INNOVATIVE JOURNAL
OF INTELLIGENT CONTROL AND OPTIMIZATION FOR ELECTRICAL SYSTEM

Vol. 2 (2025) No. 2

Publisher by AITEKS : Aliansi ahlI TEKnologi dan Sains
https://ijcioes.org/index.php/ijcioes

Position and Orientation Estimation of Omniwheel Robot Using IMU

M. Naufal Yaafi[a], Firman Hermanto[a], Efrizon[a,*], Reza Nandika[a], Iqbal Kurniawan Asmar P[a]

[a] *Electrical Engineering Department, Politeknik Negeri Padang, Limau Manis, Padang, 25176, Indonesia*

Corresponding author: efrizon_s@yahoo.co.id

***Abstract*— This research successfully designed and implemented a position and direction estimation system (odometry) on a four-wheeled omnidirectional robot by integrating a rotary encoder sensor and an Inertial Measurement Unit (IMU). The goal is to create autonomous robots capable of moving holonomically (in any direction without changing body orientation) with high accuracy for the ABU Robocon 2024 competition. The research method uses a Research and Development (R&D) approach by building a prototype of a robot controlled by the Arduino Mega 2560. The navigation control and trajectory tracking system is designed with a gyrodometry algorithm that fuses position data from two rotary encoders (X and Y axes) and orientation data from MPU6050 IMU sensors. The test results showed that the robot was able to follow a rectangular pattern trajectory (1000 mm x 500 mm) with a maximum position error of 50 mm on the X axis and 70 mm on the Y axis. The discussion of results identified that the integration of encoder and IMU data through complementary filters successfully improved the system's resistance to gyroscope drift and wheel slippage. It was concluded that this prototype of an omnidirectional robot with a fusion sensor-based odometry system has met the criteria of sufficient stability and accuracy for robotics competition applications.**

***Keywords*— IMU MPU6050, Navigation Control, Odometry, Omni Robot, Sensor Fusion, Trajectory Tracking.**

Manuscript received 25 Oct. 2025; revised 15 Dec. 2025; accepted 30 Dec. 2025. Date of publication 31 Dec. 2025.

I. INTRODUCTION

Omnidirectional robots are a class of holonomic mobile robots capable of performing translational and rotational movements in any direction without changing their body orientation. This unique capability allows such robots to maneuver efficiently in confined and dynamic environments, making them well suited for applications that require high mobility and precise motion control, including robotics competitions and industrial automation systems [1].

In the ABU Robocon 2024 competition, participating robots are required to execute complex tasks that demand fast, stable, and accurate navigation. In such scenarios, even small errors in position and orientation estimation can significantly degrade overall system performance and affect task completion. Therefore, the development of a reliable autonomous navigation system with accurate localization and heading control is a critical requirement for competition robots [2].

One of the most commonly used navigation methods for mobile robots is rotary encoder-based odometry, which estimates position by integrating wheel rotation data. Although this approach is computationally simple and efficient, it is highly susceptible to cumulative errors caused by wheel slippage, uneven surfaces, and mechanical tolerances [3]. These limitations can lead to significant drift in both position and orientation over time. To address this issue, inertial sensors such as Inertial Measurement Units (IMUs) are often integrated with odometry data using sensor fusion techniques, a method commonly referred to as gyrodometry [4][5].

In this study, a position and direction estimation system for a four-wheeled omnidirectional robot is proposed by integrating rotary encoder measurements with an MPU6050 IMU sensor. A complementary filter is employed to combine odometry and inertial data in order to improve orientation accuracy and reduce drift. The effectiveness of the proposed system is evaluated through trajectory tracking experiments, including linear, rectangular, and zig-zag paths, to assess positioning accuracy and orientation stability. The results demonstrate the suitability of the proposed approach for autonomous navigation in competitive robotics applications[6].

II. METHOD

A. System Design and System Architecture

The omnidirectional robot system is designed with an integrated hardware and software architecture to achieve autonomous navigation. The block diagram of the whole system is shown in Figure 1. This system consists of several main subsystems: (1) Input in the form of target coordinate set points (x, y, θ) and velocity; (2) Processing Unit that uses Arduino Mega 2560 as the main controller; (3) Sensor System which includes two rotary encoders (X and Y axis) and MPU6050 IMU sensors; (4) Actuation System in the form of four PG45 24V DC motors with BTS7960 drivers; and (5) Feedback Loop that closes the control system based on sensor readings. The System Design and System Architectureare shown in Figure 1.

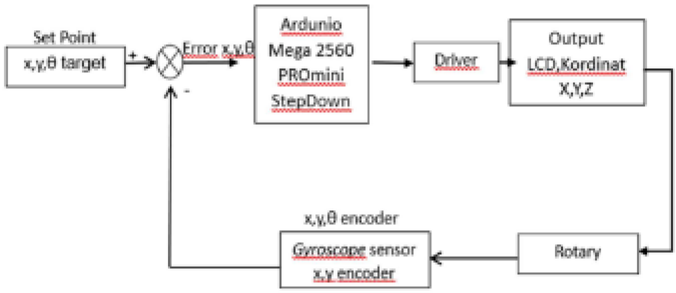

FIGURE 1 System Design and System Architecture Automatic Robot Omniwhell Wheels Using Imu Sensors

In the diagram, The robot is designed with a configuration of four-piece omnidirectional wheels that are arranged holonomically at angles of 45°, 135°, 225°, and 315° to the axis of the robot's body. The robot's mechanical design follows the maximum limit of the ABU Robocon 2024 competition with a size of 500 mm x 500 mm

B. Hardware

The hardware configuration of the robot system consists of several main subsystems, including actuators, motion sensors, inertial sensors, a controller, and a power supply unit. Each component is selected to support accurate motion control, position estimation, and stable system operation. The detailed hardware components and their respective functions are summarized in Table 1.

TABLE I
Device Specifications for Bird Pest Monitoring and Repellent Systems

No.	Device	General Specifications	Function
1	DC Motor	PG45 24V DC Motor	Main processing units Functions as the main actuator for robot movement.
2	Encoder	Internal Encoder (400 PPR)	Provides wheel rotation feedback for motion control.
3	Motor Driver	BTS7960 Motor Driver	Repelling bird pests
4	Motion Sensor A	Rotary Encoder (Free Wheel)	SMS notification delivery
5	Motion Sensor B	Rotary Encoder (Free Wheel)	Actuator Operator
6	IMU Sensor	MPU6050 IMU	System resources
7	Arduino	Arduino Mega 2560	Early motion detection triggers
8	Battery	6S 24V 6200 mAh Li-Po BAT	
9	Step-Down	LM2596 Step-Down Module	
10	Voltage Regulator	7805 Voltage Regulator	

The system is geared for use in rice fields, so field power supply options (e.g. solar panel–battery–charge controller–step-down) can be used to keep the system working when away from the power source. The components in table I were chosen because they are easy to obtain and sufficient to support *real-time* detection on embedded devices.

C. Control Algorithm and Trajectory Tracking

The control system is designed to move the robot from an initial position to a specified target with high precision. The process begins with system initialization and target input, where the microcontroller, sensors, and variables are configured, and the target position along with the desired speed (PWM) is defined. Real-time position and orientation data are obtained from the X–Y rotary encoders and the MPU6050 IMU. Encoder pulses are converted into translational displacement, while angular velocity from the IMU is integrated and fused with odometry using a complementary filter to estimate the robot's orientation [7]. Position and orientation errors are then computed and processed by a proportional controller to generate global velocity commands ((V_x, V_y, \omega)), which are transformed into individual wheel angular velocities through inverse kinematics. The resulting wheel velocities are mapped to PWM signals and rotation directions and applied to the motors via the BTS7960 drivers. This feedback loop runs continuously until the position and orientation errors fall below the predefined tolerance, indicating that the robot has reached the target, as illustrated in Figure 2.

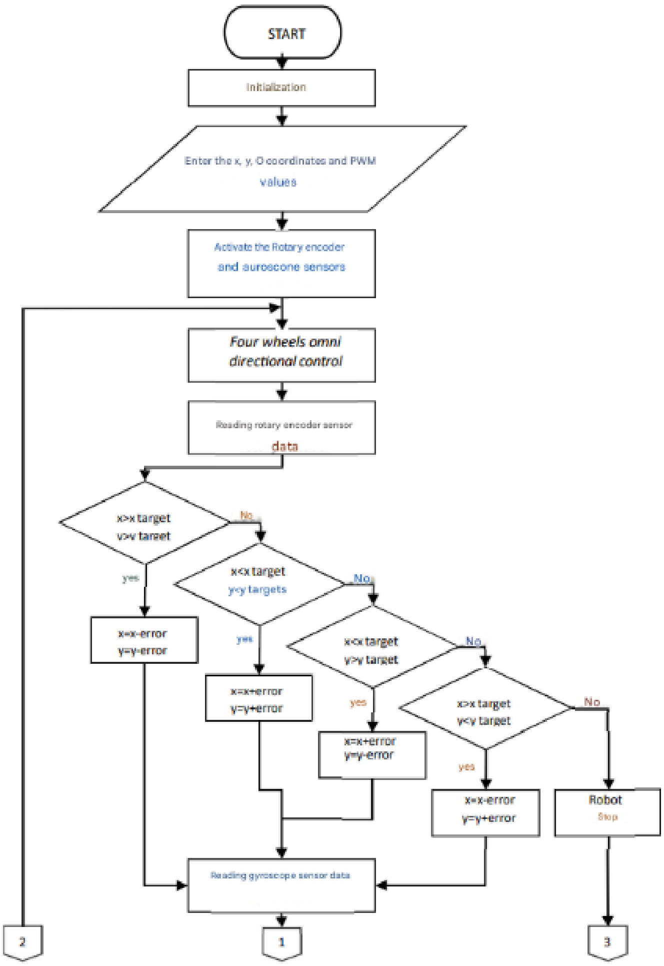

FIGURE 2 Flowchart of Automatic Robot Omniwhell Wheels Using Imu Sensors

The flowchart shows the robot control system. The robot moves omnidirectionally by continuously reading encoder and gyroscope data, comparing the current position with the target coordinates, and updating the position error. This control loop is repeated until the robot reaches the target position, after which the robot stops.

The flowchart further elaborates on the decision-making logic used to minimize the distance between the robot's current coordinates and the target. Once the rotary encoder and gyroscope sensors are activated, the system enters a continuous monitoring cycle. The controller evaluates the robot's position relative to the target by comparing x and y values. If the current position exceeds or falls short of the target coordinates, the system calculates the specific error and adjusts the movement direction accordingly—adding or subtracting the error value to refine the trajectory. This proportional logic ensures that the omnidirectional wheels can compensate for any drift in real-time. The process repeats iteratively, constantly updating the motor outputs through the BTS7960 drivers. Only when all conditional checks for x and y are satisfied (meaning the error is within the allowed tolerance) does the algorithm exit the loop and trigger the "Robot Stop" command, ensuring a precise arrival at the destination.

D. Datashet Training

The training datasheet in this study is constructed from experimental data collected through repeated robot motion trials conducted under controlled conditions. The dataset comprises rotary encoder measurements used to estimate translational displacement along the X and Y axes, gyroscope readings from the MPU6050 sensor to capture angular velocity, as well as the resulting estimated robot position, orientation, and control outputs generated by the navigation algorithm [8].

These data are utilized to calibrate key system parameters, including encoder conversion factors, complementary filter coefficients for sensor fusion, and proportional controller gain constants. Parameter tuning is performed iteratively by analyzing the system response during linear and trajectory-based motion tests to minimize position error and orientation drift. The training process focuses on achieving a balance between responsiveness and stability to ensure smooth robot motion without excessive oscillation.

As a result, the training datasheet plays a critical role in improving the overall performance of the autonomous navigation system by enabling accurate parameter adjustment. This approach ensures stable trajectory tracking, precise positioning, and reliable orientation maintenance, which are essential for omnidirectional robot operation in competition environments.

E. Error Correction and Neutralization Response Mechanism

The error correction and neutralization mechanism is implemented through a closed-loop feedback control system. Positional errors along the X and Y axes and orientation errors are continuously calculated by comparing the estimated state with the target reference. When deviations are detected, the proportional controller generates corrective velocity commands that adjust the individual wheel speeds via inverse kinematics [9]. Orientation errors detected by the IMU are neutralized by reducing or increasing wheel velocities on specific sides of the robot, allowing the heading to be restored toward the desired orientation. This mechanism ensures stable motion and prevents accumulated drift during navigation.

F. Testing Scenarios

Several testing scenarios are conducted to evaluate the performance of the proposed system. The robot is commanded to move to predefined target coordinates with varying distances and directions. Each test evaluates positional accuracy, orientation stability, and response time. The experiments are repeated multiple times to assess consistency and robustness. Performance metrics such as maximum error, average error, and orientation deviation are recorded and analyzed to validate the effectiveness of the control and sensor fusion approach.

G. Integration of IMU and Complementary Filters for Direction Estimation

To overcome drift in odometry and obtain a more reliable estimate of direction, gyroscope data from MPU6050 is integrated. The angular change rate (ω) of the gyroscope is integrated against time to obtain the angular change (Δθ_gyro). However, gyroscopes are susceptible to long-term bias drift [10]. Therefore, complementary filters are used to combine angle estimates from the gyroscope (short-term precision) and from odometry (long-term accuracy). The estimated end angle (θ) is calculated by Equation 1.

$$\theta_k = \alpha\theta_{k-1} + \omega\Delta t + (1-\alpha)\theta_{odom} \quad (1)$$

Where α is the tunable constant ($0 < \alpha < 1$) that determines the weighting, and θ_odom is the angle calculated from the change in the position of the odometry show (Equation 5).

H. Robotic Kinematics and Odometry Calculations

The control algorithm is designed to guide the robot accurately from the initial position to a specified target position. The overall control flow follows the scheme shown in Figure 3. At the beginning, the microcontroller, sensors, and system variables are initialized. The target position and orientation are defined as x_target, y_target, and θ_target, along with a desired speed reference value.

Position feedback is obtained from two rotary encoders installed orthogonally to measure displacement along the X and Y axes, while orientation data are provided by the MPU6050 IMU sensor. Encoder pulse data are converted into translational displacement using odometry equations, whereas angular velocity data from the gyroscope are integrated to estimate the robot's heading. To reduce drift and noise, the angular estimation from odometry and IMU is combined using a complementary filter, resulting in a more robust orientation estimate θ_est.

Where Kpx, Kpy, and Kpθ are predetermined proportional gain constants. The global velocity commands are subsequently converted into individual wheel angular velocities using the inverse kinematics model of the omnidirectional robot. The resulting wheel speeds are mapped into PWM values and rotation directions (clockwise or counterclockwise) and sent to the BTS7960 motor driver modules to drive the DC motors. This control loop runs continuously, updating sensor feedback and control outputs, until all position and orientation errors fall below predefined tolerance limits, indicating that the robot has successfully reached the target point

III. RESULT AND DISCUSSION

A. System Design and System Architecture

Preliminary testing was conducted to ensure that all four PG45 motors and the encoder system were functioning stably under PWM control. Table 1 presents the results of the motor speed (RPM) test read by the motor's internal encoder compared to the standard tachometer readings at various PWMT values

The results show a linear relationship between the PWM value and the motor speed. At PWM 255 (100% duty cycle), the motor speed is close to 106-113 RPM, according to the characteristics of PG45 motors at 24V voltage. A small difference between the encoder and tachometer readings (average error <5%) indicates that the motor's internal encoder is reliable for speed feedback. The stability of the four motors also looks good, with insignificant RPM variations, thus supporting symmetrical robot kinematics.

Furthermore, the linear correlation between the PWM duty cycle and the rotational speed is critical for establishing a predictable control model for the robot's movement. By maintaining an error rate below the 5% threshold, the system demonstrates high precision in its internal feedback mechanism, which reduces the dependency on external measuring tools during operation. This level of stability across all four PG45 motors is essential for omnidirectional robots, as even minor discrepancies in RPM between wheels could lead to unwanted drifting or orientation errors. Consequently, these findings validate the reliability of the hardware architecture, ensuring that the proportional controller discussed in the previous section can execute trajectory tracking commands with consistent and balanced motor responses.

B. Odometry System Accuracy on X and Y Axes

The fundamental accuracy of the odometry system was tested by commanding the robot to move 100-1000 mm straight on the X and Y axes separately. The results of the comparison between the mileage of the odometry estimate and the physical measurement of the bar are presented in Table 2.

TABLE II

Results of Odometry Calibration and Accuracy on the X and Y Axes

No.	**Target (mm)**	**X-Axis**			**Y-axis**		
		Yield (mm)	**Error (mm)**	**Error (%)**	**Yield (mm)**	**Error (mm)**	**Error (%)**
1	100	100.57	0.57	0.57	107.28	7.28	7.28
2	500	501.49	1.49	0.30	502.86	2.86	0.57
3	1000	1009.53	9.53	0.95	1032.53	32.53	3.25
Average			6.45	1.30		10.66	2.30

The data shows the average accuracy of the odometry system of 98.7% on the X axis and 97.7% on the Y axis. The error on the Y-axis is larger, thought to be caused by a slight mechanical misalignment in the installation of the Y free wheel encoder. Overall, errors below 5% are acceptable for robot navigation on a scale of several meters and can be further corrected by closed control algorithms as well as IMU integration.

C. Mamdani's Fuzzy Logic Validation

The main test of the system was carried out by instructing the robot to follow two closed trajectory patterns: rectangular and zig-zag. The actual trajectory results are compared to the ideal trajectory.

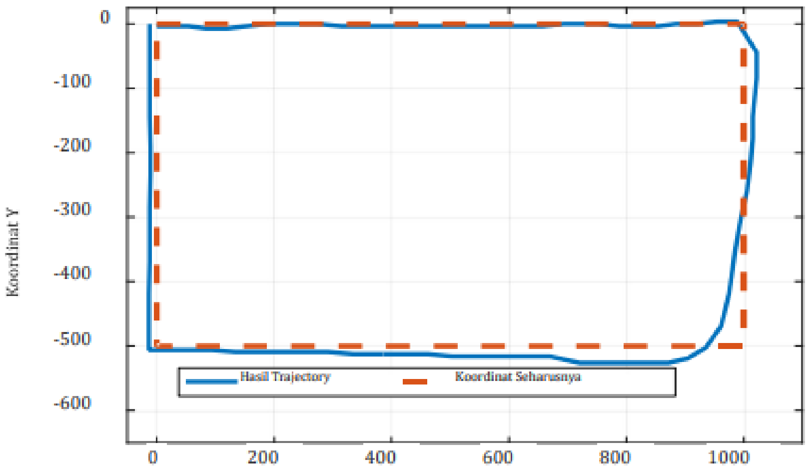

FIGURE 3. Results of Robotic Trajectory in Rectangular Patterns

Based on the test data presented in Table 2, the overall performance of the system can be summarized as follows. The maximum positioning error occurs at the coordinate point (1000, 500), with an error of 50 mm along the X-axis and 70 mm along the Y-axis. In terms of accuracy, the system achieves an average error of 4.75% on the X-axis and 4.71% on the Y-axis, indicating relatively consistent positional estimation across the test area. Regarding orientation control, the IMU sensor is able to detect angular deviations of up to 5° during robot motion. When such deviations are detected, the control system applies corrective actions by adjusting and reducing the wheel speeds on specific sides of the robot, allowing the heading to be restored and maintained close to 0°.

The robot was commanded to follow a zig-zag trajectory defined by the sequence of points (0,0) → (400,0) → (800,400) → (400,800) → (800,800). The visualization of the experimental results indicates that the robot was able to track the reference path with a consistent motion pattern and relatively small deviations from the desired trajectory. The recorded path closely follows the predefined waypoints, demonstrating the capability of the system to execute complex directional changes smoothly. This trajectory visualization, as presented in Figure 4, provides an overview of the robot's movement behavior during the zig-zag motion test and serves as a qualitative representation of the system's tracking performance based on Figure 4.

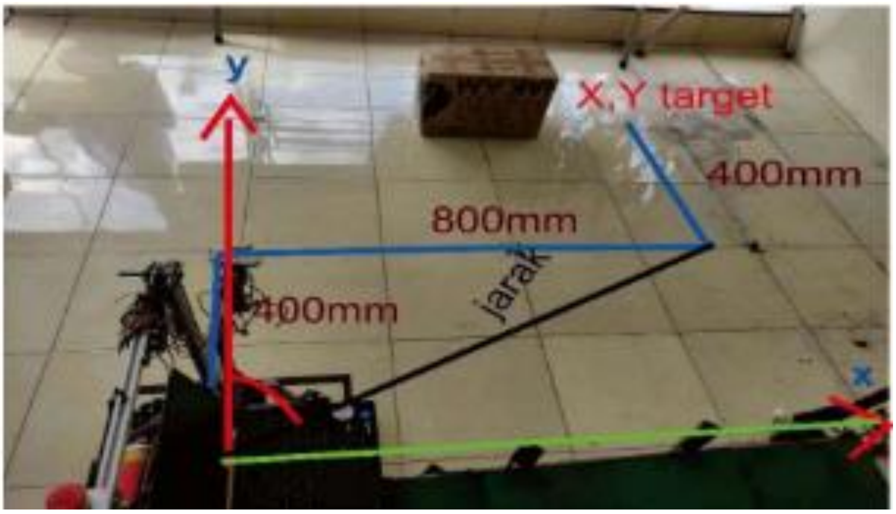

FIGURE 4. Robotic Trajectory Results in a Zig-Zag Pattern

Based on the test data reported in Table 4.6, the quantitative performance of the system during the zig-zag pattern experiment can be considered satisfactory. The maximum positional error observed was 20 mm on both the X and Y axes, indicating that the tracking error remained within acceptable limits. In addition, the average error was lower compared to the rectangular trajectory test, which suggests that the control system performs better when handling gradual changes in direction rather than sharp turns. From the perspective of motion stability, the robot was able to maneuver through directional transitions without excessive oscillations or instability. This result confirms the effectiveness of the reverse kinematics–based speed control approach in maintaining smooth motion and reliable trajectory tracking under dynamic path conditions.

D. Analysis of IMU Integration and Complementary Filters

The integration of gyroscope data from the MPU6050 IMU with wheel odometry is intended to enhance the reliability of heading estimation, particularly under conditions where wheel slippage and mechanical disturbances occur. In systems that rely solely on odometry, orientation errors tend to accumulate over time due to uneven wheel-ground interaction and encoder inaccuracies. Experimental results from the rectangular trajectory test show that, without IMU-based correction, the accumulated orientation error can reach up to 15° after completing the trajectory.

To address this issue, a complementary filter is applied to fuse angular velocity data from the IMU with orientation information derived from odometry. A filter coefficient of $\alpha = 0.98$ is selected to emphasize short-term gyroscope responsiveness while allowing low-frequency odometry information to correct long-term drift. This configuration effectively suppresses gyroscope drift and reduces noise introduced by odometry measurements.

With the implementation of the complementary filter, the orientation error during trajectory tracking tests is consistently maintained below 5°. These results demonstrate that the fusion of odometry and IMU data significantly improves both the stability and accuracy of direction estimation when compared to the use of odometry alone. Consequently, the proposed approach provides a reliable solution for maintaining heading stability in omnidirectional robots operating in competition environments

IV. CONCLUSION

Based on the results of this study, an integrated odometry–IMU-based position and orientation estimation system for an omnidirectional mobile robot has been successfully designed and implemented. The holonomic four-wheel configuration enables the robot to perform omnidirectional motion without changing its body orientation, which is highly suitable for navigation tasks in competitive robotics environments such as the ABU Robocon 2024 competition. The application of trajectory tracking control using wheel odometry combined with orientation correction from the MPU6050 IMU through a complementary filter demonstrates stable and reliable robot motion. Experimental results show that the system achieves high positioning accuracy, with average odometry accuracies of 98.7% along the X-axis and 97.7% along the Y-axis during linear motion tests. In trajectory tracking experiments, the robot successfully follows a rectangular path of 1000 mm × 500 mm with a maximum positional error of 70 mm and an average error of 4.73%, as well as a zig-zag trajectory with a maximum error of 20 mm. Furthermore, the integration of IMU data effectively maintains orientation stability during movement, with angular deviations consistently remaining below 5°. This indicates that the proposed sensor fusion approach is capable of reducing orientation drift and improving heading control during complex maneuvers. Overall, the combination of wheel odometry, IMU-based orientation estimation, proportional control, and inverse kinematics results in an accurate and stable autonomous navigation system. These results confirm that the proposed system is suitable for application in omnidirectional competition robots. Future work may focus on implementing more advanced control strategies, such as PID or adaptive control, improving sensor fusion techniques, and enhancing robustness under dynamic competition conditions

REFERENCES

[1] S. J. Johnston et al., “City Scale Particulate Matter Monitoring Using LoRaWAN Based Air Quality IoT Devices,” pp. 1–20, doi: 10.3390/s19010209.

[2] J. Jurnal et al., “Penerapan Rotary Encoder sebagai Sistem Odometry untuk Navigasi Robot Sepak Bola Beroda,” 2025.

[3] I. K. Wibowo et al., “Penentuan Posisi Robot Menggunakan Odometry Omniwheel,” pp. 8–10, 2018.

[4] R. Susanti, R. Nofendra, M. Syaiful, and M. Ilhamdi, “The Use of Artificial Neural Networks in Agricultural Plants The Use of Artificial Neural Networks in Agricultural,” vol. 2, pp. 62–68, 2022.

[5] H. Muhammad and F. Al, “Rancang Bangun Sistem Positioning Mobile Robot Omnidirectional Wheel Menggunakan Stm32 Berbasis Fuzzy Logic Controller Muhamad Syariffuddien Zuhrie , I Gusti Putu Asto Buditjahjanto , Lilik Anifah Abstrak,” Pp. 547–555.

[6] Hidayaturrahman, Hadi, R., & Yulherniwati. (2020). Prototipe Robot Pengangkut Barang dengan Sistem Pengalamatan Lokasi di dalam Gudang. JITSI : Jurnal Ilmiah Teknologi Sistem Informasi, 1(3), 80 - 86. https://doi.org/10.62527/jitsi.1.3.13

[7] M. H. Gustiana et al., “Perancangan Sistem Kendali Nirkabel Robot Pengantar Barang,” vol. 5, no. 2, pp. 64–71, 2023.

[8] K. Arief, D. Purwanto, H. Kusuma, and A. L. Fuzzy, “Algoritma Menghadang Bola dengan Metode Fuzzy Logic untuk Robot Penjaga Gawang Sepak Bola Beroda,” vol. 7, no. 2, 2018.

[9] D. U. Rijalussalam, M. Budiyanto, and A. Mayub, “Trajectory Tracking pada Robot Omni dengan Metode Odometry,” vol. 8, no. 1, 2019.

[10] Zamri, Nurul Aqilah, M. Izham Jaya, Siti Salwani Yaakob, Hidra Amnur, and Shahreen Kasim. “Multidimensional Indicator for Data Quality Assessment in Wireless Sensor Networks: Challenges and Opportunities.” International Journal on Advanced Science, Engineering and Information Technology 14, no. 5 (2024): 1663–72. https://doi.org/10.18517/ijaseit.14.5.11492.

[11] D. E. Ardana, Y. Z. Maulana, and G. Wibisono, “Jurnal Nasional Teknik Elektro Digital PID Trainer Based On Arduino For DC Motor Speed Control With Ziegler Nichols Method,” vol. 3, 2025.

[12] R. Setiawan, H. H. Triharminto, and M. Fahrurozi, “Gesture Control Menggunakan IMU MPU 6050,” vol. 3, no. November, pp. 24–25, 2021, doi: 10.54706/senastindo.v3.2021.133