computational requirements is provided in Table VIII. The analysis highlights that the PID controller's logic is based on simple algebraic operations, while the MRAC is inherently more complex, requiring the assembly of a regressor matrix and several matrix-vector multiplications in each cycle. We measured the average execution time for a single control loop on the Rosmaster X3's onboard processor. The results show that the PID controller took approximately 0.08 [ms] per cycle, while the more demanding MRAC took 0.27 [ms]. Although this makes the MRAC about 3.4 times more computationally intensive, its execution time is still extremely low, allowing for a theoretical operating frequency of over 3,700 Hz. This rate is far higher than the mechanical limits of the robot, confirming that the proposed MRAC is a highly efficient and practical solution for real-time control.

In summary, the experimental validation confirms that the proposed MRAC is not only stable in theory but also highly effective in practice. By explicitly addressing the sim-to-real gap with practical compensations and leveraging a sound adaptive control law, the controller provides a robust solution for high-precision motion in unpredictable environments.

VI. CONCLUSION

An adaptive controller for an FMWMR with unknown CoG offset and unknown slope inclination has been implemented and experimentally validated on a Rosmaster X3 platform. A linearly parameterized dynamic model that captures CoG-induced coupling and gravity on an incline is employed, and

TABLE VIII. COMPUTATIONAL COMPARISON OF PID AND MRAC

Aspect	PID Controller	MRAC
Core Calculation	- Error calculation - Summation (Integral) - Differentiation	- Regressor matrix assembly - Matrix-vector multiplication
Parameter Updates	Fixed offline-tuned gains	Online adaptation law via gradient descent
Memory Usage	Low (stores error states)	Moderate (stores error states and parameter vector $\hat{\theta}$)
Avg. Execution Time*	0.08 [ms]	0.27 [ms]

*Average per-cycle execution time measured on the Rosmaster X3.

stability is ensured through a Lyapunov-based adaptation law. To bridge the gap between theory and experiment, deployment-oriented measures that address motor deadzones, command rate limits, and adaptation transients are incorporated. Across 12 scenarios, consistent improvements are demonstrated over a carefully tuned PID benchmark. On average, the position RMSE is reduced by 52.7% and the position ITAE is reduced by 61.5%. Heading accuracy is kept bounded and precise even under the most challenging combinations of slope and CoG shift. These results indicate that the controller achieves high-precision tracking under coupled uncertainties in real time, demonstrating its suitability for operation in unstructured environments.

Future work will focus on improving the robot's long-term autonomy and robustness by addressing three key areas. The current reliance on internal sensors, while effective for short maneuvers, is susceptible to cumulative drift. To overcome this, our primary goal is to develop a robust global localization system by integrating a 2D LiDAR sensor and implementing a Simultaneous Localization and Mapping (SLAM) algorithm, with an Extended Kalman Filter (EKF) for sensor fusion. Building on this improved state estimation foundation, we will then rigorously test and adapt the controller for more complex and unstructured terrains, including surfaces with varying friction and irregularities, to potentially extend the adaptive model to account for wheel slip. Ultimately, the enhanced controller and localization system will be integrated as the low-level execution layer within a complete navigation stack. By coupling our work with a high-level motion planner capable of real-time environmental perception, we aim to transition from trajectory tracking to full autonomy, enabling the robot to independently generate and modify paths to safely navigate around obstacles in real-world applications.

ACKNOWLEDGMENT

This work is partially supported by JSPS KAKENHI Grant Number JP21K03930 and (C) 23K03898.

REFERENCES

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

[2] G. Klančar, A. Zdešar, S. Blažič, and I. Škrjanc, *Wheeled Mobile Robotics: From Fundamentals Towards Autonomous Systems*. Oxford, U.K.: Butterworth-Heinemann, 2017.

[3] T.-N. Manh, M.-P. Xuan, P.-N. Doan, and T.-P. Quoc, "Tracking control for mobile robots with uncertain parameters based on model reference adaptive control," in *Proc. Int. Conf. Control, Autom. Inf. Sci. (ICCAIS)*, Danang, Vietnam, Nov. 2013, pp. 18–23.

[4] Z. Hendzel and Ł. Rykała, "Modelling of dynamics of a wheeled mobile robot with Mecanum wheels with the use of Lagrange equations of the second kind," *Int. J. Appl. Mech. Eng.*, vol. 22, no. 1, pp. 81–99, Mar. 2017.

[5] I. Zeidis and K. Zimmermann, "Dynamics of a four-wheeled mobile robot with Mecanum wheels," *Z. Angew. Math. Mech.*, vol. 99, no. 12, Art. no. e201900173, Dec. 2019.

[6] S. O. Onyango, Y. Hamam, K. Djouani, and G. Qi, "Dynamic control of powered wheelchair with slip on an incline," in *Proc. 2nd Int. Conf. Adapt. Sci. Technol. (ICAST)*, Accra, Ghana, Jan. 2009, pp. 278–283.

[7] N. H. Thai, T. Ly, and L. Dzung, "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.

[8] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1991.

[9] X. Lu, X. Zhang, G. Zhang, and S. Jia, "Design of adaptive sliding mode controller for four-Mecanum wheel mobile robot," in *Proc. 37th Chinese Control Conf. (CCC)*, Wuhan, China, Jul. 2018, pp. 3983–3987.

[10] V. Alakshendra and S. Chiddarwar, "Adaptive robust control of Mecanum-wheeled mobile robot with uncertainties," *Nonlinear Dyn.*, vol. 87, no. 4, pp. 2147–2169, Feb. 2017.

[11] X. Yue, J. Chen, Y. Li, *et al.*, "Path tracking control of skid-steered mobile robot on the slope based on fuzzy system and model predictive control," *Int. J. Control Autom. Syst.*, vol. 20, no. 4, pp. 1365–1376, Apr. 2022.

[12] H. B. Santos *et al.*, "Model predictive torque control for velocity tracking of a four-wheeled climbing robot," *Sensors*, vol. 20, no. 24, Art. no. 7059, Dec. 2020.

[13] C. Chaichumporn *et al.*, "The dynamical modeling of four mecanum wheel mobile robot on typical unstructured terrain," in *Proc. 21st Int. Conf. Electr. Eng./Electron., Comput., Telecommun. Inf. Technol. (ECTI-CON)*, Khon Kaen, Thailand, May 2024, pp. 1–5.

[14] K. Ogata, *Modern Control Engineering*, 5th ed. Upper Saddle River, NJ, USA: Prentice Hall, 2010.

[15] Y. Luo *et al.*, "Trajectory tracking control of an orchard robot based on improved integral sliding mode algorithm," *Agriculture*, vol. 15, no. 17, p. 1881, 2025.

[16] A. D. Abdullsahiba and H. M. Alwan, "Experimental investigation of applying multi-stage (PID-MRAC) controller for trajectory tracking of four mecanum wheeled mobile robot," *Eng. Technol. J.*, vol. 43, no. 08, pp. 641–658, 2025.

[17] S. Ibrayev and B. Omarov, "Designing an autonomous mobile robot featuring self-localization through 3D LiDAR technology," *Eng. Technol. Appl. Sci. Res.*, vol. 15, no. 5, pp. 28224–28231, Oct. 2025.

[18] M. Burkacki, I. Łysy, S. Suchoń, M. Chrzan, and R. Kowolik, "Systematic review of Mecanum and Omni wheel technologies for motor impairments," *Appl. Sci.*, vol. 15, no. 9, Art. no. 4773, Sep. 2025.

[19] G. F. Franklin, J. D. Powell, and M. Workman, *Digital Control of Dynamic Systems*, 3rd ed. Menlo Park, CA, USA: Addison-Wesley, 1998.

[20] L. Biagiotti and C. Melchiorri, *Trajectory Planning for Automatic Machines and Robots*. Berlin, Heidelberg: Springer, 2008.

[21] W. A. Farag and M. M. Mahmoud, "Trajectory optimization for autonomous highway driving using quintic splines," *World Electr. Veh. J.*, vol. 16, no. 8, p. 434, 2025.