Localization and Mapping for Subsea Docking

Daniel Lines,[1] Aaron Marburg[2]

[1] Cornell University, Ithaca, NY dal337@cornell.edu
[2] University of Washington Applied Physics Laboratory, Seattle, WA amarburg@uw.edu

Abstract

From monitoring tidal turbines to surveying telecommunication cables, subsea wireless charging and data transfer is widely applicable. The main advantage of these systems are their independence from a tether to a topside computer. Naturally, this raises the question of how to control the ROV without human input. The goal of this project is to achieve a reliable solution to one of side of this question, autonomous docking.

The BlueROV Heavy was the ROV platform used for this project. Constrained by the compute power of the board Raspberry Pi 4, a lightweight solution for localization was needed. Our approach was to use GTSAM to first map the AprilTag marker-rich environment and then use trilateration to localize the ROV.

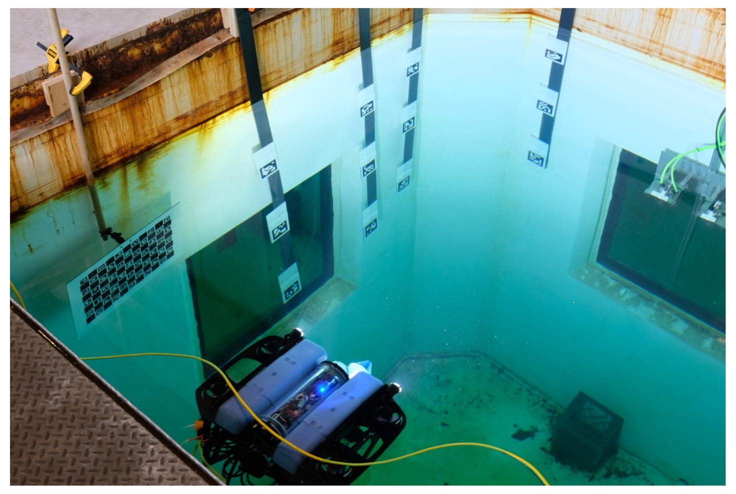

Figure 1. Testing and Data collection in saltwater tank

Methods

System Design

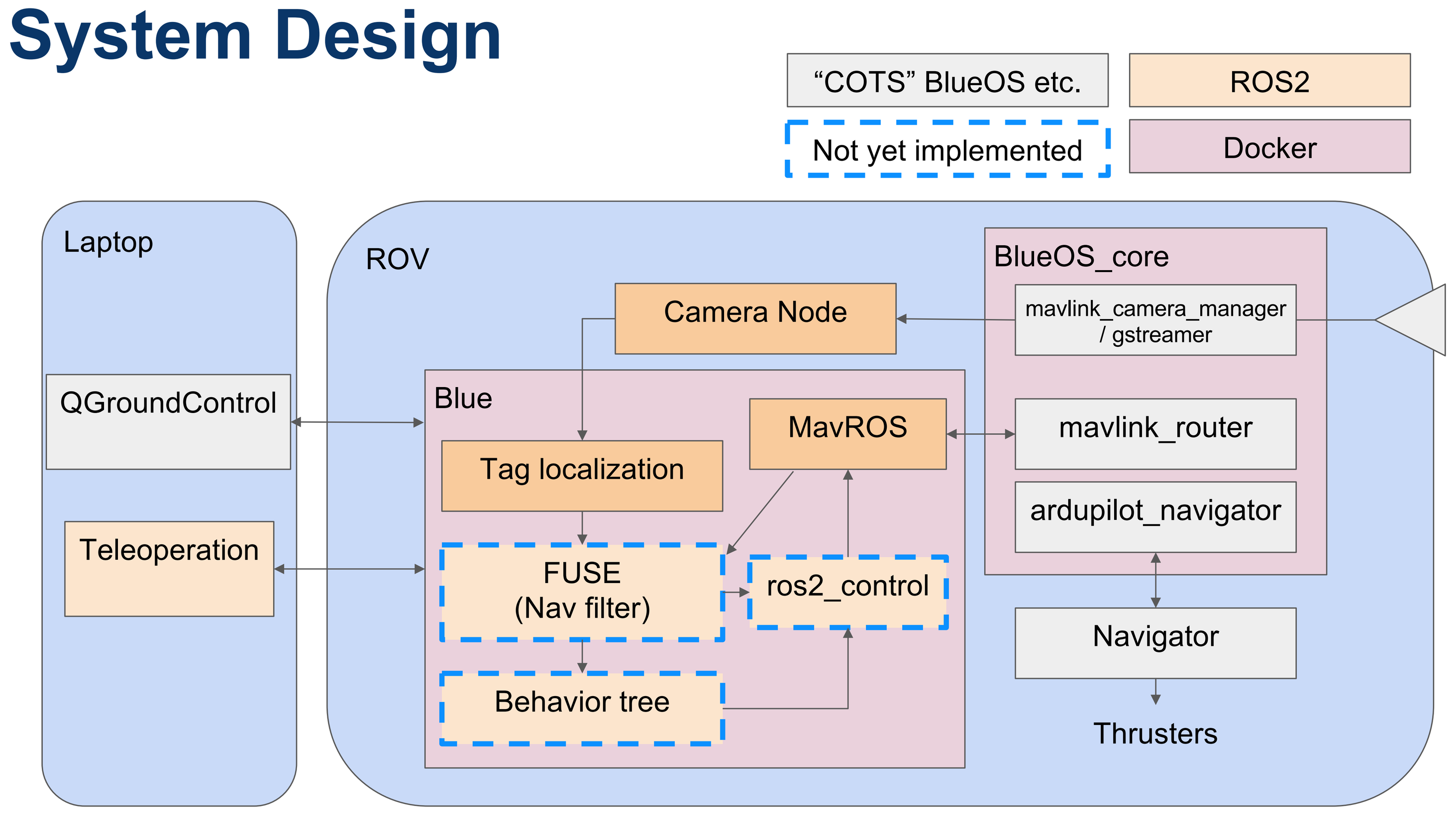

Figure 2. The diagram depicts the relationships between each elements of the system. Note that the ROV can run in headless mode which excludes the elements under the "Laptop" section.

Ros Architecture

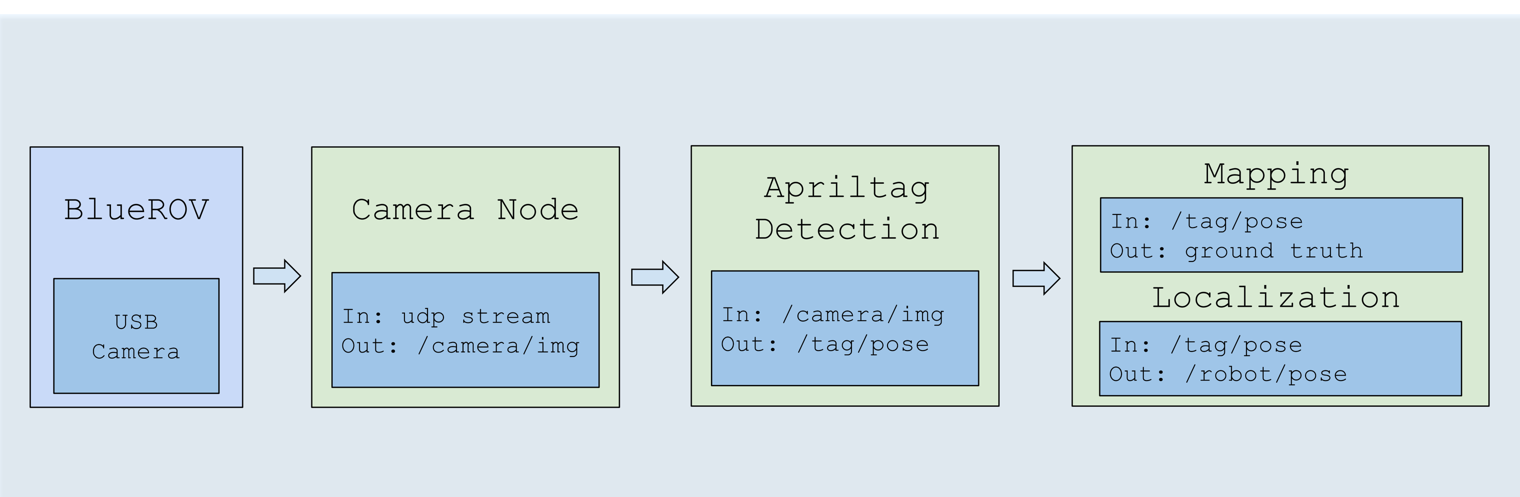

Figure 3. From left to right, the camera stream is converted into an image topic, any AprilTags in the image are detected, using the detection(s) either the SLAM algorithm is updated, or a localization is performed.

Mapping (SLAM)

GTSAM is a tool for building and optimizing factor graphs. A factor graph is composed of variables and with factors between them and can be optimized to find the most probable value for each variable. Our SLAM algorithm uses GTSAM to optimize a factor graph with:

Variables:

$X : \{x_1, \ldots, x_n\}$: where x_i represents the robot pose in the ith frame

$T : \{t_1, \ldots, t_{12}\}$: where t_i represents the pose of the ith AprilTag

Measurements:

Z : $\{z_{1,1}, \ldots, z_{n,12}\}$ where $z_{i,j}$, is distance from the robot at x_i to the AprilTag t_j

Factors:

$$F : \{f_{distance}(x_i, t_j, z_{i,j}) | \forall z_{i,j} \in Z\}$$

Objective Function:

$$Maximize\ P(X, T|Z)$$

The approach of using a SLAM algorithm allowed for both a mapping (ground truth values for AprilTag poses) of the fiducial environment to use for localization and ground truth values for the robot pose over time to test the localization algorithm. Additionally, this system provides forward compatibility for a full SLAM solution.

Localization

Once the environment has been mapped, we can use the ground truth poses of the AprilTags to estimate the ROV pose for each incoming frame. Trilateration proved to be an effective method for this task. The residual function for the algorithm (p: possible pose, t: ground truth, d: observed distance) is defined as:

$$r(p,t,d) = \sqrt{(p_x - t_x)^2 + (p_y - t_y)^2 + (p_z - t_z)^2} - \|d\|$$

First, the ground truth and observed poses are saved for each tag in the frame. Then, the algorithm optimizes over the set of possible camera points, minimizing the residual function and returning the ROV pose.

```
observed = []
landmark = []

for each tag:
   observed ← tag.observed
   landmark ← tag.truth
end for
OPT = Nonlinear_OPT(residual)
estimated_pose = OPT(observed, landmark)
```

Results

The University of Washington Ocean Science test tank (Fig.1) was set up for environment mapping by positioning arrays of AprilTags (Fig 4) across one corner of the tank. The tag detection is robust and can handle different angles, lighting, and distances. The reconstruction of the environment accurately captures the tags positions and their orientation relative to each other.

Figure 4. Detected AprilTag borders and ids. The green borders represent the detection algorithm's estimate for the corners of each tag and the number in the top right tag, signifies the unique AprilTag id used by the mapping algorithm to differentiate between tags.

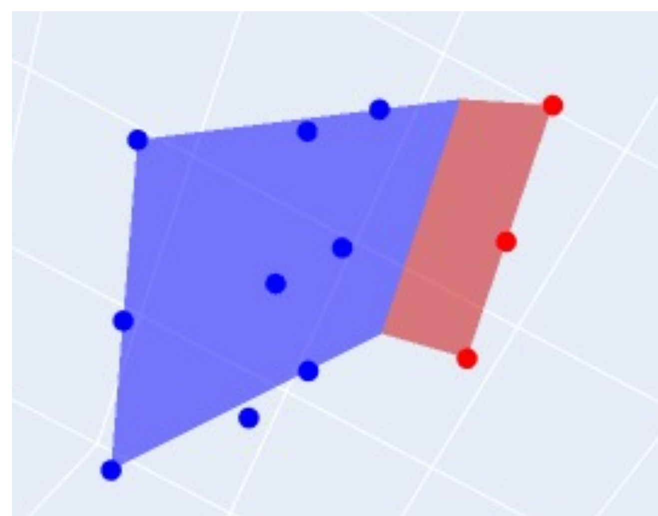

Figure 5. 3-D reconstruction of test environment. From the AprilTag detections above the GTSAM mapping algorithm reconstructs the environment in 3-dimensions. Each blue dot represents the center point of an AprilTag.

Real-Time ROV Pose Estimation

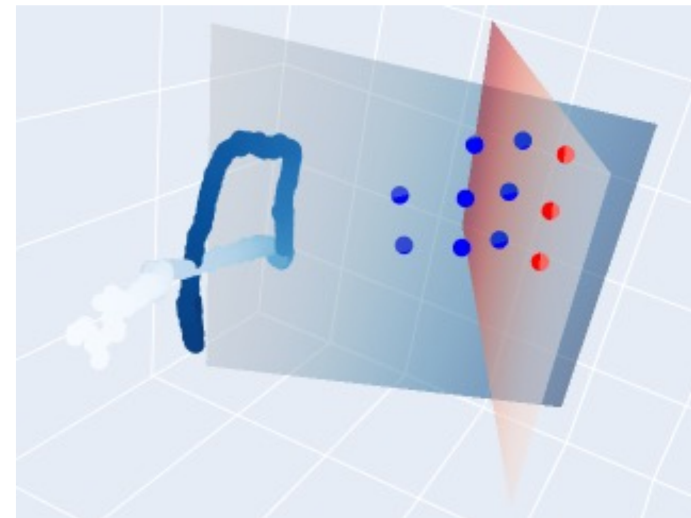

Figure 6a. Estimated ROV trajectory from trilateration

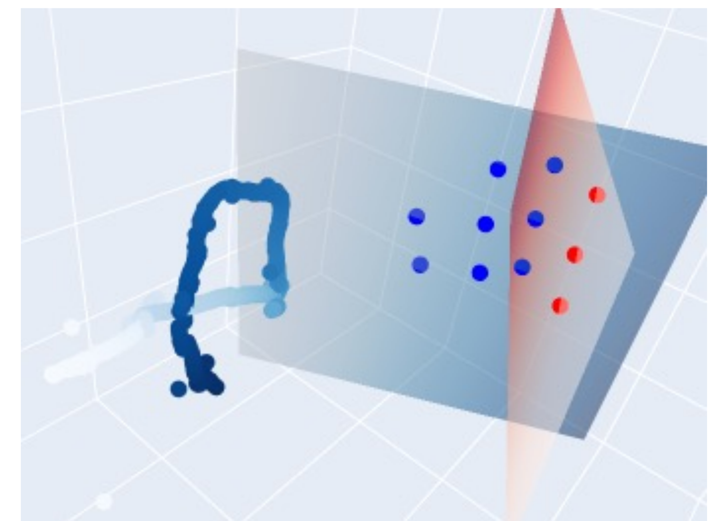

Figure 6b. Ground truth ROV trajectory from SLAM

Using trilateration with prior landmarks is very effective. The algorithm produces ROV position estimates that are accurate with respect to both previous ROV poses and ground truth AprilTag poses. Additionally, the real-time trajectory bears a strong resemblance the ground truth

Trajectory Incompleteness

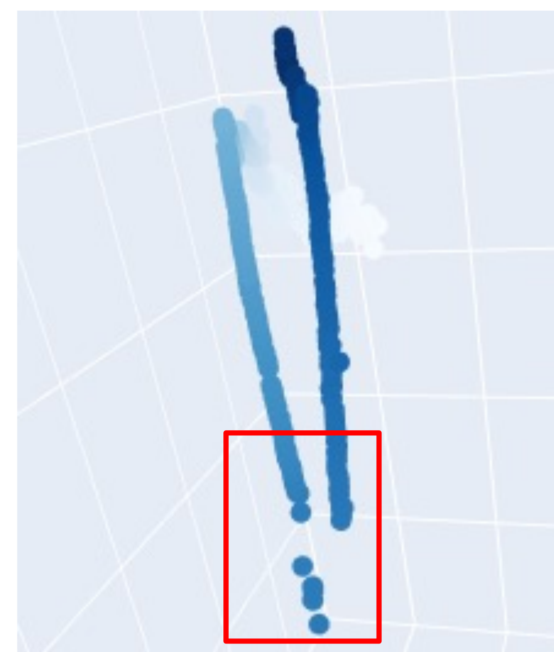

Figure 7a. Estimated ROV trajectory from trilateration

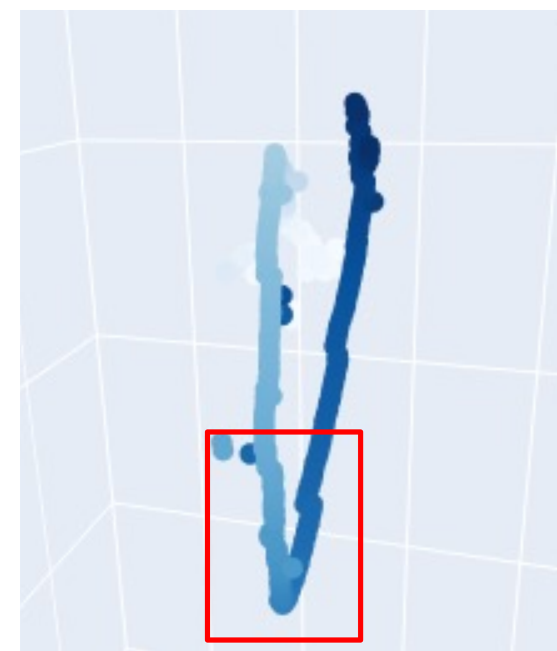

Figure 7b. Ground truth ROV trajectory from SLAM

Since the trilateration algorithm is solvable only when 4 or more AprilTags are seen, during frames with less then four tags, the algorithm does not yield a result. This pose drop out can be seen in the red boxes above.

Conclusion

The method described here, successfully provides accurate real-time ROV localization, lightweight enough to run on single board computers. To improve this method, we could:

- Upgrade the onboard computer to run real-time (full or batch) SLAM
- Complete the ROV trajectory in environments of <4 landmarks
- Incorporate the ROV odometry data into Mapping and Localization

References

1. Wang, J., Olson, E. (2016). AprilTag 2: Efficient and robust fiducial detection. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
2. Dellaert, F., Kaess, M. (2017). Factor Graphs for Robot Perception. Foundations and Trends in Robotics, Vol. 6. http://www.cs.cmu.edu/~kaess/pub/Dellaert17fnt.pdf

Acknowledgments

1. Funding for this work was provided by Schmidt Marine Technology Partners

APPLIED PHYSICS LABORATORY SUMMER RESEARCH POSTER SESSION 2024