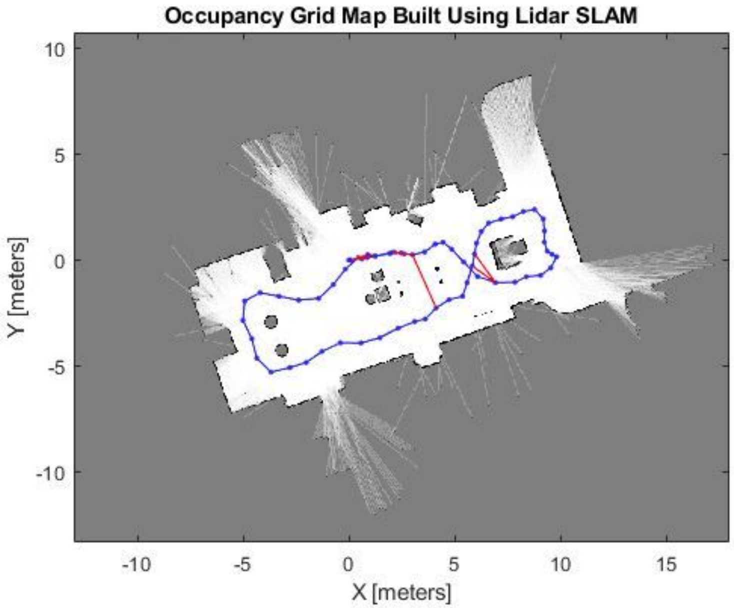
Developing a robot that autonomously collects LIDAR data and improving loop closure detection in SLAM

Gabor Szita

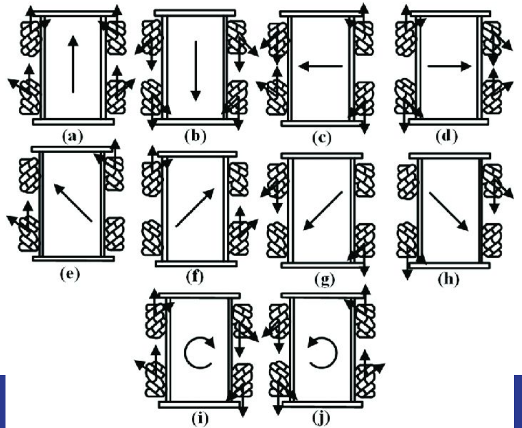
What is SLAM?

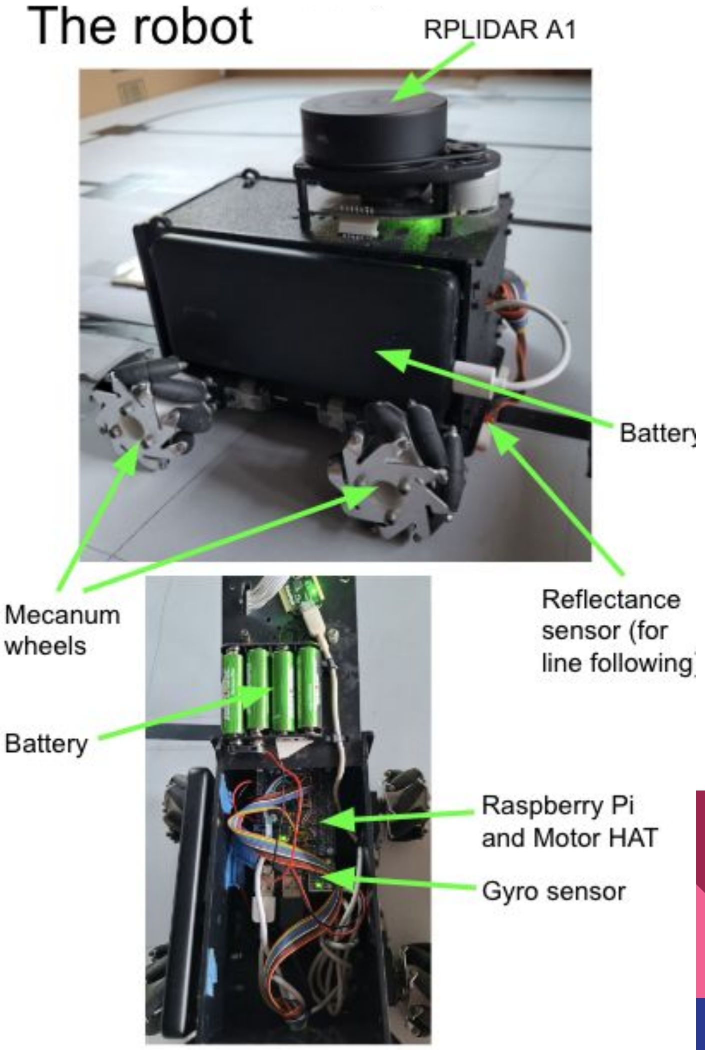
- Simultaneous Localization and Mapping
- Localization: keeping track of a robot's position
- Mapping: saving the environment of a robot
- Devices that can be used for SLAM
    - Radars
    - LIDARs
    - Cameras



The robot for data collection

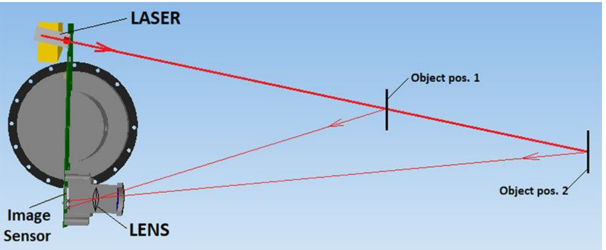
- Three main sensors:
  - Lidar
  - Gyro
  - Camera from AprilTag detection
- Raspberry Pi computer of robot
- Mecanum wheels allow movement in every direction:



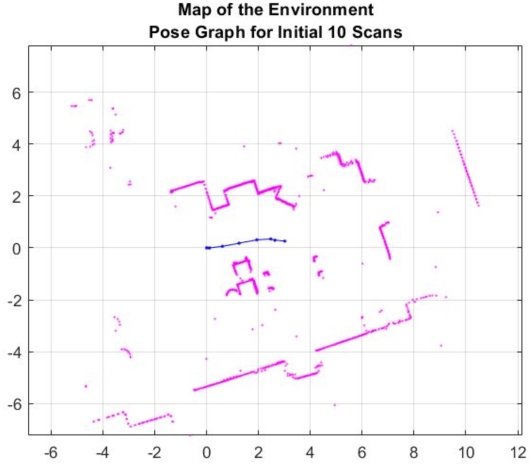


What is a LIDAR?

- LIDARs emit pulses of light and measure time for the light to bounce back from an object. Based on this time it can calculate how far the object is.
- LIDARs also rotate, so they “see” in every direction



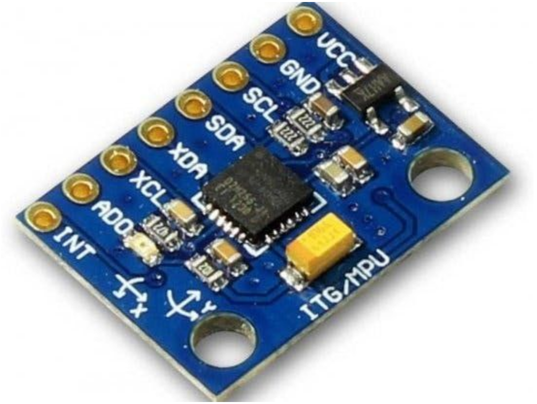
Example of what a LIDAR sees:



What is a gyroscope (gyro)?

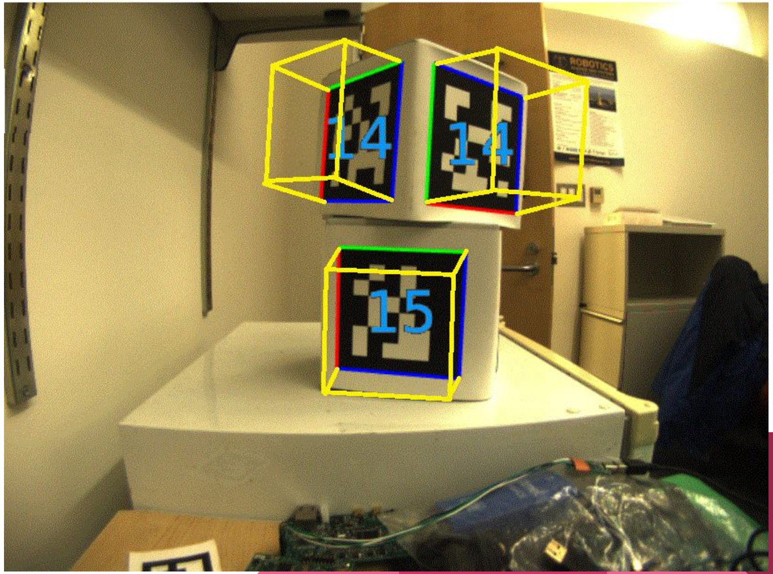
- A gyro measures the rotation and acceleration of the robot
- I.e. if the robot turns, the gyro will detect how many degrees the robot turned

I'm using the MPU6050 gyro:



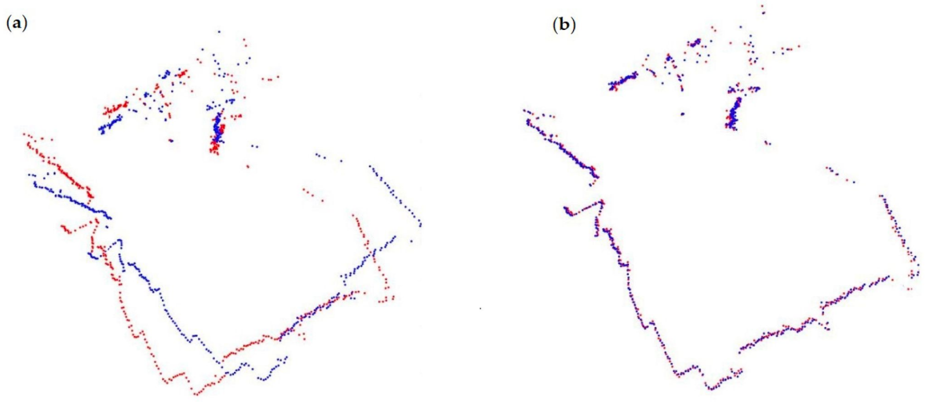
What are Apriltags?

- Apriltags are like QR codes
- If the robot sees an Apriltag and knows the position of the Apriltag, the robot can determine its position



My project

- Making SLAM localization more accurate
- ICP algorithm determines transformation between one LIDAR reading and another
    - By determining the transformation between each subsequent LIDAR reading, you can determine the robot's position
- Problem with SLAM is that it makes small error
    - Sensors make errors
    - Point clouds cannot be aligned with 100% accuracy
- Accumulation of these errors results in drift

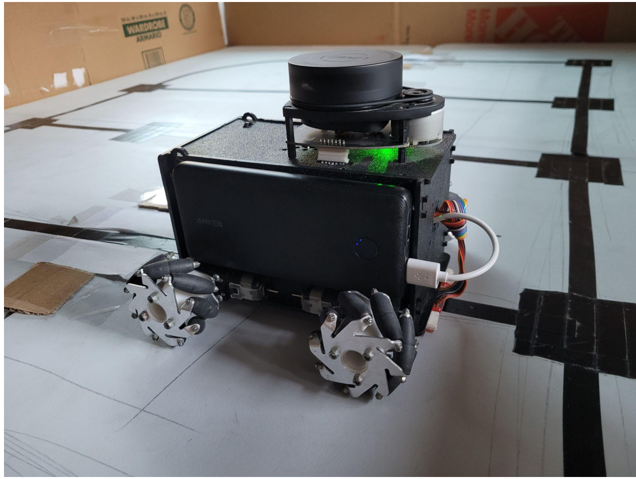


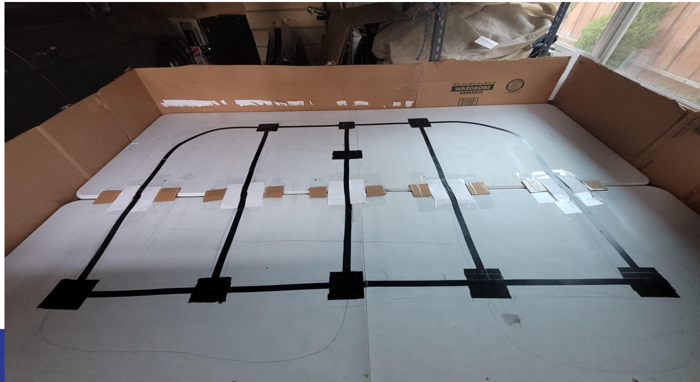
For example:

- Transformation between the first and the second LIDAR reading: (2, 0)
- Transformation between the second and the third LIDAR reading (3, 1)
- Transformation between the third and the fourth LIDAR reading: (0, -1)
- -> Robot's position: (2, 0) + (3, 1) + (0, -1) = (5, 0)

How I do my research

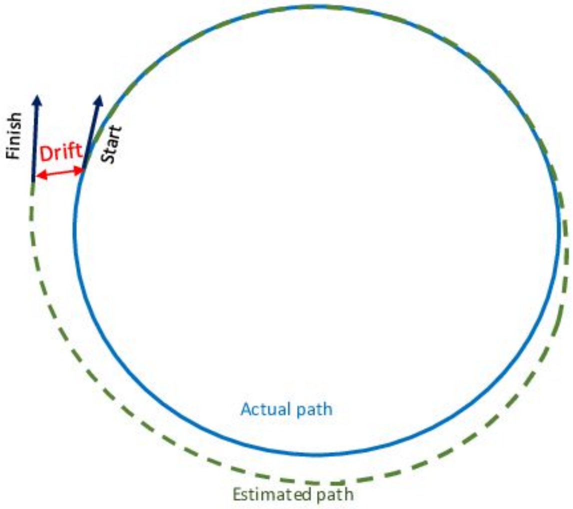
- First robot drives a path and collects data
    - It was a lot of work to get the robot reliably running on the path because it often made a mistake and went off the path, so I made an algorithm that automatically drives the robot back to the start position if it drives off the path
- After the robot finished driving on the path, I transfer the log file to my computer, which contains the data from all sensors
- I use MATLAB with MATLAB's Lidar Toolbox to analyze the data on my computer





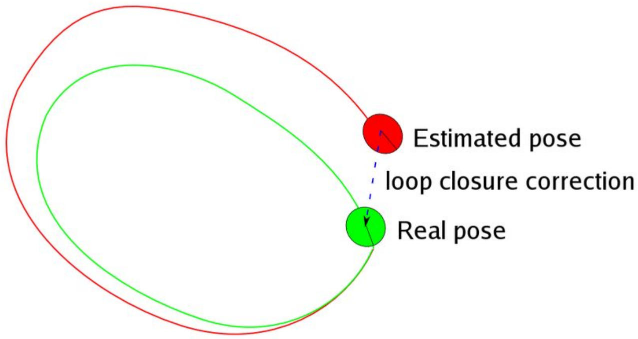
My project (initial plan)

- The more the robot drives, the more drift is, and the more inaccurate the robot position is
- You never know with 100% accuracy where the robot is, you always just have a good estimation
    - The robot position from the ICP algorithm is not robot's actual position, it is only the best estimation
- My goal was to estimate how much this drift can be (drift bound) using the machine learning
- By estimating the drift bound, you know the region where the robot probably is
    - For example, if the ICP algorithm calculates that the robot is at the (10, 10) coordinate and the drift bound is 2 meters, then you know that the robot is probably somewhere within 2 meters of the (10, 10) coordinate



My project (initial plan)

- The drift bound can be used for loop closure:
  - When a robot returns to a position it has already visited, it can correct this drift
  - Many ways to do loop closure, one method is to check if its environment is similar to a previously visited environment
  - Environments can look similar e.g. two rectangular rooms with the same dimensions -> How does it know if this is an already visited position, or another place with a similar environment?
  - Using the drift bound, it can determine if the difference between the calculated current position and the location of a previously visited location is reasonable
- This was very hard to do and I faced many challenges, so I decided to take a different approach

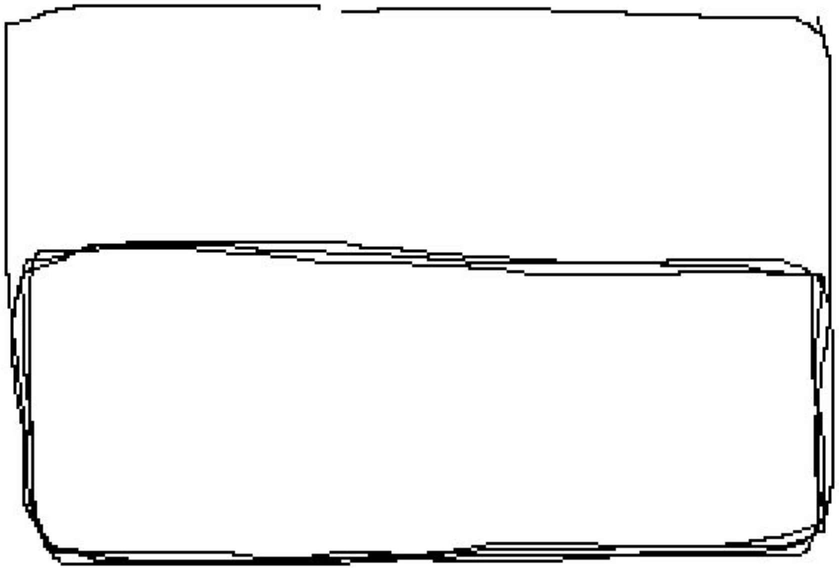


My project (what I actually did)

- One way to make SLAM more accurate is by combining data from multiple sensors
- I decided to combine data from the robot's Lidar, gyro, and AprilTag detection to improve robot localization

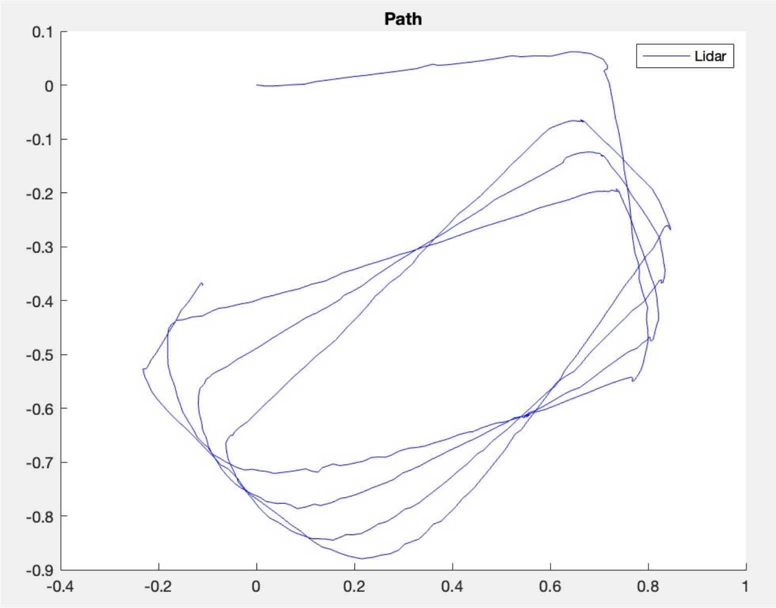
Results

Path the robot drove on (ground truth):

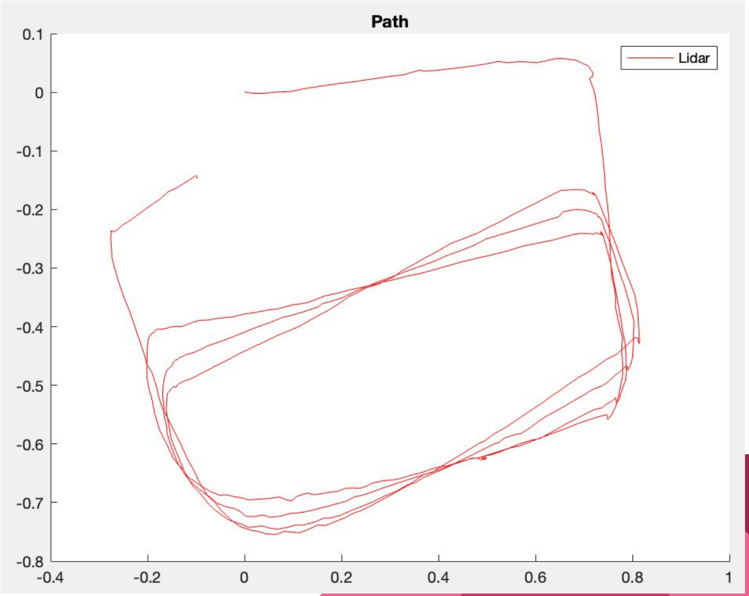


Results

Using only data from the Lidar to calculate the path:

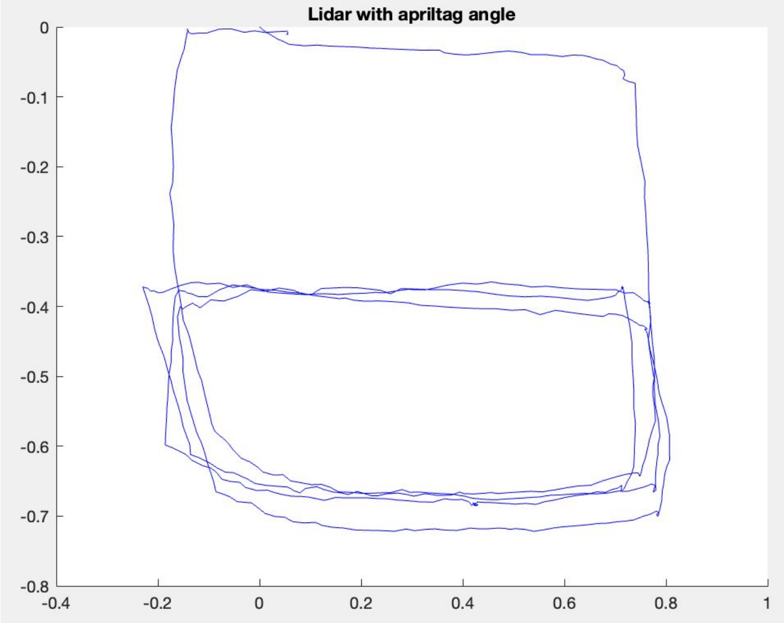


Calculating the path by combining Lidar data with rotation data from the gyro:



Results

Calculating the path by combining Lidar data with rotation data from the AprilTags:



Conclusion

- When I used data only from the Lidar to calculate the path of the robot, the path was very inaccurate because the ICP algorithm wasn't able to accurately determine the rotation at sharp turns.
- When I combined data from the Lidar with rotation data from the gyro sensor, the accuracy of the path improved, but it still wasn't perfect.
- When I combined data from the Lidar with rotation data from the AprilTags, the path was very accurate.
- Hence, I demonstrated that by combining data from multiple sensors the accuracy of SLAM can be improved.

Where can my research be used?

- Self-driving cars: Self-driving cars need to know their location accurately and also have multiple sensors, so my research could be used as an example of how to improve localization of self-driving cars by using data from multiple sensors
- Warehouse robots: Robots in large warehouses also have multiple sensors and need to know their location in the warehouse

Code

- Project's code is on GitHub
- Robot code: https://github.com/gaborszita/asi-project-robot-code
- Data analysis (MATLAB) code: https://github.com/gaborszita/asi-project-matlab-code

References

1. J. Chen, Y. Gao and S. Li, "Real-time Apriltag Inertial Fusion Localization for Large Indoor Navigation," 2020 Chinese Automation Congress (CAC), Shanghai, China, 2020, pp. 6912-6916, doi: 10.1109/CAC51589.2020.9326501.
2. 4. Corso, N.; Zakhor, A. Indoor Localization Algorithms for an Ambulatory Human Operated 3D Mobile Mapping System. Remote Sens. 2013, 5, 6611-6646. https://doi.org/10.3390/rs5126611
3. Xu, X.; Zhang, L.; Yang, J.; Cao, C.; Wang, W.; Ran, Y.; Tan, Z.; Luo, M. A Review of Multi-Sensor Fusion SLAM Systems Based on 3D LIDAR. Remote Sens. 2022, 14, 2835. https://doi.org/10.3390/rs14122835
4. Y. K. Tee and Y. C. Han, "Lidar-Based 2D SLAM for Mobile Robot in an Indoor Environment: A Review," 2021 International Conference on Green Energy, Computing and Sustainable Technology (GECOST), Miri, Malaysia, 2021, pp. 1-7, doi: 10.1109/GECOST52368.2021.9538731.
5. "Introduction to Loop Closure Detection in SLAM." Think Autonomous, 18 Apr. 2023, www.thinkautonomous.ai/blog/loop-closure/. Accessed 20 Aug. 2023.