ISSN 2087-3336 (Print) | 2721-4729 (Online)

TEKNOSAINS: Jurnal Sains, Teknologi dan Informatika
Vol. 12, No. 1, 2025, page. 110-121
http://jurnal.sttmcileungsi.ac.id/index.php/tekno
DOI: 10.37373

Omni wheel robot movement exploration using a control system for military surveillance with integrated sensor

Jeffri Kurniawan[1*], Mokhammad Syafaat[1], Kasiyanto[1], Dekki Widiatmoko[1], Rafi Maulana[1], Zihar Nanggala Putra[2]

[1*] Politeknik Angkatan Darat, Desa Pendem, Junrejo, Kota Batu, Jawa Timur-Indonesia, 65324
[2] Pusat Kesenjataan Artileri Medan, Jalan Baros, Cimahi Tengah, Kota Cimahi, Jawa Barat-Indonesia, 40521

[*] jeffrikurniawan080696@gmail.com

Submitted: 19/06/2024 Revised: 22/08/2024 Accepted: 19/09/2024

ABSTRACT

Industry 4.0 is bringing about a rapid advancement in technology, particularly in the military industry. This robot was designed to reduce danger to TNI members while monitoring military operations more swiftly and safely. The application of IMU sensors for acceleration movements, measurement of control system error values, and trajectory tracking with multiple pathways are the three modes of the robot control system observed in this study. We assess each of these modes using an analytical observational method. In addition to an MPU 6050 ultrasonic and gyro sensor to detect movement using an accelerometer and gyroscope, the robot is equipped with a rotational encoder sensor to link the DC motor rotation. The PID control approach uses inverse kinematics to regulate the DC motor. The exam is administered on a track that has squares, triangles, and circles as obstacles. According to test results, the tachometer's inaccuracy rate in RPM is about 1.2% higher than the rotary encoder. The movement success percentage for the omni-wheel robot is 89.31%. Square trajectories have an average error rate of 11%, circular trajectories of 15%, and triangular trajectories of 4.17%. The motor rotation speed is most stable along the triangle path. Overall, this study demonstrates the accuracy and speed with which a mobile omni-wheel robot equipped with a control system can peek. This could raise military operations' level of efficiency and safety.

Keywords: Omni-wheel; Sensor IMU; proportional integral derivative (PID), trajectory control system, autonomous robot technology

1. INTRODUCTION

In the era of Industrial Revolution 4.0, technology is advancing quite quickly [1]. Combat robots are examples of new military hardware that makes use of automation and communication technology advancements to support Indonesian Army operations from a distance [2]. Robots used for military reconnaissance lower the danger of casualties and material and people losses as compared to traditional approaches [3]. The loss of military personnel minimizes material and personnel losses. Regular work is hampered by human physical constraints [4]. Urban warfare takes place in urban areas, where operations will cease if the region becomes immobilized [5].

The term "Internet of Things" (IoT) often refers to a concept that involves devices acting as internet-based communication media [6]. The IoT may now be used to remotely operate the Omni-wheel robot instead of relying on human Bluetooth control [7]. The Indonesian Army uses Omni Wheel, an AI-powered robotics system, to track adversaries and minimize man-losses [8]. The gyroscope and accelerometer combo determines the robot's relative location, DC motor rotation, and acceleration values [9]. Six degrees of freedom are available for measuring acceleration and rotation with the IMU

sensor. Data is transmitted using an antenna transceiver [10]. With the use of GPS and inertial navigation devices, troops are tracked and managed [11]. Sensors and IMU monitors are needed for unmanned vehicles to send data and remotely monitor objects.

The goal of waypoint design is to enable mobile robots that are autonomous to read and identify direction and position using the earth coordinate system [12]. Without the assistance of an odometer, autonomous robots can maneuver and avoid obstacles on their own. Omni-wheel robots are equipped with IoT sensors and gyroscopes to track their position, stabilize, and spy on adversaries [13]. The HY-SRF05 claims that robot motion and path adjustment are improved by the IoT gyroscope and ultrasonic sensor.

The Omni wheel robot uses fuzzy logic to boost speed, prevent collisions, and convert GPS for DC motors to PWM [14]. The degrees of this fuzzy logic range from 0 to 1 [15]. This study examines the movement of Internet of Things-based autonomous combat robots.

2. METHOD

This analytical observational study logs signs on the research item to gather specific data. a) Track the trajectory to analyze the distance; b) use the IMU sensor to accelerate the autonomous robot's motion; and c) calculate the error using the PID control system's angle of degree of freedom on the x and y axes. The scope of this investigation is restricted to reconnaissance in an area of one kilometer. The results of this investigation are displayed in a flowchart in the following graphic:

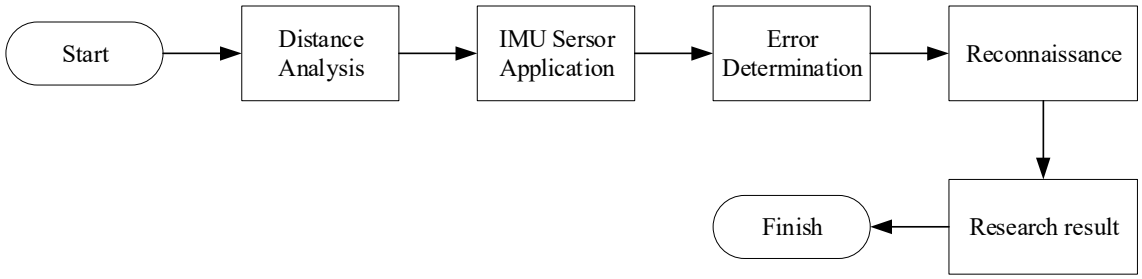


Figure 1. Omni wheel robot sensor and control analysis flowchart.

3. RESULTS AND DISCUSSION

System design for autonomous Omni-wheel robot

The concept and design of an autonomous reconnaissance robot with an IoT-based Omni-wheel and IMU make use of the materials and photographs that are already available.

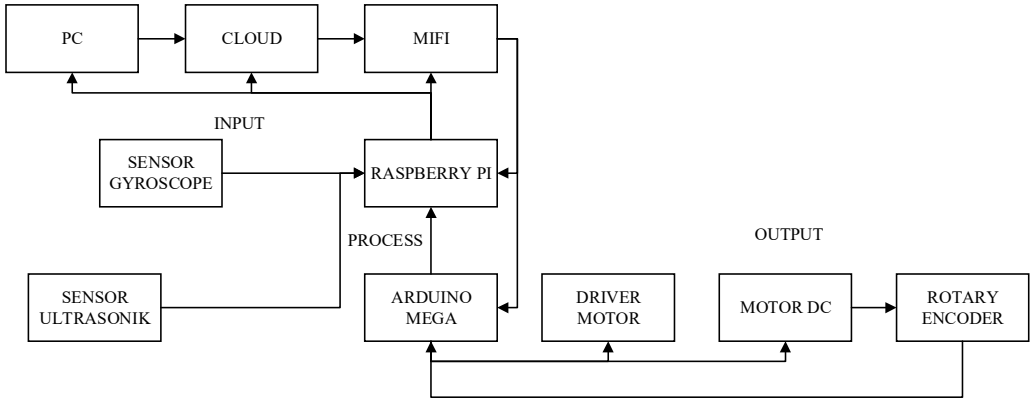


Figure 2. System block diagram.

The structure and operation of a control system based on sensors, microcontrollers, and DC motors are depicted in a block diagram in Figure 2. The PC, which serves as a control center for running applications and monitoring sensor data, the cloud, which is utilized for analysis and remote access, and the MIFI, which wirelessly links the Raspberry Pi to the PC and cloud, are some of the crucial parts of this robotic system. While the ultrasonic sensor ensures safe navigation by determining object distance,

the gyroscope sensor detects direction and rotation. Arduino Mega can operate a DC motor using a motor driver. The rotation encoder gives precise feedback by transmitting motor position and speed data to Arduino Mega. Diagram for the design of an autonomous robot system

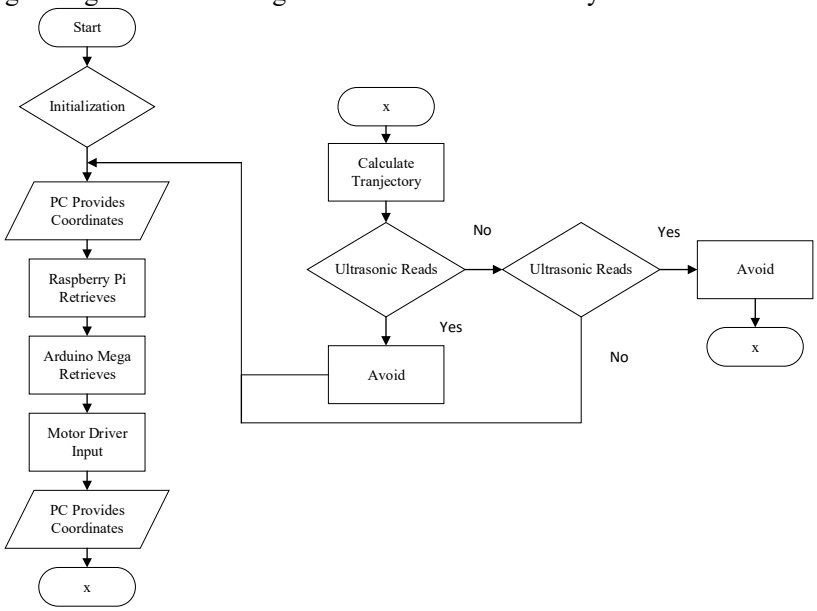


Figure 3. System flow diagram.

The system's methods and operational mechanisms are depicted in the flowchart. The research findings are compiled in the flowchart. The "Start" phase of the system occurs before the "Initialization" phase, ensuring that the hardware and software are operating as intended. Destination coordinates are sent from the PC to the Raspberry Pi so that the Arduino Mega can be used as an extra control device. The motor position is kept on course by the motor driver's input, Rotation Based on Encoders While Trajectory Calculation determines the optimal path to the target, Ultrasonic Reads ensure safety by detecting obstructions in the way. Input guarantees that the motor position is on track. If an obstacle is present, "Avoid" it by veering off course to prevent a collision. The system will be on target and prepared to halt if it is aligned with the goal. "Stop" signals the cessation of movement once the target is reached, while "End" indicates the system's conclusion.

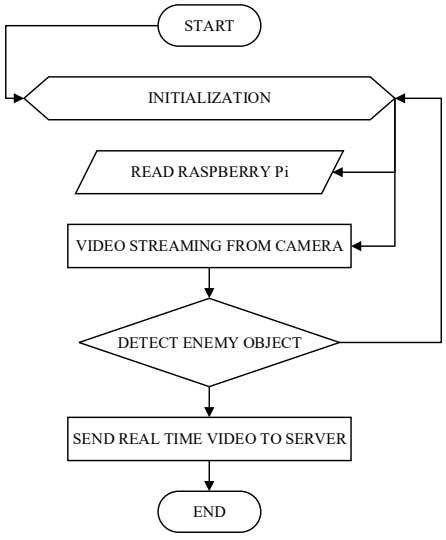


Figure 4. Camera tracking flowchart.

An illustration of the process flow in a system based on a Raspberry Pi is called a trace flowchart. It goes into great depth about each process, and an explanation is available here: Pressing the "Start" button initiates the process and turns on the system. After that, "Initialization" gives the Raspberry Pi module's hardware, software, and settings. The "Reading Raspberry Pi" gathers data from the camera and other linked sensors and modules. Next, "Sending Video from Camera" is initiated by the system, sending a video for watching or educational purposes from the Raspberry Pi camera. The Raspberry Pi algorithm is used in "Detecting Enemy Objects" to find targets. If a target is found, the system reaches "End" once all tasks have been finished. You have the option to end the procedure now or to begin over from the beginning.

MPU6050 sensor testing (accelerometer and gyroscope)

The MPU-6050 tracks the robot's angular velocity and balance while transmitting data to the Arduino at a baud rate of 9600. For increased precision, a flywheel is employed [16]. Measurements of acceleration along the x, y, and z axes were made during this test. Inertial Measurement Unit Sensor implementation on the x, y, and z axes.

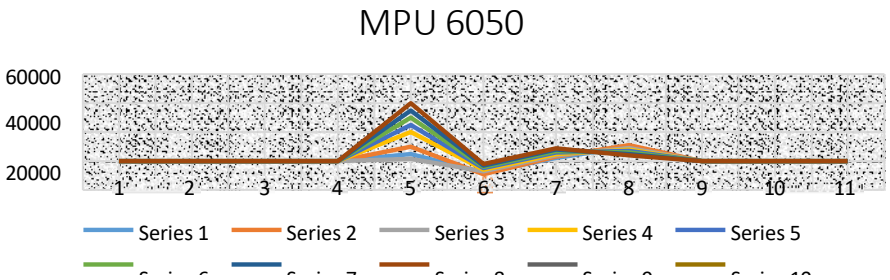


Figure 5. MPU 6050 sensor test.

The MPU-6050 tracks the robot's angular velocity and balance while transmitting data to the Arduino at a baud rate of 9600. It makes use of a flywheel for accuracy. Measurements of acceleration along the x, y, and z axes were made during this test. Inertial Measurement Unit Sensor implementation on the x, y, and z axes [17]. The accelerometer and gyroscope test results for the MPU6050 sensor combination are displayed in Figure 5.

Series 1 displays acceleration for X, Series 2 for Y, and Series 4 for the combination of acceleration and rotation for X, Y, and Z. The data's order is displayed on the X-axis, while the rotation rate (°/s) or acceleration (m/s2) is displayed on the Y-axis. The sensor's reaction to the abrupt change is shown by the dramatic peak at cycle five, which is followed by the situation returning to normal. The variations in the series, particularly at the fifth cycle, demonstrate how the MPU6050 sensor reacts to rotation and acceleration. Therefore, as Figure 5, illustrates, the MPU6050 sensor response is critical to the stability of autonomous vehicles, drones, and robotics.

Table 1. Testing the MPU6050 sensor on an autonomous robot.

No	Slope	Accelerometer Sensor Data			Gyroscope Sensor Data			Roll Angle	Pitch Angle
		Ax	Ay	Az	Gx	Gy	Gz		
1	5,8	5523	3460	15088	-515	219	140	84,77	18,79
2	7,2	2250	-755	32799	-1140	1877	18930	72,77	12,93
3	8,3	-300	288	32799	-2690	7740	6449	-21	27,36
4	9,9	1357	950	11149	-1488	-460	1	81,03	78,74
5	11	433	410	-8351	-830	88	13872	72,99	70,19
6	11,3	802	820	-5650	-671	391	12019	81,61	46,9
7	12,1	477	61	-2788	-788	-1098	8393	83,57	85,9

Table 1displays the differences in the measures of pitch (-12.93 to 85.9 degrees), roll (-21 to 84.77 degrees), and tilt (5.8–12.1 degrees). The Omni wheel robot exhibited sensor constancy and accuracy, which are critical for military surveillance.

RPM and tachometer testing on the motor wheel.

To monitor changes in the robot's location and determine whether the tachometer and rotary encoder's RPMs are appropriate, the rotation encoder detects the PWM value and error produced by each motor rotation [18].

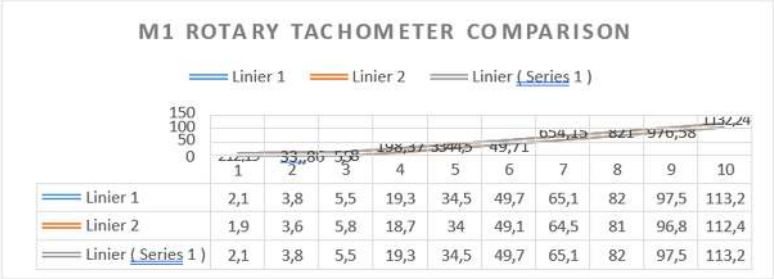


Figure 6. Comparison of rotary RPM and M1 tachometer.

Figure 6 shows the tachometer rotational speed versus time with error bars, as well as blue, orange, and black linear trend lines.

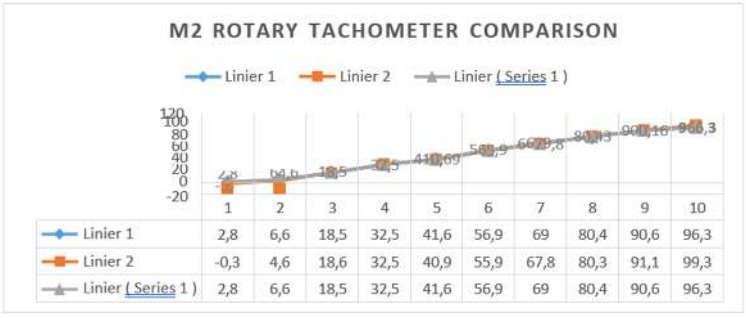


Figure 7. Comparison of rotary RPM and M2 tachometer.

The tachometer speeds for Series 1 and Series 2 are shown differently in Figure 7 using blue and orange, respectively. The black-colored linear trend indicates that the outcomes are comparable despite the little variations.

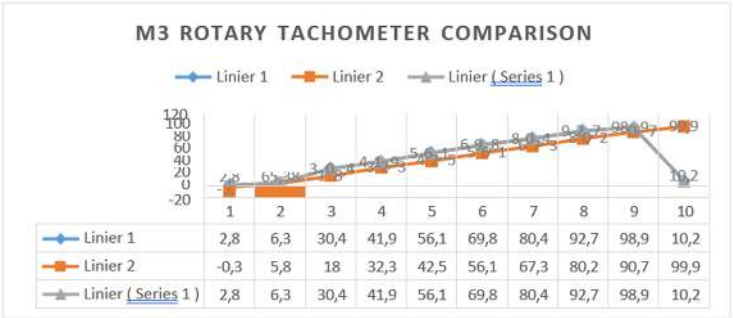


Figure 8. Comparison of rotary RPM and M3 tachometer.

The rotational speed differential between Series 1 (blue) and Series 2 (orange) is displayed in Figure 8. Series 1's linear trend is depicted by the black line, while uncertainty is indicated by the error bars. The outcomes align with a linear pattern.

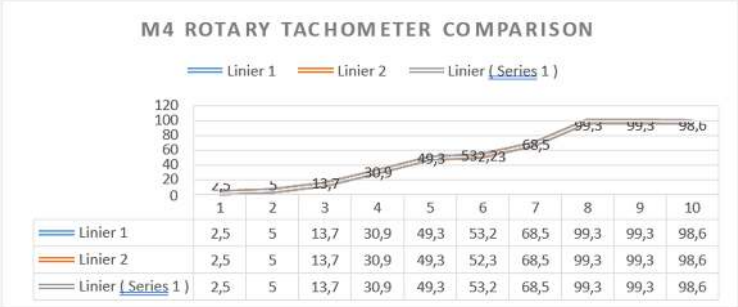


Figure 9. Comparison of RPM rotary encoder and tachometer M4.

Figure 9 shows a comparison of the rotational speeds of Series 1 (blue) and Series 2 (orange), with the black line indicating the linear trend of Series 1. The results of both tachometers show a consistent linear trend.

Table 2. Comparison of rotary encoder and tachometer RPM.

PWM	Rotary Encoder				Tachometer				Voltage
	M1	M2	M3	M4	M1	M2	M3	M4	
10	2,4	2,4	2,4	2,4	2,5	2,5	2,5	2,5	0,4777
15	6,5	6,6	6,5	5,6	6,3	6	6,2	5,9	0,9411
21	18	18	19	17	17	16,6	16	15	1,8823
46	33,3	32,5	30,1	30,9	30,4	29	30,4	30,5	3,7566
85	45,3	44,4	44,9	39,9	39,9	39,8	40,3	41,7	4,7044
103	55,7	55,3	50,1	52,4	56,6	56,9	55,1	53,3	5,6489
155	67,5	67,9	68	68,5	68,1	69,9	65,9	68,2	7,0533
183	82,7	78,2	78,4	79,3	79,1	80,4	80,3	84,2	8,4771
209	91,1	91,8	92,7	90,3	91,6	90	90	92,5	9,4123
230	100,2	98,3	98,3	98,6	97	97	97	98,5	12,1

Data analysis:

The PWM setting affects the readings from the tachometer and rotary encoder; low PWM falls between 2.4 and 2.5, and high PWM reaches 97 to 100. The tachometer readings at PWM 209 are consistent, with tachometer values ranging from 90.3 to 92.5, according to a comparison of M1, M2, M3, and M4. Furthermore, from 0.4777 at PWM 10 to 12.1 at PWM 230, the voltage rise is proportionate to the motor speed, showing that higher motor speeds demand more power.

Table 3. Error calculation from RPM comparison on Rotary encoder and Tachometer.

PWM	Rotary Encoder $\bar{x}$ M1-M4	Tachometer $\bar{x}$ M1-M4	Error
10	2,4	2,5	0,1%
15	6,3	6,1	0,2%
21	18	16,15	1,85%
46	31,7	30,075	1,63%
85	43,625	40,425	3,2%
103	53,375	55,475	2,1%
155	67,975	68,025	0,05%
183	79,65	81	1,35%
209	91,475	91,025	0,45%
230	98,85	97,375	1,475%
Rata-Rata Error			1,2%

Table 3 shows the TA module error of 1.2%. Good robot performance is also shown, with PWM 85 (3.2 RPM) and PWM 155 (0.05 RPM) having the largest difference.

PID control testing on the track

PID testing on motors uses sampling time and rotation encoder to calculate speed [19]. PID control testing compares accelerometer and gyroscope data and controls the speed of the omni wheel robot motor [20]. The robot's directional angle is controlled by PID control, which improves its ability to know the current position. Based on research [21]. The stability of the robot is optimized with a PID controller with Kp: 5000, Ki: 1000, and Kd: 500.

A. Square path

Table 4. Testing on a square track.

Position Coordinates				Error Tracking	
Target		Real			
X	Y	X	Y	X	Y
0	0	0	0	0	0
1.001	-429	9	-559	10	10
699	0	710	1	16	1
0	-711	24	407	25	4
0	0	1	1	0	1

Each row of data has a measurement error, as well as X and Y coordinates for the target and real-time positions. For example, the second row shows the target position (1.001, - 429) and the real-time position (9, -559), with an error of 10 for both X and Y.

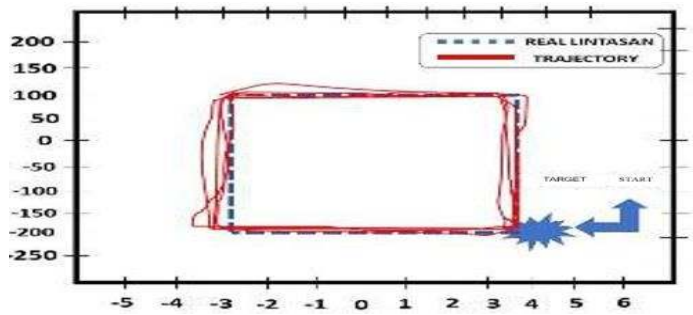


Figure 10. Square path.

Figure 10 shows the differences between the actual and desired trajectory of the omni wheel robot. Errors in the corners and straight segments indicate problems with the control system.

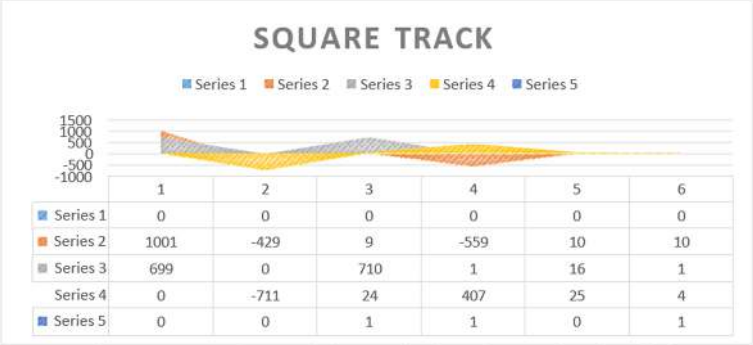


Figure 11. Tracking map on a rectangular track.

Explanation of Figure 11 graph:

a. Series 1: Displays data for a specific group or parameter.
b. Series 2: Displays data for other groups or parameters.
c. Series 3: series 4, and series 5: Different data sets with different categories or parameters.

Figure 11 shows the DC motor speed control and the robot's route shift, along with the formula for calculating the position error:

$$error = \frac{visual\ value - target\ value}{target\ value}\ 100\%\ (Percentage\ formula\ for\ errors\ or\ mistakes)$$

Table 5. Robot square trajectory position error (X, Y) based on visual observation.

Set Point	Error X (%)	Error Y (%)	Error Average (%)
1	0	0	0
2	8	31	19,5
3	1,6	1	1,3
4	24	42,75	33,75
5	1	1	1

There is no mistake in the X, Y, or average coordinates at Point One because the X and Y coordinates are exactly on target. Point Two has an average inaccuracy of 19.5%, with X coordinate error being 8% and Y coordinate error being 31%. At point three, the average inaccuracy is 1.3%, with the X coordinate error being 1.6% and the Y being 1%. At Point Four, each X and Y coordinate has inaccuracy of 1.6% and 1%, respectively.

B. Circle Path

Table 6. Testing on a circular track.

Position Coordinates				Error	
Target		Real		Tracking	
X	Y	X	Y	X	Y
0	0	0	0	0	0
0	-602	9	-600	9	1
1001	0	991	1	4	2
0	602	79	365	79	26
0	0	20	1	0	1

The table shows the target coordinates and errors. Example: (0, -602) compared to (9, -600), error X 9, Y 1.

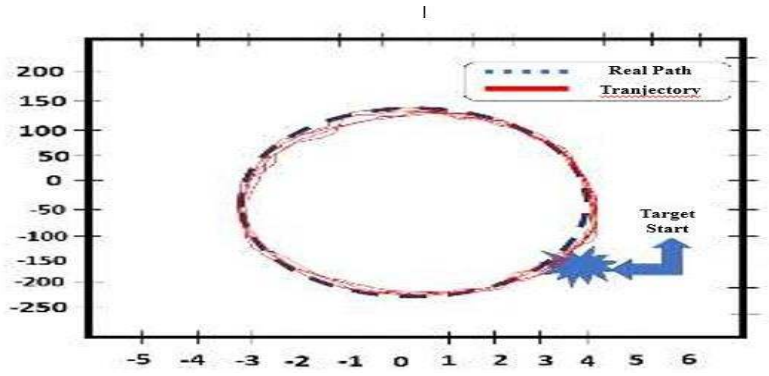


Figure 12. Circle path.

Figure 12 shows the difference between the actual and planned robot trajectories (red lines). The robot follows a circular path rather than a rectangular path, indicating that calibration is needed to improve accuracy.

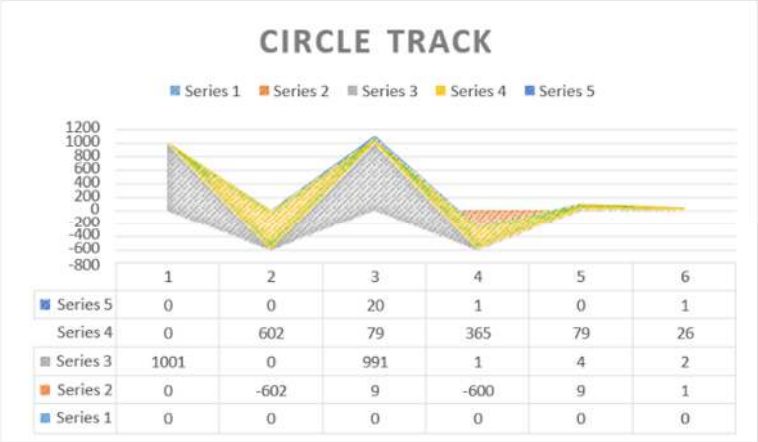


Figure 13. Tracking path map on a circular path.

Explanation of Figure 13:

This stacked area chart displays many data series with colored areas beneath the plot line. As Series 1 has no value at all, it is unchanged. The lowest values are seen in Series 2 (orange), with -602 at Point 2 and -600 at Point 4. The greatest values in Series 3 (gray) are found at Points 1 (1001) and 3 (991). Point 2 (602) is the apex of Series 4 (yellow), while Point 4 (365) is a notable value. There is minimal variation in Series 5 (dark blue), with a maximum value of 20 at point 4. The vertical Y-axis spans from 800 to 1200, while the X-axis displays data points or categories from 1 to 6. The computation of the error in the x and y coordinates on a circular path—that is, the path the robot takes that veers off course—is as follows.

Table 7. Robot circular trajectory position error (X, Y) based on visual observation.

SET POINT	ERROR X (%)	ERROR Y (%)	ERROR AVERAGE (%)
1	0	0	0
2	9	0,33	4,6
3	0,9	1	0,95
4	79	39,4	59,2
5	20	1	10,5
TOTAL ERROR AVERAGE			**15,05**

The percentage position errors for the five set points (X, Y, and average error percentages) are shown in Table 7. For example, set point 2 has an X error of 9% and a Y error of 0.33%.

C. Triangle Path

Table 8. Triangular track testing.

POSITION COORDINATES				ERROR	
TARGET		REAL		TRACKING	
X	Y	X	Y	X	Y
0	0	0	0	0	0
0	-391	9	-409	9	9
699	0	701	1	16	1
0	-392	21	407	22	4
0	0	1	1	0	1

Table 8 compares the actual and target coordinates with X and Y errors. At the second point, the target (0, -391) and the actual result (9, -409) show the distance between the actual and desired positions.

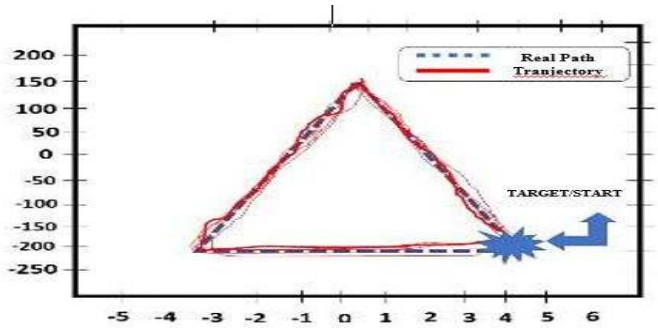
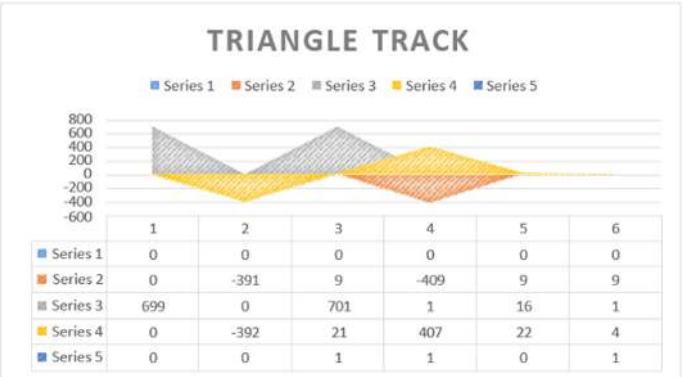


Figure 14. Triangle path.

Figure 14 shows the difference between the actual trajectory (dashed blue) and the desired one (red). There are errors in the straight and corner segments caused by mechanical or measurement constraints.



	1	2	3	4	5	6
Series 1	0	0	0	0	0	0
Series 2	0	-391	9	-409	9	9
Series 3	699	0	701	1	16	1
Series 4	0	-392	21	407	22	4
Series 5	0	0	1	1	0	1

Figure 15. Tracking path map on the triangle path.

Explanation of graph 15:

With distinct colors for each section, this stacked area chart displays several data series. Series 1 is not shown on the chart; it stays at 0. There is a noticeable decline in Series 2 (orange), with values of -409 at Point 4 and -391 at Point 3. Points 1 (699) and 3 (701), which are the highest values of Series 3 (gray), dominate the chart. Variation may be seen in Series 4 (yellow), with values climbing to 407 at Point 4 from -392 at Point 2. There is not much difference in Series 5 (dark blue), where the vertical Y-axis ranges from -600 to 800 and the horizontal X-axis displays data points or categories from 1 to 6.

Table 9. Robot triangle track position error (X, Y) based on visual observation.

SET POINT	ERROR X (%)	ERROR Y (%)	ERROR AVERAGE (%)
1	0	0	0
2	9	4,6	6,8
3	0,3	1	0,65
4	21	3,8	12,4
5	1	1	1
TOTAL ERROR AVERAGE			**4,17**

The omni wheel robot continued to operate well and reach its position, even though elements such as slippery terrain and uneven grouting caused its movement to deviate slightly from expected.

4. CONCLUSION

With its Omni wheel, this mobile robot can move and rotate in any direction on a level track, allowing it to conduct reconnaissance along the predetermined course. The test revealed an 89.31% successful robot movement rate with a reading error of only roughly 1 degree, demonstrating the high quality of

the gyroscope sensor that was employed. The rotation encoder and tachometer's average error value of 1.2% is acceptable. This still falls within the acceptable range. The triangular path, with an average error rate of 11%, 15%, and 4.17%, has the maximum motor rotation speed stability, according to the experimental data. Overall, this study demonstrates that the controlled mobile omni-wheel robot can perform precise and timely reconnaissance, enhancing military operations' safety and effectiveness.

REFERENCES

[1] Ahmad Gunawan and Rini Kartika Hudiono, "Industrial Revolution 4.0's Information Technology's Impact on the Growth of MSMEs in the Manufacturing Industries Sector," *Int. Trans. Educ. Technol.*, vol. 1, no. 2, pp. 157–164, 2023, doi: 10.34306/itee.v1i2.332.

[2] A. Kurniawan and K. Wardani, "Dagu RS003B75 Chassis Mobile Robot Platform sebagai Purwarupa MINION Mobile Mines & Intelligent Remote Detonator Robot," *TELKA - Telekomun. Elektron. Komputasi dan Kontrol*, vol. 7, no. 2, pp. 120–133, 2021, doi: 10.15575/telka.v7n2.120-133.

[3] M. N. S. Usha, S. Priyadharshini, K. R. Shree, P. S. Devi, and G. Sangeetha, "Military Reconnaissance Robot," *Int. J. Adv. Eng. Res. Sci.*, vol. 4, no. 2, pp. 49–56, 2017, doi: 10.22161/ijaers.4.2.10.

[4] D. P. Wiliyanto, S. Supriyadi, A. Mukhtar, and J. T. Mesin, "Rancang Bangun Waypoint Gps Navigation Mobile Robot Dengan Menggunakan Omni-Directional Wheel Dan Sensor Ultrasonik Sebagai Alternatif Robot Pengangkut Barang," *Sci. Eng. Natl. Semin.*, vol. 5, no. Sens 5, pp. 593–600, 2020.

[5] N. Kliot and Y. Mansfeld, "Case studies of conflict and territorial organization in divided cities," *Prog. Plann.*, vol. 52, no. 3, pp. 167–225, 1999, doi: 10.1016/S0305-9006(99)00010-0.

[6] L. Atzori, A. Iera, and G. Morabito, "Understanding the Internet of Things: definition, potentials, and societal role of a fast evolving paradigm," *Ad Hoc Networks*, vol. 56, no. October 2017, pp. 122–140, 2017, doi: 10.1016/j.adhoc.2016.12.004.

[7] R. Joshi, J. Patil, S. S. Luthra, S. Gaud, and P. Giridhar, "Three Wheeled Omnidirectional Soccer Robot Modelling and Wireless controlling using Bluetooth enabled PlayStation Controller," *REST J. Emerg. trends Model. Manuf.*, vol. 5, no. 2, 2019, doi: 10.46632/jemm/5/2/2.

[8] Irvan Arianto Randa Lembang, Romie Oktavianus Bura, R Djoko Andreas Navalino, and Lutfi Adin Affandi, "Indonesia Military Research and Development in Dealing with the Sixth Generation Warfare: The Use of Artificial Intelligence in War Operations," *J. Multidisiplin Madani*, vol. 3, no. 5, pp. 1156–1163, 2023, doi: 10.55927/mudima.v3i5.3152.

[9] N. H. Ariffin, N. Arsad, and B. Bais, "Low cost MEMS gyroscope and accelerometer implementation without Kalman Filter for angle estimation," *2016 Int. Conf. Adv. Electr. Electron. Syst. Eng. ICAEES 2016*, no. November, pp. 77–82, 2016, doi: 10.1109/ICAEES.2016.7888013.

[10] P. Franček, K. Jambrošić, M. Horvat, and V. Planinec, "The Performance of Inertial Measurement Unit Sensors on Various Hardware Platforms for Binaural Head-Tracking Applications," *Sensors*, vol. 23, no. 2, 2023, doi: 10.3390/s23020872.

[11] S. S. Mali, P. P. Hajare, R. D. Mhamane, and P. R. G. Ghodke, "Soldier Tracking and Health Monitoring System," *J. Electron. Netw. Appl. Math.*, no. 35, pp. 10–17, 2023, doi: 10.55529/jecnam.35.10.17.

[12] M. Wahdan, "Motion planning for autonomous mobile robots," *AEJ - Alexandria Eng. J.*, vol. 44, no. 1, pp. 51–57, 2005, doi: https://doi.org/10.3390/s21237898.

[13] Y. R. putra Marjanto, "Pengaturan Navigasi Gerak Robot 3 Wheels Omni-Directional Menggunakan Metode Kontrol Pid," *J. Tek. Elektro dan Komput. TRIAC*, vol. 7, no. 2, pp. 46–56, 2020, doi: 10.21107/triac.v7i2.8146.

[14] K. Nurariyanto, S. T. Elektro, F. Teknik, and U. N. Surabaya, "Sistem Positioning Pada Four-Wheeled Omnidirectional Mobile Robot Menggunakan Metode Gyrodometry Berbasis PI-Fuzzy Controller Muhamad Syariffuddien Zuhrie , Bambang Suprianto , Puput Wanarti

Rusimamto," *J. Tek. Elektro*, doi: https://doi.org/10.26740/jte.v11n1.p77-87.

[15] S. S. Harahap, W. S. Simamora, and R. R. Hadistio, "Implementation of Fuzzy Logic in Detecting Air Temperature Based on Microcontroller," *Brill. Res. Artif. Intell.*, vol. 3, no. 2, pp. 155–161, 2023, doi: 10.47709/brilliance.v3i2.3023.

[16] Azhari, T. I. Nasution, and P. F. A. Azis, "MPU-6050 Wheeled Robot Controlled Hand Gesture Using L298N Driver Based on Arduino," *J. Phys. Conf. Ser.*, vol. 2421, no. 1, pp. 0–10, 2023, doi: 10.1088/1742-6596/2421/1/012022.

[17] K. Parsa, T. A. Lasky, and B. Ravani, "Design and Implementation of a Mechatronic, All-Accelerometer Inertial Measurement Unit," *IEEE/ASME Trans. Mechatronics*, vol. 12, no. 6, pp. 640–650, 2007, doi: 10.1109/TMECH.2007.910080.

[18] H. J. Djahi, S. Y. Doo, and A. M. P. Nuga, "Rancang Bangun Robot Mobil Dengan Sistem Navigasi Berbasis Odometry Menggunakan Rotary Encoder," *J. Media Elektro*, vol. VIII, no. 1, pp. 59–65, 2019, doi: 10.35508/jme.v8i1.1082.

[19] R. Rikwan and A. Ma'arif, "DC Motor Rotary Speed Control with Arduino UNO Based PID Control," *Control Syst. Optim. Lett.*, vol. 1, no. 1, pp. 17–31, 2023, doi: 10.59247/csol.v1i1.6.

[20] V. N. Kadam, P. S. Jadhav, A. D. Ghatpande, G. H. S. Seshadri, D. K. Shah, and P. J. Kadam, "Closed Loop Control of Unstable Omni Directional Assisting System," *2018 4th Int. Conf. Converg. Technol. I2CT 2018*, no. October, pp. 1–5, 2018, doi: 10.1109/I2CT42659.2018.9058054.

[21] H. Purnata, S. Ramadan, M. A. Hidayat, and I. Maulana, "PID Control Schematic Design for Omni-directional wheel mobile robot Cilacap State of Polytechnic," *J. Jartel J. Jar. Telekomun.*, vol. 12, no. 2, pp. 89–94, 2022, doi: 10.33795/jartel.v12i2.322.