# Assignment 1

## Goal

Robust depth control is an essential capability of underwater robots. In this assignment, you will design a depth control scheme that enables our robot to stabilize at desired depth levels and follow a given depth profile in simulation and in experiments. You will use data from a pressure sensor to compute the vehicle's depth. In particular, you will explore the challenges when switching from simulation to a real-world implementation.

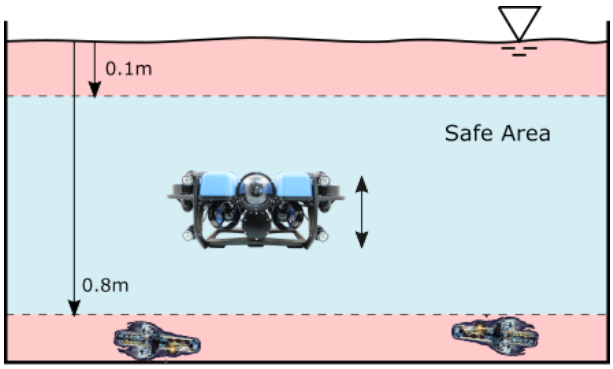


Figure 1: Experimental setup and safe operating area of the BlueROV2.

The following steps will guide you through the assignment.

## At Home - Individual Work

### Part 1a - Until 06.11.2024

1. Download the template ROS package we prepared for you to use here: https://tinyurl.com/favassign1
2. Familiarize yourself with the given code. Read the explanation in our documentation. You can find more explanations in the section "Tutorials". You will need at least the Section "Real-Time Plotting".
3. Implement the calculation of the depth of the robot. You only need to adjust the `depth_calculator.py` file! The depth (i. e. the water column above the robot) follows directly from the pressure at the robot's depth. The pressure sensor is not

necessarily at the real robot's center of gravity, so you might want to implement some way of adjusting this parameter.

4. Use `Plotjuggler` to plot the pressure and your calculated depth while the BlueROV moves. Take a Screenshot.
5. The name of this file should follow the format: `assignment1a_lastname.png`
6. Submit via the form: https://forms.gle/FhE3P7Rnjb9YZS4u5

**Deadline:** 06.11.2024, 23:59 CET

## Part 1b - Until 13.11.2024

1. Then, implement a controller that allows the BlueROV to follow a desired depth profile or setpoint. You can start with just a P controller and then add a D-part. *Hint: This will be easier after the lecture on Applied Control on 07.11.2023.*
2. As a safety feature, your controller must send *null* actuator commands for depth setpoints outside the *safe* range as depicted in Fig. 1.
3. In the template package, you will find an example setpoint node that publishes a square wave signal. Play around with other setpoint profiles.
4. Extensively test your algorithm in Gazebo.
5. Copy the template report here: https://tinyurl.com/favreporttemplate
   Answer all questions and add a plot of your results.
6. The name of your submission should follow the format:
   `assignment1b_lastname.pdf`.
7. Zip the code (i. e. the package). The folder's name should follow the format: `assignment1b_lastname`.
8. Submit via the form: https://forms.gle/9XmMtrw2sa2YtTnu5

**Deadline:** 13.11.2024, 23:59 CET

# Group Work

## Pre-Experiment Meeting

1. Together as a group, merge your results from before and implement a joint solution for the depth control task.
2. In preparation for your time at the lab, read our tutorial on the lab workflow: https://tinyurl.com/favlabworkflow.
3. We will hold a *short* Zoom meeting with your group to discuss your controller implementation and plan for the time in our lab. Not everyone within your group has to attend this meeting. Arrange a meeting until 11.11.24, the available meeting times will be posted in Slack.
4. As a preparation for your time in the lab, consider the following aspects:
   a) What information do you want to get from the experimental trials?
   b) Formulate at least two **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.
5. Answer *all* these questions in the PowerPoint template provided via Slack and share it with us in your group's Slack channel before the meeting.
6. **Only after the meeting will you be allowed to deploy your controller on the BlueROV2 Platform.**

**Deadline Meeting Arrangement:** 11.11.2024, 23:59 CET

## Experiment - Between 18.11.-22.11.2024

1. Deploy your algorithm on the BlueROV2 underwater robot. Follow the experimental plan that you came up with beforehand.
2. Prepare to record all experimental data in a ROSbag for later evaluation. **Practice this at home!** You will want to take some pictures or videos, too.

3. Test and evaluate the performance of your controller. You decide how you use your (limited) time at the lab.

## Paper Submission - Until 27.11.2024

1. Summarize your approach and your algorithm. Describe, analyze, and critically discuss the experimental results. The paper must not exceed 3 pages using the template available in Slack. Follow the structure of the template. Finally, provide a link (Github/Gitlab/...) to your code's repository.
2. The name of your submission should follow the format: `assignment1_groupX.pdf`.
3. Submit your paper via the form https://forms.gle/HvwAg2TzThBTewWn9

**Deadline Group Paper:** 27.11.2024, 23:59 CET