EMITOR: Jurnal Teknik Elektro

Implementation of SLAM Gmapping and Extended Kalman Filter for Security Robot Navigation System

Riza Agung Firmansyah*, Yuliyanto Agung Prabowo, Titiek Suheta, Afri Nanda Dwi Utomo
Jurusan Teknik Elektro (FTETI) – Institut Teknologi Adhi Tama Surabaya
Surabaya, Indonesia
*rizaagungf@itats.ac.id

Abstract – *Robot Security is a robot that is responsible for security as well as patrolling. When patrol automatically, the robot requires a navigation system. The robot also needs a mapping system that is used to make a map of the environment and as information on its location according to the map. The sensors used are wheel odometry and LiDAR. The wheel odometry system often slips which causes errors in reading the actual position of the robot. To fix this problem, a sensor fusion between the Inertial Measurement Unit (IMU) and wheel odometry is used. To combine these sensors, namely using the Extend Kalman Filter (EKF) which runs on the Robot Operating System (ROS) operating system. Mapping and navigation system testing, carried out using IMU sensors and without IMU, towards the 5 target points that have been made. In the test without IMU, the error of the robot reaching the target was (x = 45.86%, y = 54.595%, and = 56.63%). After adding the IMU sensor, the robot error has decreased to (x = 2.02%, y = 1.796%, and = 0.22%). In conclusion, the data combined from the IMU sensor and wheel odometry could minimize the existing slip errors.*

Keywords – *Robot Security; SLAM Gmapping; Extended Kalman Filter; ROS; Inertial measurement unit.*

I. Introduction

WITHIN the unprecedented circumstances of the COVID-19 pandemic, the role of security robots has become increasingly crucial. In addition to performing general environmental surveillance functions, robots can also be used for early detection of individual health conditions by integrating temperature measuring devices on them [1, 2]. Therefore, robots can identify potential individuals with body temperatures exceeding normal limits, contributing to efforts to prevent the spread of the virus.

To perform surveillance or patrol tasks optimally, reliable navigation capabilities are essential for security robots. This navigation system enables the robot to move autonomously around the environment that needs to be monitored. The navigation system involves route recognition, obstacle avoidance calculations, and adaptation to changing environmental conditions [3].

Determining the robot's actual position is an integral element in the navigation system. Lack of knowledge about the accurate position can hinder the robot's operational efficiency. Therefore, the SLAM (Simultaneous Localization and Mapping) method approach becomes important. Through SLAM, robots can simultaneously map their surroundings and determine their actual position on the created map.

Gmapping emerges as one of the SLAM implementations that can be applied to security robots for mapping and positioning [4, 5]. This approach relies on LiDAR (Light Detection and Ranging) sensors capable of measuring distances accurately and building three-dimensional maps of the surrounding area. Additionally, wheel odometry data is used to calculate the robot's movement.

Although wheel odometry data provides an overview of the robot's movement, wheel slip is a challenge. Wheel slip, which can occur on slippery or uneven surfaces, can disrupt the accuracy of movement estimation. To address this issue, a more sophisticated solution in position determination is needed. One solution is to utilize inertia data from IMU (Inertial Measurement Unit) sensors. IMU sensors can detect changes in the robot's speed and orientation accurately, even in wheel slip situations. Integrating data from IMU sensors with wheel odometry data can provide a more precise and stable position estimate [6].

The manuscript was received on November 3, 2023, revised on June 21, 2024, and published online on July 26, 2024. `Emitor` is a Journal of Electrical Engineering at Universitas Muhammadiyah Surakarta with ISSN (Print) 1411 – 8890 and ISSN (Online) 2541 – 4518, holding Sinta 3 accreditation. It is accessible at `https://journals2.ums.ac.id/index.php/emitor/index`.

Integrating data from IMU sensors and Wheel Odometry is performed using the Extended Kalman Filter (EKF) algorithm. EKF is a mathematical algorithm used to combine data from two different sources, in this case, IMU sensors and Wheel Odometry [7, 8]. EKF estimates the uncertainty level of each data and produces an optimal estimate of the robot's position and orientation.

The navigation system can be implemented within the Robot Operating System (ROS). ROS has advantages in hardware management, sensor data processing, and communication systems. By using ROS as its framework, the security robot navigation system can be implemented more easily and structurally.

A security robot is a robot used to perform environmental surveillance to maintain security in a place [9, 10]. This robot mimics the tasks of a security guard when performing security duties. To support its performance, the robot is equipped with cameras and a navigation system. The navigation system can be implemented on the condition that the robot knows its actual position and has a map of the monitored area.

The navigation system in the robot is a system that enables the robot to move autonomously, identify its location, and plan the movement path to a specific destination [11]. The navigation system is essential to make the robot autonomous and efficient in carrying out its tasks. One part of the navigation system is environmental mapping. The robot needs to understand the structure and characteristics of its environment, such as walls, doors, or other obstacles. Mapping can be done statically, where the environmental map is made beforehand, or dynamically, where the robot maps the environment while moving. Besides mapping, localization is also a main part of the navigation system. This localization aims to know the robot's actual position. This system can be achieved using techniques like odometry or the global positioning system (GPS). However, in indoor conditions, GPS does not function well [12].

Research on robot navigation systems in indoor environments conducted by several researchers generally focuses on the development of mapping and navigation systems using SLAM (Simultaneous Localization and Mapping) technology. These studies generally aim to understand how robots can move automatically, create environmental maps while knowing their actual position. Some SLAM algorithms such as Gmapping [13, 14], Hector SLAM [15, 16], and Karto SLAM [17] are tested and evaluated to understand the efficiency and accuracy of mapping. These studies are conducted to select optimal parameters to achieve the best mapping and localization results. Among these techniques, Gmapping is often used.

Gmapping is a SLAM algorithm that creates a map of the robot's surroundings and updates the map with sensor information while recording the robot's movement path. Gmapping uses a loop closure algorithm and relies on odometry data to build the environmental map. The technique used in Gmapping is loop closure which can update the map when the robot returns to its original position. Gmapping is known for its higher efficiency and lower computational power requirement compared to other techniques, making it suitable for security situations [18]. The Gmapping method uses information from lidar sensors and robot photometric. The localization process in Gmapping uses Monte Carlo Localization (MCL) also known as particle filter [19]. When mapping with Gmapping, the occupancy grid map method is used to create a two-dimensional (2D) map. The occupancy grid map represents the grayscale level on a scale of 0 to 255.

The use of sensors such as 2D lasers and Inertial Measurement Unit (IMU) sensors can help in obtaining accurate information about the surrounding environment [20]. However, challenges arise when the robot has to move outdoors, facing disturbances such as slip and uneven terrain. Thus, the use of motors and some sensors need to be optimized to reduce reading errors. The sensors used to support the performance of LiDAR are generally wheel encoders. Wheel encoders are sensors commonly used in odometry systems to estimate position changes in robots. Additionally, Inertial Measurement Unit (IMU) sensors are often used to determine the robot's heading. IMU has advantages in providing accurate and real-time information, especially when combined with the Extended Kalman Filter algorithm.

In this article, the research aims to test the reliability of the developed navigation system and contribute to the application of EKF in security robots. It is expected that the results of this research can significantly contribute to the development of more efficient and reliable security robots in performing surveillance and monitoring tasks.

II. Research Methods

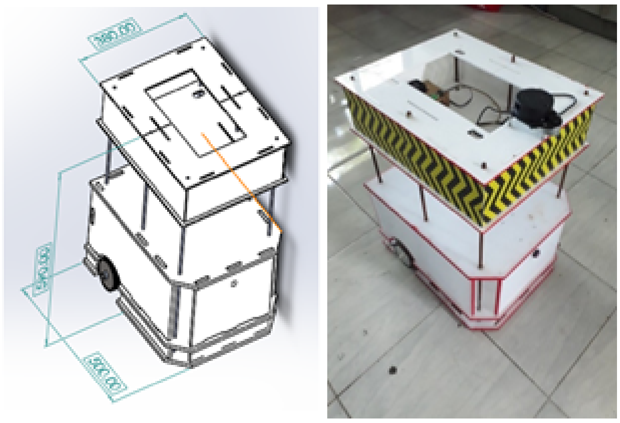
This chapter discusses the planning and creation steps of the system, which include the planning of hardware design, robot design, and software design.

i. Mechanical System Design

The Security Robot is built using a 1 cm thick acrylic frame, assembled in layers. The Security Robot has dimensions of 500 cm in length, 380 cm in width, and

570 cm in height. The robot design is illustrated using 3D drawing software, and the actual appearance of the robot can be seen in Figures 1a and 1b.

To support the odometry system using a rotary encoder, the wheel layout in the robot's geometry needs to be considered. The robot is constructed using a differential drive mobile robot mechanism with a wheelbase (d) of 38 cm. The robot's base frame is set between the two driving wheels. The combination of the left wheel speed (vL) and the right wheel speed (vR) will result in changes in the robot's speed (v) and heading direction (θ). The kinematics of the mobile robot are shown in Figure 2.



(a) **(b)**

Figure 1: Security Robot (a) Design (b) Design realization

To support the odometry system using a rotary encoder, the wheel layout in the robot's geometry needs to be considered. The robot is built using a differential drive mobile robot mechanism with a wheelbase (d) of 38 cm. The base frame of the robot is set between the two driving wheels. The combination of the left wheel speed (vL) and the right wheel speed (vR) will result in changes in the robot's speed (v) and heading direction (θ). The kinematics of the mobile robot are shown in Figure 2.

ii. Hardware Design

The hardware design includes the creation of the hardware connection design. This includes connecting the IMU sensor to the Jetson Nano, connecting the DC motor to the Arduino microcontroller, and connecting the LiDAR to the Jetson Nano. Figure 3 explains the design of the system workflow to be created. Initially, the power source comes from a Lithium-Polymer (Li-Po) battery. The Li-Po battery used is a 3-cell battery with a voltage of 12 volts and a capacity of 2200 mAh. The battery voltage is used to supply several components, including the Arduino Mega, Jetson Nano, DC motor driver, encoder, Inertial Measurement Unit (IMU), and LiDAR A1.

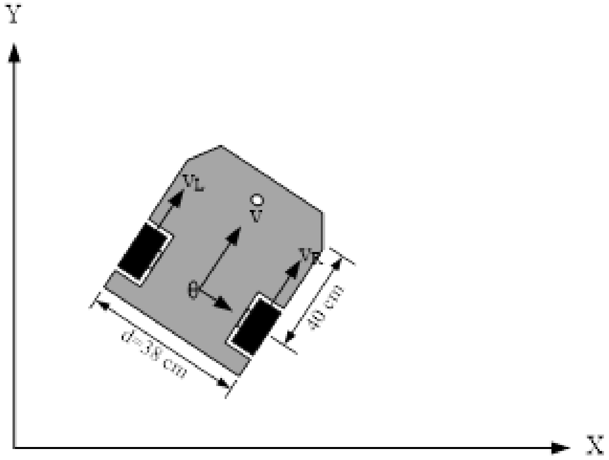


Figure 2: Kinematics of the Mobile Robot

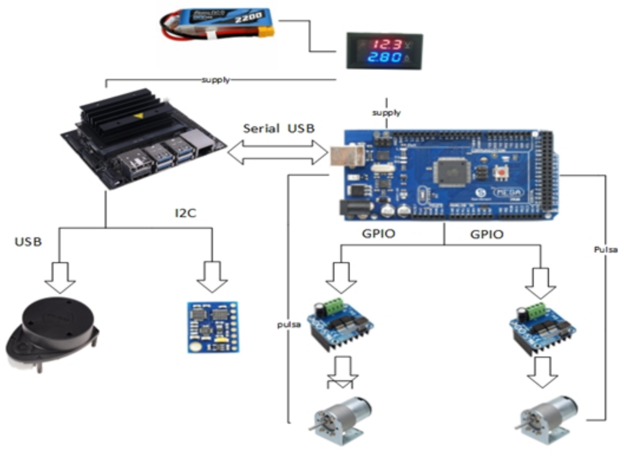


Figure 3: Hardware Design of the Security Robot

Next, a digital voltmeter display is installed to display the voltage used, which is 12 V. This display uses a seven-segment display to show the value and has 3 output cables: 2 cables for positive voltage source and 1 cable for data. The 2 source cables are parallel to the battery, and the data cable is parallel to the positive battery line.

The Arduino Mega is used to control and process data from several components, including the DC motor driver, DC motor, and encoder, and serves as the RPM data reader for the motor. The Arduino Mega is a microcontroller board based on the Atmega2560. The operating voltage for this microcontroller is 5V, but it can be supplied with a maximum voltage of 12V due to the external power supply circuit. The encoder data processed by the Arduino is then sent to the Jetson Nano via serial communication.

The wheel encoder sensor is used to measure the wheel speed. The encoder operates at 5V and is connected to the PWM pin of the Arduino Mega. The microcontroller processes this data to obtain the RPM value, which is then sent to the Jetson Nano via serial communication. The Jetson Nano serves as the main processor for mapping and navigation. It is a System On Module (SoM) and developer kit from Nvidia Jetson, featuring a 128-core Maxwell GPU, quad-core ARM A57 64-bit CPU, and 4GB LPDDR4 memory. It has 4 USB 3.0 ports, 1 Ethernet port, 1 display port, and 1 HDMI port. The operating voltage for the Jetson Nano is 5V, but it can be supplied with a maximum voltage of 12V due to the external power supply circuit. The Jetson Nano processes all inputs received from the IMU sensor, LiDAR A1, and RPM information.

The Inertial Measurement Unit (IMU) sensor is used to provide information about the robot's direction, including the X, Y, Z axes, and motion information such as rolling, pitching, and yawing. This directional data is processed in the ROS system as additional information for controlling the robot. The LiDAR A1 sensor is used for scanning the room. It can rotate 360 degrees and emits a laser to the surroundings, with a receiver on the LiDAR to receive the reflected laser. The reflected laser is calculated to provide distance parameters for the robot's system.

iii. EKF Design

The Extended Kalman Filter (EKF) is an extension of the Kalman Filter method used to estimate values. The Kalman Filter was initially developed to address problems in discrete control processes that are stochastic and linear. However, in its implementation in sensor fusion, measurements from these processes are often non-linear. The EKF provides updates to the Kalman Filter method to handle non-linear measurements.

Non-linear value estimation is processed using the EKF. In the estimation stage, the robot's position is predicted based on data obtained from the IMU sensor and wheel odometry. This stage consists of two steps: the prediction step and the update step. Estimation is performed using two models: the state space model and the observation model.

The state space model is a mathematical equation that describes the robot's movement from one time step to the next. Since the robot moves on a flat plane, the z, roll, and pitch components are excluded. The state or system state is defined by the pose equation (position and orientation) as in Equation (1).

$$P = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{1}$$

where x and y represent the position on the flat plane, and θ represents the yawing state in the global coordinate system. When the robot moves, it creates a new position and orientation. The state vector is continuously updated to determine the actual position. The new position is estimated using Equation (2).

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = A_{t-1} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix} + B_{t-1} \begin{bmatrix} v_{t-1} \\ \omega_{t-1} \end{bmatrix} \tag{2}$$

A represents how the system state $P[x, y, \theta]$ changes from $t-1$ to t when no control command is executed. B represents how the system state P changes from t-1 to t due to the control command, where v_{t-1} is the robot's linear velocity in the robot's reference frame, and ω_{t-1} is the angular velocity in the robot's reference frame. The observation model is a mathematical equation that calculates measurements from sensors to estimate the actual state, including a noise vector at the end, resulting in Equation (3).

$$\begin{bmatrix} y_1^t \\ y_2^t \\ \vdots \\ y_n^t \end{bmatrix} = H_{nx3}^t \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} + \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_n \end{bmatrix} \tag{3}$$

where t is the current time, n is the number of observations or measurements at time t. H is the measurement matrix with n rows and 3 columns, with one column for each state variable. The value of ω is the sensor noise during the measurement. In mobile robots, Equation (3) is expanded to Equation (4).

$$\begin{aligned} \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix} \\ &+ \begin{bmatrix} \cos\theta_{t-1} \cdot dt & 0 \\ \sin\theta_{t-1} \cdot dt & 0 \\ 0 & dt \end{bmatrix} \begin{bmatrix} v_{t-1} \\ \omega_{t-1} \end{bmatrix} + \begin{bmatrix} \omega_{t-1} \\ \omega_{t-1} \\ \omega_{t-1} \end{bmatrix} \end{aligned} \tag{4}$$

iv. ROS Framework Design

In the software design for the Security Robot, several accesses are needed to use and integrate the sensors. The operating system used is Ubuntu 18.04, and the ROS version used is ROS Melodic. To monitor the working ROS Framework, the RQT application provided by ROS can be used. RQT is a framework from ROS that implements various working nodes.

In the design created, the robot uses ROS node configurations for several tasks. The node configurations include running manual mode, mapping mode, and automatic mode. Manual mode is used to operate the robot with remote operator commands. Mapping mode is used to create and save maps obtained from LiDAR sensor readings and wheel odometry. In this mode, the robot is operated manually by the operator using a remote at a lower speed. Automatic mode is used to operate the robot's navigation automatically based on the operator's commands. The command given by the operator is the final pose or target within the map's reach.

The operator's computer (as a remote) publishes a node called /teleop. The /teleop node publishes a topic named /cmd_vel. This topic contains message information of type .msgs, which is used to send linear and angular movement direction commands to the robot. The message on the /cmd_vel topic is subscribed by the /serial_node. The message subscribed by the /serial_node contains the value information used to drive the wheel rotation direction. The /serial_node publishes topics named /right_ticks and /left_ticks, which contain the number of pulses from the rotary encoder. The /right_ticks topic contains the right wheel pulse count, while the /left_ticks topic contains the left wheel pulse count.

The /ekf_odom_pub node subscribes to three topics named /initial_2d, /right_ticks, and /left_ticks. The /initial_2d node is obtained when setting the robot's initial direction in RViz. The information from these three topics is processed by /ekf_odom_pub to determine the robot's direction. The robot's direction value after being processed is published with a topic named /odom_data_quat.

The /imu/imu_node produces information from the IMU sensor, containing coordinate direction and rotation values. These values are published by /imu/imu_node with the topic /imu_data. The /robot_pose_ekf node subscribes to two topics: /odom_data_quat and /imu_data. The /robot_pose_ekf node processes the parameters obtained from these two topics to determine the robot's actual direction. The node configuration for manual mode is shown in Figure 4.

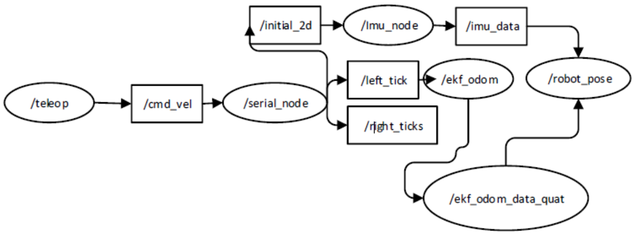


Figure 4: Node Configuration for Manual Mode

The room mapping process is performed using the /rplidarNode, while the robot is controlled to navigate the room using the /teleop_twist_keyboard node. The /rplidarNode is obtained from the LiDAR A1 sensor, which processes the laser scan data. This data is published with the /scan topic, containing parameters used to create a map. The /SLAM_gmapping node subscribes to the /scan topic, containing parameters to compose the gmapping map. Once the map is created, the /SLAM_gmapping node publishes a topic named /map. The /map topic is used to store the data created by gmapping. The node configuration during mapping mode is shown in Figure 5.

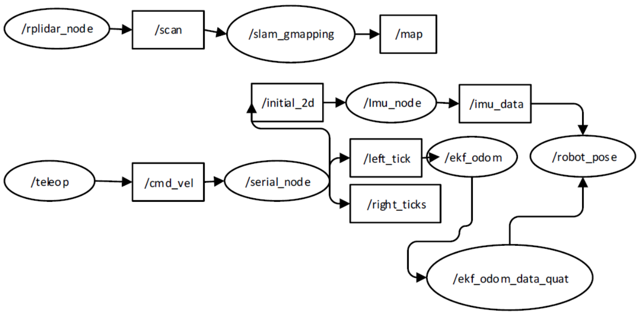


Figure 5: Node Configuration for Mapping Mode

The /map_server node is used to access the previously created map. The node containing the map information is published with the topic named /map . This topic is subscribed by the /move_base node, which is used as a parameter for the robot's automatic movement. The /rplidarNode is obtained from the LiDAR A1 sensor, which processes the laser scan data. This data is published with the /scan topic, containing parameters to be subscribed to the /amcl node. The /initialpose topic contains the global path formation process when initially setting the robot's direction in RViz. These parameters are subscribed by the /amcl node.

The /amcl node calculates to form the local path that the robot will traverse during navigation. The /amcl node always performs localization using laser data according to the environmental conditions, estimated by the IMU and encoder data. Thus, if the robot encounters new obstacles, it can recalculate to obtain a new local path to reach the goal. As a visualization of obstacles displayed in RViz, the /amcl node publishes a topic named /particlecloud. The visualization is displayed as point clouds forming obstacles according to the actual conditions. The node configuration for automatic mode is shown in Figure 6.

III. Results and Discussion

The automatic testing of the robot was conducted in the robotics lab of ITATS, as shown in Figure 7, with a total of 60 data collection trials. This testing aimed to determine the accuracy between the current goal (target point) on Rviz and the robot's pose (current state). The

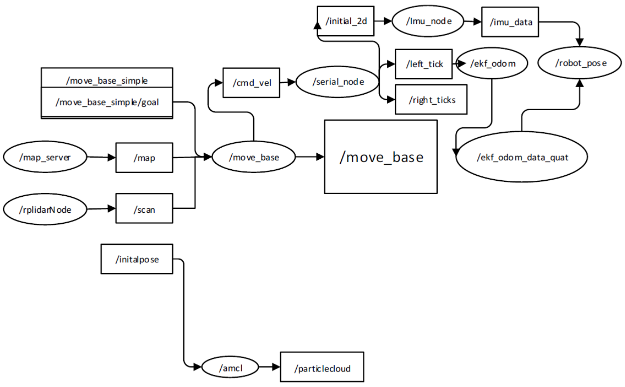
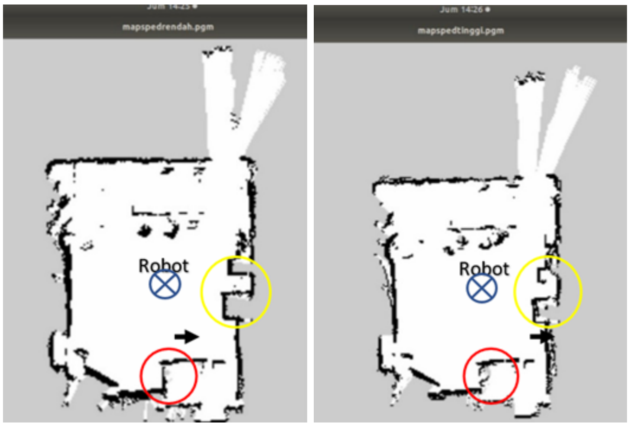


Figure 6: Node Configuration for Automatic Mode

Figure 7: Robot Testing Room



(a) (b)

Figure 8: Map generation result (a) with low speed (b) with high speed

testing was conducted without using the IMU sensor and with the IMU sensor.

i. Creating an Occupancy Grid Map

The map creation test was performed to determine if the robot could effectively map the room. The testing was conducted twice, using high speed and low speed. Figure 4.13 shows the condition of the ITATS robotics lab room where the map creation experiment was conducted.

In Figure 8(left), the map creation at maximum speed is shown, with a linear speed of 0.5 m/s and an angular speed of 1 rad/s. The resulting map shows some areas that do not form accurately, with some parts exceeding the actual room shape due to the unstable (non-constant) speed of the robot. In Figure 8(right), the map creation at low speed is shown, with a linear speed of 0.10 m/s and an angular speed of 0.11 rad/s. The resulting map closely resembles the actual room conditions, with the black lines formed accurately representing the room's layout. From these two map creation tests, it is evident that map creation is better performed at low speed. Figures 8 show that the parts of the room that should be straight are not detected at high speed, as indicated by the yellow and red circles.

ii. Automatic Navigation Testing Without EKF

Testing without EKF is a method where the robot operates without sensor fusion between the Inertial Measurement Unit (IMU) and wheel odometry. The IMU is typically used to measure and track the robot's orientation, acceleration, and angular velocity. Without the IMU, the robot relies solely on wheel odometry data to determine its position and orientation. This testing involves providing the robot with a target position to reach. In Figure 9, the target is shown in blue. After the robot moves towards the target, it does not always precisely reach it, resulting in a discrepancy between the actual position (red) and the target.

The testing results indicate an average error in the comparison between the target position (current goal) and the initial pose, with X=45.863%, Y=54.595%, and Theta=56.63%. This means the robot experiences significant errors in determining its position and orientation without using the IMU sensor. The primary cause of this error is wheel slip due to the uneven path of the robot. Wheel slip occurs when the robot's wheels do not have perfect traction with the surface. If the robot moves on uneven or slippery surfaces, its wheels may slip or experience unexpected shifts, leading to inaccurate odometry information.

When the robot relies solely on wheel odometry data for movement, the error tends to accumulate over time. This is because each movement step or wheel rotation contains a slight error, which accumulates over time. As a result, the longer or farther the robot moves, the greater the error becomes.

To address this issue, using an IMU sensor can be an appropriate solution. With the IMU, the robot

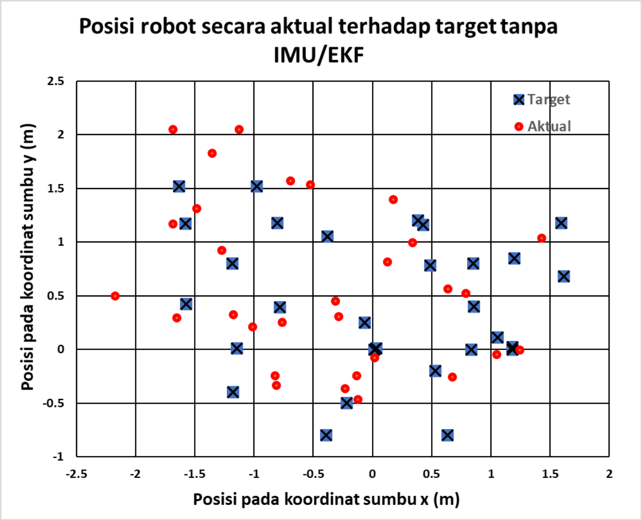


Figure 9: Target Position Achievement Without EKF

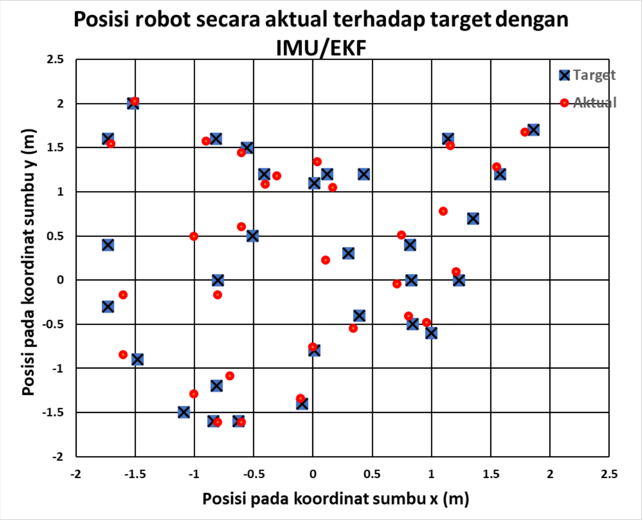


Figure 10: Target Position Achievement With EKF

can obtain additional data on orientation and angular acceleration, allowing it to correct odometry errors and improve movement and positioning accuracy. This enables the robot to move more efficiently and accurately even on uneven surfaces.

iii. Automatic Navigation Testing With EKF

To improve the robot's movement accuracy, the next test incorporated an IMU sensor. The IMU sensor was combined with wheel odometry using EKF. The results of this test can be seen in Figure 10. In Figure 11, which shows the results of testing with the extended Kalman filter on the robot, it is evident that the robot's movement when reaching the target (current goal) and its initial position (initial pose) has been thoroughly tested. The results indicate that the average movement error of the robot, measured in percentages along three different axes (X, Y, and Theta), is relatively small, with values of 2.02% for the X-axis, 1.796% for the Y-axis, and only 0.22% for the Theta axis. Therefore, it can be concluded that the extended Kalman filter significantly reduces the robot's movement errors, resulting in better accuracy in reaching the specified target.

The success of the extended Kalman filter in improving the robot's movement accuracy can also be attributed to the method used for robot movement. In this case, the robot moves by comparing data from two sources: the IMU sensor and wheel odometry. By leveraging data from both sensors, the robot can compensate for errors that may occur from each sensor, resulting in more precise and accurate movements in reaching the desired target.

Additionally, the results of the standard deviation analysis are noteworthy. The analysis shows that the Theta axis has the smallest standard deviation of 0.08463, indicating that the measurement data for the Theta axis has lower variation and is closer to the average error value. As a result, the robot's movement along the Theta axis is more consistent and reliable in reaching its final target.

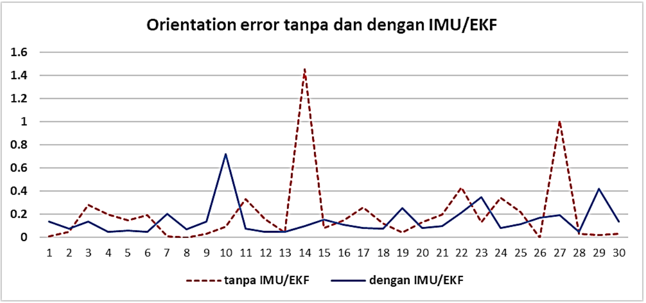


Figure 11: Comparison of Orientation Error With and Without EKF

Besides testing target achievement, the 2D trajectories generated by both methods were also compared. In Figure 12, it can be seen that the 2D trajectory resulting from the fusion of the encoder and IMU has a deviation close to the reference, while the wheel encoder alone results in a greater deviation.

Overall, the results of this test provide a positive outlook on the robot's performance in reaching targets with the help of the extended Kalman filter. The low movement errors along the three axes, especially the Theta axis, indicate that the robot can move more precisely and accurately. These results offer hope that using the extended Kalman filter technology and integrating data from the IMU sensor and wheel odometry can enhance the robot's overall performance in various tasks and applications involving movement and navigation.

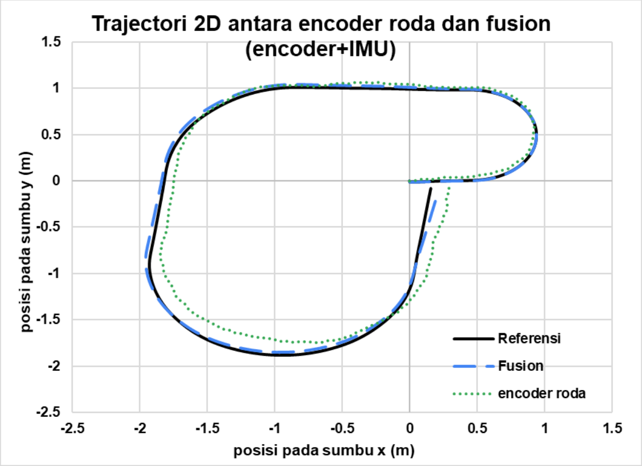


Figure 12: 2D Trajectory Comparison Between Wheel Encoder and Encoder+IMU Fusion

IV. Conclusion

The automatic movement test results of the robot show better performance when combining IMU sensor data with wheel odometry data compared to relying solely on wheel odometry data. This experiment was conducted with 60 different trials. Testing with only wheel odometry data showed significant errors when the robot reached the target, with errors of (x=45.86%, y=54.595%, θ=56.63%), indicating that the robot was far from the desired target position. However, when testing was conducted by combining IMU sensor data and wheel odometry data, the results showed much lower errors, with (x=2.02%, y=1.796%, θ=0.22%). This indicates that the robot moves closer to the target position with higher accuracy. From this experiment, it can be concluded that combining IMU sensor data with wheel odometry data synergistically reduces errors when the robot reaches the target point. Using the IMU sensor provides additional information about orientation and angular acceleration, helping to correct inaccuracies in wheel odometry data and improving the robot's movement accuracy.

Acknowledgment

The author expresses heartfelt gratitude to the Directorate of Research, Technology, and Community Service (DRTPM) of the Ministry of Education and Culture of the Republic of Indonesia for the 2023 PTUPT (Penelitian Terapan Unggulan Perguruan Tinggi) grant scheme with master contract number 077/E5/PG.02.00.PL/2023 dated April 12, 2023, derivative contract number 040/SP2H/PT-L/LL7/2023 dated April 12, 2023, and 05/KP/LPPM/ITATS/2022 dated May 31, 2023.

References

[1] R. A. Firmansyah, Y. A. Prabowo, and T. Suheta, "Thermal imaging-based body temperature and respiratory frequency measurement system for security robot," *Przeglad Elektrotechniczny*, vol. 2022, no. 06, p. 126, 2022.

[2] ——, "Rancang bangun pengukur detak jantung non kontak menggunakan pencitraan termal untuk robot security," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 6, no. 1, p. 1, Jan 2022. [Online]. Available: https://doi.org/10.30865/mib.v6i1.3416

[3] R. A. Firmansyah, W. S. Pambudi, T. Suheta, E. A. Zuliari, S. Muharom, and M. B. S. Hidayatullah, "Implementation of artificial neural networks for localization system on rescue robot," in *2018 Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS)*, Oct 2018, pp. 305–309. [Online]. Available: https://doi.org/10.1109/EECCIS.2018.8692861

[4] A. Rahman, "Penerapan slam gmapping dengan robot operating system menggunakan laser scanner pada turtlebot," *Jurnal Rekayasa Elektrika*, vol. 16, no. 2, p. 2, Aug 2020. [Online]. Available: https://doi.org/10.17529/jre.v16i2.16491

[5] F. Rohman and W. Harsanti, "Analisis kinerja teknologi slam - gmapping untuk robot beroda menggunakan sensor hokuyo laser scanner," *PENA TEKNIK: Jurnal Ilmiah Ilmu-Ilmu Teknik*, vol. 3, no. 1, p. 1, Mar 2018. [Online]. Available: https://doi.org/10.51557/pt_jiit.v3i1.168

[6] T. A. S. R. A. Firmansyah and R. Mardiyanto, "Peningkatan akurasi adaptive monte carlo localization menggunakan convolutional neural network," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, vol. 12, no. 3, p. 3, Aug 2023. [Online]. Available: https://doi.org/10.22146/jnteti.v12i3.7432

[7] M. B. Alatise and G. P. Hancke, "Pose estimation of a mobile robot based on fusion of imu data and vision data using an extended kalman filter," *Sensors*, vol. 17, no. 10, p. 10, Oct 2017. [Online]. Available: https://doi.org/10.3390/s17102164

[8] X. Z. I. Ullah, X. Su and D. Choi, "Simultaneous localization and mapping based on kalman filter and extended kalman filter," *Wireless Communications and Mobile Computing*, vol. 2020, p. e2138643, Jun 2020. [Online]. Available: https://doi.org/10.1155/2020/2138643

[9] Z. C. Y. L. Y. Z. B. Zhou, M. Du and Y. Wang, "Design and implementation of intelligent security robot based on lidar and vision fusion," *J. Phys.: Conf. Ser.*, vol. 2216, no. 1, p. 012013, Mar 2022. [Online]. Available: https://doi.org/10.1088/1742-6596/2216/1/012013

[10] H. Z. T. Mashrik and M. F. Karin, "Design and implementation of security patrol robot using android application," in *2017 Asia Modelling Symposium (AMS)*, Dec 2017, pp. 77–82. [Online]. Available: https://doi.org/10.1109/AMS.2017.20

[11] R. A. Firmansyah and T. Odinanto, "Algoritma pencarian jalur terpendek menggunakan jaringan syaraf tiruan pada aplikasi robot penyelamat kebakaran," *Telcomatics*, vol. 2, no. 2, p. 2, Oct 2017. [Online]. Available: https://journal.uib.ac.id/index.php/telcomatics/article/view/229

[12] W. L. H. L. Y. Wu, Y. Li and R. Lu, "Robust lidar-based localization scheme for unmanned ground vehicle via multisensor fusion," *IEEE Transactions*

on Neural Networks and Learning Systems, vol. 32, no. 12, pp. 5633–5643, Dec 2021. [Online]. Available: https://doi.org/10.1109/TNNLS.2020.3027983

[13] H. F. H. W. A. S. Norzam and K. Kamarudin, "Analysis of mobile robot indoor mapping using gmapping based slam with different parameter," *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 705, no. 1, p. 012037, Nov 2019. [Online]. Available: https://doi.org/10.1088/1757-899X/705/1/012037

[14] Q. Z. P. Wang, Z. Chen and J. Sun, "A loop closure improvement method of gmapping for low cost and resolution laser scanner," *IFAC-PapersOnLine*, vol. 49, no. 12, pp. 168–173, Jan 2016. [Online]. Available: https://doi.org/10.1016/j.ifacol.2016.07.569

[15] S. Nagla, "2d hector slam of indoor mobile robot using 2d lidar," in *2020 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS)*, Dec 2020, pp. 1–4. [Online]. Available: https://doi.org/10.1109/ICPECTS49113.2020.9336995

[16] M. T. S. Saat, W. Abd Rashid and M. Saealal, "Hectorslam 2d mapping for simultaneous localization and mapping (slam)," *J. Phys.: Conf. Ser.*, vol. 1529, no. 4, p. 042032, Apr 2020. [Online]. Available: https://doi.org/10.1088/1742-6596/1529/4/042032

[17] Y. L. Z. Liu, Z. Cui and W. Wang, "Parameter optimization analysis of gmapping algorithm based on improved rbpf particle filter," *J. Phys.: Conf. Ser.*, vol. 1646, no. 1, p. 012004, Sep 2020. [Online]. Available: https://doi.org/10.1088/1742-6596/1646/1/012004

[18] A. B. S. H. S. Y. Abdelrasoul and P. Sebastian, "A quantitative study of tuning ros gmapping parameters and their effect on performing indoor 2d slam," in *2016 2nd IEEE International Symposium on Robotics and Manufacturing Automation (ROMA)*, Sep 2016, pp. 1–6. [Online]. Available: https://doi.org/10.1109/ROMA.2016.7847825

[19] G. P. et al., "An improved amcl algorithm based on laser scanning match in a complex and unstructured environment," *Complexity*, vol. 2018, p. 2327637, Dec 2018. [Online]. Available: https://doi.org/10.1155/2018/2327637

[20] H. T. T. et al., "Extended kalman filter (ekf) based localization algorithms for mobile robots utilizing vision and odometry," in *2022 IEEE 21st Mediterranean Electrotechnical Conference (MELECON)*, Jun 2022, pp. 91–96. [Online]. Available: https://doi.org/10.1109/MELECON53508.2022.9843066