Autonomous Navigation System Using Indoor GPS on Turtlebot Burger Robot

Yeni Riska Pratiwi[1], Hendawan Shoebhakti[2], Faiz Albar Risi[3], Rahel Yulianti[4]
{ yeniriskap@gmail.com, hendawan@polibatam.ac.id, , albarrisifaiz@gmail.com, racheljulianti@gmail.com}

Politeknik Negeri Batam, Department of Elecrical Engineering, Study Program Robotics Engineering, Ahmad Yani Street, Batam Centre, Batam 29461, Indonesia

Abstract. The Robot Turtlebot3 Burger was developed as an autonomous mobile robot, a robot that can run automatically or only requires one command. So that the robot can run automatically, a navigation system is needed. In this study, an autonomous navigation system was developed for the turtlebot3 burger robot that moves on land on a flat surface with Indoor Positioning System (IPS) technology through an ultrasonic approach using Marvelmind robotics so that the robot can recognize the position and direction based on the Earth's coordinate system in the room. Position-based IPS uses the waypoint method to determine the movement of the robot from the initial coordinate point to the destination coordinate point based on the Earth coordinate system. The test results show that the autonomous navigation system using indoor GPS on the turtlebot3 burger robot has an accuracy of ± 1 cm. Based on the RMSE calculation, navigation using indoor GPS has a lower error rate than without using indoor GPS. Navigation using the waypoint method can regulate the movement of the autonomous robot according to the robot's point and mission.

Keywords: Turtlebot3, autonomous, Indoor GPS, waypoint, navigation

1 Introduction

As the company's needs in providing products to serve consumers are growing. Products must be produced in sufficient quantities based on a certain period of time. This requires the company team to work faster with optimal results. Goods distribution activities are one of the important parts of making a product. But this will take more time and energy if done personally and repeatedly. Therefore, an automated device for dispensing goods is needed, such as a goods delivery robot.

To make an delivery robot project that suits the Company's problems, a learning media and prototype are needed that can run autonomously with a good navigation system. Many companies use delivery robots with navigation using magnetic tracks. However, this navigation has some drawbacks. Such as having the potential to damage the surrounding environment, it is necessary to set up the field manually by making a layout so that if the robot is moved to another room, it is necessary to set up the field again, and this requires extra costs. The prototype must also be able to run according to field conditions in the company. In this

ICAE 2022, October 05, Batam, Indonesia

DOI 10.4108/eai.5-10-2022.2327481

study, the author uses the Turtlebot3 Burger as a learning medium for the manufacture of a delivery robot system.

Researchers Hendrik J. Djahi, et al offer a solution using the rotary encoder method, the data received by the sensor in the rotary encoder will be processed by the microcontroller to regulate the movement of the robot according to the program. However, there are factors that affect the final result of the robot's movement, namely the presence of vibrations in the robot and the uneven surface of the trajectory. This greatly affects the reading of the number of pulses on the rotary encoder, so that it affects the results of the odometry calculations on the robot [7]. Researcher R. Sridhar, offers a solution using LED and LDR as sensors to detect the color of the lines installed on the floor as a reference for movement. The robot movement route is distinguished by color [1]. Most of these systems have been widely used in Indonesia. However, it has several drawbacks, such as having the potential to damage the surrounding environment and needing to set up the field manually. The prototype must also be able to run according to field conditions in the company. In this study, the author uses the Turtlebot3 Burger as a learning medium for the manufacture of a delivery robot system.

The Turtlebot3 Burger Robot offers a solution to solving the company's problems as a medium of learning and making product prototypes [2]. The Indoor GPS localization system is used to get coordinate data on the robot, which is expected to have a better system response. With indoor GPS, the robot can also be determined in the direction of its movement by the waypoint method that adapts to field conditions so that the robot can move to the intended point and return to its original place.

2 Literature Review

On the Turtlebot3 Burger robot, the robot is required to run autonomously. Therefore, methods are needed that can support research. Research in the section Autonomous Navigation System Using Indoor GPS on the Turtlebot3 Burger Robot discusses the mobile navigation system of the robot through the Turtlebot3 burger prototype media so that the robot can move autonomously using the waypoint method, which uses IPS from marvel mind robotics as a positioner and directioner in its navigation system.

2.1 Autonomous Navigation of Robots

An autonomous robot is a robot that can move automatically or only requires one command [4]. Autonomous robots are divided into two, namely autonomous stationary robots, applied to robots that are stationary so that they have a limited range of movement, then developed into autonomous mobile robots, namely robots that can move positions so that they have a wider range [8].

A navigation system is needed to guide the movement of the robot by determining the coordinates of the position and direction of the robot [3]. In this study, autonomous navigation is applied to a robot that moves indoors on flat ground. **Figure 1** shows the configuration of the mobile robot with differential wheels mounted on the left and right sides of the robot using a servo dynamixel and additional freewheels behind it. In this configuration, the left (VL) and right (VR) wheel speeds are independent to move to a certain point (x,y) and a certain direction (ϕ) in a two-dimensional area. In this study, an indoor navigation system was designed using Indoor Positioning System (IPS) technology).

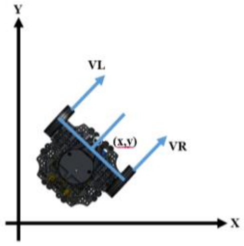


Fig 1. Differential wheel configuration on a mobile robot

2.2 Indoor Positioning Systems (IPS)

Indoor Positioning System (IPS) is a wireless technology used to detect the position/presence of objects in the room [10]. There are two possible topologies for IPS, as shown in **figure 2**. In topology 1, the transmitter is placed at the four corners while the receiver is placed at the object to be tracked. In topology 2, the receiver is placed at the four corners, and the transmitter is placed at the object to be tracked. Topology 2 is more suitable to be applied because it saves electricity.

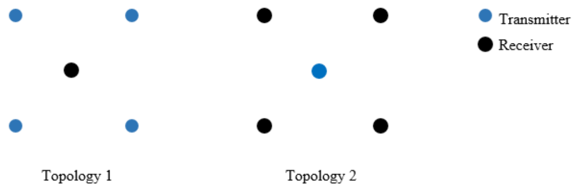


Fig 2. Possible topologies for IPS

IPS implementation can be done using several approaches. Researcher C. K. M. Lee, et al used Bluetooth as their IPS system. However, Bluetooth has a proposed effective range of 9.5 meters [9]. IPS products, such as marvel mind robotics, offer an ultrasonic-based approach with a range of up to 50 meters [5]. The ultrasonic frequency signal consists of a transmitter & receiver placed separately between the robot and several sensor units, which are mounted statically to form a General Positioning System (GPS). In this study, we use the marvel mind robotics series starter set industrial NIA 01 as the IPS system.

Fig 3. Marvelmind robotics starter set NIA-01

Figure 3 is part of the Marvelmind robotics starter set industrial NIA 01, which consists of the following:

1. 4x industrial RX beacons as an ultrasonic signal receiving static units.
2. 1x industrial TX beacon as mobile beacon transmits an ultrasonic signal.
3. 1x HW V5.1 modem as a control center for the entire system connected to a PC.

The layout of the Marvelmind robotics looks like **figure 4**. The power supply of the TX beacon attached to the Turtlebot3 burger comes from an 11.1V lippo battery. While the power supply of the wall-mounted RX beacon comes from the IP67 converter 110 - 220 VAC to 12 VDC. For the modem, the power supply is taken from the USB port attached to the monitor.

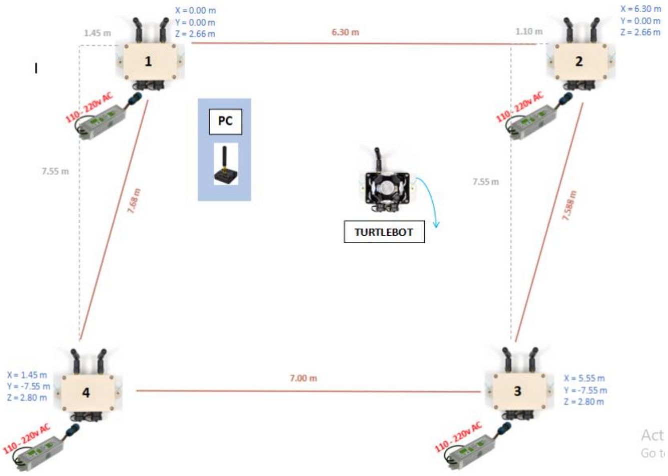


Fig 4. Marvelmind robotics layout

2.3 Inertial Measurement Unit (IMU)

An inertial Measurement Unit (IMU) is a sensor to measure the speed, orientation, and force of gravity. It consists of an accelerometer sensor, a gyroscope, and a magnetometer. The turtlebot3 robot uses an open CR microcontroller that has a MPU9250 type IMU installed in the middle of the microcontroller with a 3-axis gyroscope and a 3-axis accelerometer [6].

2.4 Waypoint Robot

Waypoint is a navigation method to adjust the robot's motion from an initial set point to a destination set point, using coordinate points (longitude & latitude if based on earth coordinates) [4] to determine the destination set point. Each set point of the movement destination is assumed to be 1 point in a certain coordinate system. The waypoint navigation method is made so that the robot can know the direction of motion and the direction it is facing so that it can increase accuracy to reach the destination set point on a predetermined route. **Figure 6** shows the field and position of the robot waypoint. The field is made with a 22 cm high dividing wall made of 3D printed filament as the foundation and cardboard as the wall. The orange circle is the set point of the robot's journey. The black circle represents the robot's destination set point, and the blue box represents the robot's home base position.

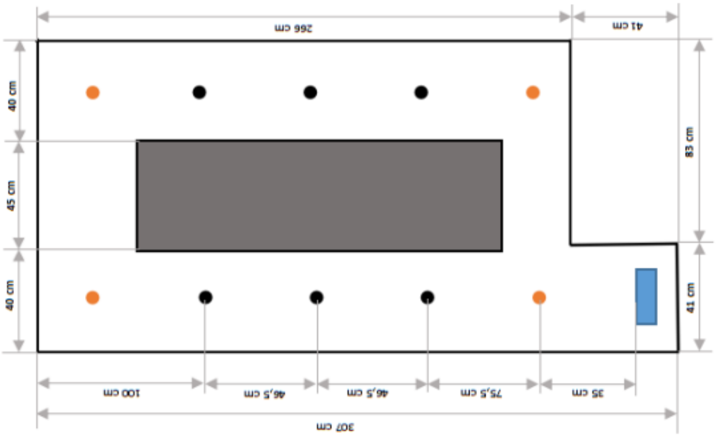


Fig 6. Robot field and waypoint position

In the waypoint method, by knowing the coordinates of the initial set point & destination set point, the difference between the direction of the initial face and the direction of the destination face is obtained so that the robot will know the extent to which the robot will rotate towards the direction of the destination face. The rotation coordinates of the starting point and the destination point must be known, as described in **Figure 7**.

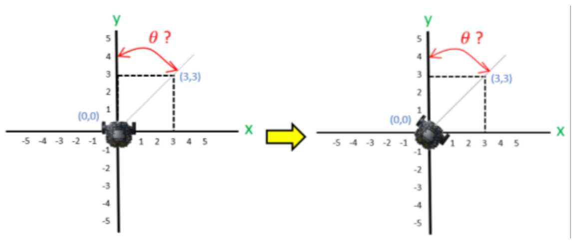


Fig 7. Illustration of determining the facing direction of the robot

To determine the extent to which the robot rotates towards its destination, trigonometric functions are used. This function is a branch of mathematics that deals with the angles of triangles [12]. The formula is as follows:

$$\theta = \text{atan2}(x, y) \tag{1}$$

This formula returns the value of the angle in radians between the x-axis and the line segment from the coordinates of the robot's position to the coordinates of the robot's destination (x,y).

The coordinates of the initial set point & the destination set point can also make the robot know the distance it has to travel to reach the specified destination set point and continue its destination in the same way. The coordinates of the initial set point & the destination set point must be known, as described in **Figure 8**.

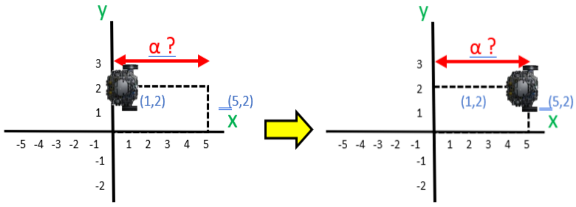


Fig 8. Illustration of determining the robot's mileage

To determine the distance that must be traveled to reach the desired set point, the Pythagorean function is used. This function states that the square of the hypotenuse is equal to the sum of the squares of the other two sides [11]. The formula is as follows:

$$\alpha = \sqrt{(x^2+y^2)} \quad (2)$$

This formula calculates the length of the line that lies between two coordinate points. The linear distance is the square root of the square of the x-axis distance plus the square of the y-axis distance between the two points.

3 Method

3.1 Block Diagram System Design

A system block diagram design is needed to find out the physical components that will be used in the turtlebot3 burger robot so that the algorithm in the software can be applied later.

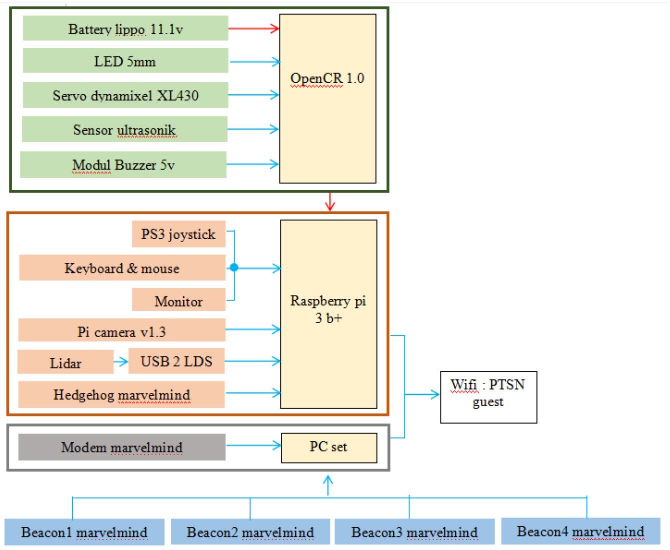


Fig 9. Block diagram system design of turtlebot3

Figure 9 shows the design of a modified turtlebot3 system block diagram with the following information:

1. OpenCR, low-level control microcontroller for programming.
2. Raspberry, a single-board computer, is a control center that will send the robot's movement data to the microcontroller. Robots are programmed in python. In it is installed ROS and several turtle bot burger packages from the web manual as well as homemade packages to run the joystick as a remote control, lidar as obstacle detection, and Hedgehog to display the coordinates of the robot's position.
3. Indoor GPS, used as a robot navigation system to move from one place to another using X, Y, and angular waypoint coordinates.
4. PC set, as a robot controller. PC and raspberry must be connected to the same wifi.

3.2 Turtlebot3 Navigation System Design

Based on the company's request for a project delivery robot through the Turtlebot3 burger robot prototype, the robot is asked to walk autonomously to the set point, then return to the original place. The robot is required to be able to sort the points that are intended so that the robot does not move back and forth and complete all its missions in one trip only.

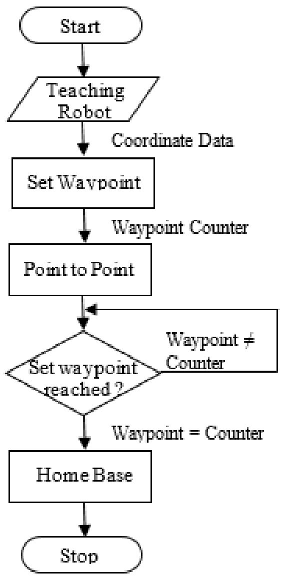
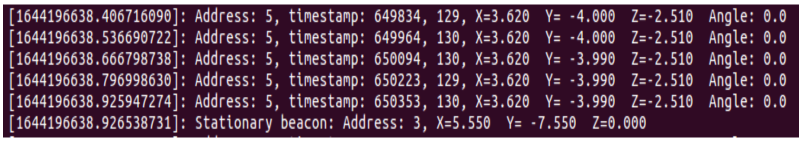


Fig 10. Flowchart with turtlebot navigation system3 burgers

Figure 10 shows a flowchart of the turtlebot3 burger navigation system. The first step is to teach the robot first using a remote control to find out the coordinates of the robot's destination set point. This teaching produces coordinates of the robot's x,y, and z position data through the RX beacon installed on the robot. **Figure 11** shows the position of the coordinates for placing the TX beacon, the real-time value of the x,y, and z position of the robot.



```
[1644196638.406716090]: Address: 5, timestamp: 649834, 129, X=3.620  Y= -4.000  Z=-2.510  Angle: 0.0
[1644196638.536690722]: Address: 5, timestamp: 649964, 130, X=3.620  Y= -4.000  Z=-2.510  Angle: 0.0
[1644196638.666798738]: Address: 5, timestamp: 650094, 130, X=3.620  Y= -3.990  Z=-2.510  Angle: 0.0
[1644196638.796998630]: Address: 5, timestamp: 650223, 129, X=3.620  Y= -3.990  Z=-2.510  Angle: 0.0
[1644196638.925947274]: Address: 5, timestamp: 650353, 130, X=3.620  Y= -3.990  Z=-2.510  Angle: 0.0
[1644196638.926538731]: Stationary beacon: Address: 3, X=5.550  Y= -7.550  Z=0.000
```

Fig 11. Hedgehog coordinate data

Teaching the robot will store the x and y coordinate data. As shown in **Figure 12**, the yellow circle indicates the coordinates of the road map with identities 0-4, and the black circles indicate the coordinates of the destination map with identities 0_0 to 0_2 and 1_0 to 1_2. Coordinate data from the RX beacon is saved as text in notepad. The set of coordinates stored in this notepad will be used as a waypoint later.

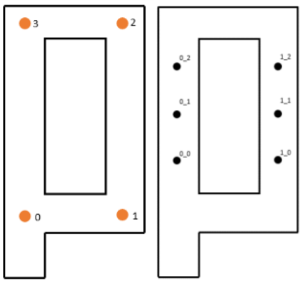


Fig 12. Set point teaching turtlebot3 burgers

After the coordinate values are stored in notepad, then call the previously saved file name. Enter the target set point. Users can input a maximum of 3 set points. The input set point is also set as a counter. If the input waypoint value entered is random, it will be sorted by line number first. The line on the left is counted as the first line with a value of 0. Then it is sorted by the destination number. If the robot is at the destination set point, the user will ask for confirmation. And so on until all missions are completed and the set point value = counter. The robot will return to its initial standing position by finding the parking station position towards the home base.

4 Testing and Analysis

4.1 Robot Navigation Testing

Based on the company's needs and the field conditions that have been created, the robot is asked to be able to walk to complete the mission in one direction using linear maneuvers forward, right, and left. **Figure 13** shows the turtlebot3 test field. The test was carried out at a constant speed of 0.14 m/s with enough light in the room and aimed to determine the difference in the level of accuracy of the robot's position on the field using indoor GPS and without indoor GPS. The test data is measured using a meter in centimeters and monitoring from a PC.

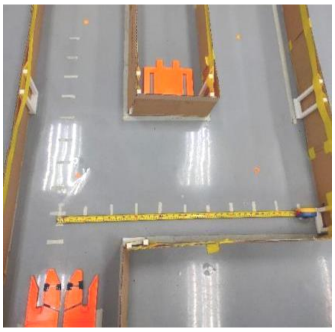


Fig 13. Turtlebot testing field

The monitored position value is compared with the actual position value to get an error value which is calculated using the RMSE (root mean squared error) formula, which is the evaluation of a linear regression model by measuring the accuracy of the estimated results of a data to measure the error of a model. The smaller the RMSE value, the better the model. The formula is as follows:

$$RMSE=\sqrt{\left[\sum (yi - \hat{y})\right]^{2}/n} \quad (3)$$

Description :

yi : Actual value
ŷ : Value of the monitor
n : Amount of data

Table 1. Forward Moving Robot Navigation Testing

No	Monitor Position (cm)		Actual Position With Indoor GPS (cm)		Actual Position Without indoor GPS (cm)	
	X	Y	X	Y	X	Y
1	0	10	0,1	10,8	0,5	10,5
2	0	20	0,2	20,7	0,8	20,9
3	0	30	0,3	30,5	1,0	31,6
4	0	40	0,1	40,3	1,0	41,8
5	0	50	0,1	50,4	1,3	50,1
6	0	60	0,5	60,5	1,2	61,9
7	0	70	0,3	70,2	1,3	71,8
8	0	80	0,3	80,5	1,6	82,1
9	0	90	0,1	90,3	1,6	93,8
10	0	100	0,5	100,5	1,9	103,6

The test results in table 1 are plotted into a graph as shown in **Figure 14.** The graph shows the results of the navigation test of the robot moving forward. While the actual testing of the robot looks like in **Figure 15** where the robot is tested in one position, namely X, Y (0, 100).

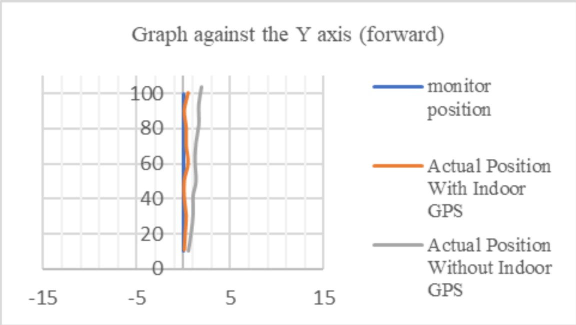


Fig 14. Forward moving robot navigation graphic

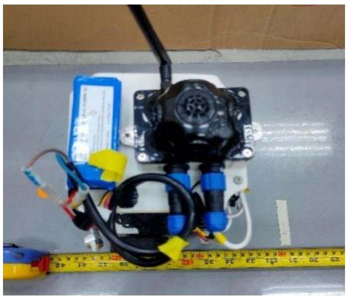


Fig 15. Robot position x,y (0, 100)

Based on the RMSE calculation, the navigation test of the forward-moving robot has an error value on the X-axis is :

with indoor GPS : $\sqrt{\frac{0,85}{10}} = \sqrt{0,085} = 0,292$ cm

without indoor GPS : $\sqrt{\frac{16,44}{10}} = \sqrt{1,644} = 1,282$ cm

The error value on the Y axis is :

with indoor GPS : $\sqrt{\frac{2,51}{10}} = \sqrt{0,251} = 0.501$ cm

without indoor GPS : $\sqrt{\frac{45,53}{10}} = \sqrt{4,533} = 2,129$ cm

Table 2. Right Moving Robot Navigation Testing

No	Monitor Position (cm)		Actual Position With Indoor GPS (cm)		Actual Position Without indoor GPS (cm)	
	X	Y	X	Y	X	Y
1	10	0	10,3	0,4	10,9	0,4
2	20	0	20,4	0,9	21,3	0,7
3	30	0	30,2	0,2	31,6	1,3
4	40	0	40,7	0,6	42,1	1,4
5	50	0	50,5	0,1	53,3	1,4
6	60	0	60,3	0,5	64,7	1,6
7	70	0	70,4	0,3	74,9	1,9
8	80	0	80,5	0,7	85,2	2,0
9	90	0	90,4	0,5	95,9	2,2
10	100	0	100,5	0,8	106,3	2,6

The test results in table 2 are plotted into a graph as shown in **Figure 16**. The graph shows the results of the navigation test of the robot moving to the right. While the actual testing of the robot looks like in **Figure 17** where the robot is tested in one position, namely X, Y (60, 0)

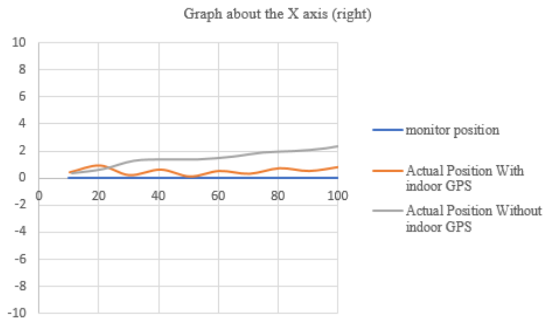


Fig 16. Right-moving robot navigation graphic

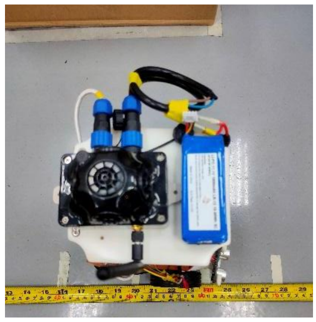


Fig 17. Robot position x,y (60, 0)

Based on the RMSE calculation, the right-hand moving robot navigation test has an error value on the X-axis is :

With indoor GPS : $\sqrt{\frac{1,94}{10}} = \sqrt{0,194} = 0,440$ cm

Without indoor GPS : $\sqrt{\frac{168}{10}} = \sqrt{1,68} = 1,296$ cm

The error value on the Y axis is :

With indoor GPS : $\sqrt{\frac{3,10}{10}} = \sqrt{0,310} = 0.557$ cm

Without indoor GPS : $\sqrt{\frac{28,03}{10}} = \sqrt{2,803} = 1,674$ cm

Table 3. Left Moving Robot Navigation Testing

No	Monitor Position (cm)		Actual Position With Indoor GPS (cm)		Actual Position Without indoor GPS (cm)	
	X	Y	X	Y	X	Y
1	-10	0	-10,4	0,2	-10,7	0,7
2	-20	0	-20,7	0,6	-21,4	0,8
3	-30	0	-30,5	0,4	-31,9	1,3
4	-40	0	-40,8	0,7	-42,7	1,5
5	-50	0	-50,6	0,5	-52,0	1,8
6	-60	0	-60,2	0,3	-63,5	1,8
7	-70	0	-70,9	0,6	-73,1	2,3
8	-80	0	-80,3	0,9	-83,8	2,7
9	-90	0	-90,4	0,4	-94,6	2,7
10	-100	0	-100,8	0,7	-105,0	2,7

The test results in table 3 are plotted into a graph as shown in **Figure 18**. The graph displays the results of the navigation test of the robot moving to the left. While the actual testing of the robot looks like in **Figure 19** where the robot is tested in one position, namely x,y (-40, 0).

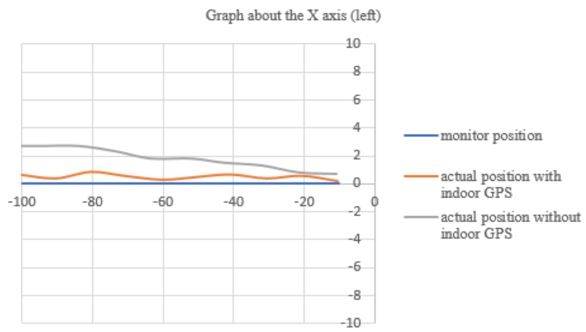


Fig 18. Left moving robot navigation graphic

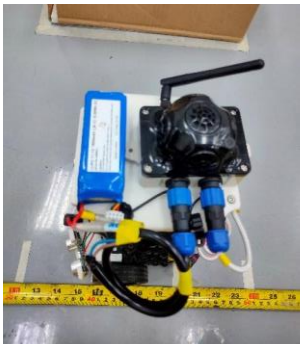


Fig 19. Robot position x,y (-40, 0)

Based on the RMSE calculation, the navigation test of the left-moving robot has an error value on the X-axis is :

with indoor GPS : $\sqrt{\frac{3,60}{10}} = \sqrt{0,360} = 0.603$ cm

without indoor GPS : $\sqrt{\frac{99,81}{10}} = \sqrt{9,981} = 3,159$ cm

The error value on the Y axis is :

With indoor GPS : $\sqrt{\frac{3,21}{10}} = \sqrt{0,321} = 0.567$ cm

Without indoor GPS : $\sqrt{\frac{38,71}{10}} = \sqrt{3,871} = 1,967$ cm

Based on the navigation test results, the robot maneuver using indoor GPS has a better position reading level with an accuracy of up to ± 1 cm and has a lower error rate than without using indoor GPS. The error value on the Y axis is greater than the X axis. This is because the maneuver testing carried out is in the form of linear motion so that the error obtained is the sum of the previous error (derivative error).

4.2 Robot Set Point Testing

Based on the needs of the company, the robot is asked to complete several missions during the travel process, namely :

1. *Set Point*, the robot is asked to move to a specific destination coordinate point with the symbol "x1_x2" x1 depicting line no. and x2 depicting the sequence on a line
2. *Confirm*, the robot will stop & ask the user to confirm if the robot is already at the destination waypoint. The robot resumes its journey after the user confirms.
3. *Home Base*, after completing all the waypoints of the given destination, the robot will return to its home base position
4. *Obstacle Detection*, during its travel process, the robot will detect and avoid an existing obstacle.

The test is carried out by changing the position of the robot from the home base position to the set point position, continuing to the next point until the mission is completed. Then return to the home base position.

Table 4. Waypoint Testing With 1 Destination Points

No	Set Point	Confirm	To Home Base	Obstacle Detection
1	0_0	✓	✓	✓
2	0_1	✓	✓	✓
3	0_2	✓	✓	✓
4	1_0	✓	✓	✓
5	1_1	✓	✓	✓
6	1_2	✓	✓	✓

From testing one waypoint, all set points were achieved, and completed the mission well.

Table 5. Waypoint Testing With 2 Destination Points

No	Set Point	Confirm	To Home Base	Obstacle Detection
1	0_0 0_2	✓	✓	✓
2	0_2 0_1	✓	✓	✓
3	0_1 0_0	✓	✓	✓
4	0_0 0_1	✓	✓	✓
5	1_0 1_2	✓	✓	✓
6	1_2 1_1	✓	✓	✓
7	1_1 1_0	✓	✓	✓
8	1_0 1_1	✓	✓	✓
9	0_0 1_2	✓	✓	✓
10	0_2 1_1	✓	✓	✓
11	0_1 1_0	✓	✓	✓
12	1_0 0_1	✓	✓	✓
13	1_1 0_1	✓	✓	✓
14	1_1 0_0	✓	✓	✓

From testing two waypoints, all set points were achieved, and completed the mission well.

Table 6. Waypoint Testing With 3 Destination Points

No	Set Point	Confirm	To Home Base	Obstacle Detection
1	0_0 0_1 0_2	✓	✓	✓
2	0_0 0_2 0_1	✓	✓	✓
3	0_1 0_0 0_2	✓	✓	✓
4	0_1 0_2 0_0	✓	✓	✓
5	0_2 0_0 0_1	✓	✓	✓
6	0_2 0_1 0_0	✓	✓	✓
7	1_0 1_1 1_2	✓	✓	✓
8	1_0 1_2 1_1	✓	✓	✓
9	1_1 1_0 1_2	✓	✓	✓
10	1_1 1_2 1_0	✓	✓	✓
11	1_2 1_0 1_1	✓	✓	✓
12	1_2 1_1 1_0	✓	✓	✓
13	1_0 0_1 1_2	✓	✓	✓
14	1_0 0_2 1_1	✓	✓	✓
15	1_1 0_0 1_2	✓	✓	✓
16	1_1 0_2 1_0	✓	✓	✓
17	1_2 0_0 1_1	✓	✓	✓
18	1_2 0_1 1_0	✓	✓	✓
19	1_0 1_1 0_2	✓	✓	✓
20	1_0 1_2 0_1	✓	✓	✓
21	1_1 1_0 0_2	✓	✓	✓
22	1_1 1_2 0_0	✓	✓	✓
23	1_2 1_0 0_1	✓	✓	✓
24	1_2 1_1 0_0	✓	✓	✓
25	0_0 1_1 0_2	✓	✓	✓
26	0_0 1_2 0_1	✓	✓	✓
27	0_1 1_0 0_2	✓	✓	✓
28	0_1 1_2 0_0	✓	✓	✓
29	0_2 1_0 0_1	✓	✓	✓
30	0_2 1_1 0_0	✓	✓	✓

From testing 3 waypoints, all set points were achieved, and completed the mission well.

Based on the results of the set point test, the autonomous navigation system using indoor GPS with the waypoint method is able to regulate the motion of the autonomous mobile robot to reach the set point of the robot's destination.

5 Conclusion

From the results of research on the Autonomous Navigation System Using Indoor GPS on the Turtlebot3 Burger Robot, the robot can navigate more accurately using the indoor GPS where the position reading is ± 1 cm so that the robot can move to the destination set point correctly. Based on the RMSE calculation, the robot has a lower error rate than without using indoor GPS. The waypoint method can regulate the movement of the autonomous robot and display the movement of the robot in real-time when the robot walks to the specified destination point.

References

[1] R.Sridhar: Design and development of material behavior of line follower automated vehicle.Department of Mechanical Engineering, Vels Institute of Science, Technology and Advanced Studies (VISTAS). Chennai, Tamilnadu, India. (2020)

[2] Emmanual robotics. from https://emanual.robotis.com/docs/en/platform/turtlebot3/overview/ Accessed on October 1st 2021

[3] Ihsan Farrassalam Ammarprawira: Implementasi Automatic Waypoint untuk Return Trip pada Autonomous Robot dengan Titik Acuan Potensi Korban Bencana. Institut Teknologi Nasional Bandung, Indonesia. (2020)

[4] Juanda Rahimatullah: Rancang Bangun Autonomous Robot Tank dengan Metode Waypoint Berbasis Raspberry Pi. Jurusan Telekomunikasi Politeknik Angkatan Darat. (2020)

[5] Precise (±2cm) Indoor Positioning and Navigation for autonomous robots, drones, vehicles, and humans. from https://marvelmind.com/ Accessed on October 5th 2021

[6] ROS robot programming, TurtleBot3 Developers, pp. 235-240. (2017)

[7] Hendrik J. Djahi, dkk: Rancang Bangun Robot Mobil Dengan Sistem Navigasi Berbasis Odometry Menggunakan Rotary Encoder. Universitas Nusa Cendana. (2019)

[8] Ahmad Sulkhan Taufik: Sistem Navigasi Waypoint pada Autonomous Mobile Robot. Jurusan Teknik Elektro Universitas Brawijaya. (2013)

[9] C. K. M. Lee, dkk: A Bluetooth Location-based Indoor Positioning System for Asset Tracking in Warehouse. The Hong Kong Polytechnic University, Hong Kong, China. (2019)

[10] H. Heru Abrianto1, dkk: Sistem Informasi Pengecekan Dan Pemesanan Ruangan Jarak Jauh Menggunakan Indoor Positioning System. Universitas Tama Jagakarsa. (2021)

[11] Muhammad Naufal Faris, dkk: Teknologi Pembelajaran Matematika Pembuktian Teorema Pythagoras Berbasis Visual. Universitas Negeri Malang. (2019)

[12] Rachmalia Vinda Kusuma, dkk: Proses Pemecahan Masalah Trigonometri Berdasarkan Teori John Dewey Ditinjau dari Gaya Kognitif. Universitas Negeri Malang. (2022)