Mapping, Localization and Navigation for an Assistive Mobile Robot in a Robot-Inclusive Space

Prabhu R. Naraharisetti[a], Michael A. Saliba[b] and Simon G. Fabri[c]
Faculty of Engineering, University of Malta, Msida, Malta

Keywords: Observability, Accessibility, Robot-Inclusive Space, Vision-Based SLAM, LiDAR-Based SLAM, GMapping, Hector SLAM, Mapping, Path Planning, ROS Navigation Stack.

Abstract: Over the years, the major advancements in the field of robotics have been enjoyed more by the mainstream population, e.g. in industrial and office settings, than by special groups of people such as the elderly or persons with impairments. Despite the advancement in various technological aspects such as artificial intelligence, robot mechanics, and sensors, domestic service robots are still far away from achieving autonomous functioning. One of the main reasons for this is the complex nature of the environment and the dynamic nature of the people living inside it. In our laboratory, we have started to address this issue with our minimal degrees of freedom MARIS robot, by upgrading it from a teleoperated robot to an autonomous robot that can operate in a robot-inclusive space that is purposely designed to adopt algorithms that are not very computationally intensive, and hardware architecture that is relatively simple. This paper discusses the implementation of suitable SLAM algorithms, to select the best method for mapping and localization of the MARIS robot in this robot-inclusive environment. The emphasis is on the development of low-complexity algorithms that can map the environment with lesser errors. The paper also discusses the 3D mapping, and the ROS based navigation stack implemented on the MARIS robot, using just a LiDAR, a Raspberry Pi processor, and DC motors with encoders as main hardware architecture, so as to keep low costs.

1 INTRODUCTION

In recent years, service robots have found their way into public environments, such as in hospitals to deliver food and medicines to patients in quarantine (e.g. UBTECH robot) (Seidita et al., 2021), or in high end restaurants to deliver food to the customers while attaining a competitive advantage over their rivals (e.g. SERVI robot (SERVI robot, 2023)). But perhaps a more pressing requirement for these service robots is in home settings, to serve the elderly and impaired as their population is increasing globally. Some companies have put their efforts to bring service robots into home spaces (Polaris-Market-Research, 2023), but these robots are not very effective in performing the daily household tasks to serve the elderly and impaired satisfactorily. A primary reason for this is the complex nature of the environment including the dynamic nature of the people living in it, that typically demand the use of highly complex robot technologies (Sosa et al., 2018). High end complex robots such as ASIMO (ASIMO robot, 2023) and SPOT (Boston Dynamics - Spot robot, 2023) may in principle be capable of performing certain demanding household tasks, but these robots would not be widely affordable. Thus, a minimally complex robot that still has the functionalities to perform daily household tasks, in particular to address the specific needs of the elderly, could be applied to operate in a robot friendly home environment called a robot-inclusive space (RIS) (Sosa et al., 2018). Prior to the present work, in the first part of our *Mobile Assistive Robot in an Inclusive Space* (MARIS) project, a survey was conducted to understand the common needs of the elderly and impaired (Aquilina et al., 2019). In subsequent work, ten representative tasks were extracted to encapsulate these needs, namely: 1) preparing or bringing

[a] https://orcid.org/0009-0007-0117-4888
[b] https://orcid.org/0000-0002-1985-3076
[c] https://orcid.org/0000-0002-6546-4362

Naraharisetti, P., Saliba, M. and Fabri, S.
Mapping, Localization and Navigation for an Assistive Mobile Robot in a Robot-Inclusive Space.
DOI: 10.5220/0012172900003543
In *Proceedings of the 20th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2023) - Volume 1*, pages 172-179
ISBN: 978-989-758-670-5; ISSN: 2184-2809

medication; 2) heating a meal in a microwave oven; 3) operating a telephone; 4) preparing a small snack; 5) getting items from a refrigerator or cupboard; 6) taking out the garbage; 7) preparing tea or coffee; 8) arranging vegetables for chopping and/or cooking; 9) drying and putting away dishes; and 10) setting and clearing a table (Naraharisetti et al., 2022). The RIS principles such as observability and accessibility will be considered in designing home environments that are suitable for mapping and navigation by the MARIS robot to perform the representative set of tasks. The objective of this research is to conduct a set of experiments to obtain quantitative evaluations that determine the robot's complexity and to illustrate the fact that by increasing the inclusiveness of the home environment, the robot complexity can be reduced. To perform these representative tasks, a robot platform should have the capability to move around the environment while avoiding obstacles and choosing the shortest routes to reach the target.

A first prototype of a mobile assistive robot (MARIS-I), that was intended to be used in a RIS, and that incorporates only teleoperated control, was introduced in (Aquilina et al., 2019). The present work discusses an autonomous system based upon the Simultaneous Localization and Mapping (SLAM) (Alsadik and Karam, 2021) approach implemented on the MARIS Omni directional three-wheeled robot base using a Light Detection and Ranging (LiDAR) sensor. The work emphasizes the implementation of autonomous robot functionalities to map the environment, to self-locate in that environment, and to navigate to a destination with minimal instructions using Robot Operating System (ROS)-based SLAM algorithms, and a ROS navigation stack that integrates with a camera or LiDAR.

The rest of this paper is organized as follows: Section 2 explains the general methodology adopted in this paper. Section 3 describes the LiDAR-based SLAM approaches implemented on the MARIS robot. Section 4 discusses the vision-based SLAM implementations on the MARIS robot. Section 5 describes the selection of the most suitable SLAM method. Section 6 describes the ROS-based navigation stack implemented on MARIS for autonomous navigation. Finally, section 7 summarizes the work and briefly discusses the ongoing work to improve the RIS and to implement pick and place functionalities to the MARIS robot.

2 GENERAL METHODOLOGY

Based on the RIS principles of observability and accessibility (Elara et al., 2018), some general features in a one-floor home environment were mandated a priori. These were: no very bright objects, no dark colour wall paints, no sharp edges on the furniture, no rough floor surfaces, a less clustered environment, non-slippery floors, no uneven surfaces, non-reflective floor surfaces, no rooms with heavy doors, and no glass or transparent environments.

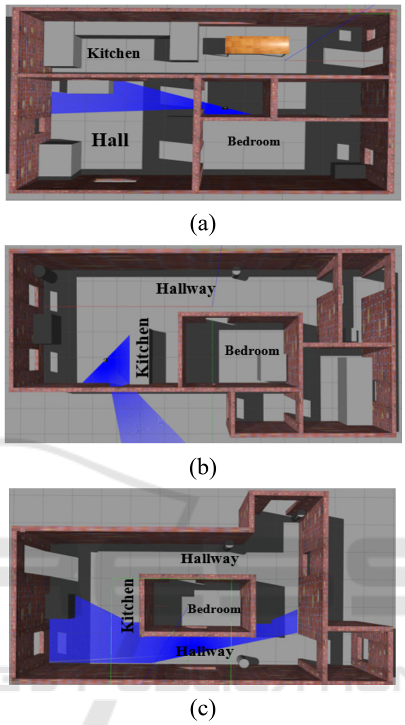


Figure 1: RIS environments (a) RIS 1 with more objects in the environment, (b) RIS 2 with a hallway, (c) RIS 3 with hallways on both sides of the bedroom.

For the LiDAR investigation, three different RIS environments were designed using Gazebo (Gazebo Building Editor, 2023) considering the general features with minor improvements from one another in terms of furniture placements, room location and hallway designs as shown in figure 1.

RIS 1 environment consists of an enclosed kitchen separated from the hall by a wall. So, eight of the ten representative sets of tasks require the robot to move from the kitchen to the hall. RIS 2 environment has a hallway with a kitchen in the hall that allows the robot to move with ease. The blue shaded regions in figure 1, represent the LiDAR rays that will be converted to 2D maps. Additionally, artificial landmarks such as familiar shapes in the environment help to map the environment precisely and help for better robot localization. RIS 3 environment has two hallways with a room at the center that allows the robot to move from any direction to find the optimal path for navigation. The LiDAR investigation was carried out firstly in the ROS-RVIZ simulation environment (Kam et al., 2015).

Both LiDAR and machine vision investigations were then carried out experimentally in the Robotic Systems Laboratory (RSL) of the Department of Mechanical Engineering of the University of Malta. In order to determine the least complex mapping suitable for our MARIS robot to perform the representative set of tasks, the approach shown in figure 2 was adopted, which will be further elaborated in the rest of the paper.

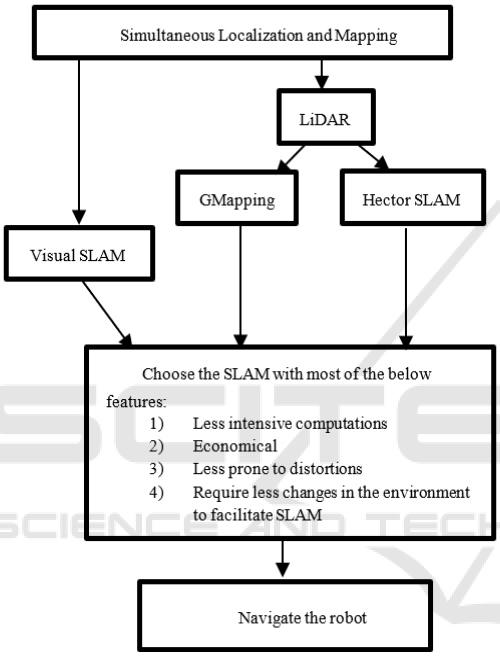


Figure 2: Approach to design a less complex SLAM to navigate the robot.

3 LiDAR-BASED SLAM APPROACH

3.1 Overview

In order to conform to our objectives to introduce a less complex and more economically accessible autonomous robot in the RIS home environments, we implemented LiDAR based SLAM using a relatively low cost 2-D RPLiDAR from SLAMTEC (SLAMTEC RPLiDAR, 2023). The SLAMTEC 2-D RPLiDAR emits and receives the reflected laser beam and measures the time the beam takes to return. This process is repeated for more than 8000 times per second, producing a map of the surroundings with a desirable point density. This method was investigated only after considering its quick and precise solution to create maps. The GMapping (Revanth et al., 2020) and Hector LiDAR-based SLAM (Sat t et al., 2020) algorithms were investigated and implemented on the MARIS Omni directional three-wheeled holonomic base, which is equipped with DC motors with encoders and control hardware that runs on the Arduino platform. To implement a portable high-level control system, a widely used ROS software framework was installed on a Raspberry Pi 4 single-board computer running the Ubuntu operating system. ROS-based SLAM techniques compatible with the SLAMTEC LiDAR software development kit provided a way to map the surroundings and localize the robot.

3.2 GMapping

GMapping is based on a Rao-Blackwellized Particle filer (RBPF) (Revanth et al., 2020; Sarkka et al., 2007) SLAM approach that is widely used for robot navigation, and uses a particle filter in which each particle carries an individual map of the environment. GMapping considers the movement of the robot and compares it with the recent environment data to decrease the inaccuracies of the robot pose. The parameters considered in the research are kernel size, linear update, resample threshold and the number of particles that combinedly determine the accuracy of the map. The inputs to the GMapping SLAM algorithm are the robot transform, laser data and odometry data that give the information of the robot pose, and generate the 2D occupancy grid or map that displays the obstacles and free spaces. The obtained map can be saved using the map server package of ROS to make the robot localize and navigate in the map. Initially, the GMapping SLAM algorithm was implemented. The laser inputs for GMapping SLAM were obtained from the SLAMTEC 2D LiDAR connected to Raspberry Pi. The odometry data that are used to estimate the robot position and orientation were obtained from the encoders of the three DC motors on the MARIS base, by determining the speed and distance travelled by MARIS.

3.3 Hector SLAM

Hector SLAM on the other hand uses a scan matching algorithm based on the Gauss-Newton approach (Cheng et al., 2021) that can be used without odometry data. The high update rate and accuracy of modern LiDAR hardware make the scan matching algorithm sufficient for a robot to achieve accurate poses. The scan matching algorithm matches the current scan to the previous scan to determine the robot movement. The parameters of the Hector SLAM including map size, map update distance threshold, map update angle threshold, laser minimum and maximum distance, can be modified to obtain a better map of the environment. The choice of these parameters makes the SLAM adaptable to the specific environment, however changing the parameters adds to the computational cost of the algorithm. The speed of the algorithm and frequency of data logging have an impact on map accuracies. The parameters under each SLAM algorithm mentioned before were modified uniformly until a better map was reached. This kind of adaptations helped us to determine the robot complexity and quantify the RIS environments in terms of inclusivity. Figure 3 shows the results of the GMapping and Hector SLAM algorithms implemented on the three RIS environments. For instance, under predefined SLAM algorithm parameters, higher complex environments will generate distorted and less accurate maps than lesser complex environments.

Environment	GMapping	Hector SLAM
Environment 1		
Environment 2		
Environment 3		

Figure 3: GMapping and Hector SLAM implemented on RIS environments 1,2,3.

The second and third columns of Table 1 compare the GMapping and Hector SLAM algorithms for the three RIS environments combinedly, based on their average values. For instance, the map error of GMapping for the three RIS environments 1, 2, and 3 are 2cm, 2.4cm, and 2.1cm respectively. Their average ((2+2.4+2.1)/(3) = 2.17cm) is taken as a measure to evaluate the accuracy of the SLAM algorithms. Other factors such as noise and features detected in the maps were assigned a value based on a widely followed 5-point Likert scale (Allen and Seaman, 2007).

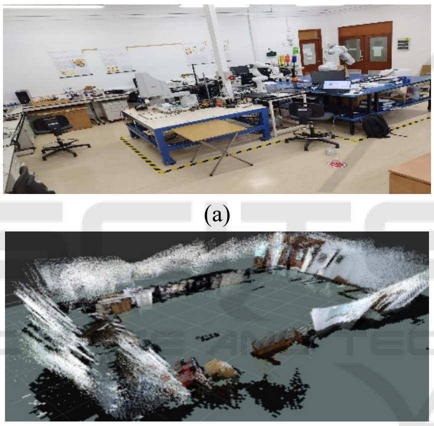
Table 1: Comparison of factors of GMapping, Hector, Vision and LiDAR fusion SLAM.

Factors	GMapping	Hector	Vision and LiDAR fusion mapping
Map accuracy (average across RIS 1, 2, 3)	2.17 cm error	2.3 cm error	0.5 cm error
Time to build a map	18 seconds	10 seconds	16 seconds
Noise of the environment	3 (Likert scale)	2 (Likert scale)	3 (Likert scale)
Number of mapping parameter changes	4	5	No changes, but under uncluttered and amiable lighting conditions
Computational load			
a) Memory in %	18	6	42
b) CPU load in %	89	16	98
Features detected in the map	3 (Likert scale)	4 (Likert scale)	5(Likert scale)

4 VISION-BASED SLAM IMPLEMENTATIONS

To move a robot autonomously to some desired location, the spatial representation of the environment should be known to the robot. The robot needs to have a sensor or sensors that save the data of the environment to enable robot localization. For the autonomous robot (MARIS-II), this mapping refers to the construction of the spatial environment to help the robot perceive its surroundings and localize itself, and to navigate accordingly. This SLAM process would, in our case, involve continuously fusing onboard sensor data from the LiDAR and/or the cameras and from wheel encoders. With the recent advancements in vision-based technologies, camera-based vision-SLAM is gaining importance because it provides 3D information of the environment. However, these systems are mostly used in an indoor environment as the camera range is limited and the machine vision is sensitive to variations in light (Debeunne and Vivet, 2020).

As the objective is to adopt a suitable algorithm with no compromise in performing the representative set of tasks or subtasks, we have also implemented and evaluated the vision-based SLAM. The vision-based SLAM uses an Intel RealSense 435i depth camera (Intel RealSense 435i depth camera, 2023) that runs on Ubuntu, an open source Linux distribution installed on a Raspberry Pi single board computer. The resulting pure 3D map of the environment of the laboratory (Figure 4 (a)), which is 9.15x5.73m in size is shown in figure 4 (b). The pure 3D map algorithm utilized only the camera to perceive the environment, but we will need a map that contains floor as well as static and dynamic obstacles for autonomous navigation. So, we have implemented vision and LiDAR fusion-based mapping on MARIS.

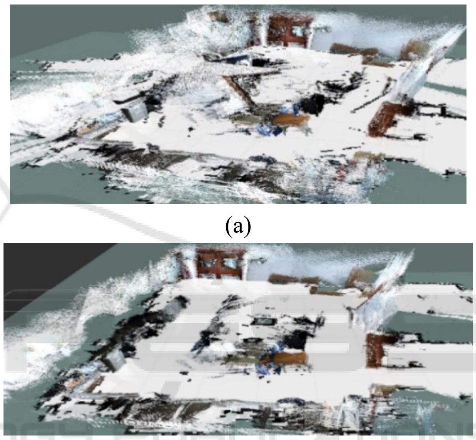


(a)

(b)

Figure 4: Image of the RSL (a) actual RSL (b) pure 3D map of RSL.

The pure 3D map is fused with the LiDAR data using the Real-Time Appearance-Based Mapping algorithm (Labbé and Michaud, 2019) to obtain the floor map thus giving a perception of the objects around as well as the obstacles in its path (Figure 5). When the RSL has a cluttered environment with objects spreading unevenly around, the vision system has a lot of noise as shown in figure 5(a). But after re-arranging the chairs and objects around properly the map shows minimal noise as in figure 5(b). The inbuilt Inertial Measurement Unit (IMU) (Ahmad et al., 2013) sensor in the RealSense camera also achieves the tasks of localizing the robot in an environment, but with the progress in the mapping, and the continuous localization of the robot in a complex environment, the usage of the central processing unit (CPU) of the robot will also increase, since feature complexity and processing time are correlated. The robustness of the CPU is questioned in this situation as real-time applications demand graphical processing units (GPU) rather than traditional processors. Since the Raspberry Pi 4, used in our research to connect to the RealSense camera, has a CPU that may not be sufficient to perform Realtime SLAM, external GPUs would have to be connected. Other ways to compensate for the load on the CPU is by using fewer, and similar, objects in the environment to facilitate their detection by the systems (Kamarudin et al., 2014). Furthermore, the challenge that limits the performance of Visual-based SLAM is due to the disruptions in the lighting conditions that introduce inaccuracies. The fourth column of Table 1 displays the factors of vision-based mapping.



(a)

(b)

Figure 5: Vision and LiDAR fusion mapping of the RSL: (a) in a cluttered RSL environment, (b) in a properly arranged RSL environment.

5 SELECTION OF THE SLAM METHOD

After studying the performance of the three SLAM algorithms as implemented separately on the MARIS robot, a selection needed to be made as indicated in Figure 2. Table 2 summarizes the main features of the three methods as extracted from Table 1. Feature 1, less intensive computations, refers to performing SLAM with minimally sophisticated processing and memory storage devices. Feature 2, economical in terms of price, refers to performing SLAM with comparatively cheaper equipment. Feature 3, less prone to distortions, refers to SLAM map with minimal unevenness. Feature 4, modification to the environment to facilitate SLAM (both LiDAR and Vision based) refers to efforts such as ensuring smooth

Table 2: Comparison of SLAM methods for the MARIS robot.

Desired Features	SLAM Algorithms		
	Vision based	GMapping	Hector
1. Less intensive computations	No	No	Yes
2. Economical in terms of price	No	Yes	Yes
3. Less prone to distortions	No	Yes	No
4. Require less changes in the environment to facilitate SLAM	No	Yes	Yes

surfaces and maintaining favourable illumination (Naraharisetti et al., 2022).

These features, including the computational load and other factors are shown in Table 2, based on which the Hector SLAM as investigated and tested on our MARIS robot will be used for autonomous navigation. Even though GMapping localizes the robot better than Hector SLAM, the latter is sufficient and proved to be reliable under restricted speed and angular velocities of the robot in RIS environments 1,2, and 3.

Hector SLAM based experiments were also performed in our RSL to evaluate the robustness of the algorithm (Figure 6). The environment of the lab with 8 office chairs, tables with almost 24 supports and wires under the tables resemble a cluttered environment. The experiments helped us to understand and tune our robot to navigate better in the RIS environments. The parameters that were tuned, shown in Table 3, were considered to adjust the robot complexity to serve in any RIS environments considering proper implementation of RIS principles.

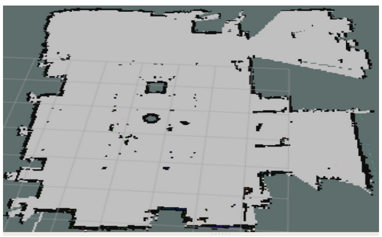


Figure 6: Hector SLAM map of our RSL under fixed SLAM parameters.

6 ROS-BASED ROBOT NAVIGATION

Robot navigation involves planning routes and moving the robot safely and conveniently through an economically beneficial route from one point to another. In order to implement autonomous navigation on the MARIS robot by integrating the LiDAR and other hardware architecture, an open source ROS navigation stack (Setup and configuration of Navigation stack, 2023) shown in figure 7 was chosen.

Table 3: Tuned mapping parameters of the robot after experiments.

Parameters	Reason	Values
Speed of the robot	Robot moving at high speed distorts the map	0.3m/s
Angular velocity	Higher velocity will distort map	1 rad/s
Map update distance threshold	The robot has to travel to have an angular change	0.2 m
Map update angle threshold	The robot has to have an angular change	0.2 rad

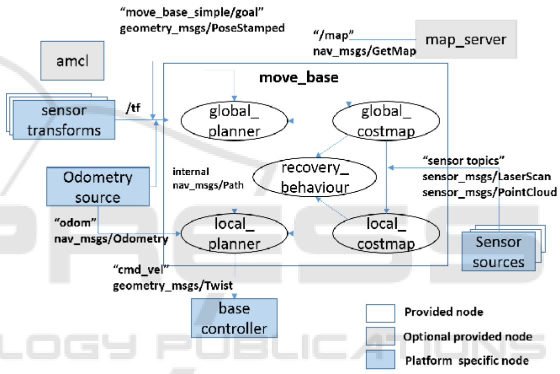
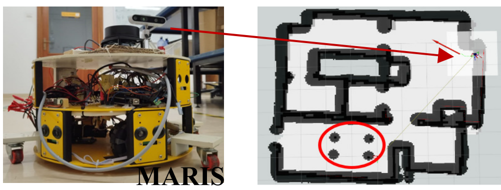


Figure 7: ROS Navigation stack architecture implemented on MARIS.

The ROS navigation stack requires the Adaptive Monte Carlo Localization (AMCL) ROS nodes to localize the robot moving in a 2D space, and AMCL takes the data from the laser scans to determine the pose of the robot (Matias et al., 2015). Sensor transforms refers to publishing the stream of LiDAR data over ROS, and the odometry information (Publishing Odometry information over ROS, 2023) can be published using the transform ("tf") (transform-tf, 2023), which tracks the coordinate frames and transforms points and vectors between the coordinate frames. The map server node (Map_server, 2023) of ROS gives the map data to the ROS move base package (Move_base package, 2023) that links global and local planners to perform the navigation task. The global cost map and the local cost map (setup and configuration of Navigation stack, 2023) store the information of the obstacles to create a long-term plan to navigate the robot without colliding with the obstacles. The base controller accepts the command velocity topic from the moving base, which gives the robot linear and angular velocities at that instant and converts them to individual wheel velocities.

The ROS navigation stack for a differential (two-wheeled) robot is supported with the required information on its website. However, our MARIS robot uses an omni directional three-wheeled architecture, which is not readily supported in the stack. So, changes in parameters such as the Odom model type from “differential” to “omni-corrected” in the AMCL launch file in ROS; changes in odom_alpha1,2,3,4 that determine the expected noise in the odometry rotation and translation estimate from rotational and translational components of robot motion; and the inclusion of a new odom_alpha5 that determines the translation-related noise parameters that are used to identify only the tendency of the robot to translate in a perpendicular direction of travel etc.,(AMCL parameters, 2023) were made so as to implement a new navigation stack for the omni directional three wheeled MARIS robot shown in figure 8 (a).

Finally, as Hector SLAM is selected instead of GMapping and vision-based SLAM, we have used the pose obtained from the Hector mapping and supplied the odometry message to the AMCL of ROS navigation stack implemented on the MARIS robot. This navigation is computationally low and accurate as the 2D-RP LiDAR is used to obtain laser scan matching. The navigation stack determines the presence of obstacles in the environment and avoids them. For instance, the four dark circles on the map labelled with a red circle shown in figure 8 (b) are the four supports of a table which, together with the navigation stack, supply the local and global map to the robot to move to the required destination.



(a) (b)

Figure 8: ROS based Navigation stack of MARIS, (a) hardware, (b) global map.

7 CONCLUSIONS

In order to perform the representative set of tasks in a one-floor home environment, the MARIS robot requires mobility features such as obstacle avoidance and collision free navigation. One of the objectives of the MARIS research project is to design a minimally complex robot that is able to perform the tasks in a robot-inclusive environment, and as such this research has involved conducting an experimental study to determine which SLAM algorithm to best adopt in the MARIS robot.

Initially, three RIS environments were considered to implement and evaluate two LiDAR-based SLAM algorithms, GMapping and Hector SLAM. The computational intensity and map quality were compared to select one SLAM that is most suitable for these RIS environments. Hector SLAM was found to be the more suitable.

Later a vision-based SLAM was also investigated and implemented, as this can generate a 3D perception or map of the environment. But with the increase in the quality of the map, the computational intensity also increases, which violates our objective of designing a minimal complex robot. So, Hector SLAM was the chosen algorithm. We thus tested the MARIS robot in our RSL environment to tune and establish the optimal Hector SLAM mapping parameters for our RIS environments. In this work, it has been shown that when the environment has been set up to accommodate a robot, i.e. as a RIS, then LiDAR-based Hector SLAM can match the functionality of the much more computationally demanding LiDAR-based GMapping or Visual-based SLAM methods. Based on the selected Hector SLAM method with tuned parameters, a new ROS based navigation stack was implemented for the MARIS robot, for further experimentation and development.

The mapping of the environment was found to have inaccuracies if the environment is featureless or has limited features, and there was also some divergence from the real map because of the accumulated inaccuracies. Thus, in a RIS environment that facilitates the use of the robot, there should be features that are recognizable. However, such features may create a cluttered environment that may further cause problems for the robot to navigate, and thus a compromise must be found between these two conflicting aspects. Ongoing work involves the development of a system to optimize the number of features that an environment should have to obtain a better mapping that enables the robot to navigate with fewer inaccuracies. The environment should obey basic RIS principles, such as observability, accessibility and manipulability.

For future developments, the MARIS robot is envisioned to also have autonomous pick and place capabilities using existing computer vision technologies that can detect and track the object. Since the MARIS robot is intended to serve the elderly and the impaired in performing household tasks, it is also intended to implement voice-based manipulation capabilities to provide the robot with wider functionality.

REFERENCES

Ahmad, N., Ghazilla, R. A. R., Khairi, N. M., & Kasi, V. (2013). Reviews on various inertial measurement unit (IMU) sensor applications. International Journal of Signal Processing Systems 1, no.2, pp. 256-262.

Allen, I. E., & Seaman, C. A. (2007). Likert scales and data analyses. Quality progress 40, no.7, pp. 64-65.

Alsadik, B., & Karam, S. (2021). The simultaneous localization and mapping (SLAM)-An overview. Surv. Geospat. Eng. J, 2, pp. 34-45.

AMCL parameters, (2023). [Online]. Available: http://wiki.ros.org/amcl.

Aquilina, Y., Saliba, M. A., & Fabri, S. G. (2019). Mobile Assistive Robot in an Inclusive Space: An Introduction to the MARIS Project. In Social Robotics: 11th International Conference, ICSR 2019, Proceedings 11 pp. 538-547.

ASIMO robot-The world's most advanced robot, (2023). [Online]. Available: https://asimo.honda.com/.

Boston Dynamics - Spot, (2023). [Online]. Available: https://www.bostondynamics.com/products/spot.

Debeunne, C., & Vivet, D. (2020). A review of visual-LiDAR fusion based simultaneous localization and mapping. Sensors 20, no.7, p. 2068.

Elara, M. R., Rojas, N., & Chua, A. (2014). Design principles for robot inclusive spaces: A case study with Roomba. In 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 5593-5599.

Gazebo Building Editor, (2023). [Online]. Available: https://www.classic.gazebosim.org/tutorials?cat=build_world&tut=building_editor.

Intel RealSense 435i depth camera, (2023). [Online]. Available: https://www.intelrealsense.com/depth-camera-d435i/.

Kamarudin, K., Mamduh, S. M., Shakaff, A. Y. M., & Zakaria, A. (2014). Performance analysis of the microsoft kinect sensor for 2D simultaneous localization and mapping (SLAM) techniques. Sensors 14, no.12, pp. 23365-23387.

Kam, H. R., Lee, S. H., Park, T., & Kim, C. H. (2015). Rviz: a toolkit for real domain data visualization. Telecommunication Systems 60, pp. 337-345.

Labbé, M., & Michaud, F. (2019). RTAB‐Map as an open‐source lidar and visual simultaneous localization and mapping library for large‐scale and long‐term online operation. Journal of field robotics, 36, no.2, pp. 416-446.

Map_server, (2023). [Online]. Available:http://wiki.ros.org/map_server.

Matias, L. P., Santos, T. C., Wolf, D. F., & Souza, J. R. (2015). Path planning and autonomous navigation using AMCL and AD. In 2015 12th Latin American Robotics Symposium and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR), pp. 320-324.

Move_base package, (2023). [Online]. Available: http://wiki.ros.org/move_base.

Naraharisetti, P. R., Saliba, M. A., & Fabri, S. G. (2022). Towards the Quantification of Robot-inclusiveness of a Space and the Implications on Robot Complexity. In 2022 8th International Conference on Automation, Robotics and Applications (ICARA), pp. 39-43

Polaris-Markert-Research. Household Robots Market Share, Size, Trends, Industry Analysis Report, (2023). [Online].Available:https://www.polarismarketresearch.com/industry-analysis/household-robots-market.

Publishing Odometry information over ROS, (2023). [Online].Available:http://wiki.ros.org/navigation/Tutorials/RobotSetup/Odom.

Revanth, C. M., Saravanakumar, D., Jegadeeshwaran, R., & Sakthivel, G. (2020, December). Simultaneous Localization and Mapping of Mobile Robot using GMapping Algorithm. In 2020 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS), pp. 56-60.

Saat, S., Abd Rashid, W. N., Tumari, M. Z. M., & Saealal, M. S. (2020). Hectorslam 2d mapping for simultaneous localization and mapping (slam). In Journal of Physics: Conference Series, Vol. 1529, No. 4, p. 042032.

Särkkä, S., Vehtari, A., & Lampinen, J. (2007). Rao-Blackwellized particle filter for multiple target tracking. Information Fusion 8, no.1, pp. 2-15.

Seidita, V., Lanza, F., Pipitone, A., & Chella, A. (2021). Robots as intelligent assistants to face COVID-19 pandemic. Briefings in Bioinformatics 22, no.2, pp.823-831.

Servi robot from bear robotics, (2023). [Online]. Available: https://www.bearrobotics.ai/restaurants.

Setup and Configuration of the Navigation Stack on a Robot, (2023). [Online]. Available: http://wiki.ros.org/navigation/Tutorials/RobotSetup.

SLAMTEC RpLiDAR, (2023). [Online]. Available: https://www.slamtec.com/en/LiDAR/A1.

Sosa, R., Montiel, M., Sandoval, E. B., & Mohan, R. E. (2018). Robot ergonomics: Towards human-centred and robot-inclusive design. In DS 92: Proceedings of the DESIGN 2018 15th International Design Conference, pp. 2323-2334.

"transform-"tf", (2023). [Online]. Available: http://wiki.ros.org/tf.

Wu, M., Cheng, C., & Shang, H. (2021). 2D LIDAR SLAM Based On Gauss-Newton. In 2021 International Conference on Networking Systems of AI (INSAI), pp. 90-94.

Wang, H., Yu, Y., & Yuan, Q. (2011, July). Application of Dijkstra algorithm in robot path-planning. In 2011 second international conference on mechanic automation and control engineering, pp. 1067-1069.

Yang, X. (2021). Slam and navigation of indoor robot based on ROS and lidar. In Journal of physics: conference series, Vol. 1748, No. 2, p. 022038.