LIDAR Navigated Robot Car

Tianhao William Jiang, Kevin Lin, Sharfaraz Ahmed, Ryan Yuen
Supervisor: Dr. Siyuan He

Abstract
This project entails the design of a LIDAR navigated robot car based upon the Mechbot used in the MEC733 course. By implementing LIDAR, this project will aim to improve the capabilities of the Mechbot, which currently uses infrared (IR) for navigation. Utilizing the Garmin LIDAR-Lite v3HP, a stepper motor, and Arduino microprocessor, a scanning mechanism was built, allowing for a live map of the unit's surroundings to be generated. This assisted the Mechbot to navigate its surroundings more accurately, due to the high resolution offered by the LIDAR system. The stopping, turning, and reaction time of the Mechbot have all improved compared to the original IR system. In conclusion, the team has accomplished the goal of designing a LIDAR navigated robot car, providing students in upcoming MEC733 courses with a more versatile learning experience.

1. Introduction

The objective of this project is to improve upon the outdated navigation system of the Mechbot, which currently uses IR. To begin tackling this design, the team will explore different scan widths, motor choices, as well as interfacing.

A prototype will be constructed using off the shelf parts in order to test the feasibility of implementing LIDAR on the Mechbot.

2. Literature Review

LIDAR stands for Light Detection and Ranging. [1] In essence, LIDAR sends out a light pulse from an emitter and then receives the reflected pulse from the environment using a receiver in the same package. With the combination of some processing and computing, the distance of the object off of which the light pulse reflected back to the LIDAR can be determined. By obtaining a large number of data points, a map of the surface can then be generated. A simple LIDAR unit typically consists of a laser and a receiver.

Stepper motors allow a simple motor to have very precise angular movements. This property is very useful in LIDAR applications, where precise control of position is necessary. Stepper motors also allow the distance that the motor has travelled to be calculated by the Arduino, by counting the number of times that the coils have been actuated (steps).

In order for the data from the LIDAR to be used, a map must be created. This map allows the vehicle to navigate through the given course autonomously. Conventional LIDAR devices usually have very slow sample rates for high-speed applications [1]. For the application of a robot car, a slow sample rate is adequate.

3. Final Design

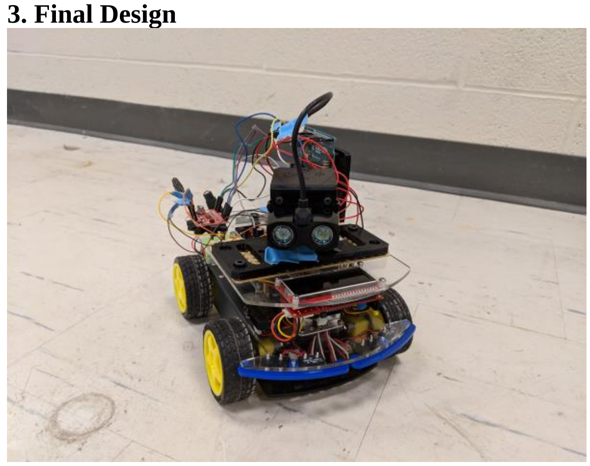

Figure 1: Prototype of Final Design

The prototype design consists of the following components:

- Garmin LIDAR-Lite v3HP
- NMB Stepper Motor
- Sparkfun Big Easy Driver
- Reflective sensor (2 LEDs)
- 3D printed mounting plates
- Miscellaneous nuts and bolts
- NiMH square battery pack

The cost of these components was a total of approximately $350 CAD.

With the LIDAR in place, it had to be spun around by the stepper motor of choice. Size of the motor had to be inside an area of 370 mm by 65 mm. The height of the motor had to be as low as possible in order to keep the LIDAR mounted close to the ground.

The NMB stepper motor chosen has a step angle of 0.198 degrees per step. Due to limitations from the speed of the motor, the motor must be stepped 9 times before the LIDAR is ready to take a reading. This would decrease our resolution from the theoretical max of 0.36° per reading to 1.782° per reading.

However, this is still adequate for the application and is ideal than operating the LIDAR setup at the theoretical limit since it tends to produce unreliable data when the LIDAR is operated at those speeds.

To control the stepper motor, the Sparkfun Big Easy Driver was chosen, due to the fact that it matched the current and voltage requirements of the NMB stepper motor. LEDs were also utilized to calibrate the LIDAR unit. All of these components were run off two Arduino microprocessors, one for the LIDAR portion, and one for navigation.

4. Implementation and Interfacing

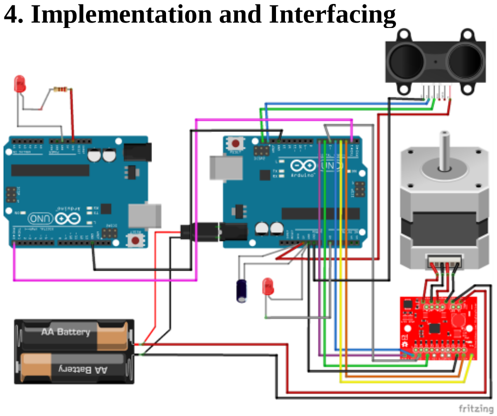

Figure 2: Standard I2C wiring for the LIDAR unit to the Arduino Uno

To interface the LIDAR unit with the Arduino, the I2C protocol was used. The wiring diagram above shows how the LIDAR unit was wired to the Arduino. To transmit the data obtained from the LIDAR, the two Arduinos communicated via the RX and TX serial communication ports.

5. Software

The main program used for coding and control of the LIDAR system was the Arduino IDE. This is largely because the controller used was the Arduino R3. Processing was for the display of the readings because of its ease of use to display serial readings to a graph.

The code for the controller for the LIDAR system had the following design points in mind; it had to be easily implemented regardless of the motor specifications, it had to be usable without encoders, it had to be capable of varying the speed of the motor which should not affect the accuracy of the camera, and it had to comply within the memory of the Arduino. To satisfy the design points, constants were used so that only they would have to be changed if any adjustments were made. The constants chosen were STEPANGLE, MAXSTEPS, DELAYAMOUNT, STEPSPERREADING, and INITANGLE.

The logic for the motor speed is determined in the LIDAR code. The reading is calculated as the max distance directly in front of Mechbot and based on a comparison to the constants STOPDISTANCE and SLOWDISTANCE, a character is written to Serial which will be read by the Mechbot. The Mechbot has a simple if-else statement and controls the motor speeds from the Serial.

6. Testing and Results

In order to qualify the performance of the LIDAR unit, a testing procedure was devised. This procedure involved placing objects within the scanning area of the LIDAR unit and comparing it to the map generated by Processing.

Figure 3: Setup for static testing

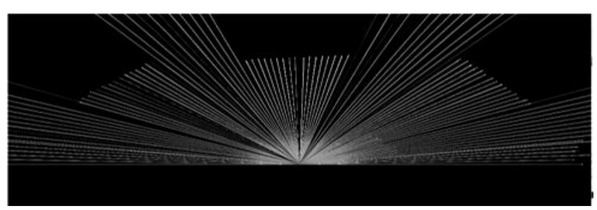

Figure 4: Real-time map

After obtaining positive test results as shown in the figures above, the next step was to test the Mechbot with real life obstacles. Programmed to slow down and stop when the LIDAR detects an obstacle within range, the following figures depict the Mechbot reacting to an obstacle while it is in motion.

Figure 5: Mechbot detecting a pedestrian crossing the hallway

Figure 6: Mechbot stopping for a pedestrian crossing the hallway

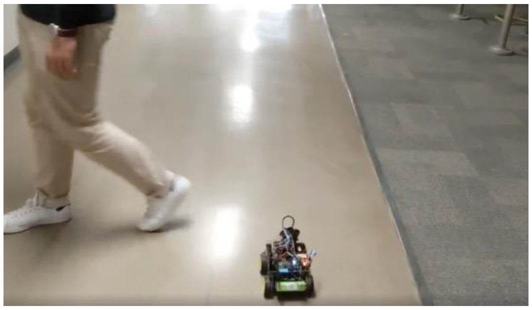

Figure 7: Mechbot resuming motion after obstacle is not in range anymore

7. Conclusions and Recommendations

In conclusion, the goal of designing and creating a LIDAR navigated robot car was successful, with quantifiable results as shown in the results section above. Future students in the MEC733 course will be able to duplicate the setup and implement LIDAR in their projects.

Throughout this project, the group was constantly refining the design in order to increase sweep speed, which would increase the accuracy of the LIDAR. This was one of the main challenges that the group faced. There are many ways to increase the sweep speed, but for this project, the group focused mainly on software changes, due to time constraints from ordering new hardware. For future recommendations, a motor with a larger step size can be used to increase the sweep speed. Using a DC motor could also be explored, but this would require an encoder to be added. At the moment, the stepper motor is rated to run at a speed up to 1kHz however in practice, only about 800Hz could be achieved.

In terms of navigation, the Mechbot was successful in accelerating and stopping in a straight line. Further refinement would allow the Mechbot to navigate around corners and obstacles, such as the maze in the MEC733 course. To increase the Mechbot's mapping and navigation accuracy, a 360° sweep angle could be implemented, through the use of a slip ring and more powerful motor. This would allow simultaneous location and mapping (SLAM) to be utilized. In a SLAM setup, the robot knows its live position within the map, which allows it to proactively navigate accordingly to the obstacles it knows are ahead, as opposed to operating in a reactive manner when something appears in front of the robot, such as the current setup.

References

[1] B. Schwarz, "Lidar: Mapping the world in 3D," *Nature Photonics*, vol. 4, *(7)*, pp. 429-430, 2010. Available: https://search-proquest-com.ezproxy.lib.ryerson.ca/docview/873168000

[2] I. Virgala et al, "Stepper Motor Control by ATMEL AVR Microcontroller," Applied Mechanics and Materials, vol. 816, pp. 321-326, 2015. Available: http://ezproxy.lib.ryerson.ca/login?url=https://search-proquest-com.ezproxy.lib.ryerson.ca/docview/1903481789?accountid=13631. DOI: http://dx.doi.org.ezproxy.lib.ryerson.ca/10.4028/www.scientific.net/AMM.816.321.

[3] D. Reiser, M. Vázquez-Arellano, D. S. Paraforos, M. Garrido-Izard, and H. W. Griepentrog, "Iterative individual plant clustering in maize with assembled 2D LIDAR data," *Computers in Industry*, vol. 99, pp. 42–52, 2018. Available: https://www-sciencedirect-com.ezproxy.lib.ryerson.ca/science/article/pii/S0166361517304748