Experimental Trajectory Tracking Control of Mecanum-Wheeled UGV Using Linear Angle-to-Gain (LA-G), Modified LA-G, and Linear Quadratic Regulator (LQR)

Izzeldeen Al-Manasra[1,2], Tamir Shaqarin[3]

1. Faculty of Engineering Technology, Al Hussein Technical University, 11831 Amman, Jordan.
2. Mechanical Engineering Department, The University of Jordan, 11942 Amman, Jordan.
3. Mechanical Engineering Department, Tafila Technical University, 66110 Tafila, Jordan.

Abstract— **Mecanum-wheeled unmanned ground vehicles (MWUGVs) offer holonomic motion with three degrees of freedom in the planar workspace, making them suitable for confined industrial environments. However, wheel slip significantly degrades localization accuracy and trajectory-tracking performance. This paper presents an experimental study of three control strategies for MWUGV trajectory tracking: the original Linear Angle-to-Gain (LA-G) controller, a modified LA-G method proposed in this work, and a Linear-Quadratic Regulator (LQR). To mitigate slip-induced localization errors, an optical mouse sensor is integrated as an independent positioning measurement. All controllers are implemented and validated on a four-wheel mecanum robotic platform. Experimental results demonstrate that the LQR achieves the highest tracking accuracy with a total two-dimensional (2D) root mean square error (RMSE) of $1.202mm$, compared to $2.322mm$ for the modified LA-G and $2.749mm$ for the original LA-G controller. The results highlight that while LQR provides superior precision, the modified LA-G offers a simplified yet effective alternative for systems with unmodeled dynamics.**

Keywords—MWUGV, Mecanum wheels, LQR, LA-G control, Optical mouse sensor, Trajectory tracking.

I. Introduction

The Mecanum wheel shown in Fig. 1 is an omnidirectional wheel design that enables a vehicle to move in any direction, sideways, diagonally, or rotationally, without traditional steering mechanisms [1]. It was invented in 1973 by Bengt Ilon. It consists of a central hub with several rollers placed around its circumference, usually oriented at a 45° angle to the wheel's rotation axis [2]. By varying the speed and direction of each wheel, the resulting force vectors allow for three degrees of freedom (3-DOF) in a 2D plane [3]. This extreme maneuverability has led to the widespread adoption of Mecanum-Wheeled Unmanned Ground Vehicles (MWUGVs) in tight industrial spaces, serving as waiter robots in restaurants [4], forklifts in factories, and automated distributors in warehouses [5].

Despite these advantages, achieving precise path-tracking remains challenging due to global localization and position tracking issues. An inherent limitation of the Mecanum wheel mechanism is the slipping or skidding phenomenon, which occurs when the coefficient of friction between the wheel and the ground is either very high or very low, such as on asphalt or oily surfaces. This is caused by the reduction of the effective driving force due to the orientation of the wheel's rollers [6]. Furthermore, Omni wheels generally suffer from vibration issues stemming from the discontinuous contact between the rollers and the ground. Some researchers have attempted to mitigate these shortcomings by increasing the number of rollers or developing complex suspension systems to force the wheels to maintain constant contact with the ground [7, 8]. Neglecting these uncertainties in the control algorithm limits the ability of the UGV to reach desired speeds and setpoints; consequently, a comprehensive analysis of both vehicle kinematics and dynamics while accounting for slippage is required to ensure tracking performance. [9, 10].

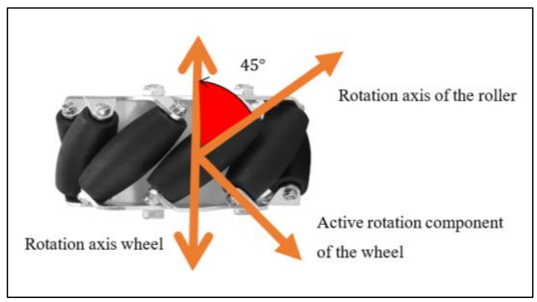

Fig. 1. Mecanum wheel.

In the literature, researchers have developed various techniques for controlling and positioning Mecanum-wheeled robots to overcome these issues. Some have utilized onboard vision cameras for visual odometry to count position movement regardless of wheel rotations [11], while others have implemented cameras to act as feedback sensors [12, 13]. Other approaches include the use of several ultrasonic distance sensors placed around the robot frame to measure the distance to walls and obstacles [14]. Control methodologies have also varied widely, ranging from conventional Proportional-Integral-Derivative (PID) controllers [15] and fuzzy logic controllers [16] to more complex model-based approaches like adaptive control [17] and the Linear-Quadratic Regulator (LQR) [11]. While model-free methods like fuzzy logic do not require explicit system modeling, they often rely heavily on the experience of the researcher to select optimum parameters, whereas model-based control can achieve more accurate results if the dynamic model is well-developed.

To obtain the most accurate position of the robot, it is often more reliable to measure the movement of the UGV regardless of the rotations of the wheels. Optical flow sensors, such as those found in computer mice, provide an efficient solution for obtaining x and y coordinates directly, which utilizes an optical flow algorithm to estimate displacement[18]. Researchers have used two mice separated by a specific distance to geometrically derive the heading angle [19]. This study utilizes a single optical mouse for position and an Inertial Measurement Unit (IMU) for heading angle observation. This combination eliminates the need for the complex geometrical derivations associated with dual-mouse setups, and the optical mouse sensor exhibits superior dust resistance relative to legacy mechanical ball sensors.

This work proposes a continuous formulation of the LA-G control law using a sinusoidal approximation. The main goal of this research is to evaluate the trajectory-tracking performance of three control methods for MWUGVs: The Linear Angle-to-Gain (LA-G) method, a modified version of LA-G, and an LQR controller, and compare their reference tracking ability. By focusing on the absolute position of the UGV and neglecting the rotations of the wheels, the proposed system aims to enhance the efficiency of trajectory-tracking and heading angle control while limiting the impact of the slipping problem inherent to Mecanum wheels. We can summarize our main contributions as follows:

1. Introducing and implementing the modified Linear angle to gain method.
2. Experimental comparison between the original and modified linear angle to gain methods with LQR.

The remainder of this paper is organized as follows: Section II describes the experimental setup. Section III presents system modeling and problem formulation, encompassing both kinematic modeling and open-loop characterization and linearity analysis. Section IV details the control design for the three evaluated methodologies: LA-G, Modified LA-G, and LQR. Section V presents and discusses the experimental results, and Section VI concludes the paper with a summary of findings and directions for future work.

II. Experiment Setup

The design and construction of the custom MWUGV platform involved the integration of specific mechanical components, a multi-layered chassis, and a hybrid sensing setup. The platform utilizes four 100 mm diameter Mecanum wheels driven by four independent brushed DC planetary geared motors.

A universal holed steel plate serves as the primary structure, reinforced by two heavyweight steel bars at the front and back. This configuration, which can be seen in Fig. 2, was specifically designed to allow slight bending at the corners, effectively acting as a suspension system to ensure all four wheels maintain constant ground contact.

For positioning and navigation, the platform utilizes an independent sensing system centered on an optical mouse sensor and an MPU-6050 Inertial Measurement Unit (IMU). To ensure a constant focal length, which maintains a stable ratio between raw sensor data and physical distance, the optical sensor was housed in a custom 3D-printed case. This case is mounted via a four-spring suspension system that maintains continuous contact between the sensor and the ground surface, allowing it to adapt to surface irregularities without inducing excessive friction. The IMU is mounted vertically above the mouse sensor at the local origin. The control architecture is managed by an Arduino Mega microcontroller.

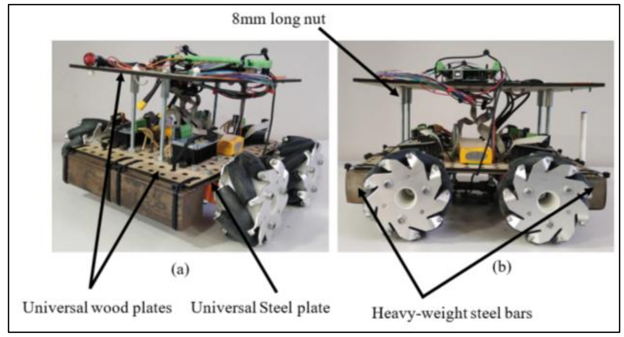

Fig. 2. Custom-build UGV. (a) Isometric view. (b) Side view.

III. System Modeling and Problem Formulation

A. Kinematic Modeling

The MWUGV utilizes four Mecanum wheels with rollers oriented at 45°. To achieve the correct holonomic motion in the desired directions, two specific combinations of Mecanum wheels are shown in Fig. 3. A, and 3. B are considered correct. These setups include two left and two right wheels in a specific configuration. Any other combination is considered incorrect, as it would require excessive power to move [20].

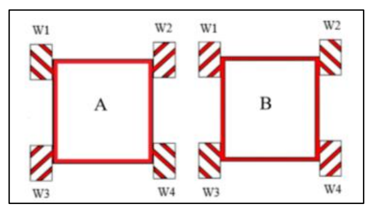

Fig. 3. Correct wheel arrangements of MWUGV.

For the Model A arrangement, the forward kinematics relate the angular velocities of the four wheels $(\omega_1, \omega_2, \omega_3, \omega_4)$ to the velocities in the local body frame (V_X, V_Y, ω_Z) as derived in [20]:

$$\begin{bmatrix} V_X \\ V_Y \\ \omega_Z \end{bmatrix} = \frac{R_W}{4} \begin{bmatrix} -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 \\ \frac{1}{(L+W)} & \frac{1}{-(L+W)} & \frac{1}{-(L+W)} & \frac{1}{(L+W)} \end{bmatrix} \begin{bmatrix} \omega_{1W} \\ \omega_{2W} \\ \omega_{3W} \\ \omega_{4W} \end{bmatrix} \quad (1)$$

In this formulation, R_W represents the wheel radius, while L and W are half the length and half the width of the robot, measured from the center to the wheel axes.

B. Open Loop Characterization and Linearity

A significant limitation of MWUGVs is the inherent wheel slippage, which introduces inaccuracies in encoder-based odometry. To address this, an independent optical mouse sensor is used. The testing setup involved performing an open-loop step response at different PWM values on the primary

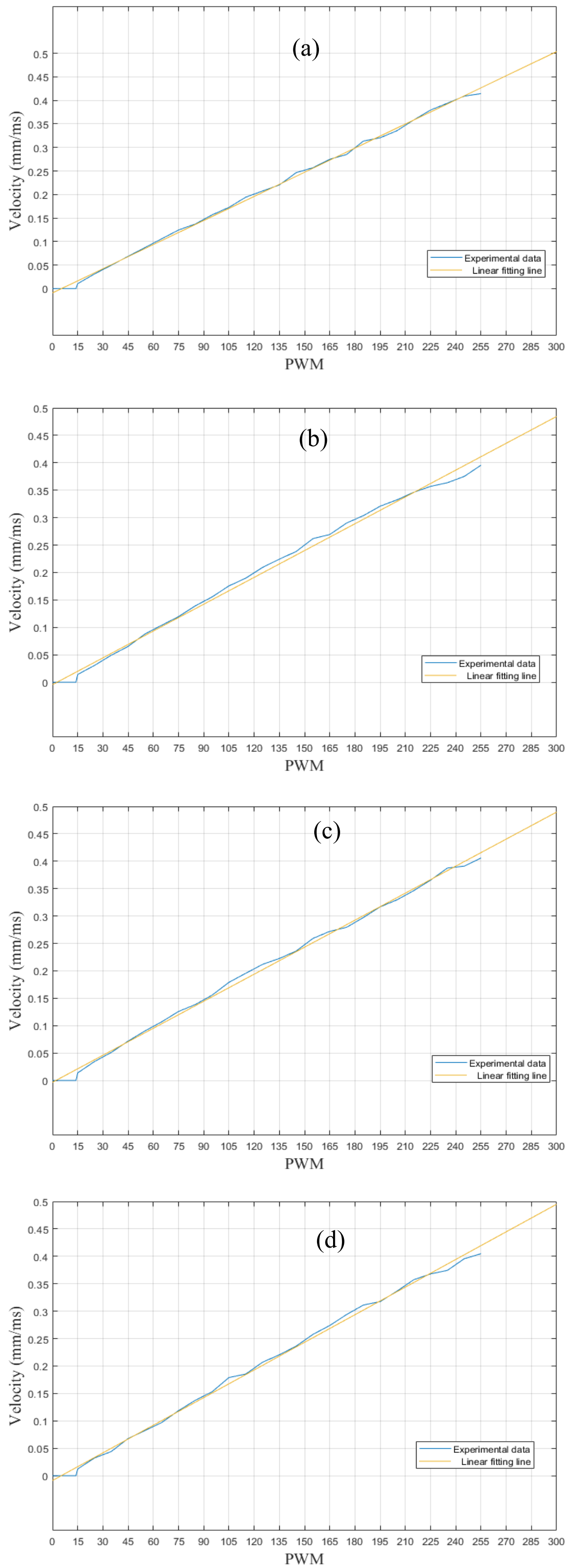

Fig. 4. Experimental open-loop step responses in terms of velocity vs. PWM values. (a) W1W4 backward. (b) W2W3 backward. (c) W1W4 forward. (d) W2W3 forward.

actuation, including moving the UGV using only two diagonal wheels to produce four main movements: W1W4 forward ($W1W4\_F$), W1W4 backward ($W1W4\_B$), W2W3 forward ($W2W3\_F$), and W2W3 backward ($W2W3\_B$). By mixing these motions, all maneuvering movements are produced.

Experimental trials with PWM values ranging from 1 to 255 revealed that the UGV started to move only at $PWM = 15$ for all four actuations. To determine the system's behavioral characteristics, each PWM value was plotted against its corresponding final 45° velocity. The resulting graphs shown in Fig. 4 demonstrate that the velocity increases proportionally with the input power, indicating that the system exhibits quasi-linear behavior within the tested operating range, which aligns with the monotonic weak non-linear dynamics found in [19]. Based on this observed correlation, the system is assumed to be linear for the purposes of the subsequent model augmentation and controller design.

Consequently, the kinematic model is augmented with the average input-output steady-state values derived from the experimental velocity-to-PWM relationship:

$$\begin{bmatrix} V_X \\ V_Y \\ \omega_Z \end{bmatrix} = \frac{\mathrm{R_W}}{4} \begin{bmatrix} -0.41415 & 0.41415 & -0.41415 & 0.41415 \\ 0.41415 & 0.41415 & 0.41415 & 0.41415 \\ \frac{0.1661}{(\mathrm{L+W})} & \frac{0.1661}{-(\mathrm{L+W})} & \frac{0.1661}{-(\mathrm{L+W})} & \frac{0.1661}{(\mathrm{L+W})} \end{bmatrix} \begin{bmatrix} PWM_{w1} \\ PWM_{w2} \\ PWM_{w3} \\ PWM_{w4} \end{bmatrix} \tag{2}$$

The control objective is to ensure the output state $d(t)$ tracks the reference trajectory $d_{ref}(t)$ with minimal error, effectively compensating for surface-induced slippage through the independent sensing data.

IV. Control Design

A direct, simplified localization approach is employed to obtain the UGV's state vector, as shown in the system block diagram shown in Fig. 5. The system utilizes an optical mouse

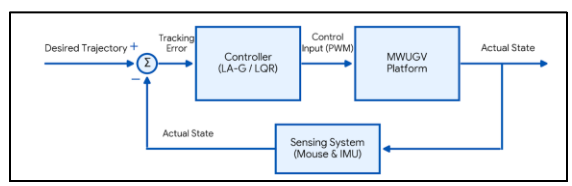

Fig. 5. Block diagram of the closed-loop path-tracking control system for the MWUGV platform.

sensor to track the x and y coordinates and an MPU6050 sensor to measure the heading angle (θ), as illustrated in the system architecture and shown in Fig. 6. This configuration reduces architectural complexity compared to previous works [19] that required dual-mouse setups to derive the vehicle's orientation.

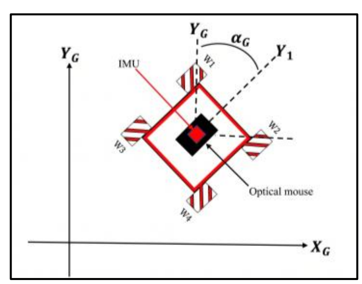

Fig. 6. Mecanum-wheeled UGV equipped with one optical mouse and an IMU at its center in the global coordinate system.

A. *Method 1: Linear Angle to Gain (LA-G)*

The LA-G method, as introduced by Keek et al. [19], serves as an intuitive displacement-controlled algorithm for MWUGVs. It manages the error between the UGV's center coordinates (x, y) and the desired target coordinates (x_{ref}, y_{ref}) using only the vehicle's position and current heading angle. The main advantage of this approach is its simplicity, as it bypasses the need for complex kinematics-based control that requires precise information regarding wheel diameters and the geometric distances between the wheels and the robot's center on both the x and y axes.

The control logic is based on the basic diagonal actuation of the Mecanum platform. By utilizing diagonal wheel pairs or a weighted combination thereof, the UGV can achieve motion in any direction, as shown in Fig. 7. The algorithm calculates a target angle θ^k required to reach the next setpoint:

$$\theta^k = \alpha^k + \alpha^k_{UGV} \quad (3)$$

Where α^k is the relative angle to the target and α^k_{UGV} is the current heading. The required gain for each wheel g_{jk} is then determined by performing linear interpolation between the boundary gains of the nearest primary actuation angles:

$$g_{jk} = \frac{\theta^k - \theta_{LB}}{\theta_{UB} - \theta_{LB}}(g_{UB} - g_{LB}) + g_{LB} \quad (4)$$

Here, UB and LB represent the upper and lower bounds of the specific movement sector. g^k_j is the output gain of J^{th} mecanum wheel at K^{th} time step, θ^k is the input angle, θ_{UB}, θ_{LB} are the input angle (θ^k) upper and lower boundaries, respectively. g_{UB}, g_{LB} are the output gain (g^k_j) upper and lower boundaries, respectively.

The gains are rescaled such that 1.0 corresponds to 100% forward power and −1.0 corresponds to 100% reverse power. Wheels not involved in a specific diagonal motion are assigned a gain of 0.

The complete framework of the LA-G method, including its full coverage, is detailed in [19]. The original methodology defines four primary cases corresponding to four spatial quadrants, determined by the location of the target position relative to the UGV's local origin. Each quadrant is bound by an upper (θ_{UB}) and lower (θ_{LB}) values of θ^k.

To obtain the gain values for each wheel, equation (5) will be used. Each case misses the value of (θ^k) to obtain the value of the gains. However, in real-world situations with uncertainties arising from various factors, the UGV's heading will change during movement. This will result in a difference between the UGV heading angle and θ^k. To overcome this issue, the current angle reading of the UGV (α^k_{UGV}) should be added to the angle between the UGV position (α^k) and the desired position to get θ^k as in (3).

Both position and orientation are managed via closed-loop feedback, with the heading controller assigned priority as per the methodology in [19].

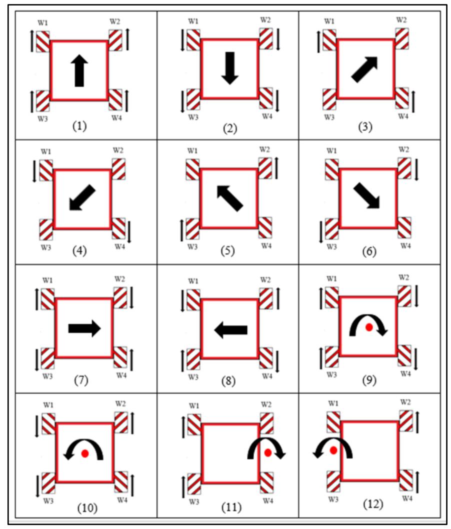

Fig. 7. Resultant MWUGV motion derived from various Mecanum wheel rotation combinations.

B. *Method 2: Modified Linear Angle to Gain (LA-G)*

To improve upon the original LA-G method, a modified version was developed to eliminate the discrete switching and manual interpolation logic between actuation sectors. This is achieved by representing the angle-to-gain relationship as a continuous function using an 8-term summation of sines, which corresponds to an R-square value of 0.9994. This modification reduces the complexity of implementing the controller on the Arduino microcontroller by providing a direct mathematical calculation for wheel gains:

$$f(s) = \sum_{i=1}^{8} a_i \sin(b_i s + c_i) \quad (5)$$

For wheels 1 and 4, the input s in (5) is θ_k, while for wheels 2 and 3, a 90° shift is applied such that $s = \theta_k + 90°$. The constants a_i, b_i and c_i have been found using the MATLAB curve-fitting tool and shown in Table 1. This modification ensures a smoother transition between states.

Table 1. Summation of sines function constants values

Constant	value	Constant	value	Constant	value
a1	0.8106	c3	-0.7908	b6	0.1918
b1	0.01745	a4	0.01657	c6	-2.33
c1	2.356	b4	0.1222	a7	0.004872
a2	0.09008	c4	0.778	b7	0.2276
b2	0.05235	a5	0.01005	c7	-0.9178
c2	-2.355	b5	0.1569	a8	0.00368
a3	0.03243	c5	2.384	b8	0.2625
b3	0.0873	a6	0.006719	c8	0.6078

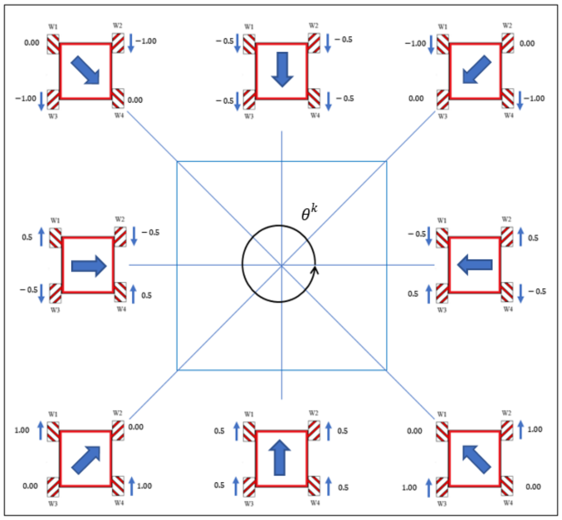

Fig. 8. Wheels gain in a continuous range approach.

The gain values for each wheel were reconsidered as a continuous range relative to the input angle θ_k, spanning from $0°$ to $360°$. This approach transitions away from a discrete quadrant-based logic to a unified mathematical model. As illustrated in Fig. 8, a normalized value of 1.00 represents the maximum (100%) gain, with the total gain range for each wheel extending from -1.00 to 1.00. Specifically, the gain profiles for Wheel 1 and Wheel 4 initiate at -0.5, decrease to a minimum of -1.00, ascend to a maximum of 1.00, and finally return to -0.5 at the completion of the rotation. Conversely, the gain values for Wheel 2 and Wheel 3 follow an identical profile but are phase-shifted by $90°$.

C. Method 3: Linear-Quadratic Regulator (LQR)

The third control strategy implemented is the Linear-Quadratic Regulator (LQR), an optimal control method designed to minimize a quadratic cost function J [21]. Consider the state space model:

$$\begin{gathered} \dot{z} = \mathrm{A}z + \mathrm{B}u \\ p = Cz + Du \end{gathered} \tag{6}$$

Where $z = [x, y, \theta]^T$ is the state vector, and $u = [PWM_{w1}, PWM_{w2}, PWM_{w3}, PWM_{w4}]^T$ is the input vector, and p is the output vector.

The cost function balances tracking performance against the control effort by solving for the controller gain K:

$$J = \int_0^\infty (z^T Q z + u^T R u)\, dt \tag{7}$$

The weight matrices Q and R are central to the design, where:

$$Q = \begin{bmatrix} q & 0 & 0 \\ 0 & q & 0 \\ 0 & 0 & q \end{bmatrix}, \quad R = \begin{bmatrix} r & 0 & 0 & 0 \\ 0 & r & 0 & 0 \\ 0 & 0 & r & 0 \\ 0 & 0 & 0 & r \end{bmatrix}$$

The weight factors $r = 1$ and $q = 1$ are selected as baseline design parameters. These matrices are systematically tuned to satisfy target transient response criteria, minimizing settling time while preventing actuator saturation. The state-feedback gain is computed as $K = R^{-1}B^T P$, where P is the positive-definite symmetric matrix obtained by solving the algebraic Riccati equation:

$$PA + A^T P + Q - PBR^{-1}B^T P = 0 \tag{8}$$

For trajectory-tracking, the model must output position and heading (x, y, θ) rather than velocities. The platform position is obtained by integrating the body velocities over time, by incorporating the robot's geometric parameters ($R_W = 50$ mm, $L = 135$ mm, $W = 75$ mm).

The vehicle is modeled using a kinematic representation in which wheel dynamics and actuator dynamics are neglected. Comparing the state space model (6) with the corrected kinematic model (2), which is based on the experimental input-output steady state map, the system state-space matrices result in $A = [0]_{3\times 3}$.

$$B = \begin{bmatrix} -1.2942 & 1.2942 & -1.2942 & 1.2942 \\ 1.2942 & 1.2942 & 1.2942 & 1.2942 \\ 0.1661 & -0.1661 & -0.1661 & 0.1661 \end{bmatrix}$$

The other state matrices, due to full state feedback, are as follows:

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, D = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

V. Results and Discussion

To evaluate the trajectory-tracking performance of the MWUGV, the three control methods were assessed by comparing the trajectory recorded by the optical mouse sensor (blue) against the target infinity-shaped path (orange), as shown in Fig. 9.

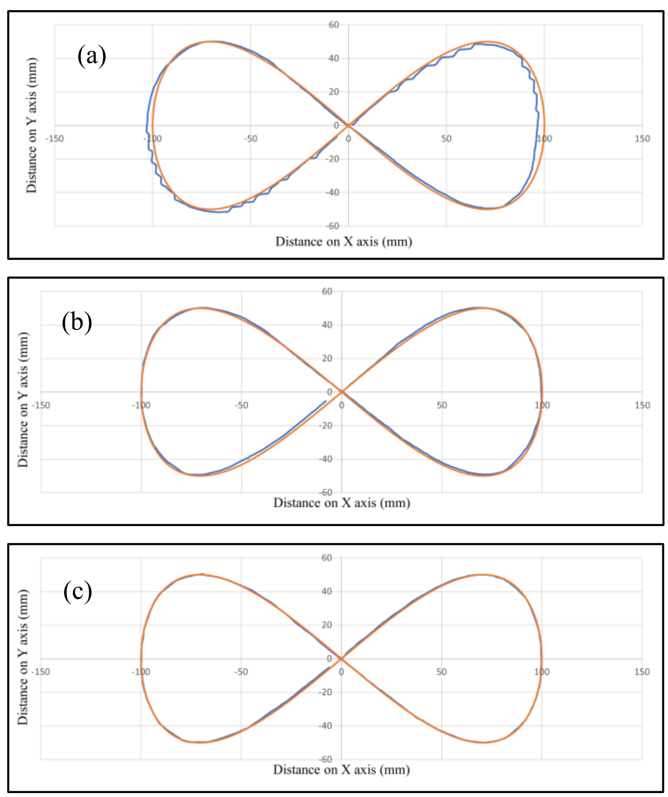

Fig. 9. Trajectory-tracking results of the (a) LA-G method, (b) modified LA-G method, and (c) LQR.

The experimental results demonstrate that the LQR method achieves the highest performance in terms of trajectory-tracking accuracy and exhibits superior precision in following the desired trajectory. It maintained stability and smoothness during the trajectory, resulting in minimal oscillations and overshoots. In contrast, the modified LA-G has better accuracy than the original.

The root mean square error (RMSE) is calculated for each method to measure the tracking error mathematically [22]. The RMSE measures the absolute average difference between the reference and actual recorded trajectories. It is the standard deviation of the distance between the reference trajectory and the recorded data points. The RMSE values for each method are presented in Table 2, which clearly shows that LQR has the lowest RMSE values, indicating the high quality of its performance in trajectory-tracking. The modified LA-G method results are better than the original LA-G, indicating that the proposed modification has enhanced the quality of the original method in trajectory-tracking accuracy.

Table 2. RMSE Values in mm for the Three Methods

Method	RMSE (x-axis)	RMSE (y-axis)	Total 2D RMSE
Original LA-G	2.275	1.543	2.749
Modified LA-G	1.593	1.652	2.322
LQR	0.789	0.907	1.202

VI. Conclusion

In this experimental study, three control strategies for MWUGV trajectory tracking were evaluated: the original Linear Angle-to-Gain (LA-G) controller, a modified LA-G method proposed here, and a Linear-Quadratic Regulator (LQR). It was concluded that, while all three methods are viable for MWUGV trajectory-tracking, they present different trade-offs in complexity and precision. The research used two SISO-based approaches, the original and modified LA-G methods, and one MIMO-based approach employing LQR. A comparative analysis of trajectory-tracking accuracy with an infinity-shaped trajectory revealed that the LQR controller achieved the highest precision, with the lowest total 2D RMSE of $1.202mm$ for both the $x-$ and $y-axes$, followed by the modified LA-G with an average of $2.322mm$, and the original LA-G with an average of $2.749mm$. Although the LA-G methods are simpler because they do not require knowledge of vehicle dimensions or kinematic derivations, the modified LA-G offers better results than the original and functions as a reliable alternative. Ultimately, the LQR demonstrated superior performance by enabling simultaneous control of the vehicle's position and heading angle without disruptions, successfully fulfilling the main objectives of this study. Furthermore, integrating an external high-fidelity tracking system would provide independent ground-truth verification of the experimental results.

References

[1] B. E. Ilon, "Wheels for a course stable self-propelled vehicle movable in any desired direction," U.S. Patent 3 746 112, Jul. 17, 1973.

[2] M. O. Tatar, C. Popovici, D. Mandru, I. Ardelean, and A. Plesa, "Design and development of an autonomous omni-directional mobile robot with mecanum wheels," in *Proc. 2014 IEEE Int. Conf. Autom. Qual. Testing, Robot.*, 2014.

[3] A. Cheong, "Development of a robotic waiter system," *IFAC-PapersOnLine*, vol. 49, no. 21, pp. 681–686, 2016.

[4] F. Adăscăliței and I. Doroftei, "Practical applications for mobile robots based on mecanum wheels - a systematic survey," *Rom. Rev. Precis. Mech. Opt. Mechatronics*, no. 40, pp. 21–29, 2011.

[5] J. B. Song and K. S. Byun, "Design and control of a four-wheeled omnidirectional mobile robot with steerable wheels," *Proc. IEEE Int. Conf. Robot. Autom.*, 2004.

[6] J. Udomsaksenee, "Glocal control for mecanum-wheeled vehicle with slip compensation," 2018.

[7] P. S. Yadav, V. Agrawal, J. C. Mohanta, and M. D. F. Ahmed, "A theoretical review of mobile robot locomotion based on mecanum wheels," *Evergreen*, vol. 9, no. 2, pp. 396–403, 2022.

[8] T. Peng, J. Qian, B. Zi, J. Liu, and X. Wang, "Mechanical design and control system of an omni-directional mobile robot for material conveying," *Procedia CIRP*, vol. 56, pp. 412–415, 2016.

[9] C. C. Tsai, F. C. Tai, and Y. R. Lee, "Motion controller design and embedded realization for mecanum wheeled omnidirectional robots," *Proc. World Congr. Intell. Control Autom.*, no. 1, pp. 546–551, 2011.

[10] L.-C. Lin and H.-Y. Shih, "Modeling and adaptive control of an omni-mecanum-wheeled robot," *Intell. Control Autom.*, vol. 4, no. 2, pp. 166–179, 2013.

[11] M. Killpack, T. Deyle, C. Anderson, and C. C. Kemp, "Visual odometry and control for an omnidirectional mobile robot with a downward-facing camera," *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, pp. 139–146, 2010.

[12] A. Shimada, S. Yajima, P. Viboonchaicheep, and K. Samura, "Mecanum-wheel vehicle systems based on position corrective control," *IECON Proc.*, vol. 2005, pp. 2077–2082, 2005.

[13] P. Viboonchaicheep, A. Shimada, and Y. Kosaka, "Position rectification control for mecanum wheeled omni-directional vehicles," *IECON Proc.*, vol. 1, pp. 854–859, 2003.

[14] B. Chu, "Mobile robot position control algorithm based on multiple ultrasonic distance sensors," *ICCAS 2015*, pp. 1238–1240, 2015.

[15] C. Bühler, R. Hoelper, H. Hoyer, and W. Humann, "Autonomous robot technology for advanced wheelchair and robotic aids for people with disabilities," *Rob. Auton. Syst.*, vol. 14, no. 2–3, pp. 213–222, 1995.

[16] S. Guo, Q. Diao, and F. Xi, "Vision based navigation for omni-directional mobile industrial robot," *Procedia Comput. Sci.*, vol. 105, pp. 20–26, 2017.

[17] V. Alakshendra and S. S. Chiddarwar, "A robust adaptive control of mecanum wheel mobile robot: Simulation and experimental validation," *IEEE Int. Conf. Intell. Robot. Syst.*, pp. 5606–5611, 2016.

[18] K. Nagatani, S. Tachibana, M. Sofue, and Y. Tanaka, "Improvement of odometry for omnidirectional vehicle using optical flow information," *IEEE Int. Conf. Intell. Robot. Syst.*, vol. 1, pp. 468–473, 2000.

[19] J. S. Keek, S. L. Loh, and S. H. Chong, "Comprehensive development and control of a path-trackable mecanum-wheeled robot," *IEEE Access*, vol. 7, pp. 18368–18381, 2019.

[20] S. F. Wani, A. Ashraf, and A. Sophian, "Prototype Development of Mecanum Wheels Mobile Robot: A Review," Appl. Res. Smart Technol., vol. 3, no. 1, pp. 38–48, 2022.

[21] T. Shaqarin, "Robustness analysis of feedback linearisation and LQR control on quarter-car model with cubic nonlinearity," *International Journal of Vehicle Noise and Vibration*, vol. 14, no. 3, p. 238, 2018.

[22] A. Bonarini, M. Matteucci, and M. Restelli, "Automatic error detection and reduction for an odometric sensor based on two optical mice," *Proc. IEEE Conf.*, no. 1, pp. 75–80, 2005.