

MAPPING AND POSITIONING SYSTEM ON OMNIDIRECTIONAL ROBOT USING SIMULTANEOUS LOCALIZATION AND MAPPING (SLAM) METHOD BASED ON LIDAR

Article history

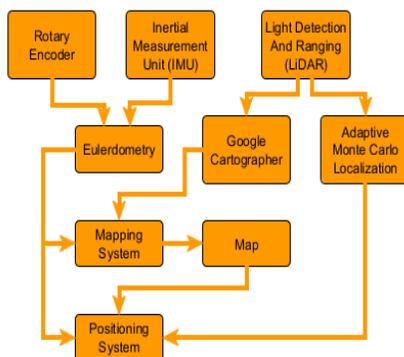
Received
20 May 2021
Received in revised form
23 August 2021
Accepted
2 September 2021
Published Online
20 October 2021

Achmad Akmal Fikri*, Lilik Anifah

*Corresponding author

Department of Electrical Engineering, Faculty of Engineering, Universitas Negeri Surabaya, East Java, Indonesia
achmad.1705874044@mhs.unesa.ac.id

Graphical abstract



Abstract

The main problem from autonomous robot for navigation is how the robot able to recognize the surrounding environment and know this position. These problems make this research weakness and become a challenge for further research. Therefore, this research focuses on designing a mapping and positioning system using Simultaneous Localization and Mapping (SLAM) method which is implemented on an omnidirectional robot using a LiDAR sensor. The proposes of this research are mapping system using the google cartographer algorithm combined with the eulerdometry method, eulerdometry is a combination of odometry and euler orientation from IMU sensor, while the positioning system uses the Adaptive Monte Carlo Localization (AMCL) method combined with the eulerdometry method. Testing is carried out by testing the system five times from each system, besides that testing is also carried out at each stage, testing on each sensor used such as the IMU and LiDAR sensors, and testing on system integration, including the eulerdometry method, mapping system and positioning system. The results on the mapping system showed optimal results, even though there was still noise in the results of the maps created, while the positioning system test got an average RMSE value from each map created of 278.55 mm on the x-axis, 207.37 mm on the y-axis, and 4.28° on the orientation robot.

Keywords: Autonomous Robot, Mapping and Positioning System, SLAM, Google cartographer, AMCL

© 2021 Penerbit UTM Press. All rights reserved

1.0 INTRODUCTION

The main problem in an autonomous robot is how the robot manage to move automatically toward its specified destination without damaging the environment in which the robot passes. In the case of the human in the maze, human need a map of the maze to move out of the room, and it was necessary for the human to know this position on the map so

that the human could determine which way to move out of the room. Autonomous robot also like that, the biggest challenge for robot to find a map of their environment operate and also get to know their own position on the map.

Autonomous robot has a system that able to detect and compute quickly, move automatically and can also communicate between robot or with operator. The ability to compute and combine

information is a challenge for a robot, where the robot must detect the surrounding environment and count the steps of the robot simultaneously to be able to form a map which is commonly called the Simultaneous Localization and Mapping (SLAM) method [1]. Meanwhile, the fundamental problem for autonomous robot to move or navigate is the knowledge of the robot's position in its environment. Many researchers have developed various systems, sensors, and techniques to determine the position of a robot which can be categorized as follows, namely odometry, inertial navigation, magnetic compass, active beacon, Global Positioning System (GPS), landmark navigation, and map matching [2]. With these two basic problems, autonomous robot need a method or system that can create a map of the robot environment and can also determine its own position on the map. Simultaneous Localization and Mapping (SLAM) is a technique or method applied to mobile robots to explore conditions in various environments [14]. SLAM is used to build and generate maps of the environment being explored (mapping). The resulting map is then used to determine the location of the robot and its surrounding landmarks and is also used to plan appropriate pathways for the robot (localization). The mapping and localization processes in SLAM are carried out simultaneously where the mobile robot relatively creates maps. Where the map that has been made is used to calculate and estimate the position of the landmark and the trajectory of the mobile robot [14].

The SLAM method has many variants depending on the use of the spatial dimensions of the sensors used, in two-dimensional (2D) space there are several variants such as Gmapping, HectorSLAM, TinySLAM, Karto, Google Cartographer, and others. Meanwhile, in three-dimensional (3D) space, there are several variants such as ORB-SLAM, LSD-SLAM, RTAB-MAP, and others [15-19]. The sensor used are also of various kinds, the LiDAR sensor is one of them. Light Detecting And Ranging (LiDAR) is a remote sensing technology that can obtain high density and high accuracy geographic reference data about the shape of the environment. LiDAR operates by scanning the field of view with one or more laser beams. The laser beam is generated to scan the measured area, when a wall or object is hit by a laser beam, the laser light will bounce back to the sensor and the sensor device will calculate the detected distance by measuring the time difference between sending the signal and receiving the signal. Because of these characteristics LiDAR is considerably suitable in autonomous vehicle applications and other similar applications, which of course are also suitable for use as input on the SLAM method [3-5]. Currently the google cartographer method is well-known among researchers in the scope of autonomous robot. The google cartographer method is based on the graph optimization approach, not the particle filter approach [20-22]. In the benchmark results that have been carried out in previous studies, mapping in 2D

space shows that the use of the google cartographer method has the highest score and has the smallest error than the results of other methods [15-18].

Whereas in the problem of positioning the robot as mentioned above, there are several categories that can be used to solve it, but in a lot of literature there is still no solution that can really solve this problem. Some of the above categories are grouped into two, namely relative position measurement or dead-reckoning and absolute position measurement or Referenced-based System. So that some robot developers often combine two or more categories of combining the two groups [2]. In studies the most commonly used categories are odometry [9-13], inertial measurement [7-8], magnetic compass [7-8], and map matching [23-31].

This odometry method is based on sensor readings computed with the robot kinematics formula of the type of robot used. Omnidirectional robot is a type of mobile robot that can move in any direction without changing the orientation of the robot, with the wheels used are omni wheels. Therefore, odometry calculations refer to the omnidirectional robot kinematics depending on the number of wheels and the type of omni wheel used in the robot [9-13]. Meanwhile, the method of Inertial Measurement and magnetic compass can be said to have the same characteristics because there are so many integrated circuits (IC) of the Inertial Measurement Unit (IMU) sensor, there are two components of the method, namely Accelerometer, Gyroscope, and Magnetometer [6-8]. In the map matching category there are many methods, one of which is the Adaptive Monte Carlo Localization (AMCL) method that is often used by autonomous robot developers. contrary to the google cartographer method, the AMCL method is based on a particle filter algorithm, which AMCL is one of the implementation variants of the Monte Carlo Localization (MCL) method. AMCL can dynamically adjust the number of particles based on Kullback-Leiber Divergence Sampling (KLD-Sampling) [23-31]. According to the authors, AMCL is almost close to the SLAM method, because when the AMCL algorithm is localized, AMCL also forms its own submap so that it manage to understand the location of the submap. The difference is that AMCL does not create its own global map from its submap and AMCL uses map information from the mapping system results to strengthen location information from its submap.

The researches above have not discussed combining the mapping system and positioning system on a global map and the implementation of combining the two systems on a robot. And also in previous research that still focuses on localization or determining the location of the robot in limited space or a local map or sub map of the environment detected by sensors, so the robot still does not know the coordinates of its position on the global map.

The purposes of this research are the robot can create maps and can determine the position independently with different environmental

conditions, so that the robot can navigate autonomously based on data from the mapping system and positioning system.

The originality of this research is, In the odometry method, the calculation of robot odometry data is combined with the euler orientation Inertial Measurement Unit (IMU) BNO055 data, where we can call this method the eulerodometry method. The Simultaneous Localization And Mapping (SLAM) method used is Hybrid, where the algorithm is used for the mapping system and the positioning system is different. The mapping system uses the google cartographer algorithm combined with eulerodometry, and the positioning system uses the Adaptive Monte Carlo Localization (AMCL) algorithm combined with eulerodometry. And all of these methods are implemented in an omnidirectional robot.

This research is a part of paramedic assistant robot project for medical personnel for covid patients. The contribution of this research can be applied to paramedic assistant robot to help medical personnel work in caring for patients, with the advantage that robots can navigate autonomously without human control, so that they can ease the workload of medical personnel.

2.0 METHODOLOGY

2.1 Hardware Design

2.1.1 Mechanical Design

The robot used is a holonomic robot, the robot can move in any direction without changing the direction of the robot's orientation. The mechanical design of the robot used has a configuration of four DC motor drives by forming an angle of 45 degrees with an X-axis and a Y-axis with dimensions of the robot chassis 50cmx50cmx25cm which can be seen in Figure 1 (a), where the wheels used are omni wheels. For the 3D Robot Design, it can be seen in Figure 1 (b).

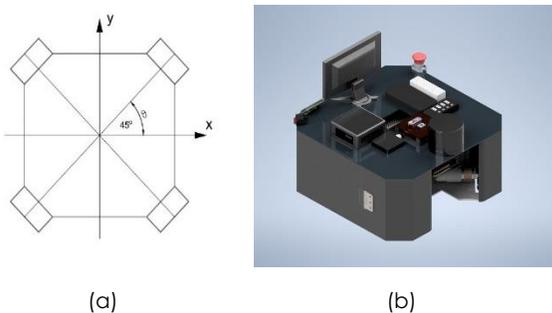


Figure 1 (a) Motor Configuration, (b) 3D Robot Design

2.1.2 Electronic Design

The electronic design used is illustrated in the block diagram as follows.

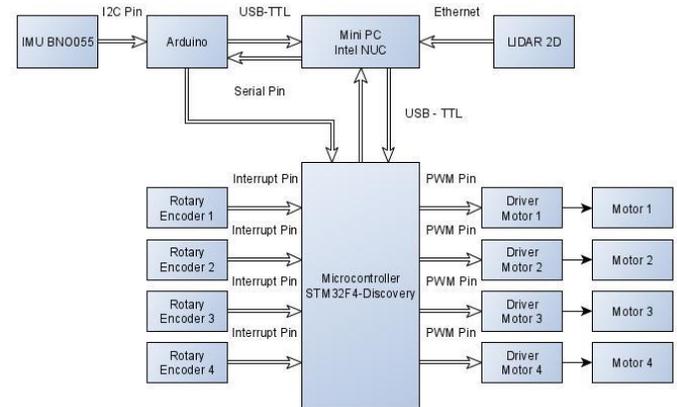


Figure 2 Electronic Block Diagram

The robot uses several components, have two microcontrollers and one minimum PC or mini PC. The microcontroller used is the Arduino Nano with an IC based on ATmega and STM32F4-Discovery with an IC based on ARM Cortex M4, while the mini PC used is the Intel NUC with the 10th generation Intel Core I7 processor. The sensors used are the rotary encoder built in the DC motor and the IMU BNO055 sensor to determine the direction towards the robot, and the two-dimensional (2D) LiDAR sensor with a measurement range of + -30m with a frequency of 20Hz. As for the drive control, four H-Bridge-based DC Motor Drivers are used to drive four DC motors.

STM32F4-Discovery as the master or control center on the robot. The MCU controls the input from the sensor and the mini PC instructions with the output of the robot actuator. Rotary Encoder is accessed by STM32F4-Discovery via an interrupt pin which will be combined with IMU data into an eulerodometry system. Meanwhile, the IMU BNO055 sensor is accessed by Arduino Nano via I2C communication and the data is forwarded to the STM32F4-Discovery and mini PC via serial communication. For communication between the STM32F4-Discovery and the mini PC via serial communication so that the mini PC can find out the robot data information which will be combined with the LiDAR sensor which is directly connected to the mini PC via ethernet communication which is processed to form a mapping system and positioning system.

2.2 System Method

2.2.1 Eulerodometry

The eulerodometry method is a combination of the odometry method with euler orientation data from the IMU sensor. The odometry method is a method for estimating changes in the coordinates of the relative position over time from the actuator movement. In a wheeled robot, the sensor used for the odometric method input is a rotary encoder to detect the number of wheel rotations.

This odometry method is based on robot kinematic, robot kinematic functions to define the direction and speed of the robot. The robots used are included in the holonomic robot so that the robot can move in any direction without changing the robot's orientation. The following is an illustration and equation of an omnidirectional robot with a four-wheel configuration.

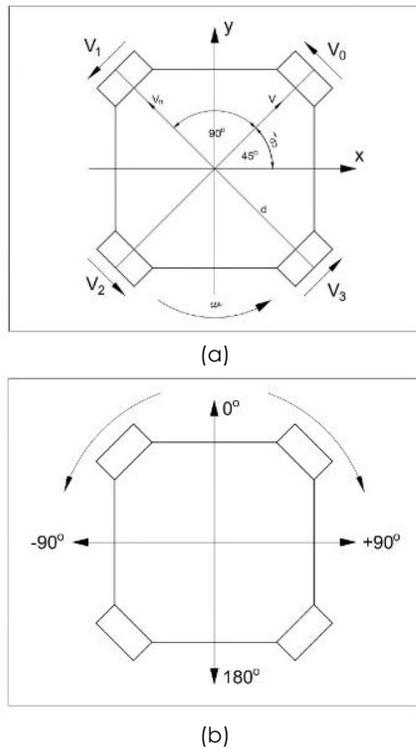


Figure 3 (a) Robot Kinematic configuration, (b) Robot Orientation

$$(x, y, \theta) \Rightarrow V_x(t) = \frac{dx(t)}{dt}, V_y(t) = \frac{dy(t)}{dt}, \omega = \frac{d\theta(t)}{dt} \tag{1}$$

$$\begin{bmatrix} V_x(t) \\ V_y(t) \\ \omega(t) \end{bmatrix} = \begin{bmatrix} \cos(\theta'(t)) & \sin(\theta'(t)) & 0 \\ -\sin(\theta'(t)) & \cos(\theta'(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} V(t) \\ Vn(t) \\ \omega(t) \end{bmatrix} \tag{2}$$

$$\begin{aligned} V_x(t) &= V(t)\cos(\theta'(t)) + Vn(t)\sin(\theta'(t)) \\ V_y(t) &= -V(t)\sin(\theta'(t)) + Vn(t)\cos(\theta'(t)) \\ \omega(t) &= \omega(t) \end{aligned} \tag{3}$$

where,
 (x, y, θ) = Local position of the robot
 (V_x(t), V_y(t), ω(t)) = Linier velocities on the static axis
 (V(t), Vn(t), ω(t)) = Linier velocities on the robot axis

In robot kinematic, the terms forward kinematic and inverse kinematic are known, forward kinematic is a method for determining the orientation or distance of a robot based on wheel speed, while inverse kinematic is the opposite of forward kinematic, which is a method for determining the value of the velocity of each wheel from the

reference distance and orientation to be addressed [9-13]. Equation (4) of robot kinematic equations are as follows.

$$\begin{bmatrix} V(t) \\ V_2(t) \\ V_3(t) \\ V_4(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & d \\ -1 & 0 & d \\ 0 & -1 & d \\ 1 & 0 & d \end{bmatrix} \begin{bmatrix} V(t) \\ Vn(t) \\ \omega(t) \end{bmatrix} \tag{4}$$

$$V(t) = \frac{1}{2}(V_3 - V_1) \tag{5}$$

$$Vn(t) = \frac{1}{2}(V_0 - V_2)$$

$$\omega(t) = (V_0 + V_1 + V_2 + V_3)/(4d)$$

where,
 (V₀, V₁, V₂, V₃) = Wheel velocities
 d = Distance between the center point of the robot and the wheel

Referring to Equation (6), the position of the robot obtained still refer to the local position of the robot, so it need to be converted from the local position of the robot to the global position. At the local position the origin point (0,0) is located at the midpoint of the robot when the calculation is started, while the origin point (0,0) of the global position is located at the location from where the robot operates when the robot starts working. Figure 4 is used to illustartes about the equation.

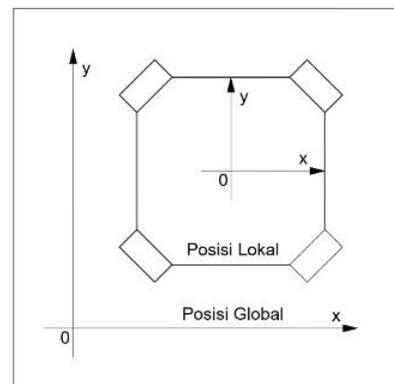


Figure 4 Robot Position Reference

$$\begin{bmatrix} x_g \\ y_g \\ \omega \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_l \\ y_l \\ \omega \end{bmatrix} \tag{6}$$

where,
 (x_g, y_g, ω) = Global position of the robot
 (x_l, y_l, ω) = Local position of the robot

To get the real robot distance value, value of the rotary encoder pulse needs to be converted using the following equation.

$$\text{Wheel circumference} = \text{wheel diameter} \times \pi \tag{7}$$

$$\text{Distance} = \frac{\text{pulse_encoder}}{\text{encoder_revolution}} \times \text{wheel circumference} \quad (8)$$

In the eulderometry method, the robot orientation data is obtained from reading the euler orientation of the IMU. As a result, this odometry method only considers the x- and y-axis positions, as well as the orientation data used to convert the robot's local position to a global position using the IMU euler orientation data. The algorithm flow of the eulderometry method is illustrated in Figure 5.

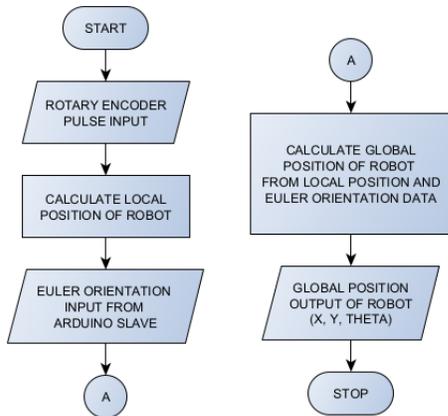


Figure 5 Eulerdometry Flowchart

2.2.2 Mapping System

The method used for the mapping system is the google cartographer method combined with the eulderometry method. The Google cartographer is a method based on the graph optimization approach by matching the scans of the surrounding environment obtained by the sensor to the submap for its best position estimate and matching scans that are only related to that submap at that time. After the submap has been successfully formed, partial loop closure will be carried out using the previously calculated branch and grid positions. And after all the submaps have been completed, the global map will be compiled.

The cartographer system consists of two parts, Local optimization and global optimization, local optimization runs the matching part of the LiDAR scanning frame and its submap, after which the submap is optimized. Meanwhile, global optimization performs global map optimization according to the position relationship between frame scans after finding the closed-loop frame scan.

Local optimization is the process of matching the scan of the LiDAR frame with the submap, and iteratively aligning the scan of the LiDAR frame and the reference of the submap frame to create the submap. Multiple iterations of the frame scan create a probability grid submap of resolution r , with each grid point the system determines the corresponding pixel. Each time a new scan is entered into the

probability grid, a set of grid points that are hit or miss will be counted. Before submitting the map scan to the submap, the frame scan position is optimized by the ceres application in the current submap, and the scan point mapping is superimposed by the nonlinear least squares optimization, and the total value of the scan reaches the hit value. After changing positions and pairing with the probability value in the submap, each place that matches the display should be a big probability to be hit. Since the least squares problem is a local optimization problem, a good initial value will have a large influence on the solution. Therefore, the use of IMU or odometry can be used to provide a variable rotation or position that matches the scan for initial initialization values.

Global optimization is achieved through closed loop detection. Since each LiDAR frame scan only matches the submap containing the most recent frame scans, errors accumulate slowly. To eliminate accumulated errors, the Sparse Pose Adjustment (SPA) method is used to optimize all positions of the scan and submap. The scan position of the LiDAR frame that is inserted into the submap is stored in memory. When the submap is created, the corresponding frame and submap scans are computed for closed-loop detection. All matching scans are performed on the back (back-end), and once a good closed-loop match is found, it is added to the global optimization. The flow of the algorithm from the mapping system method is illustrated in Figure 6.

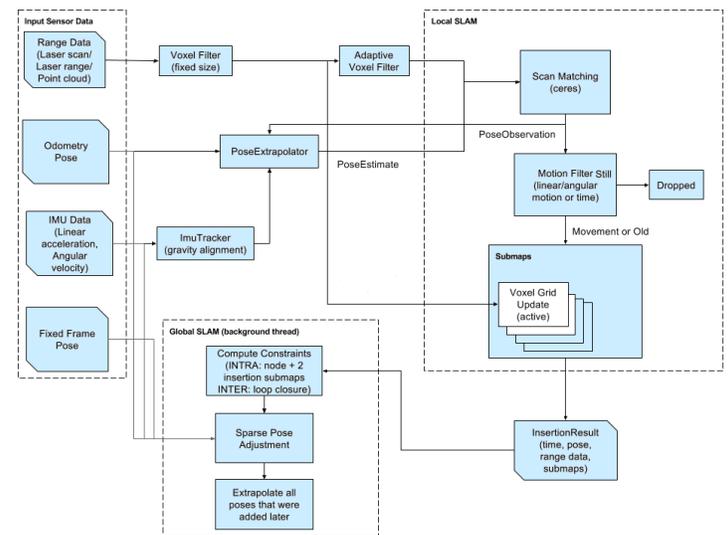


Figure 6 Google Cartographer Flowchart [14-22]

2.2.3 Positioning System

The purpose of the positioning system is to determine the current position of the robot on the map, and positioning accuracy directly affects the accuracy of subsequent navigation. The methods for positioning

commonly used are divided into relative positioning and absolute positioning. Relative positioning determines the current position by measuring the distance and direction of the robot relative to the initial position. However, time drift may occur due to relative positioning based on the accumulation of measured values to achieve accurate positioning, which is not suitable for long distance and time positioning. Absolute positioning determines position based on a single reference. Generally, this method is intended for navigation purposes, because the resulting value has high accuracy even though it still has a few errors [2]. Therefore, the positioning system used is to try to combine the relative and absolute methods, namely using the Adaptive Monte Carlo Localization (AMCL) method with the eulderometry method.

Adaptive Monte Carlo Localization (AMCL) is an implementation variant of the Monte Carlo Localization (MCL) method where AMCL can dynamically adjust the number of particles based on Kullback-Leiber Divergence Sampling (KLD-Sampling). MCL or it can be called a particle filter uses the distribution of the sample particles to represent the actual position, and continuously updates the particles and their weights according to the motion and the robot's observation model to realize the filter particle estimation. The flow of the MCL is illustrated in Figure 7.

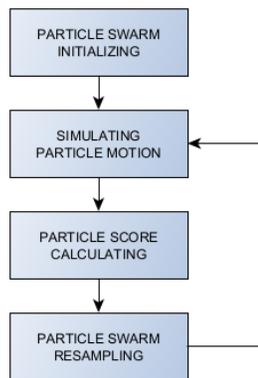


Figure 7 Monte Carlo Localization Flowchart

However, the MCL method still has shortcomings, namely the conventional MCL method will experience serious particle degradation, particle shortages and other phenomena, so that it cannot solve the problem for robot navigation. Therefore, AMCL is conceptualized by combining conventional MCL with the KLD-Sampling method to adjust the number of particles over time based on the distribution of the weight values of the particles to form an adaptive system.

In the implementation of the odometry method, there are weaknesses in the odometry method when the floor or ground surface is uneven or the wheel slips, the data error will get bigger over time, so this causes big problems in the position of the robot. So to solve the problem of positioning the robot, we tried

to combine the AMCL method with eulderometry. By providing the value of the change in position and also the position data of the robot based on the eulderometry method as a comparison of the estimated value of the robot's position on the map so that it can provide additional reference values for the AMCL method in calculating the estimated position of the robot. The flow of the positioning system algorithm is illustrated in Figure 8.

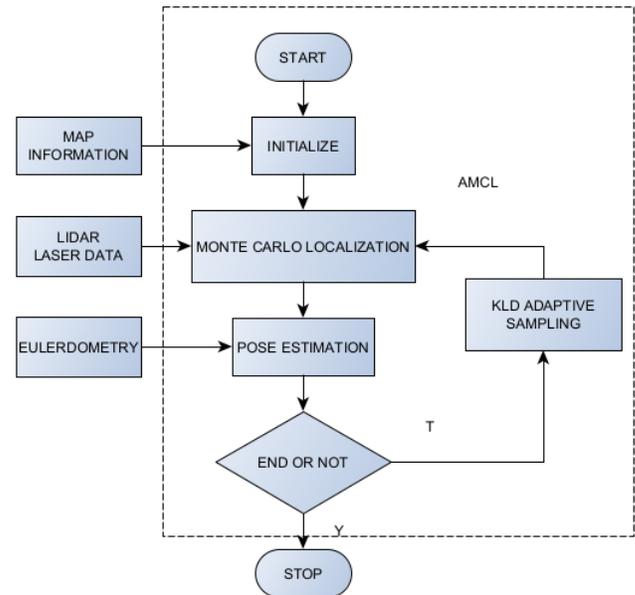


Figure 8 Flowchart From Combination of AMCL and Eulerometry

The limitation in this research is that the testing place used is an indoor room by presenting environmental conditions such as in a hospital, the LiDAR angle range used is 270 degrees, and the robot used is still a prototype with a robot shape that is made similar to a real robot, which is still in the manufacturing stage so that the system created can be directly applied to real robots.

3.0 RESULTS AND DISCUSSION

3.1 IMU BNO055 Test Results

The BNO055 IMU test is carried out to determine the direction of the robot's orientation, the data will be utilized by the eulderometry method. For the test as shown in Figure 9(a).

In this test, the reference data used for testing is to use an orientation sensor on a smartphone, at first the direction is adjusted to the direction of the robot's orientation. The data tested on the robot is yaw data from the euler orientation sensor of the IMU BNO055, namely the orientation data which refers to the z-axis which is illustrated in Figure 9(b).

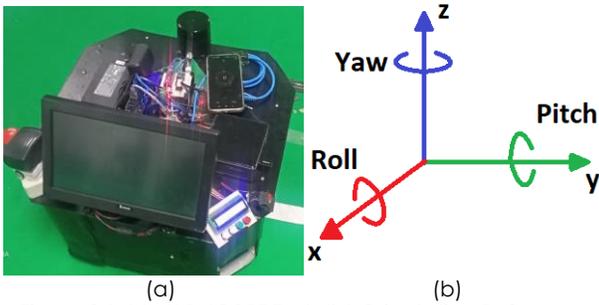


Figure 9 (a) IMU BNO055 Test, (b) Orientation Reference

Because the IMU BNO055 sensor is almost in the middle of the robot in Figure 9(a) the data generated by the IMU sensor is expected to have a high level of accuracy. While the smartphone used to compare the test data is placed near the sensor, it is expected that the orientation data from the smartphone has high precision and low deviation, resulting in more optimal test data.

The test is carried out by rotating the robot at a certain angle, then observing and recording the output value from the sensor. The results of yaw data testing from the euler orientation IMU BNO055 are presented in Table 1. From the data obtained during the test, it is obtained the Root Mean Square Error (RMSE) value or a frequently used measure of the differences between values (sample or population values) predicted by a model or an estimator and the values observed. The RMSE equation is shown in equation (9).

$$RMSE = \sqrt{\frac{\sum (Error)^2}{N}} \tag{9}$$

Where,
N = number of sample data

Table 1 Euler Orientation Yaw IMU BNO055 Results

Orientation (Degree)	Euler Orientation IMU BNO055 (Degree)	Error (Degree)
0	359,94	0,06
19	19,12	0,12
36	36,44	0,44
54	54,19	0,19
72	71,75	0,25
90	89,25	0,75
108	107,3	0,69
126	124,81	1,19
144	143,12	0,88
162	160,81	1,19
180	178,81	1,19
198	197,06	0,94
216	214,12	1,88
234	232,81	1,19
252	250,94	1,06
270	268,62	1,38
288	287,00	1,00
306	304,25	1,75
324	322,06	1,94
342	339,69	2,31
RMSE (Degree)		1,19

The RMSE value in the test is 1.19°. When the robot rotates with a degree below 90°, it still has an RMSE below 1.19°, the RMSE is higher if the robot rotates above 90°. This is because the electromagnetic field affects the magnetometer reading on the IMU sensor and the speed when the robot rotates affects the accelerometer and gyroscope readings, so the Euler orientation readings will affect the RMSE.

The solution offered to reduce the RMSE value is to keep the sensor away from the electromagnetic source by a certain distance, this requires further research. Another solution is to reduce the speed when the robot rotates, with the consequence that the speed of the robot will be reduced. From these data, it can be concluded that the data obtained is optimal and in accordance with the specifications of the BNO055 IMU sensor [6]. So that the sensor can be used in the next process.

3.2 LiDAR Test Results

LiDAR testing is carried out to determine the data from the distance measurement results from the center point of the LiDAR sensor. For the test as shown in Figure 10(a).

In this test, an obstacle is placed with a certain distance and angle according to the LiDAR reading reference. For the LiDAR angle reading shown in Figure 10(b).

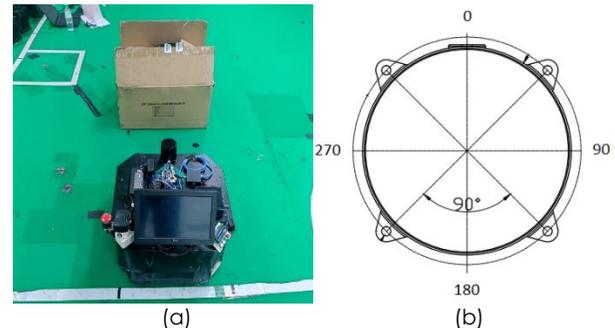


Figure 10 (a) LiDAR Test, (b) LiDAR Reference

Due to the LiDAR angle robot used is 270°, therefore, the angles used for the test are angles 0°, 45°, 90°, 270°, and 315°. The reason for using a 270° LiDAR angle on this robot is that it adjusts the design of a real robot that is still in the manufacturing process, and that angle can already cover the real robot environment. The LiDAR distance reading is in millimeters (mm) and the angle reading is in degrees in the LiDAR test.

The results of testing the LiDAR distance data are presented in the Table 2. From the data obtained during the test, the RMSE value on the test was 23.14 mm. From these data, it is concluded that the data obtained is optimal and in accordance with the specifications of the LiDAR sensor because the resulting RMSE value is very small with 23.14 mm or approximately 2 centimeters (cm), with this value it can be said that the LiDAR sensor used has a high accuracy. So that the sensor can be used in the next process

Table 2 LiDAR Results

Real			LiDAR			Error				
Azimuth (Degree)	Distance (mm)		Azimuth (Degree)	Distance (mm)		Distance (mm)				
0	500	1000	1500	0.00	548	1016	1516	48	16	16
45	707	1414	2121	45.00	680	1396	2144	27	18	23
90	500	1000	1500	90.00	504	980	1484	4	20	16
270	500	1000	1500	270.00	524	988	1476	24	12	24
315	707	1414	2121	315.00	672	1396	2132	35	18	11
RMSE (mm)									23,14	

3.3 Eulerdometry Results

Eulerdometry testing is carried out by changing the position of the robot from the origin (0,0) to a certain point, and then to other points. For reference to the origin point and the definition of the initial x and y axes can be seen in Figure 11.



Figure 11 Eulerdometry Test

In this test, the direction of the robot's orientation is set at a value of 0°, so that the value tested is only the position value on the x-axis and the y-axis with the units used are millimeters (mm). The results of the eulerdometry data test are presented in Table 3. From the data obtained during the test, the RMSE value in the test on the x-axis is 48.37 mm and the y-axis is 66.68 mm. These data indicate that the eulerdometry method produces optimal data even though it still has a large enough RMSE value, because basically the odometry method has drawbacks with poor accuracy because it calculates the estimated displacement of the distance with the number of revolutions of the robot wheel, so many factors affect the resulting value such as wheel slipping, wheel rotation frequency reading and others. With an RMSE value of 48.37 mm on the x-axis and 66.68 mm on the y-axis, the data can still be categorized as optimal. And also in Table 3, it can be concluded that the farther robot moves, the resulting error is also greater.

Table 3 Eulerdometry Results

Real (mm)		Eulerdometry Robot (mm)		Error (mm)	
X	Y	X	Y	X	Y
500	0	490	-10	10	10
1000	0	970	0	30	0
1500	0	1450	0	50	0

Real (mm)		Eulerdometry Robot (mm)		Error (mm)	
2000	0	1990	0	10	0
3000	0	3010	-10	10	10
0	500	0	470	10	30
0	1000	0	940	10	60
0	1500	0	1430	0	70
0	2000	0	1940	0	60
0	3000	10	2880	10	120
1000	1000	940	1010	60	10
1000	2000	910	2070	90	70
2000	2000	1920	1990	80	10
3000	2000	2930	1860	70	140
3000	3000	2910	2880	90	120
RMSE (mm)				48,37	66,68

3.4 Mapping System Results

Testing of the mapping system is carried out to test the results of the google cartographer method combined with the eulerdometry method. The test was carried out on the third floor of the Technology and Entrepreneurship Laboratory Building, Universitas Negeri Surabaya. Environmental conditions and the floor plan of the test site can be seen in Figure 12 and Figure 13 respectively.



Figure 12 Mapping System Test Locations From All Sides

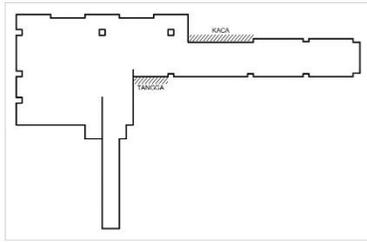


Figure 13 Ground Plan

The test is carried out by giving keyboard instructions to move the motor to run around the room. The test was carried out five times, with the map results shown in Figure 14. There is a slight mismatch with the plan where when the robot crosses the staircase area and glass walls, so that when entering the hallway there is noise on the map. Since there are two stairs, the stairs up and down, so that the barrier from the stairs down is not detected which instead detects the outer wall of the stairs, while the glass wall cannot be detected because the glass can refract the laser from LiDAR so that the laser light can penetrate the glass and detect the wall or objects outside the glass wall. Noise is also found at the very end of the hallway because the reading of the comparison of the data from LiDAR with the orientation information from eulermometry has different frequencies, so that when the robot moves rapidly the data obtained there is an error matching the scan when forming a submap. The following are the results of the map made during testing.

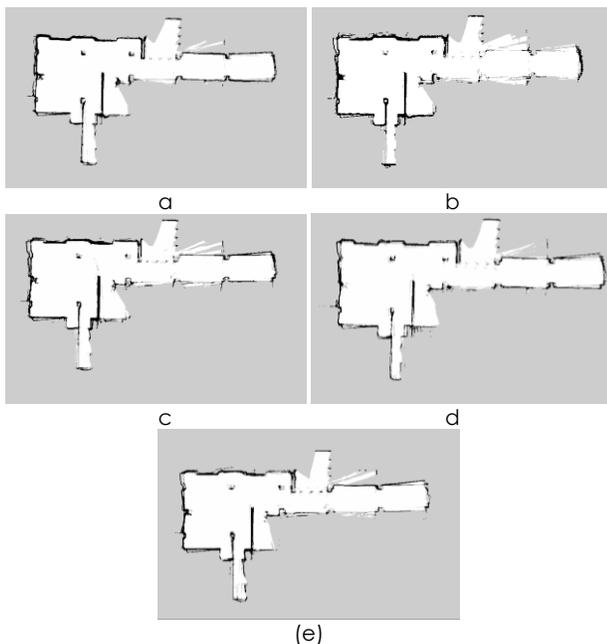


Figure 14 Mapping System Results, (a) First Map Created, (b) Second Map Created, (c) Third Map Created, (d) Fourth Map Created, (e) Fifth Map Created

3.5 Positioning System Results

Positioning system testing is carried out to test the results of the AMCL method combined with eulermometry. This positioning test uses a map that has been generated in the previous mapping system test, with the intended position determined and can be seen in Figure 15.



Figure 15 Positioning System Test Locations, (a) Start Position, (b) First Position, (c) Second Position, (d) Third Position, (e) Fourth Position, (f) Fifth Position

We designed a field robot like Figure 15, because this robot is used in certain hospitals. The fields that have been built have been adjusted to the actual field conditions. With adjustments according to actual conditions, the robot can be directly used in real conditions, with this field design the movement of the robot will be useful according to needs.

For robot control, it is same as testing the mapping system using keyboard instructions to move to a predetermined position. Because the map used as a reference has various characteristics, each test yields different results, so the position value obtained varies as well. The results of this positioning system test can be seen in Table 4. From the data obtained during the test, the average value of the RMSE for each test map obtained is 278.55 mm on the x-axis, 207.37 mm on the y-axis, and 4.28° in the direction of the robot's orientation. These data indicate that the positioning system produces optimal data even though it still has a large enough RMSE value, because in this positioning system combines two methods with map characteristics data greatly affect the results of the

system. Each map that is used as a reference map has different results, so the resulting position data varies depending on the characteristics and shape of the map grid generated in the previous mapping system. With the average RMSE value on each map,

it can be said that the data may still be tolerated because it does not surpass the robot's dimensions, thus the deviation is not too noticeable in real-world situations and this data can be a reference for the robot in navigation.

Table 4 Positioning System Results

Real			AMCL + Eulerdometry			Error		
First Map								
X (mm)	Y (mm)	θ (Deg)	X (mm)	Y (mm)	θ (Deg)	X (mm)	Y (mm)	θ (Deg)
0	0	0	54	26	0,91	54	26	0,91
5400	100	180	5810	284	175,52	410	184	4,48
5400	4200	90	5539	3870	85,50	139	330	4,5
600	12000	90	1162	12132	94,21	562	132	4,21
-900	18000	-90	-874	18026	-92,99	26	26	2,99
600	21000	90	1046	21115	88,37	446	115	1,63
RMSE						342,97	170,66	3,42
Second Map								
X (mm)	Y (mm)	θ (Deg)	X (mm)	Y (mm)	θ (Deg)	X (mm)	Y (mm)	θ (Deg)
0	0	0	-14	-68	0,91	14	68	0,91
5400	100	180	5632	-338	177,47	232	238	2,53
5400	4200	90	5360	4148	77,89	40	52	12,11
600	12000	90	752	12262	86,26	152	262	3,74
-900	18000	-90	-250	18316	-86,91	650	316	3,09
600	21000	90	693	21369	71,10	93	369	18,9
RMSE						291,51	247,86	9,43
Third Map								
X (mm)	Y (mm)	θ (Deg)	X (mm)	Y (mm)	θ (Deg)	X (mm)	Y (mm)	θ (Deg)
0	0	0	-35	-105	-1,49	35	105	1,49
5400	100	180	5701	-385	-176,87	301	285	3,13
5400	4200	90	5519	3950	86,43	119	250	3,57
600	12000	90	941	12001	91,21	341	1	1,21
-900	18000	-90	-1061	18134	-96,08	161	134	6,08
600	21000	90	596	21182	89,43	4	182	0,57
RMSE						203,39	185,21	3,25
Fourth Map								
X (mm)	Y (mm)	θ (Deg)	X (mm)	Y (mm)	θ (Deg)	X (mm)	Y (mm)	θ (Deg)
0	0	0	-70	-95	0,00	70	95	0
5400	100	180	5939	-441	174,60	539	341	5,4
5400	4200	90	5767	3926	87,56	367	274	2,44
600	12000	90	847	12230	85,78	247	230	4,22
-900	18000	-90	-493	18070	-89,59	407	70	0,41
600	21000	90	892	21172	87,81	292	172	2,19
RMSE						351,67	219	3,10
Fifth Map								
X (mm)	Y (mm)	θ (Deg)	X (mm)	Y (mm)	θ (Deg)	X (mm)	Y (mm)	θ (Deg)
0	0	0	-3	-52	-1,14	3	52	1,14
5400	100	180	5564	-362	177,82	164	262	2,18
5400	4200	90	5575	3820	86,51	175	380	3,49
600	12000	90	882	12049	90,32	282	49	0,32
-900	18000	-90	-1178	18152	-92,59	278	152	2,59
600	21000	90	783	21184	92,02	183	184	2,02
RMSE						203,23	214,12	2,20
Average RMSE						278,55	207,37	4,28

4.0 CONCLUSION

Based on the results and test analysis, the following conclusions are obtained. This research presents a combination of several systems to get maximum results in creating a robot environment map and determining the position of the robot in its environment as a reference for the robot to

navigate. From the results of the tests that have been carried out, the RMSE value on the yaw euler orientation IMU BNO055 data reading is 1.19°, the RMSE value on the LiDAR test is 23.14 mm, the RMSE value on the eulerdometry test is 48.37 mm on the x-axis and 66.68 mm on the y axis. From the test results, it becomes a reference for testing the mapping and positioning system. The map produced by the

mapping system still has a little noise because the environmental conditions tested have a glass wall that can refract the laser light from the LiDAR sensor so that it can become noise data that can affect global map making, and also the frequency difference between the LiDAR data and the direction data towards the robot of the eulardometry method can make a difference in matching scans when the robot is moving rapidly rotating. Whereas in testing the positioning system varies greatly depending on the map that is used as a reference, because the system compares the real scan data with map information with the average RMSE value for each test map of 278.55 mm on the x-axis, 207.37 mm on the y-axis, and 4.280 in the direction of the robot's orientation. From these data, we conclude that the designed system can be used to make maps of the robot's environment optimally and the resulting position estimation can be a reference for the robot in navigation, and also this system can solve the main problem of autonomous robots in recognizing their environment and knowing their position. The system created in this study can be directly applied to robots to navigate autonomously which aims to assist medical personnel in handling COVID-19 patients, and this system can also be used on autonomous robots operating in other fields.

Acknowledgement

The authors would like to acknowledgement and express much appreciation to Universitas Negeri Surabaya, Surabaya, Indonesia, especially Department of Electrical Engineering, Faculty of Engineering and related parties for the opportunity and facilities provided to carry out this research.

Reference

- [1] Correll, N. 2016. *Introduction to Autonomous Robots*. Creative Commons. <http://uiiis.unsyiah.ac.id/oer/files/original/f8bdc5ed4ad9d6c70b7923b1a7e19a7e.pdf>.
- [2] Borenstein, J., H. R. Everett, L. Feng, and D. Wehe. 1997. Mobile Robot Positioning-Sensors and Techniques. *Journal of Robotic Systems*. 14(4): 231-249. DOI:10.1002/(SICI)1097-4563(199704)14:4<3C231::AID-ROB2%3E3.0.CO;2-R.
- [3] Wang, X., H. Pan, K. Guo, X. Yang, and S. Luo. 2020. The Evolution of LiDAR and Its Application in High Precision Measurement. *IOP Conference Series: Earth and Environmental Science*. 502 012008. DOI: 10.1088/1755-1315/502/1/012008.
- [4] Li, Y. and J. Ibanez-Guzman. 2020. Lidar for Autonomous Driving. *IEEE Signal Processing Magazine*. 37(4): 50-61. DOI: 10.1109/MSP.2020.2973615.
- [5] Yang, J., Y. Li, L. Cao, Y. Jiang, L. Sun, and Q. Xie. 2019. A Survey of SLAM Research based on LiDAR Sensors. *International Journal of Sensors*. 1(1): 1003. <http://www.remedypublications.com/open-access/a-survey-of-slam-research-based-on-lidar-sensors-4870.pdf>.
- [6] Bosch Sensortec. 2016. BNO055, Intelligent 9-Axis Absolute Orientation Sensor. Datasheet BST-BNO055-DS000-14. https://www.bosch-sensortec.com/pdfdocs/BST_BNO055_DS000_14.pdf.
- [7] Aji, W. S. 2019. Kendali Berputar pada Robot Kontes Robot Abu Indonesia dengan kendali PID dan IMUBNO055. *Buletin Ilmiah Sarjana, Teknik Elektro*. 2(1): April 2020. 14-23. <http://journal2.uad.ac.id/index.php/biste/article/view/987/pdf>.
- [8] Langley, R. 2016. Development of a Self-balancing Robot utilizing FPGA. Engineering Honors Thesis. School of Engineering and Information Technology, Murdoch University, Perth, Western Australia. <http://researchrepository.murdoch.edu.au/id/eprint/38687>.
- [9] Taufiqurrohman, M. and N. F. Sari. 2018. Odometry Method and Rotary Encoder for Wheeled Soccer Robot. *IOP Conference Series: Materials Science and Engineering*. 407(2018): 012103. DOI: 10.1088/1757-899X/407/1/012103.
- [10] Phunopas, A. and S. Inoue. 2018. Motion Improvement of Four-Wheeled Omnidirectional Mobile Robots for Indoor Terrain. *Journal of Robotics, Networking and Artificial Life*. 4(4): 275-282. DOI: 10.2991/jrnal.2018.4.4.4.
- [11] Sofwan, A., H. R. Mulyana, H. Afrisal, and A. Goni. 2019. Development of Omni-wheeled Mobile Robot Based-on Inverse Kinematics and Odometry. *2019 6th International Conference on Information Technology, Computer and Electrical Engineering (ICITACEE)*, 26-27 Sept. 2019, Semarang, Indonesia. DOI: 10.1109/ICITACEE.2019.8904418.
- [12] Rijalulalam, D. U. and I. Iswanto. 2021. Implementation Kinematics Modeling and Odometry of Four Wheel Mobile Robot on the Trajectory Planning and Motion Control Based Microcontroller. *Journal of Robotics and Control (JRC)*. 2(5): Sept. 2021. DOI: <https://doi.org/10.18196/jrc.25121>.
- [13] Fahmizal, D. U. Rijalussalam, M. Budiyanto, and A. Mayub. 2019. Trajectory Tracking pada Robot Omni dengan Metode Odometry. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*. 8(1): Feb. 2019. DOI: <http://dx.doi.org/10.22146/jnteti.v8i1.488>.
- [14] Khairuddin, A. R., M. S. Talib, and H. Haron. 2015. Review on Simultaneous Localization and Mapping (SLAM). *2015 IEEE International Conference on Control System, Computing and Engineering*, 27-29 Nov. 2015, Penang, Malaysia. DOI: 10.1109/ICCSCE.2015.7482163.
- [15] Aerts, P. and E. Demeester. 2017. Benchmarking of 2D-SLAM Algorithms. ACRO Research Group, KU Leuven. Departement of Mechanical Engineering, Campus Diepenbeek. http://www.acro.be/downloadvrij/Benchmark_2D_SLAM.pdf.
- [16] Filipenko, M. and I. Afanasyev. 2018. Comparison of Various SLAM Systems for Mobile Robot in an Indoor Environment. *9th IEEE International Conference on Intelligent Systems 2018*, 25-27 Sept. 2018, Medeira, Portugal. DOI: 10.1109/IS.2018.8710464.
- [17] Vanelli, B. 2019. Comparison and Benchmarking for SLAM in Mobile Robots. Automation and Control Engineering of the Universidade Federal de Santa Catarina. <https://repositorio.ufsc.br/handle/123456789/196836>.
- [18] Yagfarov, R., M. Ivanou, and I. Afanasyev. 2018. Map Comparison of Lidar-based 2D SLAM Algorithms Using Precise Ground Truth. *15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 18-21 Nov. 2018, Singapore. DOI: 10.1109/ICARCV.2018.8581131.
- [19] Huang, B., J. Zhao, and J. Liu. 2020. A Survey of Simultaneous Localization and Mapping with an Envision in 6G Wireless Network. arXiv:1909.05214. <https://arxiv.org/pdf/1909.05214>.
- [20] Hess, W., D. Kohler, H. Rapp, and D. Andor. 2016. Real-Time Loop Closure in 2D LIDAR SLAM. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 16-21 May 2016 Stockholm, Sweden.

- DOI: 10.1109/ICRA.2016.7487258.
- [21] Zhi, C. and S. Xiumin. 2019. Research on Cartographer Algorithm based on Low Cost Lidar. *International Journal of engineering Research and Technology (IJERT)*. 8(10): Oct. 2019. DOI: 10.17577/IJERTV8IS100060.
- [22] Nuchter, A., M. Bleier, J. Schauer, and P. Janotta. 2017. Improving Google's Cartographer 3D Mapping by Continuous-time SLAM. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Volume XLII-2/W3, 2017. DOI: 10.5194/isprs-archives-XLII-2-W3-543-2017.
- [23] Liu, X., W. Wei, Y. Gao, and X. Wei. 2019. Research on Improved Localization and Navigation Algorithm for Automatic Guided Vehicle. *IOP Conference Series: Materials Science and Engineering*. 611(2019): 012076. DOI: 10.1088/1757-899X/611/1/012076.
- [24] Rowekamper, J., C. Sprunk, G. D. Tipaldi, C. Stachniss, P. Pfaff, and W. Burgard. 2012. On the Position Accuracy of Mobile Robot Localization based on Particle Filters Combined with Scan Matching. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 7-12 Oct. 2012, Vilamoura-Algarve, Portugal. DOI: 10.1109/IROS.2012.6385988.
- [25] Peng, G., W. Zheng, Z. Lu, J. Liao, L. Hu, G. Zhang, and D. He. 2018. An Improved AMCL Algorithm Based on Laser Scanning Match in a Complex and Unstructured Environment. *Hindawi Complexity*. Volume 2018, Article ID 2327637, 11 pages. DOI: 10.1155/2018/2327637.
- [26] Das, S. 2018. Robot Localization in a Mapped Environment Using Adaptive Monte Carlo Algorithm. *International Journal of Scientific and Engineering Research*. 9(10): October-2018. <https://www.ijser.org/researchpaper/Robot-localization-in-a-mapped-environment-using-Adaptive-Monte-Carlo-algorithm.pdf>.
- [27] Supper, G., C. Aschauer, A. Gronauer, and N. Barta. 2021. Localisation and Navigation of a Robot Platform Using Laser Scanner and Adaptive Monte Carlo Localization (AMCL) for an Indoor Horticultural Application. *Informations- und Kommunikationstechnologien in kritischen Zeiten, Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2021* 313. https://dl.gi.de/bitstream/handle/20.500.12116/35693/GIL2021_Supper_313-318.pdf?sequence=1&isAllowed=y.
- [28] Yatim, N. M., and N. Buniyamin. 2015. Particle Filter in Simultaneous Localization and Mapping (SLAM) Using Differential Drive Mobile Robot. *Jurnal Teknologi (Sciences and Engineering)*. 77(20): 91-97. DOI: 10.11113/jt.v77.6557.
- [29] de Miguel, M. A., F. Garcia, and J. M. Armingol. 2020. Improved LiDAR Probabilistic Localization for Autonomous Vehicles Using GNSS. *MDPI Journal Sensors* 2020. 20: 3145. DOI: 10.3390/s20113145.
- [30] Zhou, J., C. He, and J. Fang. 2020. Positioning System Based on Lidar Fusion. *2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI)*. 1: 32-35. DOI: 10.1109/IWECAI50956.2020.00013.
- [31] Ffreira, L. 2013. Localization and Navigation in an Autonomous Vehicle. Universidade de Aveiro, Departamento de Electrónica, Telecomunicações e Informática, 2013. <https://core.ac.uk/download/pdf/19448422.pdf>.