Jurnal Integrasi
Vol. 16 No. 1, April 2024, 11-20
e-ISSN: 2548-9828

Article History
Received January 2024
Accepted April 2024

Simulation of Mobile Robot Navigation System using Hector SLAM on ROS

Hendawan Soebhakti*, Robbi Hermawansya Pangantar*

Politeknik Negeri Batam
Electrical Engineering Program Study
Jl. Ahmad Yani, Batam Centre, Batam 29461, Indonesia
E-mail: hendawan@polibatam.ac.id, robbihermawansya1@gmail.com

Abstract

The ability to move from one point to the destination point autonomously is very necessary in AMR robots, to be able to meet this, the robot must be able to detect the surrounding environment and know its location to the environment, the Hector SLAM algorithm is added using the LIDAR sensor, and to find out the ability of the LIDAR sensor with the Hector SLAM and computer specifications in order to process properly, a simulation of the HECTOR SLAM with the LIDAR sensor was made. Simulation is carried out by creating an environment map on the Gazebo. Then explore environmental mapping using Hokuyo LIDAR which has been added to the turtlebot3 model waffle_pi to the simulated environment map. In this study, a model of the second floor lobby environment and Brail of the Batam State Polytechnic was used which was made in the form of a simulation on the Gazebo, where robots that have used LIDAR will be controlled with a keyboard around the simulation environment, where simultaneously the mapping and localization process runs and the process can be seen on the Rviz in real-time, where LIDAR will send data in the form of distance readings that will be received by Hector SLAM. The results of this study are expected that Hector SLAM using LIDAR sensor simulation can produce environmental mapping and localization in the simulation environment and obtain a minimum computer specification to process data from the SLAM Hector process using LIDAR sensors.

Keywords: Autonomous, AMR Robots, Gazebo, Hector SLAM, LIDAR Sensor, Rviz, Turtlebot3.

1. Introduction

Autonomous Mobile Robot (AMR) is a mobile robot that can run on an unspecified path, one of the benefits of AMR robots is to move objects or goods from one starting point to the destination point. Another purpose of making this robot car is to reduce the number of work accidents caused by worker negligence. The ability to move from the starting point to the destination point with autonomous is very necessary in the AMR robot, to be able to meet this, the robot must be able to detect the surrounding environment and know its location in the environment.

To detect the environment and location, the AMR robot uses the Simultaneous Localization and Mapping algorithm by means of Hector SLAM and uses LIDAR Hokuyo UTM-30lx, to support SLAM LIDAR testing on AMR robots and to find out the capabilities of LIDAR sensors with Hector SLAM and computer specifications in order to process properly, created a simulation tool mapping unknown locations using a modified turtlebot3 waffle_pi simulation model using the Hokuyo UTM-30lx LIDAR sensor on ROS. This simulation is carried out in a closed space to map the environment at an unknown location with the SLAM Hector algorithm on Gazebo and Rviz. Based on the information already explained, this study aims to produce mapping at unknown locations and the position of the robot against environmental maps.

Simultaneous Localization and Mapping (SLAM) is a method that can be applied to autonomous mobile robots, where this method can help the robot to create a map and at the same time indicate its position on the map created [1]. There are several ways to apply the SLAM algorithm, one of which is Hector SLAM [2]. SLAM hector is an SLAM method that processes mapping & localization without using odometry data. Hector SLAM generates a 2D grid map based on the data transmitted by the LIDAR sensor [3].

To help use the Hector SLAM method, a sensor data processing system is needed that is used to control AMR, namely the Robot Operating System (ROS). ROS is a middleware that can connect robot hardware with software on the robot [4]. ROS has many packages that can help run the SLAM Method [5]. One of the purposes of using ROS is to make it easier for robot developers to create the desired robot system without coding from scratch and can develop source code together [6].

2. Theoretical Basis

2.1 SLAM

The SLAM (Simultaneous Localization and Mapping) method allows robots to map [7-9]. SLAM can also be used for real-time localization and building maps in completely unknown environments, and mobile robots can ensure the position of the robot against the environment being explored, autonomous Mobile Robot is one of the most common robots, where the robot can autonomously solve "where I am" which is a prerequisite for automatic navigation and other functions [10-12]. Figure 1. rqt_tf_tree from Hector SLAM.

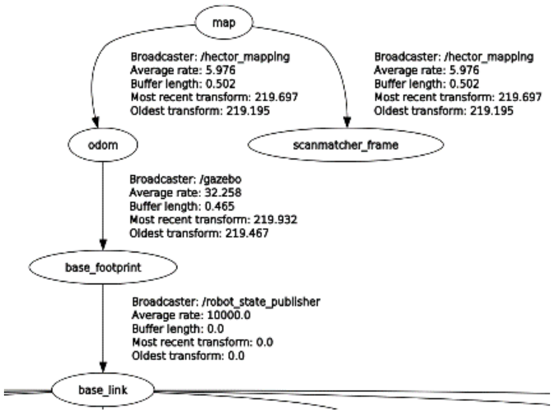


Figure 1: rqt_tf_tree from Hector SLAM

2.2 Hector SLAM

In this study, the Hector SLAM algorithm became the process of making environmental maps. mapping using high-frequency lidar lasers without odometry information. Mapping Hector with a scan matching algorithm developed using aligning sensor data to create an environmental map. Figure 2 shows the mapping process of Hector, Geotiff Hector, and other forms of the SLAM Hector schema and its associated process flow [13,14].

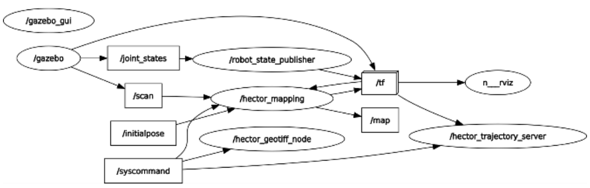


Figure 2: Rqt_graph from SLAM Hector

2.3 LIDAR

Light Detection and Ranging (LIDAR) works by reflecting the target with a laser beam and measuring the reflection with a sensor. Furthermore, the data is used to perform various tasks such as 2D mapping. Lidar's high resolution and accuracy have improved it for use in mapmaking. Also used in road, building, and vegetation mapping [15,16].

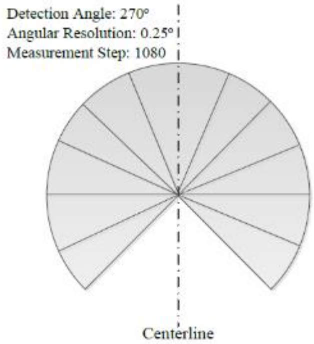


Figure 3: Hokuyo Range

2.4 ROS

Robot Operating System (ROS) one of the program frameworks is usually used in robotics. ROS functions to create software for robots that can run against other robots, where to create functions for robots that can be shared and reused against other robots by not having to recreate their functions from scratch to be suitable for the robots used. ROS provides standard operating system facilities, one of which is hardware abstraction, low-level robot control, execution of common functions used, sending commands between processes, and package management. Based on the architecture on ROS is based on a centralized topology where processing occurs on nodes that can capture inputs or output outputs, among them such as a multiplex sensor, actuator, control, and so on [17]. Robot Operating System (ROS) is not a real operating system, but a framework and set of tools that provide operating system functionality on heterogeneous computer clusters. Its usefulness is not limited to robots, but most of the tools provided are focused on working with peripheral hardware [18].

2.5 Gazebo

Gazebo is an open source simulation application, which can process the development of dynamic engines developed by the Open Source Robotics Foundation [19]. Gazebo simulates robot design, as well as trains artificial intelligent systems using simulation capabilities accurately, simulation design can be designed either indoors or outdoors complex [20]. In this final project, Gazebo is used as a simulation of designing real locations that you want to make in the form of simulations.

2.6 Rviz

Rviz is one of the 2D visualization software tools for robots, sensors, and algorithms. Used to see the perception of robots (real or simulated). The function of Rviz is to visualize the state of the robot on the Hector SLAM test. By using sensor data to try to make an accurate picture of what is happening in the robot's environment [21].

3. Research Methods

3.1 SLAM Implementation Design

To control the simulation system, an integrated ROS of several nodes is used. Laser scanner data that handles communication with the Hokuyo LIDAR laser scanner in the form of a simulation that allows to observe the simulation environment map, then the laser scan data is used Hector SLAM and localization visualization and mapping integrated in ROS, can be seen in Figure 4.

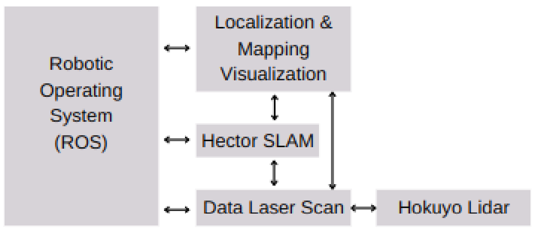
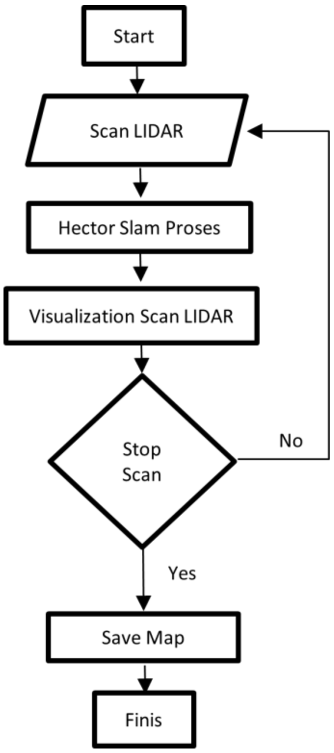


Figure 4: Block Diagram System of SLAM

3.2 Flowchart System Design



3.3 Simulations Design

The simulation is carried out using a gazebo to create a map design that you want to simulate. To produce a map, there are several SLAM simulation processes with Rviz as a visualization of localization and mapping and mapping results, can be seen in figure 5.

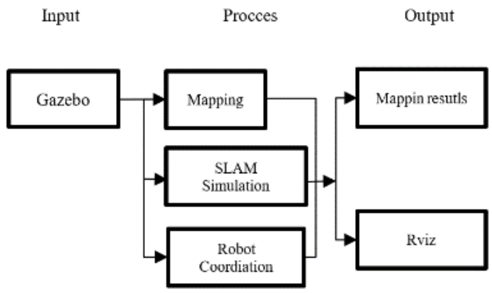


Figure 5: Simulations Design

Simulation using a turtlebot3 robot model waffle_pi, with the addition of LIDAR Hokuyo modeling, In the creation of an environmental map simulation model, using the size of the real environment map you want to simulate, can be seen in figure 6. To get a map from the simulated environment map, a mapping process is carried out by controlling the robot using a teleop key with a constant speed of 0.26 m/s so that the robot can trace the environmental map and the lidar sensor can map the environment which can be seen in real-time on the Rviz after the mapping process is completed, deviations from the environmental map in the form of pgm and tif are carried out. To test the saved map displayed again on Gazebo and Rviz, the test is carried out with a point top key by entering the destination x, y point. In the mapping results that have been carried out on SLAM LIDAR Hokuyo with direct tools and mapping SLAM LIDAR simulation, a comparison of mapping results is carried out.

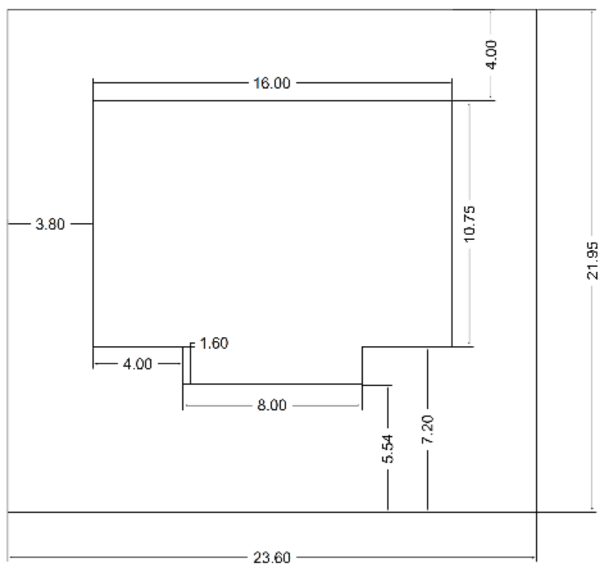


Figure 6: Lobby Environment Map (unit meter)

3.4 Hector_Mapping Parameters with LIDAR

In order for the environment scan to run optimally using Hokuyo LIDAR, the Hokuyo LIDAR parameter is created in Hector_Mapping as in table 1.

Table 1: Parameter Lidar Hokuyo

Parameter Laser	Details	Value
laser_min_dist	The minimum	0.2 [m]

	distance for the laser scan endpoint to be used by the system. Scan endpoints closer than this value is ignored.	
laser_max_dist	The maximum distance of the laser scan endpoint to be used by the system. Endpoint scans farther away from this value are ignored. Close to this value is ignored.	30.0 [m]
laser_z_min_value	The minimum height relative to the frame of the laser scanner for the laser scan endpoint to be used by the system. Scan endpoints lower than this value are ignored.	-1.0 [m]
laser_z_max_value	The maximum height relative to the frame of the laser scanner for the laser scanning endpoint to be used by the system. Scan endpoints higher than this value will be ignored.	1.0 [m]

4. Testing and Results

4.1 LIDAR Hokuyo Simulation Testing

In the LIDAR test, Hokuyo simulation has the aim of testing the accuracy value of LIDAR in reading the distance from objects or obstacles around LIDAR, where in this test using an equilateral rectangular environment size with a size of 5m x 5m and take values from each angle of 0°, 90°, -90°, 135°, and -135° against the distance of the detected obstacles can be seen in table 2. Figure 7 shows the range of the LIDAR laser.

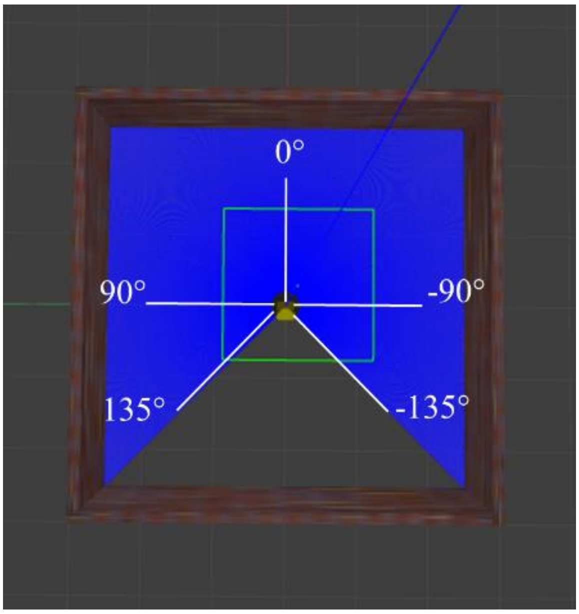


Figure 7: Displaying the Lidar Laser Range

Table 2: Data Testing of LIDAR Hokuyo Simulation

No	Corner	Measurement (m)	Measurable (m)	Error (%)	Accuracy (%)
1	0	2,50	2,51	0,40	99,60
2	90	2,50	2,51	0,40	99,60
3	-90	2,50	2,50	0,00	100,00
4	135	3,53	3,54	0,28	99,72
5	-135	3,53	3,54	0,28	99,72
Average				0,27	99,73

In the table above, the average error value from the simulated Hokuyo LIDAR sensor reading was 0.27%, and the average accuracy was 97.73% from the 5 angle samples taken.

4.2 Mapping Simulation Testing

In the simulation mapping test, an environmental map has been prepared on the Gazebo as shown in figure 8. In this simulation mapping test, it will produce a simulation environment map using the SLAM Hector algorithm, aiming to find out whether LIDAR with the Hector SLAM algorithm can map the environment, and environmental map results will be obtained in the form of pgn and tif.

Unit : Meters

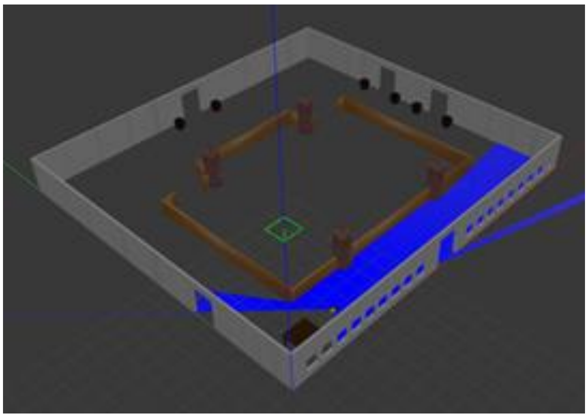


Figure 8: Map Environment Simulation Gazebo

The simulated environment map on the gazebo uses almost the same size as the size of the original environment map, which here uses the lobby environment of the main building on the 2nd floor. Figure 9 shows the placement of a simulation robot that has used Hokuyo's LIDAR on the simulation map.

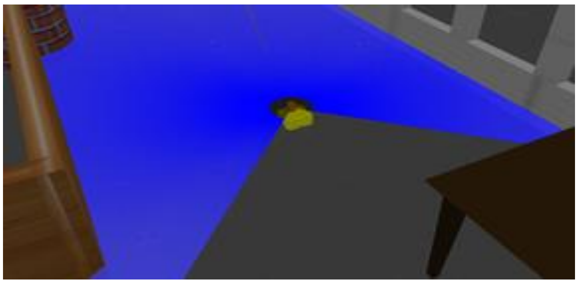


Figure 9: Robot Simulation on Environment Simulation

In the mapping results that have a pgm & tif format, the two mapping results are carried out in two different mapping processes, in figure 10 displays the mapping results from the simulation environment map that has been tested by Hector SLAM, obtained a length of 23.30 and a width of 21.73 in one of the tests from 5 tests with the pgm storage format. For the accuracy value of the simulation mapping can be seen in table 3.

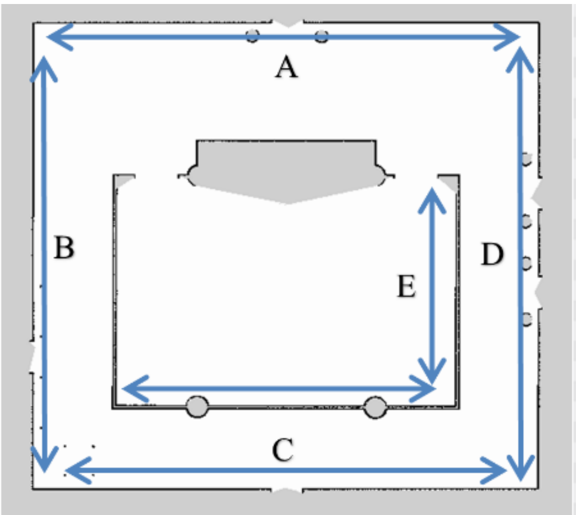


Figure 10: Mapping in Environment Lobby pgm Format

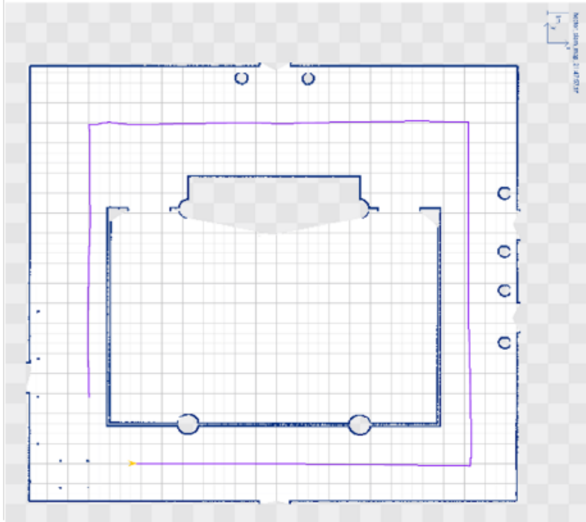


Figure 11: Mapping in Environment Lobby tif Format

Table 3: Accuracy Value of Mapping Hector SLAM on Lobby Map

Measurement (m)		Measurable (m)	Error (%)	Accuracy (%)
A	23,60	23,30	1,27	98,73
B	21,95	21,73	1,00	99,00
C	23,60	23,43	0,72	99,28
D	21,95	21,82	0,59	99,41
E	10,75	10,67	0,74	99,26
F	16,00	15,98	0,12	99,88
Average			0,74	99,26

In the table above, it is known that the average error mapping of Hector SLAM is 0.74%, and the average accuracy of Hector SLAM mapping is 99.26% of the 6 tests carried out. To find out whether the number of rooms will affect the mapping results of LIDAR Hokuyo with the Hector SLAM simulation algorithm, then next try to have a Brail environment map that has more rooms. Figure 12 shows the Brail environment folder.

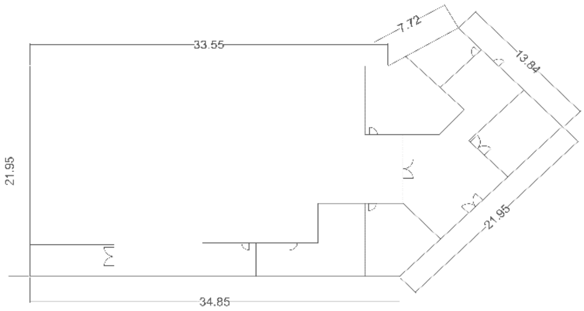


Figure 12: Map Environment of Brail

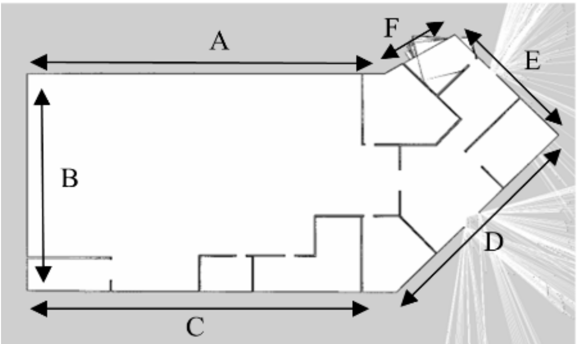


Figure 13: Mapping Result of Map Brail pgm Format

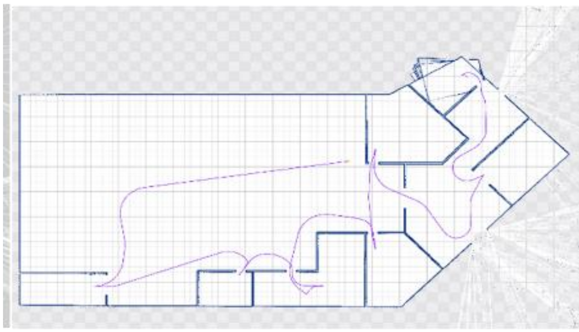


Figure 14: Mapping Result of Map Brail tif Format

Table 4: Accuracy Value of Mapping Hector SLAM on Map Brail

Measurement (m)		Measurable (m)	Error (%)	Accuracy (%)
A	33,55	33,40	0,45	99,55
B	21,95	21,55	1,82	98,18
C	34,85	34,75	0,29	99,71
D	21,95	21,45	2,28	97,72
E	13,84	13,77	0,51	99,49
F	7,72	7,55	2,20	97,80
Average			1,26	98,74

In the table above, it is known that the average error mapping of Hector SLAM on the barley environment map which is 1.26%, and the average accuracy of the Hector SLAM mapping is 98.74% of 6 tests carried out with a robot displacement speed of 0.26 m/s.

SLAM mapping simulation testing by increasing the robot's displacement speed, where using a speed of 0.5 m/s, an increase of 0.24 m/s from the initial speed causes a fairly severe overlapping and causes damage as a result of mapping, can be seen in figure 15. Therefore, a speed of 0.26 m/s is used to produce minimal overlapping mapping, where this overlapping will be very visible in Brail mapping, because the shape of the Brail map itself has a lot of space, so more obstacles and high displacement speeds can cause high overlapping errors. This condition runs on a devise computer with Intel core i3 Gen 6 processor specifications with 20gb of memory and we can see the memory and CPU usage in figure 16.

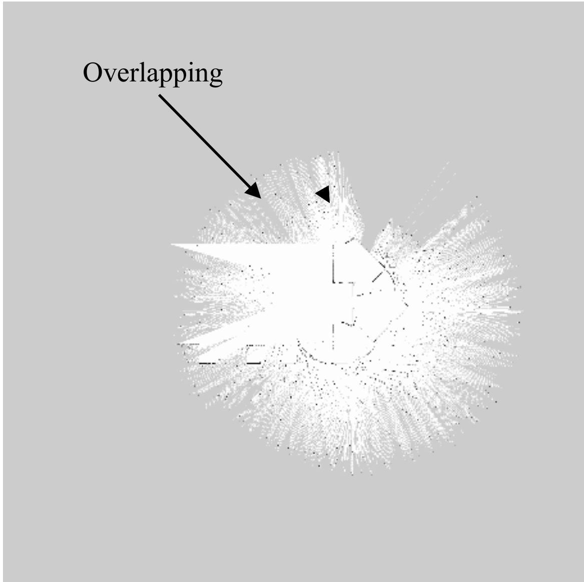


Figure 15: Overlapping Mapping Brail on Intel Core i3

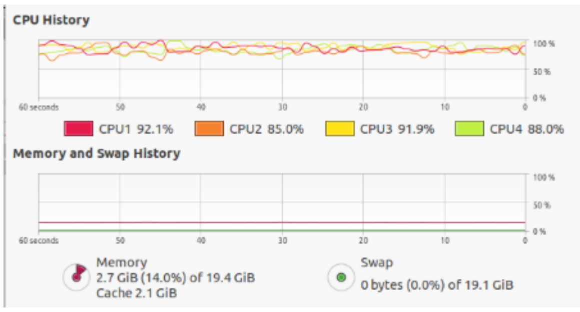


Figure 16: Usage Condition CPU on Intel Core i3

The test continued using a computer device with higher specifications than before using the same robot displacement speed of 0.5 m/s, with computer specifications, namely with an Intel core i5 Gen 6 processor with 8gb memory, then results were obtained with less overlapping conditions than the previous computer.

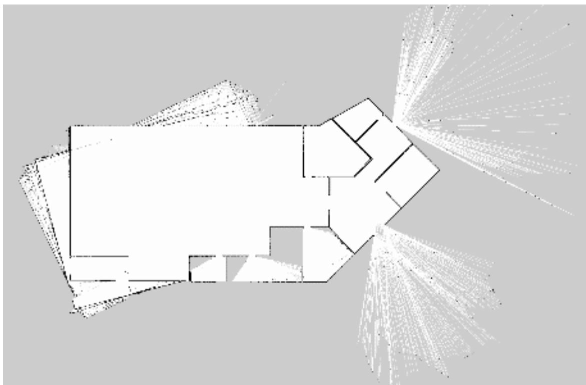


Figure 17: Overlapping Mapping Brail on intel Core i5

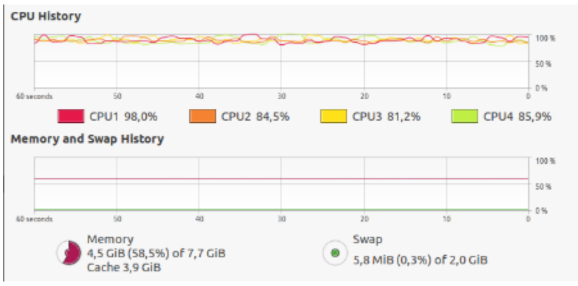
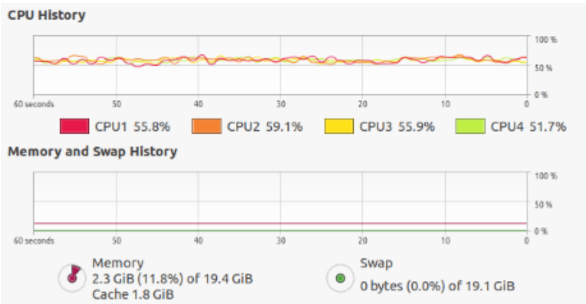


Figure 18: Usage Condition CPU on Intel Core i3

From the test results on two computer devices that have core i3 and i5 processor specifications, different mapping results were obtained, from the CPU usage on each computer to the same average usage where the CPU usage in the mapping process is quite high compared to memory usage, it can be concluded that if you want to increase the speed of robot movement in the process of mapping this requires a CPU with high specifications with a minimum of 8gb memory, then we can see that the higher the CPU specifications on a computer will result in minimal overlapping mapping at a robot displacement speed of 0.5 m/s. To strengthen the analyzes of CPU usage, the test was carried out by looking at CPU usage when the simulation was run with a non-active lidar condition, an active lidar condition, and an active lidar and hector slam condition, where the increase in CPU usage increased but memory usage only increased by 2% of the simulation conditions where lidar and hector slam were inactive to the active state, can be seen in Figure 19-21.

Figure 19: Usage Condition CPU on Intel Core i3 When Lidar



Off

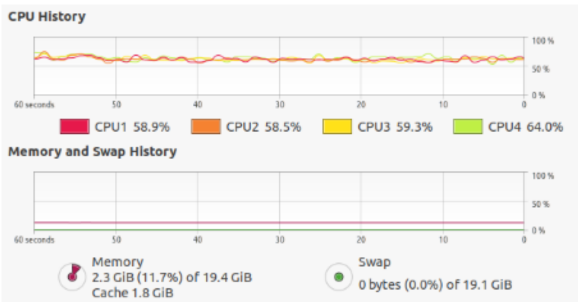
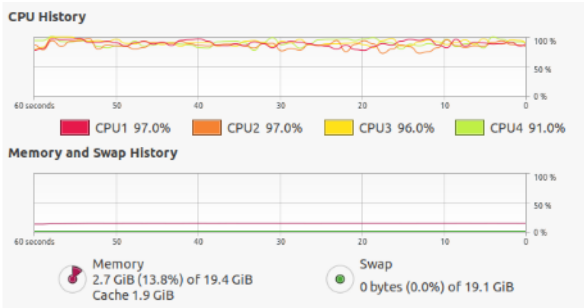


Figure 20: Usage Condition CPU on Intel Core I3 When Lidar Active

Figure 21: Usage Condition CPU on Intel Core I# When Lidar



and Hector SLAM Active

4.3 Simulation Localization Testing

In testing SLAM localization using the Hector SLAM algorithm, it aims to find out the accuracy of the points obtained from testing the Hector SLAM algorithm by moving the robot from one point to another, 2D visualization of localization is displayed on Rviz. localization with Hector SLAM utilizes one of the TF (Transform) packages, namely base_link. TF (Transform) is a package that can allow users to know some of the frame coordinates on the robot. TF base_link is a frame contained in the robot base that can display robot position data on a map. In this localization test, Hokuyo's LIDAR is placed exactly on the base_link where the lidar has a TF (Transform) base_scan.

In the localization test in the lobby environment, 7 robot location points were obtained, including the starting point (A), the first destination point (B), the second destination point (C), the third destination point (D), the fourth destination point (E), the fifth (F), and the last destination point (G). Figure 22 shows localization testing on Hector SLAM.

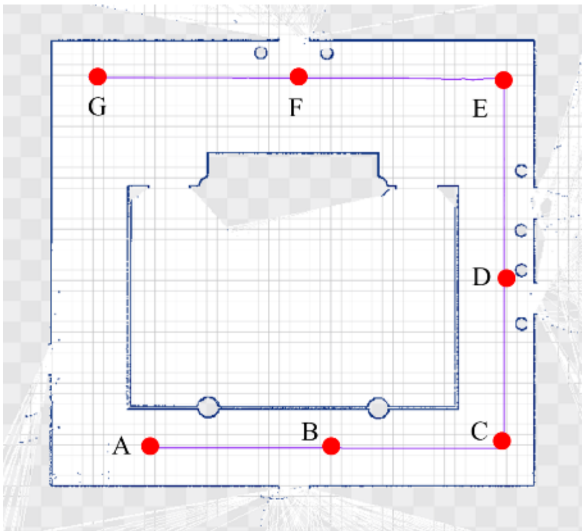


Figure 22: Testing localization Hector SLAM Map Lobby

Table 5: Testing localization Hector SLAM Map Lobby

Point	Position (m)		Position on Hector SLAM (m)		Error (%)		Accuracy (%)	
	X	Y	X	Y	P	X	Y	X
A	0,00	0,00	0,00	0,00	0,00	0,00	100	100
B	8,50	0,00	8,48	0,00	0,24	0,00	99,76	100
C	17,00	0,00	16,99	0,00	0,06	0,00	99,94	100
D	17,00	9,00	17,04	8,94	0,24	0,67	99,76	99,33
E	17,00	18,00	17,06	17,92	0,35	0,44	99,65	99,56
F	8,50	18,00	8,58	17,98	0,94	0,11	99,06	99,89
G	-3,00	18,00	-2,94	17,99	2,00	0,06	98,00	99,94
Average					0,36		99,64	

In the table above, it is known that the average error of Hector SLAM localization is 0.36%, and the average accuracy of Hector SLAM localization is 99.64% of the 7 test points carried out.

To find out whether the number of rooms will affect the localization results of LIDAR Hokuyo with the Hector SLAM simulation algorithm, then try to have a brail environment map that has more space. Figure 23 displays the results of localization on the Brail map.

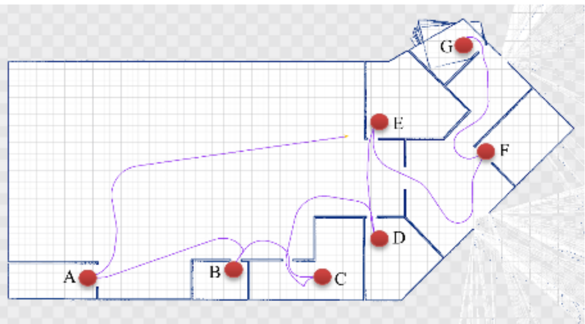


Figure 23: Result Displaying localization on Map Brail

Table 6: Testing localization Hector SLAM Map Brail

Point	Position (m)		Position Pada Hector SLAM (m)		Error (%)		Accuracy (%)	
	X	Y	X	Y	P	X	Y	X
A	-23,25	-12,50	-23,14	-12,61	0,47	0,88	99,53	99,12
B	-10,00	-11,75	-10,17	-11,77	1,70	0,17	98,30	99,83
C	-2,50	-13,00	-2,47	-13,06	1,20	0,46	98,80	99,54
D	2,25	-8,50	2,19	-8,52	2,67	0,24	97,33	99,76
E	2,25	1,25	2,23	1,22	0,89	2,40	99,11	97,60
F	12,10	-1,25	11,98	-1,26	0,99	0,80	99,01	99,20
G	10,50	8,50	10,24	8,31	2,48	2,24	97,52	97,76
Average					1,26		98,74	

In the table above, it is known that the average error of the localization of Hector SLAM on the brail environment map which is 1.26%, and the average accuracy of the localization of Hector SLAM is 98.74% of the 7 test points carried out.

In the position of the robot without any object or obstacle hitting the LIDAR sensor, the robot will not know the x and y points or error localization, therefore the robot must be around the object that hits the LIDAR sensor, at least there is an object that hits the sensor from LIDAR. The speed of the robot's movement in localization is the same as the speed of the robot's movement in mapping, because if there is an error in the mapping process, the localization will also be an error, because a lot of overlapping will affect the localization results at points x and y. Figure 24 robots are on an empty map. Figure 25 of the robot is given an object that hits the sensor from LIDAR.

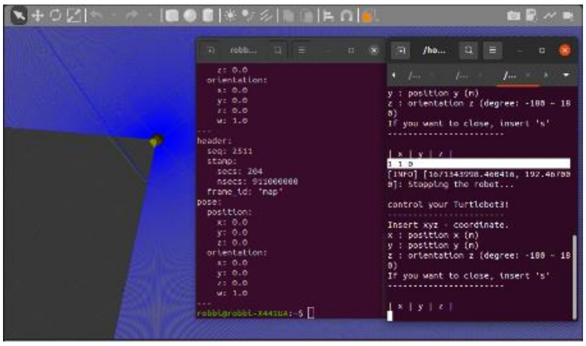


Figure 24: The Robot is on The Empty Map

Where x and y points slam_out_pose only read 0, while the robot has been moved to points x=1 and y=1 from points x=0 and y=0 on the x and y axes of the gazebo.

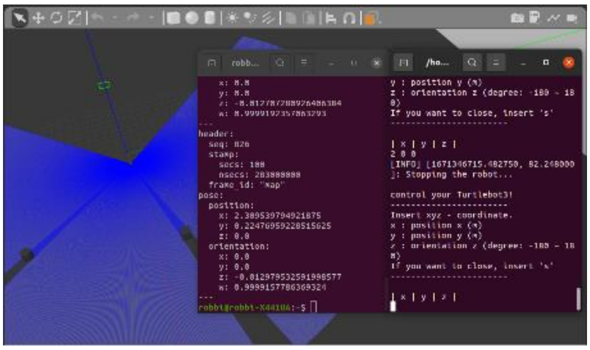


Figure 25: The Robot Given an Object That Hits the Sensor of the Lidar

When the sensor hits an object, the robot can read localization, where slam_out_pose produce x=2.3 and y=0.2 where the robot is moved from points x=0 and y=0 to points x=2 and y=0 on the x and y axes of the gazebo, this condition uses two objects or obstructions that hit the sensor, because if the sensor only hits one obstacle, the localization of the robot will be unstable and cause errors.

5. Conclusion

From the results of testing and analysis on SLAM simulation using LIDAR with the SLAM Hector algorithm against localization and mapping, it is concluded that the LIDAR simulation system can detect the surrounding environment with an error of 0.27% with the condition that the speed of robot displacement is a maximum of 0.26 m/s. SLAM Hector algorithm can map and store result maps with pgm and tif formats, From the mapping results, it can be concluded that the number of rooms in an environment and the speed of movement of the robot can affect the error value, where the more the number of rooms in the environment, the higher the error of the Hector SLAM mapping, and where the faster the movement from the robot, the higher the possibility of overlapping to occur, as well as the localization of Hector SLAM where in the lobby environment you get an error value of 0.36% where the lobby map has There is little space, while on the brail environment map, an error value of 1.26% is obtained where the brail map has a lot of space. So it can be concluded that the number of rooms/obstacles in an environment affects localization, and in order for Hector SLAM to be able to do localization, an object or obstacle that hits the sensor from LIDAR is needed, because if there is no object that hits the sensor from the lidar, Hector SLAM cannot know the location of the robot or do localization.

And it can be concluded that the specifications of the device used affect the results of the SLAM Hector process with LIDAR, where in this test the mapping results were obtained with a robot displacement speed of 0.5 m/s with a device that has a CPU core i5 Gen 6 specification with minimal overlapping compared to devices with core i3 Gen 6 CPU specifications. So to be able to achieve a high robot displacement speed must be done at high CPU specifications as well.

References

[1]. Prayoga, S. et al. 'Ekf slam menggunakan lidar', pp. 135–140, 2017.

[2]. Saat, S. et al. 'HECTORSLAM 2D MAPPING for SIMULTANEOUS LOCALIZATION and MAPPING (SLAM)', Journal of Physics: Conference Series, 1529, 2020. doi: 10.1088/1742-6596/1529/4/042032.

[3]. Putra, T. A., Muliady, M. and Setiadikarunia, D. (2020) 'Navigasi Indoor Berbasis Peta pada Robot Beroda dengan Platform Robot Operating System', Jetri: Jurnal Ilmiah Teknik Elektro, 17, p. 121, 2020. doi: 10.25105/jetri.v17i2.5447.

[4]. Jalil, A. 'Pemanfaatan Middleware Robot Operating System (Ros) Dalam Menjawab Tantangan Revolusi Industri 4.0', ILKOM Jurnal Ilmiah, 11(1), pp. 45–52, 2019. doi: 10.33096/ilkom.v11i1.412.45-52.

[5]. Sartika, E. M. et al. 'Implementasi Hector Slam Pada Robot Pencari Korban Gempa the Implementation of Hector Slam on the', pp. 383–391, 2017.

[6]. Jalil, A. 'robot operatyng system (ROS) dan gazebo sebagai media pembelajaran robot interaktif', LKOM Jurnal Ilmiah, 10, pp. 284–289. 2018.

[7]. Shahrizal Saat et al hector slam 2d mapping for simultaneous localization and mapping (SLAM) 2020 J. Phys: Conf. Ser. 1529 042032 doi:10.1088/1742-6596/1529/4/042032.

[8]. [8]. Yassin Abdelrasoul, Abu Bakar Sayuti HM Saman, Patrick Sebastian. (2017). A Quantitative Study of Tuning ROS Gmapping Parameters and Their Effect on Performing Indoor 2D SLAM. IEEE.

[9]. Xiang Gao 1, Tao Zhang 1, "Loop Closure Detection for Visual SLAM Systems Using Deep Neural Networks," IEEE Sensors Journal. Vol.13, No.1, pp. 253-262, 2015.

[10]. Wei Chen, Ting Qu, Yimin Zhou, Kaijian Weng, Gang Wang and Guoqiang Fu, (2014). "Door recognition and deep learning algorithm for visual based robot navigation," IEEE International Conference on Control and Decision Conference (CCDC)2014.

[11]. Shen, Dong & Xu, Yuhang & Huang, Yakun. (2019). Research on 2D-SLAM of Indoor Mobile Robot based on Laser Radar. CACRE2019: Proceedings of the 2019 4th International Conference on Automation, Control and Robotics Engineering. 1-7. 10.1145/3351917.3351966.

[12]. L Yan, J C Dai, J X Tan et al. SLAM laser point cloud overall fine registration pose technology [J]. Journal of Surveying and Mapping, 2019, 48(03): 313-321.

[13]. Saat, S., Abd Rashid, W. N., Tumari, M. Z. M., & Saealal, M. S. (2020). HECTOR SLAM 2D MAPPING for SIMULTANEOUS LOCALIZATION and MAPPING (SLAM). Journal of Physics: Conference Series, 1529(4).

[14]. Bambang Gunawan Tanjung. "Purwarupa Autonomous Mobile Robot dengan Hoverboard dan Sensor RPLIDAR menggunakan Algoritme

Hector SLAM dan Navfn," e-ISSN: 2548-964X, Vol. 6, No. 7, Juli 2022, hlm. 3219-3227.

[15]. Nagla, S. (2020). 2D Hector SLAM of Indoor Mobile Robot using 2D Lidar. ICPECTS 2020 - IEEE 2nd International Conference on Power, Energy, Control and Transmission Systems, Proceedings, 18–21.

[16]. M. Pierzchała, P. Giguère, and R. Astrup, "Mapping forests using an unmanned ground vehicle with 3D LiDAR and graph-SLAM," Comput. Electron. Agric., vol. 145, pp. 217 – 225, Feb. 2018.

[17]. Zaikiy Ni Starry, Agung Nugroho Jati, Faisal Candrasyah Hasibuan, OPENSLAM SIMULATION ON ROS WITH TURTLEBOT3, e-Proceeding of Engineering: Vol.8, No.5, ISSN: 2355-9365, 2020.

[18]. Zheng, K. (2021). ROS Navigation Tuning Guide. In: Koubaa, A. (eds) Robot Operating System (ROS). Studies in Computational Intelligence, vol 962. Springer, Cham. https://doi.org/10.1007/978-3-030-75472-3_6

[19]. Addison Sears-Collins, "What is the Difference Between rviz and Gazebo? – Automatic Addison." 2020, [Online]. Available: https://automaticaddison.com/what-is-the-difference-between-rviz-and-gazebo/

[20]. Rivera, Z.B.; De Simone, M.C.; Guida, D. Unmanned Ground Vehicle Modelling in Gazebo/ROS-Based Environments. Machines 2019, 7, 42. https://doi.org/10.3390/machines7020042.

[21]. Rajesh Kannan Megalingam, Sarath S, Ravi Teja Chinta, Akhil Raj "ROS based Autonomous Indoor Navigation Simulation Using SLAM Algorithm.", International Journal of Pure and Applied Mathematics, Volume 118 No. 7 2018, 199-205, ISSN: 1314-3395.