OBSTACLE AVOIDANCE CHALLENGE

FINAL REPORT

MAE 412 Mobile Robotics

Department of Mechanical and Aerospace Engineering

Benjamin M. Statler College of Engineering and Mineral Resources

West Virginia University

*Submitted: Wednesday December 15, 2019*

*Submitted To: Dr. Yu Gu*

*Submitted By:*

*Team - A*

*Mathew Bergenstock, Adam Hensley, Austin Moore, Hayden Nichols, Hyejun Park, William Smith*

1. **Executive Summary**

This document describes the technical approach, demonstration and testing plans for implementing obstacle avoidance along with the introduction, literature review of related work, and project management. The robot was tested on its ability to explore an environment without colliding with obstacles.

The overall project will be developing a fully autonomous robot that explores an area without colliding with obstacles or other robots that will perform simultaneously. The performance was held in AER 222 with many obstacles, competing with other robots to determine the most efficient robot when it comes to avoid obstacles and explore the area in given time. Detailed information can be found in the introduction and objective sections with articles researched on obstacle avoidance and path planning algorithms to implement the goals of this project.

The preliminary results of testing the proposed approach revealed some problematic behaviors of the robot. Initially, a bug in the code prevented the Vector Field Histogram (VFH) planner from controlling the wheels. As such, the reactive layer was the only functioning avoidance measure for the preliminary demo. The robot became trapped in small regions of the competition area as the reactive algorithm was triggered when the robot would try to move through small corridors and favored turning left abruptly. Improvements were made to the algorithm by fixing bugs in the VFH algorithm and removing the rigid logic behind the reactive layer.

For the demonstrated technical approach, the rigid logic behind the proposed reactive layer was removed, as it was not well suited for the dense competition environment. The robot is designed to generate a random bearing for drive direction, which is modified after the robot traverses a set distance to provide an exploratory behavior. The VFH algorithm is used to generate a new goal heading when the drive direction of the robot is obscured. This determines from the 2D light detection and ranging (LIDAR) data a series of windows that are free of obstacles. The new heading corresponds to the window closest to the global goal direction randomly generated within the script. This heading is fed into a PD controller that sends wheel velocities to the SMART robot.

The final demonstration of this refined system showed markedly better performance. This configuration of the obstacle avoidance algorithm and control system performed comparably to its competitive counterparts. The performance was consistent between all three trials, travelling around 70 meters and colliding with about 20-30 obstacles each trial. Some erroneous behaviors were revealed during the final demonstration, which were consistent with testing during development.

Implementation of obstacle avoidance was tested within demonstrating above strategies with other SMART 2.0 Robot on the same field. This document demonstrates techniques, results and essential plans that allowed above demonstrations. Individual responsibilities and project management can be found in the last section of this document along with the references.

**Table of Contents**

1. Introduction and Objective------------------------------------------------------------------------------pg. 4
2. Literature Review--------------------------------------------------------------------------------pg. 4-5
3. Proposed Technical Approach-----------------------------------------------------------------------------pg. 5-8
4. Preliminary Results---------------------------------------------------------------------------------pg. 9
5. System integration and testing plan-------------------------------------------------------------------------------------pg. 9
6. Project schedule----------------------------------------------------------------------pg. 10
7. Risk management plan--------------------------------------------------------------pg. 10
8. Team Qualification-------------------------------------------------------------------pg. 11
9. Project Management and individual responsibility------------------------------pg. 11

**2) Project Description**

1. **Introduction and Objective**

The teams are required to use a WVU SMART 2.0 Robot to navigate a field with minimal collisions. The robot is a modified roomba that has a small computer, a LIDAR for navigation, and a sensor interface board. The iRobot is controlled through code ran via Matlab. The teams will use the LIDAR system to detect objects around it and use that information to avoid collision with the objects.

The main objective of the final project is to have the team’s robot be able to autonomously traverse a confined area without colliding or bumping into moving or still objects. All robots will be competing in the same competition field at the same time. Three trials will be held and each trial will be ten minutes. The competition area will be changed and modified for each trial and each bot must be able to show “productive” signs, meaning it should not be frozen or repeating the same action over and over again or it will be disqualified from that trial. For points, the teams will be awarded one point for each meter the bot travels. Ten points will be deducted for each collision the robot has and a bonus of ten points may be awarded if the team’s robot can successfully find all the areas in the competition field.

From a learning stand-point, the main objective of the final project is to take the learned material from class and apply it to the competition through perception, decision making, and system integration of iRobot.

1. **Literature Review**

Obstacle avoidance can be viewed as a dichotomy: obstacle detection and avoidance path planning. In general, there are two approaches to obstacle detection and avoidance: one based on a global map, and one based entirely on sensors oriented about the robot’s local frame [8]. Given the possibility for reconfiguration of the environment and the dynamic nature of the environment, a global map approach may not be feasible. As such, articles detailing a local, sensor-based approach will be emphasized.

For the case of obstacle detection using the SMART robot, information about the obstacles in the robot’s environment must be gleaned from 2D LIDAR range data. One method for producing information about obstacles from 2D LIDAR in the robot local frame involves a pipeline of point cloud filtering to remove noise and reduce the complexity of clustering, segmentation and merging segments, and clustering nearby segments into obstacles. This method reports the size of the obstacle and position relative to the robot [8].

Characteristics of the obstacles in addition to size and relative position, such as relative velocity of the obstacle, can also be tracked. One approach that expands obstacle detection to non-stationary obstacles uses a Kalman-based process to predict velocities of detected obstacles [1]. Tracking obstacle velocity can allow for decision-making in a dynamic environment. This can be applied through a Bayesian approach to an occupancy grid style reactive algorithm [1,2]

Once obstacles have been detected, the avoidance path planning or reaction algorithm can be applied. One of the earliest, and perhaps simplest, algorithms is the Bug algorithm. Bug plans a direct path to the destination until it faces an obstacle. There are three varieties of the bug algorithm, based on intermediate behavior when avoiding the obstacle [9]. The bug algorithm requires a distance sensor and knowledge of the current and destination position. Although it may move the robot away from the destination or take a low-efficiency path, it is low-computation and always converges [5].

Another low-computation algorithm is the artificial potential field (APF) method, which simulates a potential field by assigning repulsive potential obstacles and attractive potential to goals. The net artificial force of these potentials on the robot are calculated based on the strength of the potential field and the robot position in the field. This net force results in acceleration actuated by the robot control system. AFP is goal-oriented and can generate the shortest path, but can also generate dead-end solutions or local minima when symmetric or U-shaped obstacles are encountered [6, 10]. Moreover, APF does not take into account robot constraints [5], such as non-holonomic constraints on a mobile robot, therefore it may be better suited for manipulator and UAV planning than for mobile robots.

A more efficient and tunable avoidance planning algorithm is the follow the gap method (FGM), which determines the maximum gap between obstacles is calculated and the center angle of the gap from the robot frame is the bearing angle for the path [7]. FGM always chooses the shortest path and does not have problems with symmetric obstacles. The calculations are simple to compute and the only needed information is an obstacle’s distance and angles. However, the solution may not be reached with dead-end obstacles [5].

A more computationally complex method of avoidance that is well-suited to nonholonomic mobile robots is the vector field histogram (VFH) method. This projects obstacles into a 2D histogram representing occupies cells and associated certainties. The 2D cartesian histogram is converted into a 1D histogram representing the polar densities of obstacles surrounding the robot. From this 1D polar histogram, the direction associated with a region whose density is below a set threshold is chosen as the goal heading, and a linear velocity is set in this direction [3]. Unfortunately, the generation of the histograms can be computationally costly, but for the case of a 2D LIDAR, this is less concerning. A method that works similarly, but avoids local traps and has been applied to two-wheeled differential drive is the obstacle restriction method (ORM), which also takes into account the proximity of a potential path to the path toward the goal location [4].

1. **Technical Approach**

Development of the avoidance and exploration capabilities will be done in the custom controls section of the provided SMART robot control code. In order to provide some means of random exploration, functionality will be developed to generate a random heading in the inertial frame for drive direction. The total distance travelled by the robot will be tracked and when the robot has traversed a set incremental distance with the current heading goal (i.e., travelled 10 m), a new random heading will be generated. This methodology will inject some random disturbances in the case of the robot getting stuck in an area, promoting exploration of new regions.

The technical approach for avoidance is an implementation of the vector field histogram (VFH) approach for path planning. During the development and testing of this algorithm, it was found to be too unstable to have full control over the robot’s heading at all times in a dense environment. This was suspected to be the result of a rough histogram (no smoothing function was used), as cited to be a potential issue in [3], or an issue with mismatched coordinate system. As such, it was determined to be more effective if the robot drove in the selected direction until the immediate path was obstructed, at which point a new direction would be generated by VFH.

To determine if the current drive direction is clear, a basic close-range obstacle detection algorithm searches for rangefinder detections within a limited immediate radius of the robot (e.g., 0.5-m radius.) in a wedge directly in front of the robot. If points are detected within this radius, the VFH algorithm is triggered to find a new heading.

A simplified vector field histogram (sVFH) method, based on [3], is used to plan a path if when this condition is met. VFH was chosen due to the facility of generating a polar histogram from the laserscan data. Since the points returned from the scanner are already given in polar coordinates (i.e., each array entry representing a distance at the corresponding angle in degrees), much of the computational work of VFH (projection of points into an occupancy grid and generating a polar grid) is avoided. The simplification of the VFH used involves removal of the detection certainty mechanic, and simplification of the direction from active cell calculation described in [Borenstein and Yoren, et. al.]. The certainty will be set to unity due to the planar detection method and lack of scan processing to determine accuracy of detection. Moreover, the direction from the active cell of each point is already known, so calculation of polar obstacle density will only depend on the angular location of the point and its distance, using the following equation:

M\_j = a-bd\_j

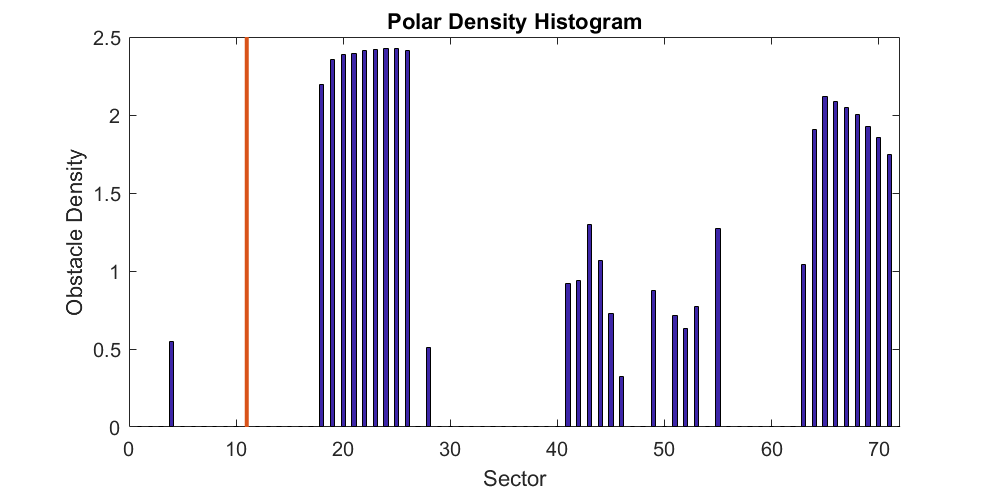
Where m\_j is the magnitude of the obstacle vector in the *j*th direction, a and b are positive constants, d\_j is the distance in the *j*th direction. Constants a and b are selected such that a-b\*d\_max = 0. These are then grouped into sectors of size **alpha**, as follows

k = INT(β\_j/α)

Where β\_j is the angle of *j*th direction relative to the robot and k is the sector index to which β\_j corresponds. The polar density of each sector is calculated as the sum of the magnitudes of each obstacle vector corresponding to the sector.

H\_k = sigma\_j(m\_j)

Plotting of the polar densities with respect to the sector direction generates a polar histogram, an example of which can be seen in Figure XX. The blue bars indicate the density (a function of distance from the robot) of obstacles in each sector. 72 sectors are shown, meaning each sector represents a 5-degree wedge. The orange bar delineates the selected direction, which is chosen from the center of the selected window (discussed in further detail below).



<INSERT FIGURE LABEL>

Once the histogram is generated, each sector is checked against a threshold for polar density to determine the valleys of the histogram. The width of each valley is determined by counting the number of consecutive sectors that have densities below the threshold. Usually, two or more sectors will be candidates so the algorithm will select the candidate sector closest to the global target direction. Selection of the portion of the window to traverse depends on window size. For small windows, the mid-line angle of the candidate region will be determined and set as the desired heading angle in the control loop. However, when large windows (i.e., all of the region behind the robot is open), to avoid repeatedly turning the opposite direction, it was found most advantageous if the left third of the window is chosen as the desired heading. This encourages exploratory behavior toward dense environments, as the robot will be less likely to “bounce” between two parallel walls in an open area.

An additional feature added while testing is a “bump” reaction that makes the robot turn when any of the bumpers are triggered. This keeps the robot from stopping when an obstacle collision occurs.

1. **Demonstration**

In the first demonstration, the SMART robot began the test pointing toward a corner of the enclosed area and had an issue detecting this obstacle. The robot had to be manually rotated in order to end the process of driving straight into the course’s boundary. However, after intervening the robot performed well by traveling a total of 69 meters and fully exploring the confined test area with a total of 29 collisions. This collision total would be the highest out of all three tests for our robot due to the unexpected behavior at the start of the test.

For the second demonstration the testing area was increased in size while also reducing the stationary objects within the region. This new orientation provided the best overall results of the robots testing with a total distance of 72 meters, full site exploration, and significantly dropping the collision total to 18 over the ten-minute testing period.

For the third and final test area, a dramatic increase in test area obstacles was added while still holding the area from test two constant. The largest obstacle in the new area was several crates that divided the large area into two smaller sections while leaving a small corner open connecting them. For the first several minutes of the test, the robot entered the open pathway to explore the other half of the test area only to be disturbed by another robot and eventually turn back to the starting area. One of the believed causes for this is the robots heavy favoring to rotating counter clockwise no matter the direction of the detected obstacle. However later in the testing period the robot successfully made it through the threshold and begun its exploration of the second side. In the final test the robot completely explored the area while traveling 73 meters and tallying 25 collisions.

One issue that may have resulted in better performance if addressed was the instability of the robot heading selection when up against an obstacle in a narrow corridor. It seemed to spin for a couple seconds before selecting/reaching the intended heading and driving straight. This cost the robot time and distance.

1. **Project Management**

|  | **Percent Contribution (%)** | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Category** | **Mathew** | **Adam** | **Austin** | **Hayden** | **Hyejun** | **William** |
| **Document**  **Writing** |  |  |  |  |  |  |
| **Perception** |  |  |  |  |  |  |
| **Path Planning** |  |  |  |  |  |  |
| **Robot Control** |  |  |  |  |  |  |
| **Software**  **Integration** |  |  |  |  |  |  |
| **Testing &**  **Debugging** |  |  |  |  |  |  |

1. **Conclusion**

WVU SMART 2.0 Robot was modified to avoid obstacles including other robots and explore the field in this project. Preliminary demonstration revealed some problematic behaviours of the robot including bug in Vector Field Histogram planner from wheel control, preventing the robot to escape when it was trapped due to reactive algorithm that was triggered. This was fixed for final demonstration by fixing bugs in the VFH algorithm and removing the rigid logic behind the reactive layer.

The overall demonstration featured full exploration of the field that was given with some amount of collisions. This was achieved by implementing vector field histogram and setting the random heading that is generated in the inertial frame for drive direction along with tracking total distance traversed setting incremental distance with the current heading goal. However, there were some behaviours observed that can be enhanced if we were to design the project again.

Improving the instability of the robot heading selection when up against an obstacle in a narrow corridor that spins for a couple seconds before selecting or reaching the intended heading and driving straight will enhance the exploration of the robot. This will cost the robot time and distance. Another solution for this is implementing multi-stage planning, or planning out two turns to develop a path. Adding other stages to the current robot to eliminate the behaviour of exploring the same path for a while can be removed. Other solutions can be increasing the time tuning the weights, constants to get the intended behaviour.

Obstacle avoidance with exploring project has improved many skills we have learned throughout this semester. This project has taught us how robot can encounter unexpected circumstances when it was put on the field compared to its theory behind. Many fixing and trials were held with time management. Furthermore, Researching articles to plan the algorithms to achieve the goal included many sources that this course has taught including many sensors and Kalman-based process to predict velocities of the detected obstacles.

**3) References**

[1] A. Cherubini, F. Spindler and F. Chaumette, "Autonomous Visual Navigation and Laser-Based Moving Obstacle Avoidance," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2101-2110, Oct. 2014.  
doi: 10.1109/TITS.2014.2308977

[2] C. Fulgenzi, A. Spalanzani and C. Laugier, "Dynamic Obstacle Avoidance in uncertain environment combining PVOs and Occupancy Grid," *Proceedings 2007 IEEE International Conference on Robotics and Automation*, Roma, 2007, pp. 1610-1616. doi: 10.1109/ROBOT.2007.363554

[3] J. Borenstein and Y. Koren, “The vector field histogram-fast obstacle avoidance for mobile robots,” IEEE Transactions on Robotics and Automation, vol. 7, no. 3, pp. 278–288, 1991.

[4] J. Minguez, "The obstacle-restriction method for robot obstacle avoidance in difficult environments," *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Alta., 2005, pp. 2284-2290.  
doi: 10.1109/IROS.2005.1545546

[5] M. Zohaib and M. Pasha and R. A. Riaz and N. Javaid and M. Ilahi and R. D. Khan, “Control Strategies for Mobile Robot With Obstacle Avoidance,” 2013.

[6] Oroko, J., & Ikua, B. (2012). Obstacle Avoidance and Path Planning Schemes for Autonomous Navigation of a Mobile Robot: A Review. Sustainable Research and Innovation Proceedings, 4.

[7] Sezer, V., & Gokasan, M. (2012). A novel obstacle avoidance algorithm:“Follow the Gap Method”. Robotics and Autonomous Systems

[8] Y. Peng, D. Qu, Y. Zhong, S. Xie, J. Luo and J. Gu, "The obstacle detection and obstacle avoidance algorithm based on 2-D lidar," *2015 IEEE International Conference on Information and Automation*, Lijiang, 2015, pp. 1648-1653.  
doi: 10.1109/ICInfA.2015.7279550

[9] Yufka, A., & Parlaktuna, O. (2009, May). Performance comparison of bug algorithms for mobile robots. In Proceedings of the 5th international advanced technologies symposium, Karabuk, Turkey.

**Appendix:**

***Project Code:***