Improving Quality of Life Using ICT, IoT and AI (HONET)

Charles Koduru
Kennesaw State University

Independent Optimization for Robot Path Planning and Dynamic Obstacle Avoidance

1st Terrance Hall
Dept. of Robotics and Mechatronics Engineering
Kennesaw State University
Marietta, USA
thall110@students.kennesaw.edu

2nd Christopher Johnson
Dept. of Robotics and Mechatronics Engineering
Kennesaw State University
Marietta, USA
cjohn567@students.kennesaw.edu

3rd Brighton Swales
Dept. of Robotics and Mechatronics Engineering
Kennesaw State University
Marietta, USA
mswales@students.kennesaw.edu

4th Charles Koduru
Dept. of Robotics and Mechatronics Engineering
Kennesaw State University
Marietta, USA
ckoduru@students.kennesaw.edu

5th Dr. M. Hassan Tanveer
Dept. of Robotics and Mechatronics Engineering
Kennesaw State University
Marietta, USA
mtanveer@kennesaw.edu

***Abstract* — Autonomous robots can be assigned with various tasks such as moving payload, analyzing terrain, and capturing data in an environment. For an Autonomous Mobile Robot (AMR) to execute such tasks the robot (Hussarion ROSbot) will require efficient algorithms and techniques to reference its current location. The robot is relative to surrounding obstacles in its predetermined path. The conducted research study explains the coordinated method used to successfully allow a robot to identify its position in the environment (Gazebo Simulation) and avoid obstructions with increasing velocity - contingent on nearby surroundings. The results show multiple robots individually tasked with distinct roles, while incorporating an obstacle avoidance function used to avoid both static and dynamic obstacles. Such results can be used in the applications of a high-capacity warehouse environments.**

***Keywords*—Autonomous Robots, Gazebo Simulator, Multi Robot System.**

I. Introduction

Navigation strategies for autonomous vehicles have come a long way in the past few decades through research and development [1]. As we move towards a world where autonomy of ground vehicles is integrating into workplaces with humans, obstacle avoidance is one of the most crucial factors to account for. Though the collaboration of humans and robots can increase efficiency, without proper safety features implemented such as obstacle avoidance, the environment can become dangerous. However, path planning in a dynamic environment such as a workspace is difficult [2]. Throughout the years [3], researchers have developed different methodologies on implementing efficient ways autonomous ground vehicles could be used in different environments. Methods such as Model Predictive Path Integral (MPPI) have shown promise by enabling controllability for fast vehicles such as racing vehicles[4], [5]. However, excessive disturbance to the initial state can cause the robot odometry values to exceed design specifications making the controller no longer useful. Although most controllers are specialized for fast controls, the evolution for Path Integral Controllers is moving towards performance and accuracy rather than speed [6]. Continuing this trend, the experiments conducted were for a more controlled method that allowed efficient path planning and still accounted for some standard deviation. During this research work, a method of implementing independent efficient navigation controls for autonomous ground vehicles that can avoid obstacles in a dynamic environment was tested and analyzed. Leveraging the use of an occupancy grid, we are able to accurately track our independent robots and analyze their behavior under different scenarios. This allowed us to take a graphical approach to illustrate the response of the Gaussian function. We validated the practicality of the equation through static and dynamic obstacle avoidance scenarios. The use of Gazebo allowed for multiple experiments to be conducted rapidly to validate results. Experiments include, validating the objective path without an obstacle, with a static obstacle, and a dynamically changing obstacle. The ROSbot used in the Gazebo simulation uses its LiDAR sensor to generate environment data, which is used to determined moving and nonmoving obstacles.

The next section goes over our methodology of the approach and section 4 goes into a deeper analysis of the experiments and delving into the results. Our ambition with these experiments is to transition 2D navigation techniques to 3D where multi-collaborative robots can maneuver autonomously and move efficiently on land and in air.

II. METHODOLOGY

A. Robot Odometry

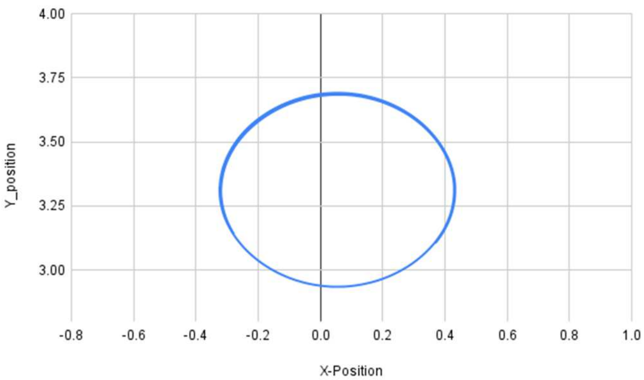

Fig. 1: Occupancy grid of tracked path

The following steps allow the ROSbot to avoid obstructions and take path respective to the environment based on x and y coordinates of both the ROSbot and obstacle. Here we stored the odometry positions of the ROSbot (X,Y) and implemented a circular equation (1) for the robot to take a continuous path centered at a point. Hence, we can control where the robot moves and curves, by recording the robot's current positions in Fig. 1 and incorporating the x and y coordinates to make the robot take a path relative to an equation.

$$x_j{}^2 + y_j{}^2 = R^2 \tag{1}$$

$$O_j(x,y) = A_j e \frac{(x-x_j)^2+(y-y_j)^2}{\sigma^2} \tag{2}$$

- x_j, y_j is the position of obstacle;
- A_j is the height of the Gaussian curve;
- σ is the standard deviation.

B. Environment Scan

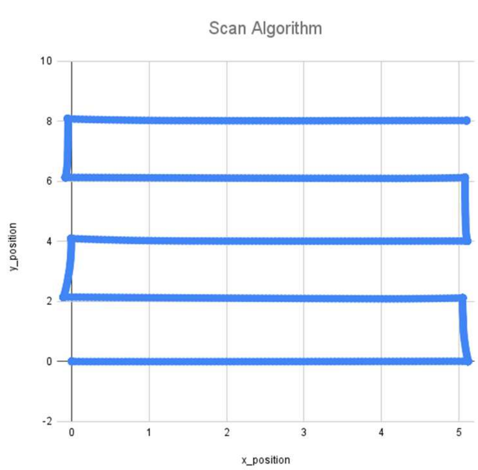

Fig. 2: Occupancy grid of scanning empty environment for mapping

We constructed a virtual environment via Gazebo and Rviz simulations to assess virtual characteristics of robot navigation and path planning. This graph shows the coordinates of a scan algorithm of a ROSbot in Fig. 2. The 'S' type path allows the robot to navigate all points in the environment. Additionally, we stored the x and y values of the robots to create a grid of the environment. The ROSbot can use these values to move to a certain position within the grid space. The scan algorithm uses the position of the ROSbot and outputs the coordinate values to a file. This, used in conjunction with the wall follow algorithm, can scan an unknown room and create a space where it can move Fig. 2. The utilization of occupancy grids allow the robot to develop a coordinate system of the scanned environment.

C. Static Obstacle Detection

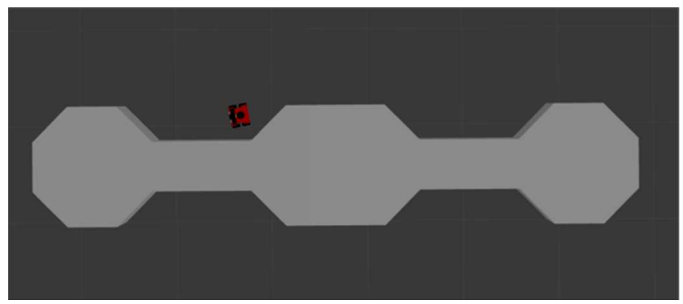

Fig. 3: Static Wall Obstacle

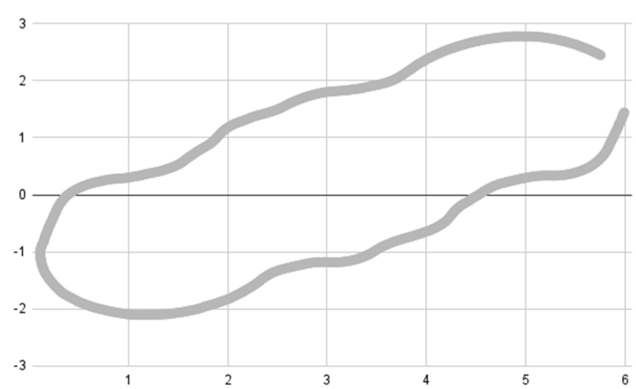

Fig. 4: Occupancy grid of right wall follow on obstacle

The experiments illustrated in Fig. 3 and Fig. 4 are performed using a static object as a point to implement the wall following algorithm. The points shown in the occupancy grid correspond to the x and y positions the ROSbot is following. As the ROSbot approaches different angles on the obstacle, the ROSbot adjusts its position accordingly to maintain the user's desired distance and continue following the coordinated path. Compared to Fig. 2, Fig. 4 shows that as the ROSbot approaches an obstacle, it can efficiently alter its path to avoid the obstacle. Simultaneously, the ROSbot recognizes an obstacle in the environment by scanning the sides and edges of objects. In reference to equation (2) the obstacle coordinates are represented as (Xj and Yj) and (X, Y) being the current position of the robot. Using this equation allow the robot to take a curved path around obstructing edges and corners. Sigma squared effects the robot's angles of avoidance around obstacles.

D. Dynamic Obstacle Avodance

This experiment was to test the Gaussian function in Fig 5. on a dynamic moving object. Below the two ROSbots are performing independent tasks but need to avoid each other and continue. The ROSbot's goal following the green path is to map an area while maintaining a specified distance on the ROSbot's left side. The ROSbot that's following the red path is following the same actions but avoiding on the ROSbot's right side. Once either ROSbot detects a discrepancy in the x and y initial and final positions, the ROSbots will individually change paths based on the values given using the Gaussian function (2).

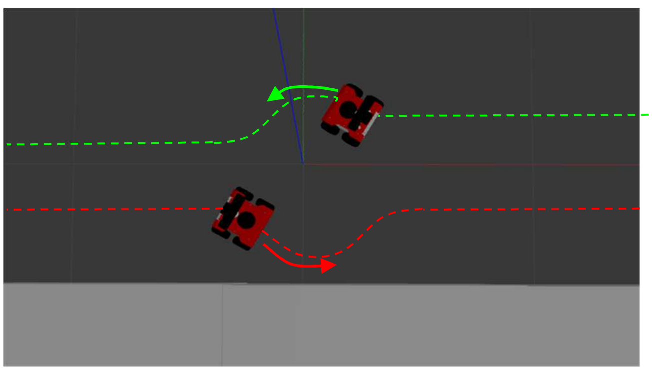

Fig. 5: Independent bots avoiding collision and returning to planned path

After both ROSbots dynamically avoid each other by following another path created by the Gaussian function, the ROSbots would continue on their initial paths. In this particular case, both robots are programmed to follow a straight line. The dotted line illustrates the path what the robots take.

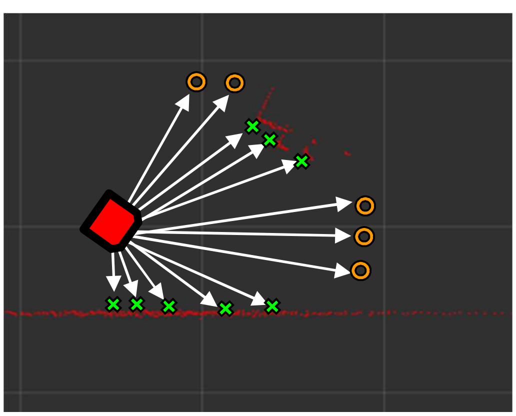

Fig. 6: LiDAR sensor detecting moving object and static object

This view represents LiDAR data, in this scenario the ROSbot initially encounters the wall obstacle on the right side and later is avoiding the incoming robot with similar velocity on the left side. In Fig 6. the ROSbot's LiDAR detection is shown by X's and O'. The X's are where the LiDAR hits an object, and it reads the distance value The O's are where it reads a value too high to record (inf), therefore analyzing the data as an empty area and allowing the the ROSbot to continue its path in that space. The readings from the left will follow the standard distance maintaining program and move along the designated path while maintaining a set distance. However, the path will be altered due to the detections from the separate ROSbot to its left. Since the change in x and y final position compared to their initial position is higher than expected, the ROSbot can determine that the obstacle is dynamic and not static. Similar to Fig. 5, it uses equation (2) by taking the x and y final and initial points measured from the LiDAR to make a slight curve from its original path enough to avoid the dynamic obstacle.

E. Individual Dynamic Path Planning

Fig. 7: Multi Robot Warehouse Simulation

Fig. 7 ties in our implementation of obstacle avoidance and path planning by mapping out the environment and avoiding both static and dynamic obstacles. Each ROSbot has an assigned area to map, but each robot is individually tasked with it object to move from one point in the warehouse environment to another point with some overlay into others. In Fig. 7, the green and red mapping areas overlap slightly. To avoid collision with the other robot, they both integrate the (2) function in equation (2) to find the most efficient path for evading the other and return to its original course. As the light blue, dark blue, and yellow paths approach an obstacle, the ROSbot uses the reading from the LiDAR data to approach and maintain the predetermined distance from the obstacle. The fixed distance in our experiment was half a meter.

III. ANALYSIS

The first part of this research began with a simple wall-following algorithm seen in Fig 3. This will be used for moving around static objects within the non-conventional environment used in Fig 7. Expanding on this algorithm allows implementing another ROSbot sought to see the relationship between the two robots when dynamically encountering each other on the grid space. When encountering each other, we tested out how different object-avoidance algorithms would interact when both ROSbots are using the same algorithm, Fig 5, and working dynamically to avoid each other. We settled to use a gaussian function (2). This function is used when ROSbots encounter each other instead of a wall follow or object avoidance because of its ability to have the ROSbots nearest to their path they were on after the function is done. This saves time so the ROSbot isn't constantly moving and readjusting like in an avoidance algorithm. The use of the Gaussian function (2) allows the ROSbots to change its targeted path in order to avoid an obstacle. After this encounter the ROSbots would continue on their respective objective path.

To incorporate this method toward real-life application, we created a warehouse environment in the Gazebo simulation to see how individual independent path planning could be used in a non-uniform setting, shown in Fig. 7. This shows multiple ROSbots moving to separate target points with the green and red paths avoiding each another in a non-conventional setting. While the rest of the robot are following their own paths, at the same time they are remapping their path based on obstructing obstacles. Most warehouses are grid shaped and/or planned out to suit the ROSbots needs, where the ROSbots are given specific tasks that need to wait in a queue with others to wait for their turn to deliver/move across a warehouse floor. In settings where this is not an option or time constraints prevent the organization of objects within said warehouse, this system could be used to operate ROSbots in such environments. The importance of Autonomous Mobile Robots to effectively follow paths and avoid dynamic obstacles Fig. 7. For example, if there is debris on the floor, this system could be also used on construction sites where materials need to be delivered across uneven ground in order to aviod obstacles.

IV. CONCLUSION

The goal of the research is to produce autonomous robots with individual objective accomplishment. In a warehouse environment, humans and robots can work effectively and independently when focused solely on task accomplishment that is independent of another's actions. This will minimize waiting periods that would result from a curated operational routine. Through our method of environment mapping and avoiding dynamic and statics obstacles, robots can detect and move around humans while working efficiently in order to accomplish a task. This methodology of environmental navigation is uniquely suited for businesses with a less formal and repetitive structure in which autonomy normally inhabits. A consistent and predictable work environment allows the meticulous planning, organization, and execution of autonomous actions, but not all operations have the same predictability. We plan to further develop our avoidance algorithms to operate effectively in a 3-dimensional environment through the use of a Motion Capture System. As aerial systems become more ubiquitous, the need for safe navigation through a dynamic air space is important. The premise of efficiency through individual goal accomplishment need not be limited to ground applications but will need to include ariel environments as well. The expansion of this approach to a 3D space would allow for multiple robots to occupy the same area without a central coordinated structure (e.g., multiple logistics companies could enter delivery drones into already occupies service areas without prior coordination of other participants). Akin to other types of decentralized models, independent path planning, optimization, and dynamic avoidance offers many benefits to consistently varying environments where other methods may struggle.

V. ACKNOWLEDGEMENTS

We would like to thank Kennesaw State University's Department of Robotic and Mechatronics Engineering for providing the opportunities, resources, and support that enabled us to conduct this research.

REFERENCES

[1] T. Litman, *Autonomous vehicle implementation predictions*. Victoria Transport Policy Institute Victoria, BC, Canada, 2017.

[2] O. Mazhar, B. Navarro, S. Ramdani, R. Passama, and A. Cherubini, "A real-time human-robot interaction framework with robust background invariant hand gesture detection," *Robotics and Computer-Integrated Manufacturing*, vol. 60, pp. 34–48, 2019.

[3] B. B. K. Ayawli, R. Chellali, A. Y. Appiah, and F. Kyeremeh, "An overview of nature-inspired, conventional, and hybrid methods of autonomous vehicle path planning," *Journal of Advanced Transportation*, vol. 2018, 2018.

[4] J. Yin, Z. Zhang, E. Theodorou, and P. Tsiotras, "Trajectory distribution control for model predictive path integral control using covariance steering," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 1478–1484.

[5] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *Journal of Field Robotics*, vol. 37, no. 3, pp. 362–386, 2020.

[6] Y. Xiong, C. Peng, L. Grimstad, P. J. From, and V. Isler, "Development and field evaluation of a strawberry harvesting robot with a cable-driven gripper," *Computers and electronics in agriculture*, vol. 157, pp. 392–402, 2019.