NTNU

Norwegian University of Science and Technology
Department of Marine Technology

TMR4120 - Underwater Engineering

Assignment 6

Blueye in the laboratory

Assignment Summary

1. Get to know the Blueye Vehicle
2. From simulation to real operations
3. Blueye with keyboard controller from assignment 0
4. Blueye with waypoint controller from assignment 2

Delivery

1. Demonstrate that you are able to make the Blueye fly autonomously in a square.

The Blueye Vehicle

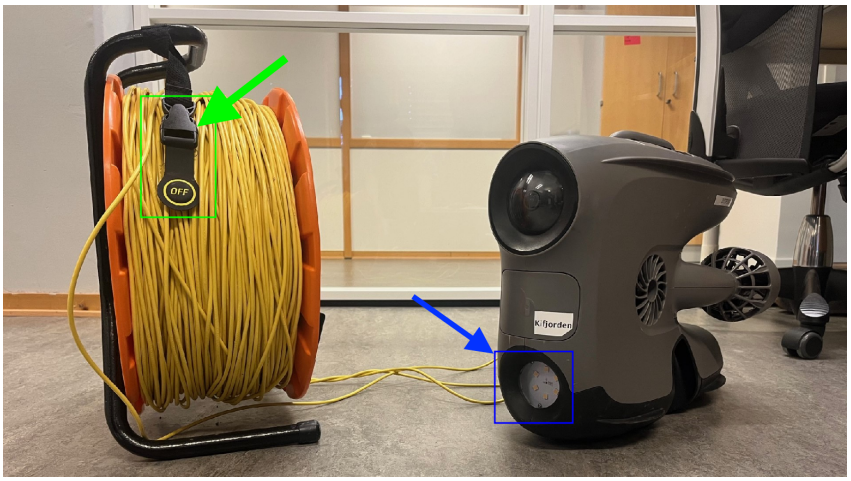

Figure 1: One of our Blueye drones with its reel.

The Blueye is a small underwater drone with a monocular camera (see Fig. 1). It is well suited for inspection tasks, and due to its small size it is easy to deploy in small teams. It has four thrusters set in a configuration that allows us to reliably control the surge, sway, heave and yaw motions of the vehicle. A figure showcasing the entire Blueye ROV system pipeline can be seen in Fig. 3.

To power on the Blueye, unclip the magnetic tag shown by the green box in Fig. 1. The magnetic tag has either ON or OFF inscribed on the sides. Place the tag with the side marked with *ON* toward the drone on the sensor highlighted by the blue box. The drone will play the classic tune which translates to *I am a drone and I am powering on*. The boot process can take up to a minute, it's completion is marked by the drone briefly spinning the thrusters at a low RPM for a couple of seconds.

Figure 2: The Blueye reel and its buttons, courtesy of Blueye.

The standard method for controlling the drone is using the official Blueye App. It's available in the official Android and iOS app stores. To connect with the App to the vehicle, the phone must be logged into Blueyes WiFi. To achieve this, turn on the wireless router on the reel casing by pushing the power button (see Fig. 2), and wait for a network called Blueye_XXXX, where XXXX is replaced by some digits and letters, to appear. The network password is written on the reel case. When you are connected, you will see the drone's camera feed and be able to control its movements using the joysticks on screen.

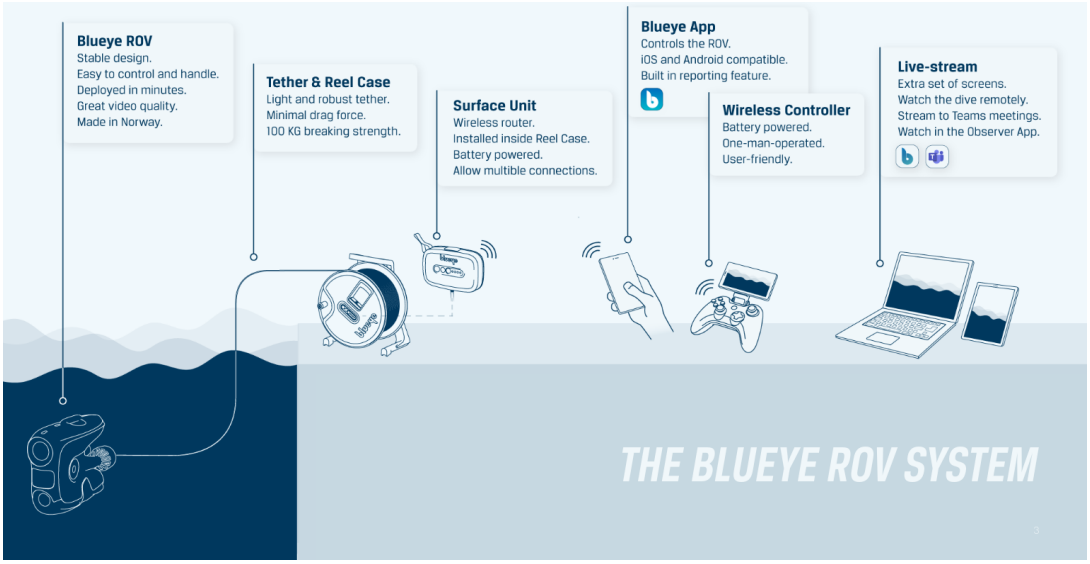

Figure 3: The Blueye ROV system, courtesy of Blueye.

The Blueye has an IMU, a depth sensor, and a magnetometer, thus depth and heading hold functionality are accessible in the App, employing logic similar to what you've used in some of the assignments.

From simulation to real operations

In prior tasks, you wrote software that interacted with a simulated Blueye drone. A popular method in robotics is to thoroughly test software in a simple simulated environment before gradually increasing the complexity of the simulated environment before testing it in the actual world. Some of the benefits are as follows:

1. **Safety:** Robotic equipment is typically rather pricey. Simulations enable us to uncover potential software faults without causing damage to our equipment, the working environment, or human operators.

2. **Cost-Effectiveness:** Conducting real-world testing can be extremely expensive. Especially for underwater robotics, which frequently require both a support vessel and a skilled crew. There is also the matter of wear and tear on the robotic system. We should constantly endeavor to spend as little time in the field debugging code as possible.

3. **Controlled Environment:** Simulations create a controlled environment in which certain parameters (such as water currents, pressure, and barriers) can be repeated and adjusted. This enables extensive testing under a variety of circumstances, including extreme conditions that are impossible or dangerous to recreate in the real world.

4. **Rapid prototyping and iterations:** Simulation facilitates the quick development and testing of new designs, methods, and strategies. Engineers can quickly iterate over designs and configurations to improve performance and reliability without the effort and cost of building many physical prototypes.

5. **Extensive data collection:** A simulation allows you to collect precise data on every element of the robot's performance, including parameters that would be difficult or impossible to assess in the actual world. This data is extremely valuable for debugging, performance analysis, and future development.