3D Mapping using 2D Lidar

UGV Lidar Team

Britanya Wright, Doyun Lee, Alexander McGirt
North Carolina State University
Department of Electrical Engineering, Department of Civil Engineering, Department of Industrial & Systems Engineering

***Abstract-* Welding processes on a construction site expose individuals to harmful fumes, elevated temperatures, demanding physical labor, as well as the risk of injury from the surrounding work area. Due to this, combining robotics and welding can greatly contribute to construction safety and efficiency. Within a construction site, an autonomous robot must have the ability to use various sensors to avoid obstacles, perform navigation to the welding material, and lastly perform welding on desired material using machine learning. This report focuses on how the navigation and obstacle avoidance of the Husky Robot using a 2D LiDAR and the NVIDIA Jetson Nano computer was achieved. Once the robot has reached its desired location (location of welding material), status updates are sent via MQTT to a robotic arm to perform the welding process.**

***Index Term*– RViz, robotic operating system (ROS), lidar, adaptive monte carlo localization (ACML), simultaneous localization and mapping (SLAM)**

I. Introduction

Developments in the autonomous robotic realm have a broad reach. Although that is the case, there are still many improvements to be made in that area. The benefits associated with automation can be obtained in almost every level of society if implemented correctly. In the case of construction environments, which present varying levels of risks and hazards, proper implementation can be challenging. Welding processes in a construction environment are scenarios of high benefit potential. This is the case because welding processes expose individuals to harmful fumes, elevated temperatures, demanding physical labor, as well as the risk of injury from the surrounding work area.

One way to carry out this process in an automated fashion is through the use of robotics. In this case, a Husky Unmanned Ground Vehicle (UGV) robot will be utilized to carry out the welding process. The ultimate objective of this study is to develop an automated robotic welding system that consists of a ground vehicle, manipulator, welding machine, and visual sensors. This research includes the hardware integration of embedded platforms, sensors, and networking with visual sensors for machine vision. The planning of the development of software will include the design of a simulation environment, integration of visual Simultaneous Localization and Mapping (SLAM) algorithms, and design of feedback and motion planning strategies.

As a result of this research, these components will be incorporated into a platform that will navigate autonomously in a construction scene while creating a map of the scene and following a trajectory generated automatically from a user-defined target destination, detecting welding joints using a visual sensor, and performing welding. Coupled with the Husky UGV there will be several subsystems in place to help streamline the welding process. This includes not only the actual welding steps themselves but also locating the welding location(s).

A. Motivation

Building construction mainly employs welding in the creation of structural frameworks from metal components [3]. Welding is primarily used for connecting steel I-beams, trusses, columns, and footers to support the walls, roof, and floors of a building. These huge metal components cannot be welded using a fixed robotic system. The integration of a UGV with developments made by the ARM team will be incorporated to add mobility to the welding robot.

By integrating UGV with the developed welding robotic system, the limitation of movable ranges becomes unlimited to the x and y-axis, as shown in Figure 1. This complete proposed robotic system can physically cover most of the structural components on the ground in the construction project. The trajectory plan for UGV and the robotic arm will be re-designed to perform scanning and welding tasks for large metal components. The study on portable and automatic welding systems will be conducted by integrating diverse technologies such as the camera, the laser scanner, the robotic arm, the welder, the LiDAR, and the UGV.

Figure 1: Conceptual design of complete mobile welding robot

II. Background

Although the Husky UGV system is quite robust, various types of sensors and actuators will need to be added to it before autonomous welding operations can be carried out. To make the robot move to the desired position, map generation using LiDAR is needed. The 2D Lidar system utilized is a current product on the market created by

SlamTec, called the RPLidar A2 laser range scanner. This sensor generates light energy reflected from various surfaces where the LiDAR sensor measures the return energy. With the information on the direction and range of the laser beam, in conjunction with the device's position at which the sensor is attached, a 2D representation of the desired space can be formed. Providing a scan output of up to a 16-meter range radius, RPLidar A2 is to be utilized to generate a 2D rendering of the environment around it. Gmapping, one of the most widely used Simultaneous Localization and Mapping (SLAM) methods, is used in this research. It will be utilized to help navigate the Husky around the construction environment while also aiding in collision avoidance efforts when applicable.

A. Related Work

Expanding autonomous approaches to carrying out tasks can be seen in almost every segment of society. Whether it be in the interest of enhancing productivity levels, improving quality, or even reducing risks to humans, autonomous system development is a continuous process. Efforts to reduce intensive and repetitive construction-related tasks with a ROS-based platform can be found in [8]. In the realm of robotic control, relevant research involving lidar-based localization of ground robots [1], along with the mapping of unknown environments [7], is a growing topic. Various approaches have been taken to combine these approaches and obtain real-time positioning and navigation, such as in [9].

III. Methodology

This project is broken down into multiple teams, as shown in Figure 2. Team one focuses on joint detection and trajectory planning using Deep Learning. Team two focuses on the movement of the robotic arm for joint tracking and welding. Team three focuses on 3D mapping using visual SLAM. Team four focuses on 3D mapping using LiDar and wheel encoders.

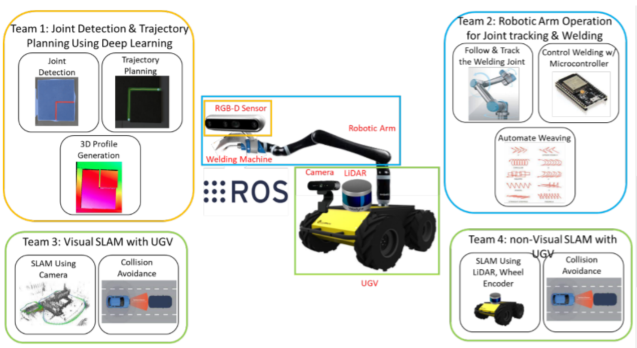

Figure 2: Pictorial representation of different teams in the development of the welding robot.

Our focus is to execute the tasks of the UGV-SLAM team and in order to generate a map of the space, we chose to use a 2D RPLidar due to an existing 3D Lidar not working sufficiently.

To generate a map with a 2D RPLidar, perform navigation, and execute obstacle avoidance, the outline in Figure 3 was followed.

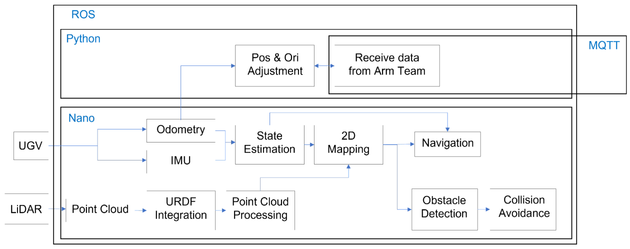

Figure 3: Outline of the project

An overall diagram of the robot's objective destination and intended path can be seen in Figure 4.

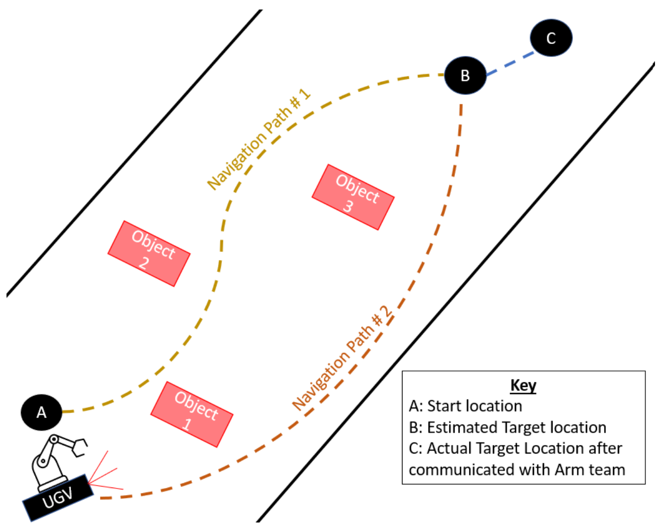

Figure 4: Outline diagram of the overall objective during each stage of the robot's operation

In order to execute each stage of the project, the NVIDIA Jetson Nano is used as the main computer for the Husky Clearpath robot. First, Ubuntu Bionic 18.04 is installed and configured to the Jetson Nano using a flashed micro SD card. Then, ROS melodic along with the necessary APT packages are utilized to get the Husky functioning is installed by running 'install.sh' file provided by Clearpath. This process is needed to configure the Jetson Nano for use with Husky.

On the Jetson Nano, the Husky package for ROS melodic is installed using the command line:

sudo apt install ros-melodic-Husky*

This package is inclusive of local control, simulator, and navigation; all of which are used throughout the course of this project's development.

A. 2D Map Generation with LiDAR

The environment in which the robot is to move autonomously must be mapped in a teleoperator mode first when the robot uses sensors to scan the environment. The robot then navigates to the specified waypoint, mainly according to the map [10].

Since the LiDAR is attached to the Husky, both odometry and Inertial Measurement Units (IMU) data from the Husky are used for mapping with point clouds generated by the LiDAR. As a preliminary test for mapping, the RPLiDAR is used for generating a 2D map (see Figure 5). After generating a map, the robot can be moved to any desired position in the generated map.

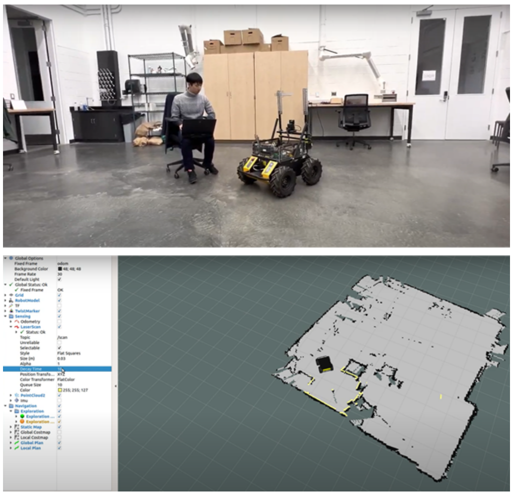

Figure 5. 2D mapping using UGV and LiDAR

A URDF file was created in order to assist RViz in pointing to the specifications of the Lidar. Specifications are inclusive of the Lidar's dimensions, location on the Husky robot, etc. The URDF file was created and referenced in the ROS setup.bash file using this command:

export HUSKY_URDF_EXTRAS=/home/doyun/rplidarA2_2.urdf.xacro

Once the URDF file is created and linked, RViz can now run using this command:

roslaunch rplidar_ros rplidar.launch

The Lidar readings can be shown on the RViz window by running this command in a different terminal window:

roslaunch husky_viz view_robot.launch

To start mapping, this command line needs to be executed in a different terminal window:

roslaunch husky_navigation gmapping_demo.launch

The main purpose of this command line is to use move_base with Gmapping to perform autonomous planning and movement with SLAM, on a simulated Husky with the laser scanning RPlidar to publish on the scan topic.

To save the generated map for later navigation and use, while still running the Gmapping command line, this command needs to be executed in a different terminal window:

rosrun map_server map_saver -f <filename>

B. Localization

Husky needs to be localized in the virtual environment on the generated 2D map. In this step, the viewing direction and position of Husky shown in RViz should be matched to the actual Husky machine. We assisted Husky in localization by giving it a rough idea of where it should be on the map using 2D pose estimate.

C. Navigation

Once the map has been imported into RViz and the Husky has been localized on the 2D map, it is ready to perform navigation. The adaptive Monte Carlo localization (ACML), which uses a particle filter to track the pose of a robot against a known map, was implemented for automated navigation in our research [5]. AMCL takes in information from odometry, laser scanner, and an existing map and estimates the robot's pose [6]. During operation, AMCL estimates the transformation of the base frame with respect to the global frame but it only publishes the transformation between the global frame and the odometry frame, as shown in Figure 6.

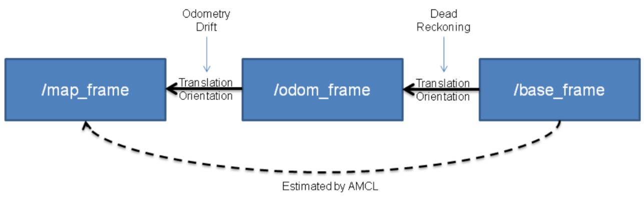

Figure 6: AMCL Map Localization

Once the goal pose and viewing direction are selected, Husky plans its trajectory and moves from the current pose to the selected goal pose using AMCL.

D. Obstacle Avoidance

Although the robot can be navigated to the specified goal point, it still needs to perceive its surroundings and in case of an incorrect location on the map or at the risk of collision with an obstacle. We used integrated reactive navigation for obstacle avoidance when the robot responds to conditions in the environment with its sensor system, including mapping data to increase safety. It is navigation using a local map of the environment with the lidar in this research [10]. It detects an obstacle on both the local and global map which is a pre-generated map in the mapping procedure. When it detects the obstacle within a distance of 1.5 m on either a local or global map, it stops and rotates continuously at the same position to replan the route, bypass the obstacle, and continue according to the map to the specified destination. This process can involve several iterations if necessary.

E. Communication with Other teams

Since the user needs to manually select the target position and orientation in RViz, there should be an error. Therefore, automatic position and orientation adjustment of the Husky based on the arm team's requirements is needed.

Integration between various teams is important for the development of the Husky welding robot. The main objective is to ensure a smooth communication network between all the separate systems functioning simultaneously as the robot operates. The UGV Lidar team will constantly communicate with the Arm team to give updates on when it is safe to move the arm.

The UGV Lidar team will move from the start location to the target location. During the duration of the UGV's travel, a 0 is sent to the Arm team to signal "unsafe to move." Once the target location has been reached, a 1 is sent to the Arm team to signal "safe to move, proceed". All communication is done with MQTT, a messaging protocol for IOT devices.

To send messages, the NVIDIA Jetson Nano for the UGV Lidar team is set as the publisher while the Arm team's NVIDIA Jetson Nano is set as the subscriber, as shown in Figure 7.

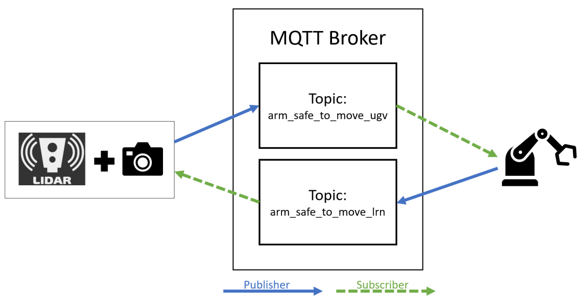

Figure 7: MQTT broker and topics connecting devices

For demonstration and testing purposes, when the up arrow on the computer is pressed, a 1 is sent, when the down arrow is pressed, a 0 is sent, and when the delete button is pressed, communications are terminated. The code can be viewed at the GitHub [11].

E. Pose & Orientation Adjustment

Since the user needs to manually select the target position and orientation in RViz, there should be an error. Therefore, automatic position and orientation adjustment of the Husky system based on the Arm team's requirements is needed, as shown in Figure 8. First, desired target position (x, y) and orientation (θ) are sent by the arm team. Based on the received desired target position (x, y) data, the required angle as well as the distance to the goal are calculated. Then the Husky rotates based on the calculated angle at the same place, and goes straight based on the calculated distance. Finally, the angle to the desired orientation (θ) is calculated and the Husky rotates. Since there is no rostopic that we can directly use for position and orientation adjustment, we developed a new Python script using the rostopic for moving the robot at the desired speed to rotate a certain angle and reach the 2D coordinate (x, y).

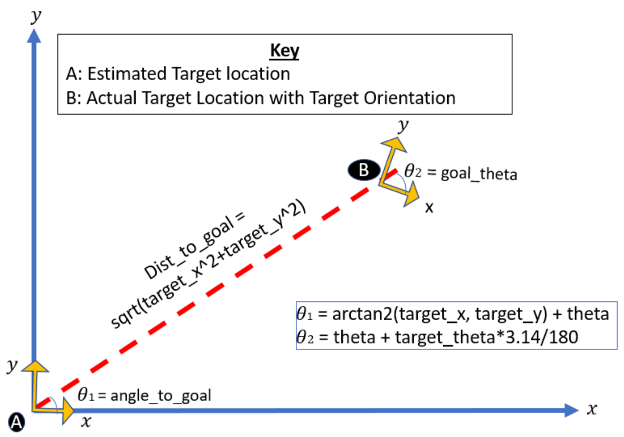

Figure 8: Schematic Design of Automatic Position and Orientation Adjustment

Figure 9 shows the detailed steps for automatic position and orientation adjustment when the received target position is (1, 1) and the orientation is 90 degrees.

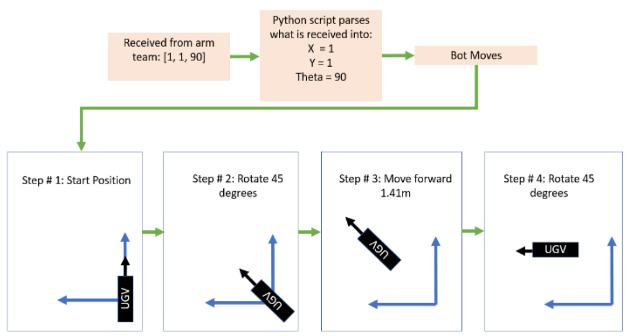

Figure 9: Sequential Steps of Automatic Position and Orientation Adjustment

IV. RESULTS

After running the Husky with the Lidar running and mapping activated, we were able to generate a 2D map of the space as seen in Figure 10.

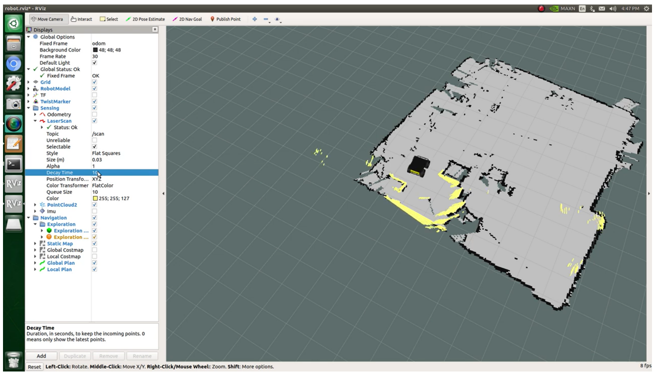

Figure 10: Final 2D Map using the 2D Lidar

There were many things we found that worked in generating an accurate 2D map using the Lidar. We found that it is best to:

1. Start in the middle of the room as opposed to a corner of the room where the Lidar cannot receive a full view
2. Move Husky slowly on turns to account for misreadings and delays image capture times
3. Stay at some points to wait for the Lidar to generate point clouds
4. Change the Lidar decay time from 0 to 10 sec. The decay time determines the rate at which the light is emitted following the excitation.

After generating the 2D map, we imported our saved map to RViz (the visualization software) to perform navigation. The Husky path planning can be seen in Figure 11.

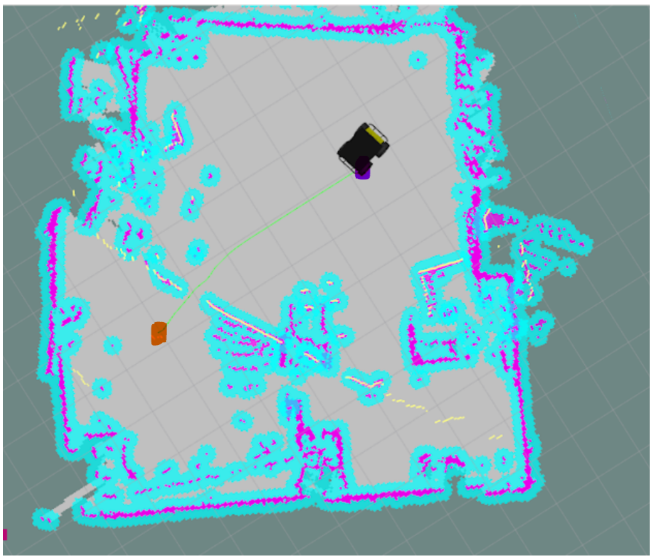

Figure 11: The start point for the Husky is point A (denoted by the orange dot) and the end goal is point B (denoted by the purple dot). The Husky planned a path from point A to point B (green line) after receiving a 2D navigation goal.

After path planning was finished, automated travel was conducted. As can be seen in figure 12, successful test trials were carried out to test for both single obstacle avoidance and navigation, as well as multiple obstacle avoidance and navigation Figure 13.

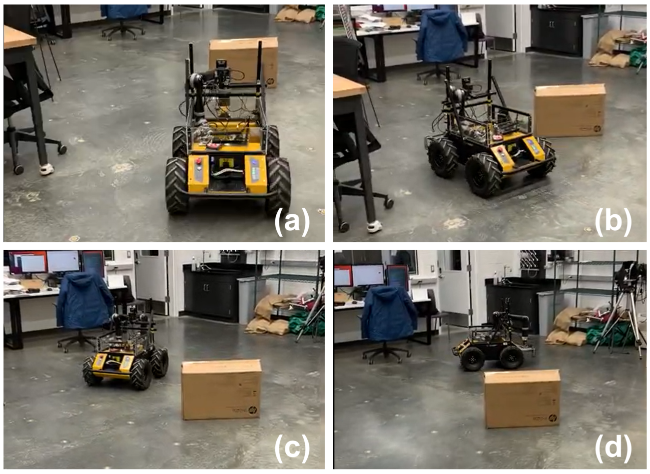

Figure 12: Obstacle Avoidance with Sequential Steps from (a) to (d)

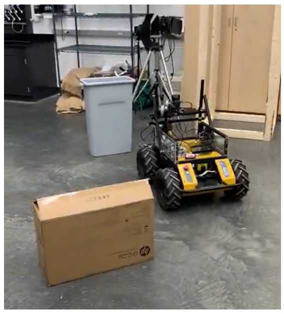

Figure 13: Multiple Obstacle Avoidance

V. FUTURE WORK

This system currently performs the necessary task of autonomous navigation and obstacle avoidance. A few things that proves to be a draw back is that the setup is in a controlled environment and to perform the necessary tasks many command lines must be ran in a specific order. In the future, to avoid this strenuous process, implementing a python code to run these command lines sequentially while waiting for different results would be more ideal.

References

[1] National Oceanic and Atmospheric Administration. (2021, February 21). What is lidar? U.S. Department of Commerce. https://oceanservice.noaa.gov/facts/lidar.html

[2] Belkin, Ilya, et al. "Real-Time LIDAR-Based Localization of Mobile Ground Robot." Procedia Computer Science, vol. 186, 2021, pp. 440–448., https://doi.org/10.1016/j.procs.2021.04.164.

[3] Xu, Xiaobin, et al. "Design of 2D LIDAR and Camera Fusion System Improved by Differential Evolutionary PID with Nonlinear Tracking Compensator." Infrared Physics & Technology, vol. 116, 2021, p. 103776., https://doi.org/10.1016/j.infrared.2021.103776.

[4] Lee, D., Lee, J. & Han, C.-H. (2018). A study of the influence factors on modular residential asset. ISARC 2018 - 35th International Symposium on Automation and Robotics in Construction and

International AEC/FM Hackathon: The Future of Building Things, Isarc. https://doi.org/10.22260/isarc2018/002

[5] ROS Wiki. http://wiki.ros.org/amcl

[6] Clearpath Robotics. "Jackal Tutorials 0.6.0 documentation" https://www.clearpathrobotics.com/assets/guides/noetic/jackal/navigation.html

[7] C. -g. Lu, J. -x. Wang and X. Cui, "Research and Implementation of Unknown Environment Map Construction Method with Lidar-based Robot," 2020 5th International Conference on Automation, Control and Robotics Engineering (CACRE), 2020, pp. 26-30, doi: 10.1109/CACRE50138.2020.9230033.

[8] R. Lankin, K. Kim and P. -C. Huang, "ROS-Based Robot Simulation for Repetitive Labor-Intensive Construction Tasks," 2020 IEEE 18th International Conference on Industrial Informatics (INDIN), 2020, pp. 206-213, doi: 10.1109/INDIN45582.2020.9442192.

[9] J. Zhu and L. Xu, "Design and Implementation of ROS-Based Autonomous Mobile Robot Positioning and Navigation System," 2019 18th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), 2019, pp. 214-217, doi: 10.1109/DCABES48411.2019.00060.

[10] Dominika Krajčovičová (2019), "We have developed autonomous map navigation and obstacle avoidance", https://robotechvision.com/autonomous-navigation-using-map-and-obstacle-avoidance/

[11] Github. https://github.com/britwright/sp22Robot