Assignment 3

Problem Statement

Consider the scenario of an underwater robot autonomously navigating in a confined volume, as depicted in Fig. 1. Navigating refers in this context to way-point tracking and path following. In order to design a path following controller for this task, accurate online information about the robot's absolute position is required. Hence, a pipeline covering both aspects, namely localization and tracking control has to be developed in order to fulfill the autonomous navigation task.

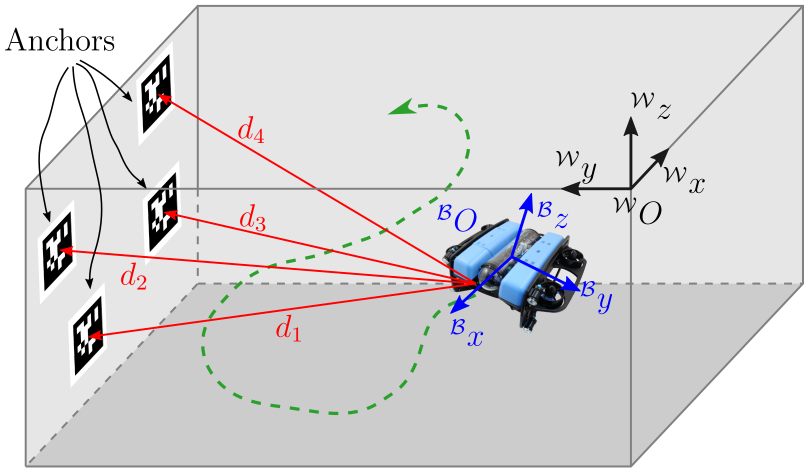

Figure 1: Underwater vehicle autonomously navigating in a confined volume using an absolute localization system based on visual range measurements towards anchors placed at known positions.

Background

The robot's absolute position can be obtained based on range measurements between the robot and reference points at known positions, referred to as anchors. For the sake of simplicity, we assume the availability of an on-board ranging-sensor which directly provides data on the distance towards the individual anchors. Note that due to the physical limitations of this sensor, we cannot assume that all anchors are detected at each measurement cycle. On-board the real robot we use a vision-based ranging system based on the BlueROV2's forward-looking camera. Hence, it has to be ensured that the anchor points lie within the camera's field of view (FOV) in order to receive range measurements.

Tasks

Find a solution!*

* The choice of your localization method and your control algorithm is up to you. However, as part of your study you should analyze and discuss the performance of your chosen approach.**

** Additionally, we always appreciate entertaining and amusing ideas. ;)

Hints

1. The tank wall is equipped with four AprilTags at fixed and known positions, see Fig. 2. These AprilTags can be used as anchors for the localization algorithm [1].

2. In order to reduce the computational burden within the simulation, you should use our provided range sensor model instead of the full-fledged AprilTag detection algorithm. In simulation and experiment, you can subscribe to the topic `/ranges` to receive distance measurements.

3. In the simulation, the position of tag 1 is $[0.50, 3.35, -0.50]^\top$ m. Make sure you can adjust the absolute position of the tags for the experiment, since we cannot exactly tell you where they will be.

4. For position control, you might want information about the BlueROV2's orientation as well.

   a) In simulation, you can use the ground truth orientation data. The Gazebo ground truth is published in `/ground_truth/state`.

   b) In the experiment, you will get the orientation data from `/mavros/local_position/pose`, where you have access to the Flight Control Unit's (FCU) pose estimation, consisting of a position and orientation quaternion. The FCU's on-board estimator fuses IMU measurements with external pose data. You can prepare a subscriber for this now to save time during the experiment.

   Note, that your controller can only rely on orientation data from these topics. Do not use the position data for your control algorithm!

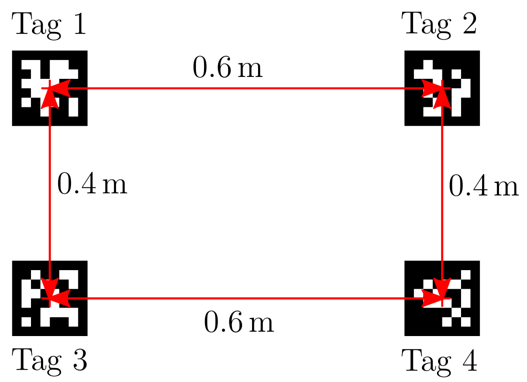

Figure 2: AprilTag positions on the test tank wall.

Challenges you might face in the experiment

- Single/multiple distance measurements getting lost
- The distance measurements range is limited
- The update rate of the ranging sensor might be low and is varying over time.
- ...

Procedure

At Home:

1. **Read the Assignment 3 section of our docs thoroughly** and follow the steps to install the packages needed for this assignment.
2. What capabilities are required to fulfill the task described in the problem statement?
3. Develop an algorithm which solves the problem described above, taking into account possible challenges.
4. Extensively test your algorithm in Gazebo!

Pre-Experiment Demo:

1. Send a short video of your controller running in simulation and a few words explaining what you implemented to Lennart and Nathalie via Slack by 07.12. the latest. Note that experimental time slots are very limited and it is **your responsibility to come well prepared and use your time as efficiently as possible**. Within your preparation you should consider the following aspects:

   a) What information do you want to get from the experimental trials?

   b) Formulate at least two to three **specific** and **relevant** questions you want to answer with the experiments.

   c) What is your scheduled timetable you want to follow? Think about the order of your actions and estimated time span for each subtask?

   Summarize these three points on a single Powerpoint slide and send it to Lennart and Nathalie, saved as `*.pdf` file.

2. Afterwards, we will coordinate a short Zoom meeting with your group to discuss your implementation and plan for the time in our lab. **Only then will you be allowed to deploy your algorithm to the BlueROV2 Platform.**

Experiment:

1. Implement your algorithm to control the BlueROV2 underwater robot.
2. Prepare to record all experimental data in a ROS-bag for later evaluation.
3. Test and evaluate the performance of your controller.

Submission:

1. Submit the simulation demo by 07.12. and schedule a Zoom-Demo-Meeting.
2. The Poster Session: Prepare a poster summarizing your approach and your learnings. Moreover, describe your team's workflow. How did you approach the problem? Submit your poster via the form `https://forms.gle/f6miejNhBf6ek1EC7`.

3. The Report: Summarize your approach and your algorithm. Describe, analyze, and critically discuss simulation and experimental results. The report must not exceed 4 pages using the template available in Slack. Follow the structure of the template. Moreover, **provide a link to your code's github-repository** with your report.

4. The name of your submission should follow the format: `assignment3_groupX.pdf`.

5. Submit your report via the form `https://forms.gle/rSMJWT5Wi3zherHB7`

As announced in the lecture, if you do not have access to a google-account please e-mail us under `formulasandvehicles@gmail.com` or reach out via Slack.

Demo Deadline: 07.12.2020, 23:59 CET

Poster Deadline: 14.12.2020, 23:59 CET

Report Deadline: 04.01.2021 23:59 CET

References

[1] Wang, John, and Edwin Olson. *AprilTag 2: Efficient and robust fiducial detection.* 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2016.