Assignment 2

Goal:

Robust depth-control is an essential capability of underwater robots. In this assignment you will design a depth-control scheme which enables an underwater robot to stabilize at desired depth levels and follow a given depth profile in simulation and in experiments. In particular, you will explore the challenges when switching from simulation to a real-world implementation.

Tasks:

The following steps will guide you through the assignment.

Before proceeding with the individual tasks, follow the guide in our docs to update our repositories and download a template ROS package we prepared for you to use:

h ttps://hippocampusrobotics.github.io/fav_docs/assignment2.html

Hint: You can find details on the BlueROV2 actuation on:

https://hippocampusrobotics.github.io/fav_docs/the_vehicle.html

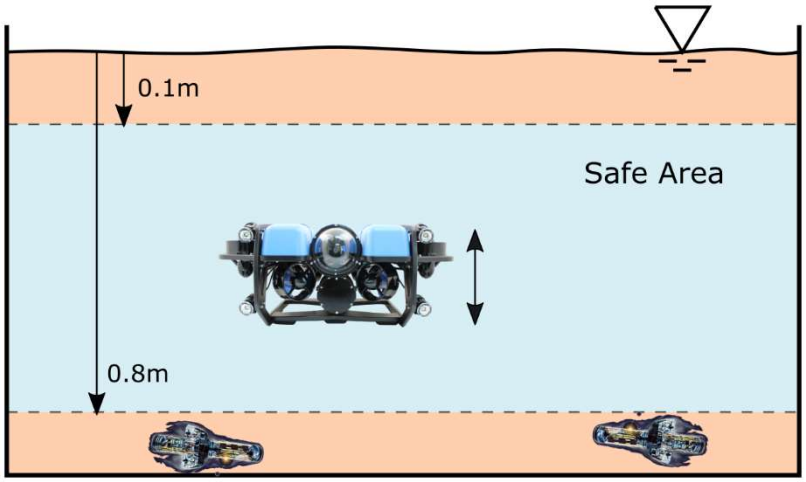

Fig.1: Safe operating area of the BlueROV2.

I) At home:

1) Familiarize yourself with the actuation of the BlueROV2. We provide you with a mixer module in our template package that you can use.
2) Design a depth-controller to **stabilize** the BlueROV2 at a desired water depth. Moreover, the controller should be able to follow a given depth profile. Your controller will receive setpoints from a topic named “depth_setpoint”. In the template package, you will find an example node that publishes a constant desired depth. Additionally, as a safety feature, your controller must send null actuator commands when the measured robot depth is unsafe, i.e. depths greater than 0.8 m and smaller than 0.1 m as depicted in Fig. 1. It also has to send null actuator commands for depth setpoints outside the “safe” range. Note

that according to ROS conventions, our world frame's z-axis is pointing upwards with zero at water level. This means desired depths will be negative.

3) Extensively test your algorithm in Gazebo. Explore the limits of your controller. What are useful ROS-tools (rqt, etc.) to analyze the control performance? What are the most important tuning parameters? Keep in mind that simulation and real world can differ immensely. Therefore, you will have to re-tune your controller for the physical experiments. Make this as easy as possible to save time (for example by using ROS parameters).

II) Pre-Experiment Demo:

1) Send a short video of your controller running in simulation to Lennart **and** Nathalie via slack (in a *single direct group message containing all of your group members*) by **23.11. 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 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 this on a single power point slide and send it to Lennart and Nathalie.
2) Afterwards, we will coordinate a short Zoom meeting with your group to discuss your controller and plan for the time in our lab. **Only then will you be allowed to deploy your controller on the BlueROV2 Platform.**

III) 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:

0) Submit the simulation demo by 23.11.
1) **The report**: Summarize your approach and your algorithm. Additionally, provide a table summarizing your mixer configuration of the BlueROV2. 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.
2) The name of your submission should follow the format: 'assignment2_groupX.pdf'.
3) Submit your report via the form https://forms.gle/9WSHY4Xp24Cgm6qv9

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.

Deadline: 23.11.2020, 23:59 – Demo of Simulation

Deadline: 07.12.2020, 23:59 – Report