Mapping AprilTags and Evaluating the Robustness of a Novel Vision-Based Feedback Controller

Liam Rivers[1,2], Mehdi Kermanshah[2], Roberto Tron[2]

Gilman School, 5407 Roland Avenue, Baltimore, MD 21210[1], Boston University Robotics Laboratory, 750 Commonwealth Avenue, Boston, MA 02446[2]

Introduction

Background

- Most classical path planning methods plan a single path based on a pre-existing map of an environment.
- A new output-feedback controller proposed by Bahreinian et al. (2024) focuses on controller generation rather than the generation of a single path.[1] The idea behind this approach is to decompose landmarks into convex regions called cells and specify an output edge for each of these cells which can be used for controller generation. The output of the controller can be seen in Fig. 1.
- This, in theory, is more robust to errors present in the map used to generate the controller, which makes landmark-based navigation more accessible to robots with poorer equipment.

Objective

- The goal of this project is twofold:
  - Design an algorithm using Python and the Robot Operating System (ROS) that maps the positions and rotations of AprilTags (Fig. 2) that are positioned around the robot, Clearpath Jackal (Fig. 3).
  - Use this mapping algorithm to analyze the robustness of the output-feedback controller.[2]

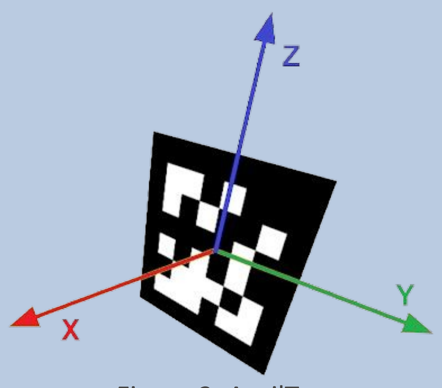

Figure 2. AprilTag

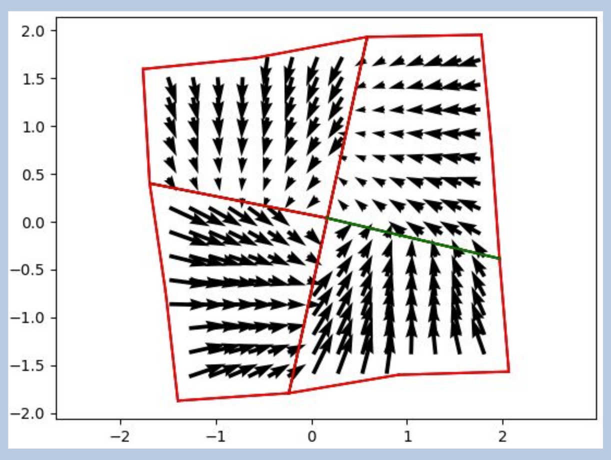

Figure 1. Vector field showing output of output feedback controller

Figure 3. Clearpath Jackal

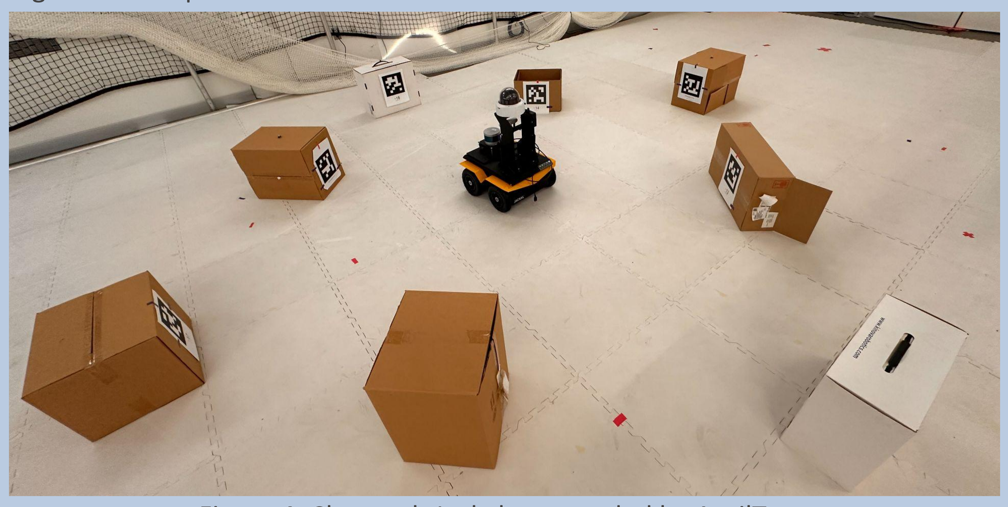

Figure 4. Clearpath Jackal surrounded by AprilTags

Results

The positions of the vertices of the map are altered by the formula $(x_i, y_i) = (x_i + (r - 0.5) \cdot w, y_i + (r - 0.5) \cdot w)$ where r is a random float on the interval [0,1) and w is the weight of the noise. The success rate of the controller with various weights of noisy data is shown in Fig. 8. I defined a successful controller as one that could navigate through all the cells without colliding with an AprilTag or getting stuck.

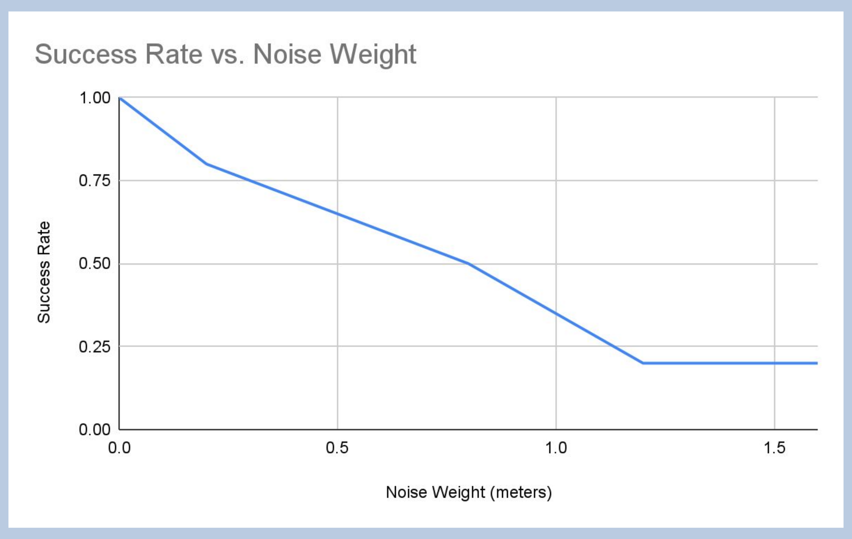

Figure 8. Success rate of controller in real life when map is exposed to different weights of noise

Noise Weights (meters)	0.1	0.2	0.4	0.8	1.2	1.6
Success Rate	.9	.8	.7	.5	.2	.2

Table 1. Success Rate of controller when generated with a map with varying degrees of noise

In order to evaluate the accuracy of the mapping algorithm, I set up the AprilTags in a square with a set size of the diagonal. For various squares with various diagonal lengths, I generated 5 maps each and calculated the average accuracy. Accuracy is defined as the average distance between the observed positions of the AprilTags and the actual positions of the AprilTags.

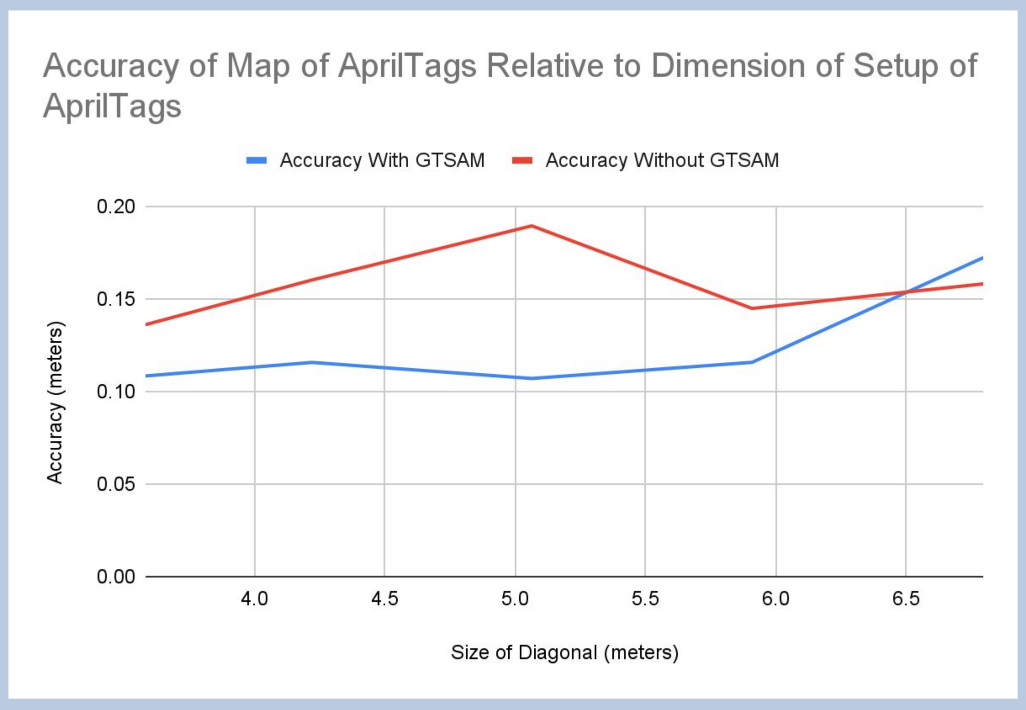

Figure 9. Accuracy of optimized and unoptimized maps when run on different sized square environments

Size of Diagonal (meters)	3.5814	4.2208	5.064	5.909	6.796
Accuracy w/o GTSAM (meters)	0.1363	0.16048	0.18974	0.14508	0.15836
Accuracy w/ GTSAM (meters)	0.10858	0.1159	0.1072	0.11594	0.1726

Table 2. Accuracy of mapping algorithm in different sized environments

Methods

- Mapping Algorithm
  - **Goal -** Find positions and rotations of the AprilTags surrounding the robot in the same arbitrary reference frame called the world frame
  - Robot aligns with first tag and begins spinning.
    - While spinning, the robot saves the rigid body transformations between each pair of adjacent AprilTags.
    - When it gets the rigid body transformation from the last AprilTag to the first AprilTag, the robot stops. This is called a loop closure.
  - Robot calculates the positions of all of the tags in the world reference frame by chaining the transformations between adjacent AprilTags together as seen in Fig. 5. The yellow lines represent rigid body transformations and the vertices represent the AprilTags.
  - Robot uses the Georgia Tech Smoothing and Mapping (GTSAM) library to correct the map based on the difference in the world position of the first tag before and after spinning in a circle.[3,4] Note in Fig. 6 how before mapping, the final tag does not line up with the first tag despite them being an estimate of the same AprilTag's position. GTSAM corrects this error as seen in Fig. 6.

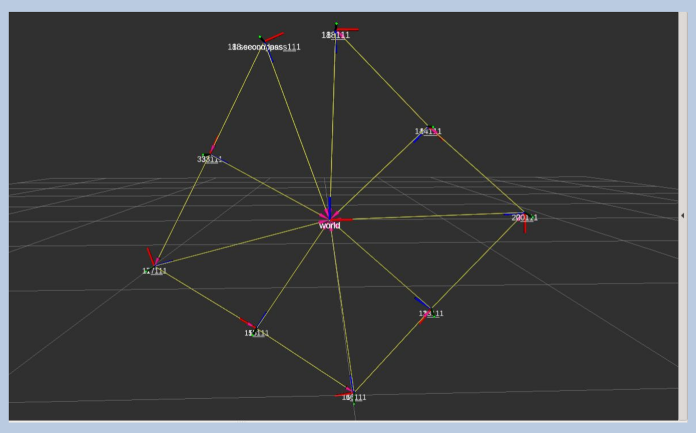

Figure 5. Mapped positions of AprilTags arranged in diamond before optimization with axis

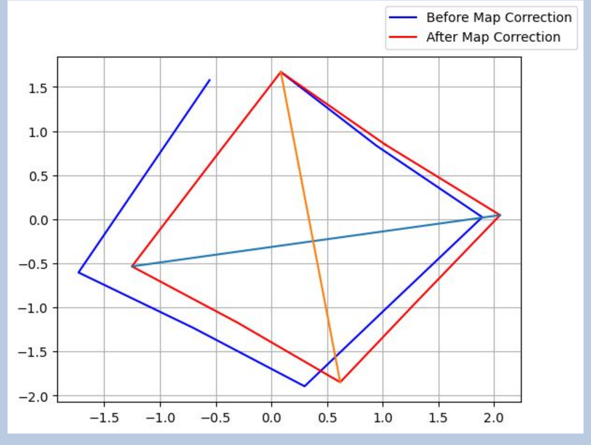

Figure 6. Mapped positions of AprilTags arranged in diamond before and after optimization

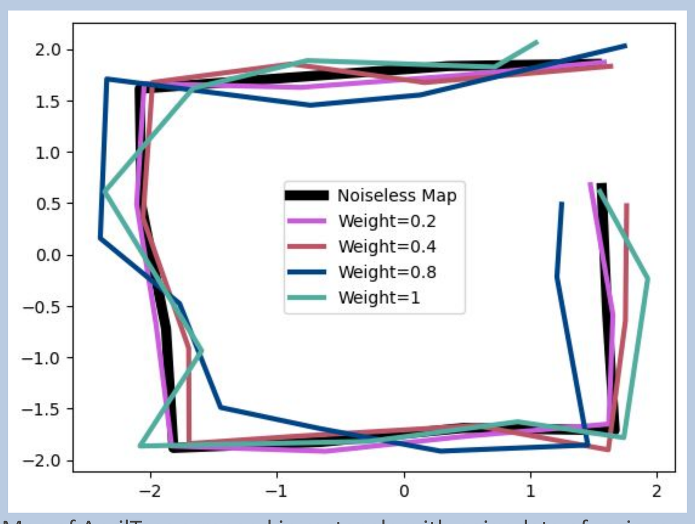

Figure 7. Map of AprilTags arranged in rectangle with noisy data of various weights superimposed

- Measure of Controller Robustness
  - Output feedback controller takes vertices and their rotations as input
  - In order to measure the controller's robustness, noise is artificially induced in the estimate of the AprilTag positions, and the success rate of the controller is determined. I induce noise by shifting the positions of the AprilTags by a random amount whose magnitude is determined by a weight value.
  - Example of the effect of varying degrees of noise can be seen in Fig. 7.

Conclusions

Controller Robustness

The output-feedback controller proved to be robust to what would normally be considered large amounts of error in the map used to generate it. Even when the vertices varied by up to 40 cm from the original position (see Fig. 7), the controller still managed to navigate the environment successfully 50% of the time. These results indicate that this controller could be used in situations where landmark positions are not known exactly or where the map is not static.

However, the controller failed 80% of the time when the positions of the AprilTags were up to 60 cm or up to 80 cm away from the actual positions of the AprilTags. The primary manner in which the controller failed was that it would receive an instruction from one tag to turn in one direction. This turn would bring another tag into the field of view of the camera that instructed the robot to turn back toward the first tag. This caused a loop to ensue, causing the controller to fail.

Mapping Accuracy

The accuracy of the mapping algorithm was greatly improved by the GTSAM library until the distance of the tags from the robot increased, which is most likely due to the larger amount of noise at that distance. (Fig. 9)

Future Directions

- Find a way to overcome bug with controller where it gets stuck in loop
- Investigate whether controller can operate with a map made by a cheaper and less reliable camera to test the effect of uncontrolled noise on the controller's function.
- Use features in the robot's environment as landmarks rather than AprilTags, which give explicit positional information, to make controller more applicable in real-world scenarios.

References

1. Bahreinian, M., Kermanshah, M., & Tron, R. (2024). Designing Robust Linear Output Feedback Controller based on CLF-CBF framework via Linear~ Programming (LP-CLF-CBF). arXiv preprint arXiv:2403.14519.
2. Rivers, L. (2024, July 29). apriltag_mapping. Bitbucket. https://bitbucket.org/burobotics/apriltag_mapping/src/main/
3. Rivers, L. (n.d.). *Rivers RISE 2024*. https://riversrise2024.blogspot.com/
4. GTSAM. (n.d.). https://gtsam.org/

Acknowledgements

I am extremely grateful to everyone who has made this research project possible. First and foremost, I'd like to thank Professor Tron and Mehdi Kermanshah for their expert guidance which has allowed me to learn so much. You have made this experience so valuable for me and for that I am incredibly grateful. I would also like to thank my parents for always supporting me and allowing me to pursue my interests at RISE. And finally, I would like to thank the RISE program for providing me with such an enriching experience.