Proceedings of of the ASME 2024
International Mechanical Engineering Congress and Exposition
IMECE2024
November 17-21, 2024, Portland, Oregon, USA

IMECE2024-144920

EFFICIENT AUTONOMOUS NAVIGATION FOR GPS-FREE MOBILE ROBOTS: A VFH-BASED APPROACH INTEGRATED WITH ROS-BASED SLAM

Amanuel Abrdo Tereda[1], **Sun Yi**[1], **Xingguang Li**[1]
[1]North Carolina Agricultural and Technical State University, North Carolina, USA
Department of Mechanical Engineering

ABSTRACT

Simultaneous Localization and Mapping (SLAM) is an autonomous localization technique used for mobile robots without GPS. Since autonomous localization relies on pre-existing maps, to use SLAM with the Robotic Operating System (ROS), a map of the surroundings must first be created, and a controller can then use the initial map. The first mapping procedure is mostly carried out manually, with human intervention. When operating manually, the person operating the robot is responsible for avoiding obstacles and moving the robot to different sections of the space to create a full map of the entire environment. The mapping process, if done manually, is time demanding, and often not feasible. To solve this constraint, which is to construct a map of the environment autonomously without human involvement while avoiding obstacles, the Vector Field Histogram (VFH) technique is implemented in this study by integrating it with SLAM. VFH is a real-time motion planning approach in robotics that uses a statistical representation of the robot's surroundings known as the histogram grid, to place a strong emphasis on handling modeling errors and sensor uncertainty. Furthermore, using range sensor values, the VFH algorithm determines a robot's obstacle-free driving directions. Aside from its real-time obstacle avoidance function, the VFH method is enhanced in this study to collaborate with SLAM to create maps and reduce localization complexity. While generating maps, the VFH approach uses a two-step data-reduction procedure to calculate the appropriate vehicle control directives. The robot's temporary location is used to generate a one-dimensional polar histogram, which is the first stage of the histogram grid reduction process. The polar obstacle density in a given direction is represented by a value in each sector of the polar histogram. In the second stage, the robot's steering is oriented in the direction of the most appropriate sector, which the algorithm determines from all the polar histogram sectors with a low polar obstacle density. Following that, further algorithms, such as Rapidly Exploring Random Tree (RRT) and A, can be used to plan autonomous pathways using the map provided by VFH. In order to put the concept into practice, MATLAB and ROS are used together in collaboration to autonomously and simultaneously map the environment and localize the robot. The combination of MATLAB and ROS provides many advantages because of their extensive feature set and ability to integrate with each other. Finally, a simulation and a real-time robot are utilized to analyze and validate the study's findings.*

Keywords: VFH (Vector Field Histogram); SLAM (Simultaneous Localization and Mapping); autonomous navigation; histogram grid, Polar Obstacle density; Masked Polar Histogram; obstacle avoidance; RRT

1 INTRODUCTION

These days, the field of robotics that deals with mobile robots is one of the fastest growing. Due to their ability to move independently, mobile robots can help humans in numerous industries. When a robot can decide how to carry out an action while using a perceptual system to help it, it is considered autonomous [1]. The fundamentals of autonomous mobile robots include locomotion, perception, cognition, and navigation [2]. Within the frame of reference, robot navigation is the ability

of the robot to identify its position and orientation. Navigating around its surroundings is essential for any mobile device since it helps prevent unpredictable events like collisions as well as hazardous situations. Determining the robot's current position and the planned location inside the same frame of reference or set of coordinates is an essential requirement for path planning, which is an extension of localization [3].

The primary application of the global positioning system (GPS) has been precise navigation since the invention of the atomic clock. After that, it developed into the best navigation system that can be used in every location on Earth, regardless of the time of day or the type of weather [4]. It has been noted that GPS guidance works well for mobile robots in several pertinent surrounding circumstances, such as agricultural location, formation movement or search, and item loading and unloading [5]. However, GPS is not always applicable, as there are numerous indoor applications where GPS is ineffective owing to signal limitations. In these cases, it is necessary to look for other navigation approaches.

Mobile robots need to be able to create a map of their environment and simultaneously localize within it in the absence of external referencing systems like GPS. Over the last twenty years, one of the most frequently studied problems in mobile robotics has been the simultaneous localization and mapping (SLAM) problem. Several efficient methods have been developed to address this problem [6]. In robotics, SLAM is a technique for determining the 3D structure of an unknown environment as well as sensor movements inside that environment. This technique was initially presented to achieve autonomous control of robots [7].

Even though SLAM is extremely useful for autonomous navigation, it is dependent on pre-existing maps. To use SLAM with the Robotic Operating System (ROS), a map of the surroundings is first constructed, and then a controller captures the original map. The first mapping process is mostly carried out manually with human intervention, and subsequently, it will become suitable for autonomous navigation. When controlling the robot manually, the operator is responsible for avoiding obstacles and moving the robot to various portions of the space in order to create a complete map of the entire environment. To overcome this limitation, which is to shift the map-making process from human involvement to a point where the robot can construct the map on its own, a Vector Field Histogram (VFH) technique of autonomous navigation is integrated with SLAM in this study.

The VFH is a real-time motion planning system in robotics that was first presented by Koren and Borenstein in 1991 [8]. Since the VFH makes use of the so-called histogram grid to provide a statistical representation of the robot's surroundings, it is imperative that modeling mistakes and sensor uncertainty be addressed [9]. In contrast to alternative obstacle avoidance algorithms, VFH considers the robot's dynamics and structure and provides platform-specific steering directives [10]. Robustness, insensitivity to misreading, and computational efficiency were the main goals of developing the Vector Field Histogram. When navigating heavily crowded obstacle courses, the VFH algorithm has shown to be reliable as well as fast in practice [11].

The VFH algorithm was also suggested by Pappas et al., as a solution to the problem of safely and effectively navigating remotely operated robots in uncontrolled and hazardous situations. The VFH+ obstacle avoidance navigation module's commands are combined with teleoperation orders sent by an operator via a joypad to create a shared control technique [12].

Chen et al. developed a technique that adheres to the fundamental ideas of the VFH algorithm and employs a global path point produced by the A* algorithm as a temporary goal point. To control the movement of the omnidirectional mobile robot to the target place, a speed control method is also provided, based on the optimal obstacle avoidance direction obtained in the suggested methodology [13].

Lluvia et al. surveyed to compile an overview of various active mapping and robot exploration techniques, focusing on the major advancements made in the field of indoor mobility robotics. The main concepts of the study center on actively computing trajectories to investigate the region and provide an error-free map. SLAM approaches tackle the challenge of creating an environment map using one or more sensors in the methods that are covered [14].

For differential-drive mobile robots, Ghamri et al. presented a navigation system that combines obstacle avoidance and trajectory monitoring. Sonar range sensors are used for obstacle detection, and kinematic control and the VFH approach are used for navigation. In their paper, they provide an example of a robot that tracked a trajectory while dodging obstacles in a simulated environment [15].

Kiat Tee et al. deployed ROS-based SLAM libraries on an experimental mobile robot equipped with a 2D LIDAR module, an IMU, and wheel encoders to examine and compare common 2D SLAM methods in an indoor static environment. Overall, the study outlines the benefits and drawbacks of SLAM algorithms and illustrates the differences through created maps, since it is crucial to choose the best system for the intended use and pinpoint prospective avenues for further optimization [16].

The publications reviewed in this paper, as well as the majority of the studies in this domain, focus on using a previously constructed map to apply SLAM algorithms, whereas other works that use VFH do not integrate VFH's obstacle avoidance advantage with autonomous mapping. Thus, combining the benefits of VFH and SLAM is advantageous for indoor mobile robot applications, and this study addresses this approach. In the proposed method, a map is built autonomously using VFH and the SLAM algorithm, and the resulting map is used for subsequent navigation utilizing additional path planning methodologies such as RRT and A*.

The methodology used to conduct the research as well as the

procedures followed to carry it out are discussed in Section 2 of the paper. Section 3 explains the results of the study and the final section summarizes the entire study and proposes prospective areas for further research.

2 METHODOLOGY

2.1 VFH algorithm

The VFH algorithm generates obstacle-free steering directions for a robot using range sensor readings. A two-dimensional Cartesian histogram grid is used as the world model to accomplish its objective. Range data collected by onboard range sensors is regularly updated in this world model. The appropriate control commands for the vehicle are then calculated by the VFH approach using a two-stage data-reduction procedure. A one-dimensional polar histogram is created around the robot's transient location in the initial stage of the histogram grid reduction process. In the polar histogram, the polar obstacle density in a given direction is represented by a value in each sector. The robot is then steered in the direction of the most appropriate sector, which the algorithm has determined to be among all polar histogram sectors with a low polar obstacle density [8].

To compute steering directions, information regarding the robot's size and driving capabilities should be provided. The VFH algorithm requires four input parameters from the robot: Robot Radius, Safety Distance, Minimum Turning Radius, and Distance Limits. Robot Radius defines the smallest circle's radius that may include every component of the robot. By setting this radius, the robot is guaranteed to avoid obstacles according to its size. The term Safety Distance refers to an extra space beyond the robot's radius. When navigating an area, this feature is employed to increase safety. The robot moving at the intended speed has an appropriate turning radius, which is specified by the Minimum Turning Radius. At high speeds, the robot might not be able to make sudden turns. This characteristic allows it to avoid obstacles and provides sufficient space for movement. The distance range that must be taken into account to avoid obstacles is specified by distance limits.

A two-element vector with a lower limit and an upper limit must have its bounds specified. Sensor noise, short-range sensor errors, and sensor intersections with robot components are all ignored by setting the lower limit. The upper limit is dependent on the application that has been defined or the effective range of the sensor by taking into account the obstacles in the whole sensor range. VFH treats each active cell's contents as an obstacle vector in the histogram grid, with the direction of each obstacle vector being defined by the direction 'β' from the cell to the Vehicle Center Point (VCP). The important VFH method parameters are displayed in Figs. 1 and 2.

Trigonometry is used to determine the direction of β from a cell to the VCP. The cell's coordinates and the VCP are required to compute the direction of beta. If the coordinates of the vehicle's center point are written as $(x\_vehicle, y\_vehicle)$, and the coordinates of the cell are written as (x_cell, y_cell), the direction of β can be computed using the equation:

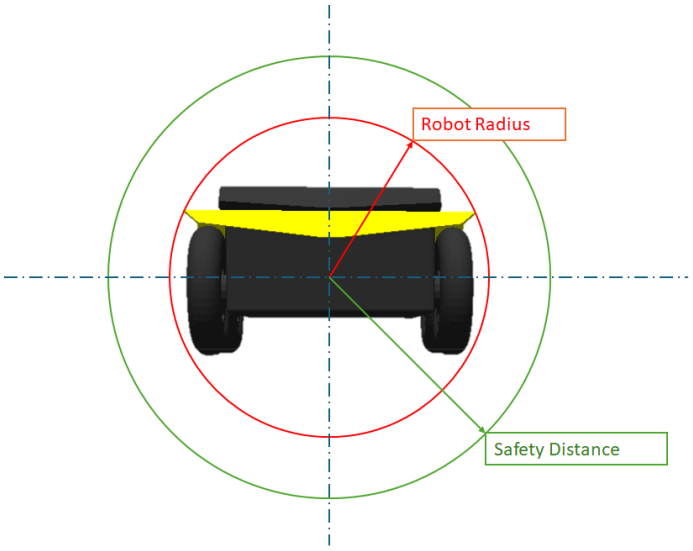

FIGURE 1. Robot Radius and Safety Distance.

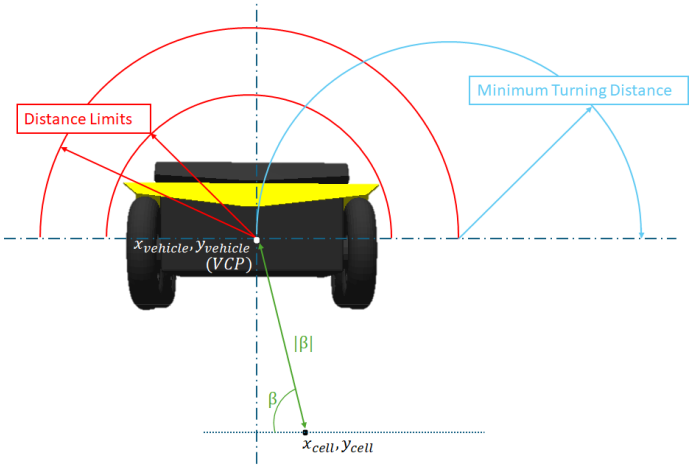

FIGURE 2. Distance Limits, Minimum Turning Distances, and VCP

$$\beta = tan^{-1}\left(\frac{y\_vehicle - y\_cell}{x\_vehicle - x\_cell}\right) \quad (1)$$

The distance between the cell of interest and the vehicle's center point is usually represented by the magnitude of β ($|\beta|$). It is frequently used to assess whether a cell can serve as a nav-

igational aid or an obstruction. The Euclidean distance formula, which determines the separation between two points in a two-dimensional space as indicated in the equation below, can be used to determine the magnitude of β ($|\beta|$).

$$|\beta| = \sqrt{(y\_vehicle - y\_cell) + (x\_vehicle - x\_cell)} \quad (2)$$

The distance between active cells and the VCP ($|\beta|$), the certainty value of active cells (C_{ij}), the current coordinates of the VCP (x_0, y_0), the coordinates of the active cell (i, j), and positive constants (a and b) can all be used to determine the magnitude of the obstacle vector (m_{ij}) at any cell. The value of m_{ij} can be computed as follows by utilizing these values:

$$m_{ij} = (C_{ij})^2(a - b|\beta|) \quad (3)$$

The positive constants 'a' and 'b' are essential in forming the potential field that directs the robot's motion. The attractive potential of the VFH algorithm is associated with the constant 'a' and it establishes the degree of attraction to the objective point. The increased value of 'a' intensifies the attraction force, propelling the robot in the direction of the objective point. Conversely, a lower value of 'a' produces a weaker attraction, which permits the robot to travel through more paths before arriving at the destination. The constant 'b' is associated with the repulsive potential of the VFH algorithm, and it regulates the degree of resistance to obstacles. The robot will avoid obstacles more insistently if 'b' is larger since it will have a stronger repelling force. On the other hand, if 'b' is smaller, the repulsive force is smaller, which would enable the robot to go closer to obstacles. Various sections of the histogram grid are shown in Fig. 3.

The behavior of the VFH algorithm can be modified to suit various navigation scenarios and robot capabilities by varying the values of 'a' and 'b'. In situations with few barriers and well-defined paths, higher values of 'a' and 'b' might be appropriate for safe and straightforward travel towards the goal. Lower values of 'a' and 'b' advance careful navigation in congested surroundings with small paths, allowing the robot to maneuver through confined regions without colliding. The structure of the VHS algorithm is given in Table 1.

2.2 Overall approach

The MATLAB and ROS platforms collaborate to create a map of the environment and navigate the robot autonomously. Because of their wide feature set, built-in toolboxes, and ability to connect, MATLAB and ROS give numerous benefits. MATLAB offers an environment for numerical calculation, data analysis, and visualization, whereas ROS provides a versatile frame-

TABLE 1. STRUCTURE OF THE VHS ALGORITHM

Algorithm 1: Autonomous Navigation Using VFH Algorithm

Input: *Lidar scan data*

Output: *Velocity commands*

1 ***Initialization***

- *Initialize a subscriber to receive LiDAR scan data*
- *Initialize a publisher to send velocity commands*

2 ***VFH Setup***

- ***Define the input parameters:***
  - *Distance Limits, Robot Radius, Minimum Turning Radius, Safety Distance*

3 ***Main Loop***

- ***while (time*** $<$ ***max***$_{time}$***)***
  - *Receive LiDAR scan data and convert it to a LiDAR Scan object*
  - *Compute steering direction based on scan data and target direction*
  - *Calculate desired linear and angular velocities based on steering direction:*
    - ***If*** *steering direction is valid:*
      - *Set desired velocity to constant linear velocity*
      - *Calculate angular velocity based on steering direction*
    - ***If*** *steering direction is invalid:*
      - *Stop robot*
      - *Set desired velocity to backward velocity*
      - *Set angular velocity to constant value*

4 ***Sending Velocity Commands***

- *Set linear and angular components of velocity message based on calculated desired velocities*
- *Send velocity message using publisher*

5 ***Stop the robot***

- *Send zero-velocity command to stop the robot after main loop*

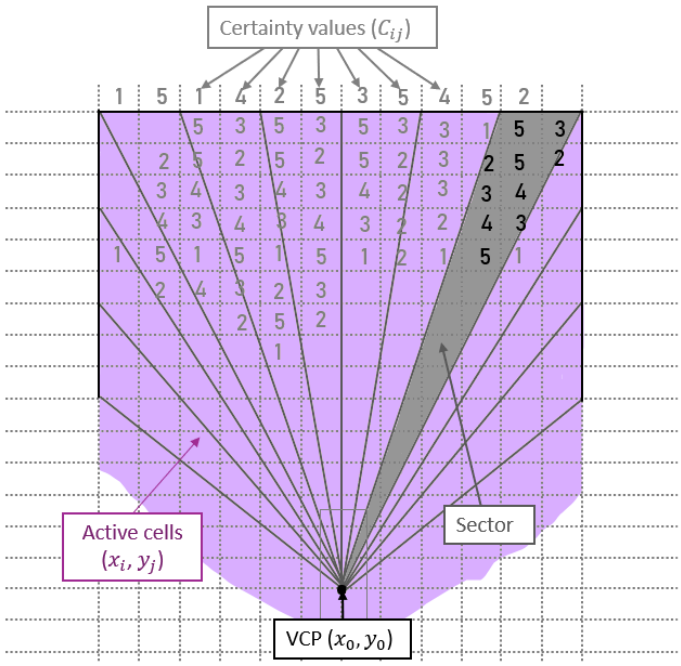

FIGURE 3. Histogram grid of the VHS algorithm

work for robot control, simulation, and communication. Furthermore, MATLAB's Robotics System Toolbox and ROS support package allow for smooth communication between MATLAB and ROS, allowing users to integrate the benefits of both platforms in a single workflow and making it simple to prototype, test, and deploy robotic systems.

The process begins by connecting MATLAB to ROS. There are several ways to accomplish this, including using a Windows or Mac operating system and running ROS from an Ubuntu machine on another computer or within the same computer via a virtual machine. When using a virtual machine, it is necessary to assign an IP address to the virtual machine or another computer where ROS is installed. However, when running both MATLAB and ROS on the same machine with the UBUNTU operating system, connecting via IP address is not required.

After running MATLAB and ROS, the connection can be enabled using MATLAB's *'rosinit'* function, which allows users to transmit commands straight from MATLAB to ROS. starts the global ROS node with the default MATLAB name and attempts to connect to a ROS master operating on localhost and port *'11311'*. If the global ROS node is unable to connect to the ROS master, *'rosinit'* launches a ROS core in MATLAB, which includes a ROS master, a ROS parameter server, and a *'rosout'* logging node [17].

After establishing the connection, the workflow began by scanning the area with a ground robot's Lidar. To start the Lidar on the ground robot, a ROS launch file to launch the robot's gazebo simulation file or an actual ground robot is used. In this study, a ground robot named Jackal is employed for both gazebo simulation and actual testing. In addition to launching the gazebo file, another program named '*gmapping*' is launched to activate the SLAM. The *'gmapping'* package includes a ROS wrapper for SLAM mapping. The *'gmapping'* package includes laser-based SLAM as a ROS node named *'slam_gmapping'*. Using *'slam_gmapping'*, it is feasible to produce a 2-D occupancy grid map (like a building floorplan) using laser and pose data gathered by a mobile robot [18].

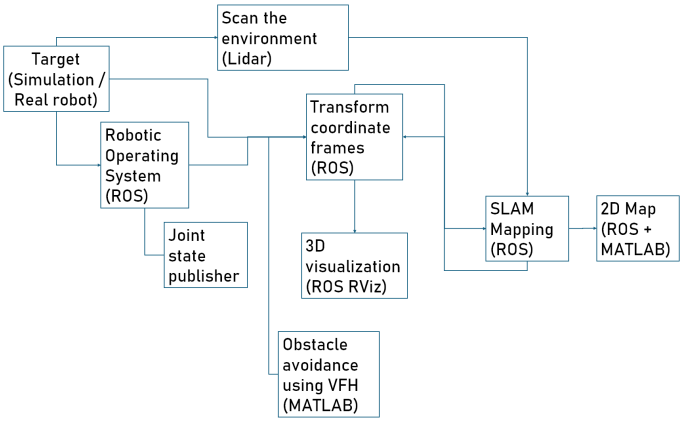

FIGURE 4. Block diagram of the overall procedure

This is where the VFH algorithm comes in practical, assisting the robot with simultaneous obstacle avoidance and mapping. MATLAB is used to run the VFH algorithm. Because MATLAB can now publish and subscribe to ROS nodes, it is possible to obtain detailed map information and view the results in MATLAB. The VFH algorithm guides the robot through the environment by detecting obstacles in real time. While navigating, the ROS node *'tf'* is launched automatically, allowing the robot to perform direct coordinate frame transformations. Using the coordinate frames from *'tf'*, the *'ROS-gmapping'* continuously records data, which is then utilized to create a full map of the environment. Figure 4 shows a block diagram that outlines the overall process.

3 Results and Discussions

The overall process is tested and validated using a simulation and real-time robot named Jackal. Jackal is an entry-level robotics research platform equipped with an Inertial Measurement Unit (IMU), GPS, and onboard computer that are linked with ROS to enable autonomous operation. First, the test is carried out using a pre-built simulation environment based on a simulator called a Gazebo. Gazebo is an open-source 2D/3D

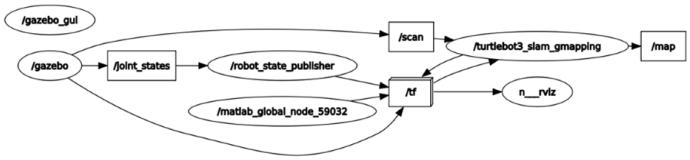

FIGURE 5. Overall computation graph from the *'rqt_graph'* tool

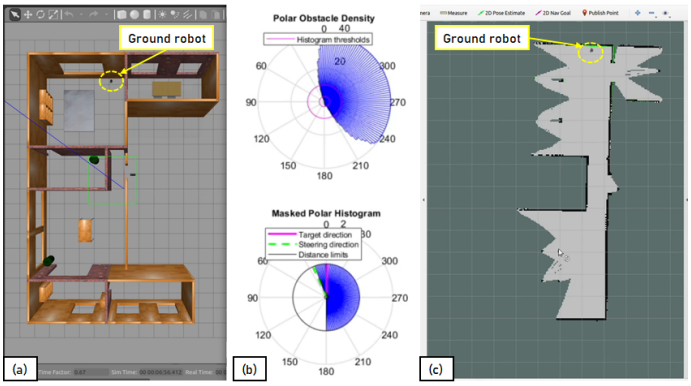

FIGURE 6. Mapping environment (a), polar obstacle density and masked polar histogram (b), and map generated by VHS (c) at the middle of the map-making process.

robotics simulator that combines simulation with a set of development frameworks and cloud services. It also provides the capability to approximate a range of robots in complex indoor conditions [19].

A ROS tool known as *'rqt_graph'* can be used to visualize the complete workflow of the overall autonomous mapping and localization process. The computation graph can be viewed using a GUI plugin included in the *'rqt_graph'* package of ROS. All the nodes *(/gazebo, /matlab_global_node, /robot_state_publisher, /gmapping)* launched by MATLAB and ROS are listed in the graph, along with the topics *(/joint_states, /scan, /tf, /map)* that are subscribed to and published. Figure 5 displays the overall computation graph.

Using the approach mentioned in the methodology the jackal ground robot is allowed to navigate and create a map of the environment based on the VFH algorithm and SLAM. Figures 6 and 7 depict the simulation environment to be explored with the current robot position in (a), the Polar Obstacle Density and Masked Polar Histogram in (b), and the map generated by the VHS and SLAM methods in (c). The figures provide the previously specified details at the middle and finale of the map-making process, aiding in the analysis of various case scenarios.

A Polar Density Histogram is computed over angular sectors in VFH using the range sensor data gathered from the Lidar. The angular sectors are shown in blue on this histogram, while the histogram thresholds are shown in pink. The lower and upper values of the Masked Histogram are determined by the two-element vector known as the Histogram Thresholds attribute. In the Masked Histogram, Polar Obstacle Density values greater than the upper threshold are shown as occupied space (1). Free space is displayed for values that are less than the lower threshold (0). The values in the previous binary histogram are applied to values that fall between the bounds, with free space (0) being the default. A Masked Polar Histogram plot corresponds to the Polar Density Plot. This plot displays the target and steering directions, range readings, and distance limits at various parts of the explored environment.

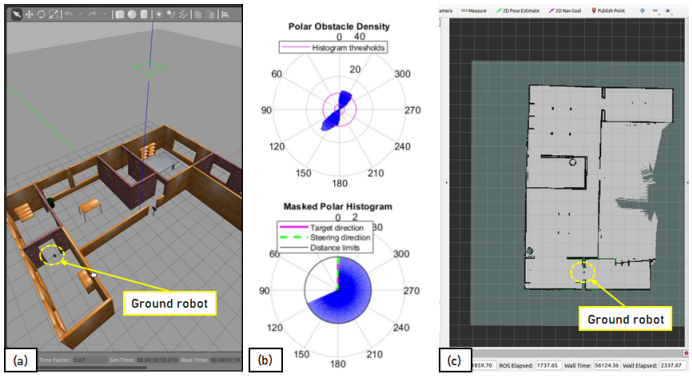

FIGURE 7. Mapping environment (a), polar obstacle density and masked polar histogram (b), and map generated by VHS (c) at the finale of the map-making process.

Parameter tuning can also be used to prototype the obstacle avoidance application. For instance, it is advised to change the histogram thresholds to the proper values in the Polar Obstacle Density plot if specific obstacles are absent from the Masked Polar Histogram display. The range sensor values, indicated in red, should line up with spots in the masked histogram (blue) once the modifications have been made to the Masked Polar Histogram plot. Additionally, the person using it can specify the goal direction and access the steering directions in the Masked Polar Histogram.

Since the steering direction is the primary output of the VFH algorithm, fine-tuning the final steering direction output can be achieved by varying the Cost Function Weights. Based on the present, prior, and target directions, the VFH algorithm takes into account several steering directions. The robot's steering behavior can be adjusted by adjusting the attributes for Current Direction Weight, Previous Direction Weight, and Target Direction Weight. Modifying these weights has an impact on the robot's responsiveness and obstacle-reaction capabilities. The Target Direction

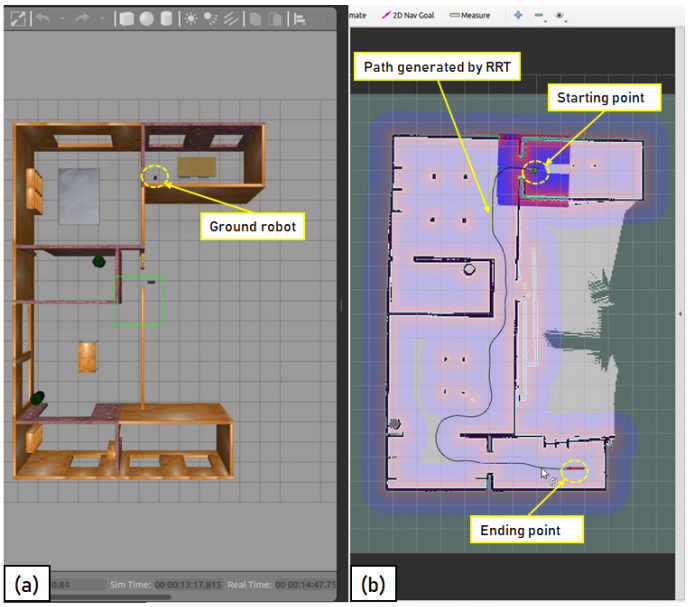

FIGURE 8. Autonomous path planning using RRT.

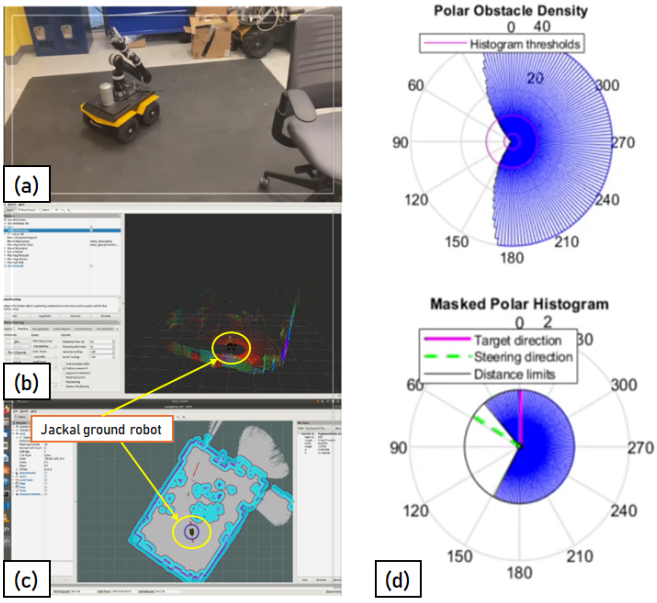

FIGURE 9. Jackal ground robot with Kinova Gen2 robotic arm (a); 3D point cloud visualization of the environment (b); map created by the VHS and SLAM methods (c); Masked Polar Histogram and Polar Obstacle Density at the robot's current position (d).

Weight is set greater than the total of the other weights to direct the robot toward its destination. To guarantee that the computed steering direction is near to the target direction, a high Target Direction Weight value is used.

Using the map that the VFH and SLAM algorithms simultaneously created, the RRT path planning approach uses it to navigate autonomously from the start point to the completion point while avoiding obstacles. Alternative path planning techniques, such as A*, can also be used for this kind of path planning. The autonomous path planning using the RRT method is displayed in Figure 8. The path that the ground robot takes to get to the destination is indicated by the curved line in Fig. 8.

The next step is to assess how well the methodology performs in a real-world scenario after using it in a simulation environment. Real-time testing is conducted using a Ground Robot called Jackal that has a robotic arm called Kinova Gen2 mounted on it. Even though it is not used in this study, the robotic arm can increase the ground robot's capabilities to the point where both can be used in the same application.

It should be noted that the results of the study do not guarantee that any solution developed for the simulation environment would directly work for a real robot and ground surface. When using real-time applications, there are a few things to consider, one of which is the friction between the ground robot and the ground surface. When working in a simulation environment, a rough ground surface with a coefficient of friction of one is typically assumed. But in actual ground conditions, this is not the case, and the ground robot's tires' friction with the ground must be taken into account.

The nature of the ground has an impact on the mapping process as a whole, particularly when turning the robot or applying angular velocity. For instance, if the robot is expected to turn 5 degrees per second for 5 seconds, the intended result is 25 degrees and, in a simulation setting, the same outcome is achieved. On the other hand, in real-time testing, if the ground surface is comprised of a ceramic floor, a rotation of less than 25 degrees is typically reached after 5 seconds. In this study, the ceramic floor is replaced with a high-friction mat to solve this problem, and the outcome is pretty much what was intended. However, the outcome is still not as pleasing as the simulation, which has an impact on the robot's ability to avoid obstacles and create a map promptly. This provides valuable information for future studies that will enable the inclusion of a coping mechanism for the integrated method that accounts for friction between the robot and the ground.

Figure 9 displays the jackal ground robot with the Kinova Gen2 robotic arm (a); the 3D point cloud visualization of the environment (b); the map created by the VHS and SLAM methods (c); and the Masked Polar Histogram and Polar Obstacle Density plots at the robot's current position (d).

4 CONCLUSION

The objective of this study is to provide a technique that uses the VFH approach in conjunction with the SLAM algorithm to generate a map autonomously while avoiding obstacles. The SLAM approach and the VFH algorithm are effectively integrated, and the outcome is examined and verified using both a real-time robot and a simulation. Furthermore, the generated map is employed for further autonomous navigation using an additional path-planning technique known as RRT.

One of the difficulties this research has encountered is the amount of time required to make maps. Because the VHS approach only uses the robot's lidar and previously mapped data as input to create new portions of the map, the robot may repeatedly go through an area without realizing it has already been navigated. Since the robot builds maps by coming from several locations and lacks a prediction ability that would allow it to avoid mapping the same locations repeatedly before arriving at a position, it realized it was mapping the same area after navigating the environment. It is advised to use a nonlinear model predictive control (NLMPC) technique in conjunction with the SLAM and VHS algorithms to get around the time issue [20] [21]. By providing the robot with a prediction about the places it needs to go based on the data it has already collected, this integrated strategy will assist the robot avoid exploring the same sites and will significantly reduce the time it takes to obtain the whole map.

The ground robot Jackal is equipped with a robotic arm known as Kinova Gen2, as seen in the results section. By using the robotic arm with the ground robot, its capabilities can be improved to the point where both can be used in the associated operation, and this is a useful step with a wide range of applications such as pipe inspection [22]. The collaboration between the ground robot and the robotic arm can also be applied in transporting goods, picking and placing objects, rescuing people, and operating in hazardous environments such as nuclear plants by using the map produced by the VHS and SLAM methods in combination with a technique that can be used for path planning and sensing of robot manipulators [23]. Through the integration of machine learning, the collaboration between the ground robot and the robotic arm makes a substantial contribution to the application's risk assessment and consumption prediction [24].

ACKNOWLEDGEMENTS

The work is supported by NSF Engineering Research Center (EEC-2133630), Hybrid Autonomous Manufacturing Moving from Evolution to Revolution (HAMMER), and Oak Ridge National Lab. This work is also supported by the Department of Energy Minority Serving Institution Partnership Program (MSIPP) managed by Savannah River National Laboratory under BSRA contract 0000602156.

REFERENCES

[1] F. Rubio, F. Valero, and C. Llopis-Albert, "A review of mobile robots: Concepts, methods, theoretical framework, and applications," *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, p. 1729881419839596, 2019.

[2] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*. MIT press, 2011.

[3] C. Stachniss, *Robotic mapping and exploration*, vol. 55. Springer, 2009.

[4] R. J. Aughey, "Applications of gps technologies to field sports," *International journal of sports physiology and performance*, vol. 6, no. 3, pp. 295–310, 2011.

[5] R. Willgoss, V. Rosenfeld, and J. Billingsley, "High precision gps guidance of mobile robots," in *Proceedings of the the 2003 Australasian Conference on Robotics and Automation (ACRA 2003)*, University of Southern Queensland, 2003.

[6] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based slam," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.

[7] R. Chatila and J. Laumond, "Position referencing and consistent world modeling for mobile robots," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, pp. 138–145, IEEE, 1985.

[8] J. Borenstein, Y. Koren, *et al.*, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE transactions on robotics and automation*, vol. 7, no. 3, pp. 278–288, 1991.

[9] M. B. Alatise and G. P. Hancke, "A review on challenges of autonomous mobile robot and sensor fusion methods," *IEEE Access*, vol. 8, pp. 39830–39846, 2020.

[10] R. Van Breda, *Vector field histogram star obstacle avoidance system for multicopters*. PhD thesis, Stellenbosch: Stellenbosch University, 2016.

[11] Y. Rabhi, M. Mrabet, F. Fnaiech, and P. Gorce, "Intelligent joystick for controlling power wheelchair navigation," in *3rd International Conference on Systems and Control*, pp. 1020–1025, IEEE, 2013.

[12] P. Pappas, M. Chiou, G.-T. Epsimos, G. Nikolaou, and R. Stolkin, "Vfh+ based shared control for remotely operated mobile robots," in *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 366–373, IEEE, 2020.

[13] W. Chen, N. Wang, X. Liu, and C. Yang, "Vfh based local path planning for mobile robot," in *2019 2nd China Symposium on Cognitive Computing and Hybrid Intelligence (CCHI)*, pp. 18–23, IEEE, 2019.

[14] I. Lluvia, E. Lazkano, and A. Ansuategi, "Active mapping and robot exploration: A survey," *Sensors*, vol. 21, no. 7, p. 2445, 2021.

[15] S. E. Ghamri, N. Slimane, and F. Nezzar, "Trajectory tracking and vfh obstacle avoidance for differential drive mobile

robot,"
[16] Y. K. Tee and Y. C. Han, "Lidar-based 2d slam for mobile robot in an indoor environment: A review," in *2021 International Conference on Green Energy, Computing and Sustainable Technology (GECOST)*, pp. 1–7, IEEE, 2021.
[17] MathWorks, "rosinit." `https://www.mathworks.com/help/ros/ref/rosinit.html`, 2024. [Online; accessed 16-February-2024].
[18] ROS, "gmapping." `https://wiki.ros.org/gmapping`, 2024. [Online; accessed 17-February-2024].
[19] S. Noh, J. Park, and J. Park, "Autonomous mobile robot navigation in indoor environments: Mapping, localization, and planning," in *2020 International conference on information and communication technology convergence (ICTC)*, pp. 908–913, IEEE, 2020.
[20] A. A. Tereda and S. Yi, "Predictive control of the kinova gen3 robotic manipulator using a nonlinear model," in *ASME International Mechanical Engineering Congress and Exposition*, vol. 87639, p. V006T07A021, American Society of Mechanical Engineers, 2023.
[21] S. C. Dekkata, S. Yi, M. Muktadir, S. Garfo, X. Li, and A. A. Tereda, "Improved model predictive control system design and implementation for unmanned ground vehicles," *Journal of Mechatronics and Robotics*, vol. 6, pp. 90–105, 2022.
[22] S. Hamoush, S. Yi, A. Megri, Y. Seong, H. ElSherif, M. Muktadir, S. Garfo, X. Li, B. Keshinro, H. Khoury, *et al.*, "Technology of mapping and ndt for pipes inspection," tech. rep., North Carolina Department of Transportation. Research and Development Unit, 2023.
[23] A. A. Tereda, *Path Planning and Sensing for Autonomous Control of Robot Manipulators*. PhD thesis, North Carolina Agricultural and Technical State University, 2021.
[24] Y. Ayalew, W. Bedada, A. Homaifar, and K. Freeman, "Data-driven urban air mobility flight energy consumption prediction and risk assessment," in *Intelligent Systems Conference*, pp. 354–370, Springer, 2023.