Design of the Steelhead Autonomous Underwater Vehicle for the International RoboSub Competition 2025

Viktor Moreno*, Brenna Sullivan*, Jonathan Wu*, Clark Jeffrey*,
Gyan Zesiro*, Ryan Nashota*, Joshua Han*, Jared Chan*, Hashaam Zafar*,
David Chung*, Abhishek Raghuwanshi*, Dorson Tang*, Fei Kuan*
**UBC Subbots*
University of British Columbia
Vancouver, Canada
Email: ubc.subbots@gmail.com

***Abstract*—UBC Subbots' submission to RoboSub 2025 is the Steelhead Autonomous Underwater Vehicle (AUV). A competition strategy is introduced, which influences the following sections on design and is followed by a breakdown of design strategy. A discussion regarding the in-house design and production of novel elements highlights aspects such as machined aluminum enclosures and lids, custom neoprene seals and internal mounting, actuating and torpedo systems, system control, navigation procedures, and an object recognition pipeline. Methods of physical and analytical testing such as prototypes, experiments, and simulations are explored, and results are analyzed in context.**

***Index Terms*—robotics, navigation, autonomous underwater vehicles, controls, acoustics, undergraduate, ROS, materials**

I. COMPETITION STRATEGY

In preparation for RoboSub 2025, the team's strategic vision has evolved to align with the AUV's current stage of development. While last year's approach focused on designing and building a new vehicle from the ground up, this year's strategy is characterized by iteration. Our aim now is to complete as many tasks as we can, thus we prioritized general functionalities such as pathfinding, computer vision, maneuverability, and object manipulation. During the competition, we expect the AUV to be able to complete Collecting Data (Gate), Navigate the Channel (Slalom), Drop a BRUVS (Bin), and Tagging (Torpedoes). These tasks were strategically chosen to play to the strengths of our actuators and software sub-teams, who have conducted extensive physical and virtual testing throughout the year.

This shift is also driven by our team's desire to prioritize student learning while attempting each design challenge. In choosing to continue developing each part of our AUV rather than relying on off-the-shelf components, we've accepted a trade-off between time constraints and increasing team knowledge of underwater robotics. Over the past year, the electrical team has focused on improving system reliability and modularity, the mechanical team has developed multiple working prototypes for each task, the software team has rebuilt its simulation tools and mission planning algorithms, and the sound localization team has developed a working hydrophone-based prototype, which we hope to integrate into future iterations. Designing and manufacturing all major subsystems has increased the AUV's overall complexity, but has also provided valuable experience that will inform our future work. This method also aids in troubleshooting, as the team has an intimate understanding of our custom-built systems. A complete CAD model of Steelhead can be found in Figure 1.

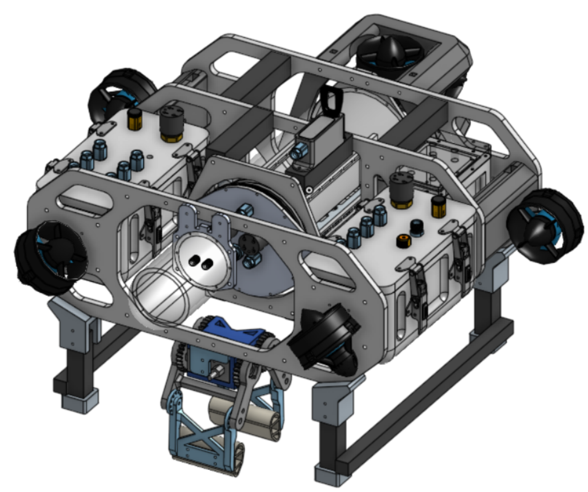


Fig. 1: Full Steelhead CAD Model

II. NOVEL DESIGN ELEMENTS

A. Initial Design

The original Steelhead concept was developed for RoboSub 2023 and featured a uni-body machined hull. Due to tight time constraints and complex machining needs, the original Steelhead concept was revised and produced the following year ahead of RoboSub 2024. For more information about the

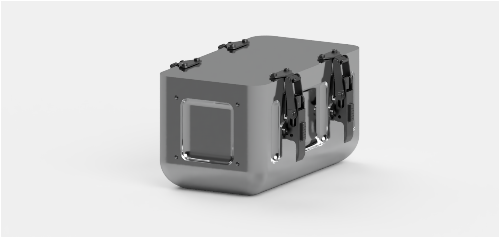


Fig. 2: Steelhead Auxiliary Hull

design and testing of the 2023 Steelhead concept, please refer to UBC Subbots' RoboSub 2023 Technical Report [6].

B. Revised Design

Steelhead's current design serves as a middle ground between previous years' AUV designs and the complexity of the original Steelhead concept. It was developed with the team's competition strategy and adaptability in mind, and allows for seamless integration of all subsystems. Over the past year, as the team's vision has evolved, Steelhead has undergone key upgrades to multiple mechanical systems. Primarily, the team developed and manufactured a claw in-house to replace the off-the-shelf device detailed in UBC Subbots' RoboSub 2024 Technical Report [7], a new enclosure is being designed to protect the hardware for the actuating components, and both external and internal mounting schemes were revisited for ease of access. After RoboSub 2025, Steelhead will be retired, fulfilling the team's future needs by acting as a test bench for the software and mechanical teams.

1) Design Goals: The main design goals of the Steelhead AUV are affordability, manufacturability, and adaptability. Firstly, Steelhead is low-cost and was manufactured over a short period of time. Most materials and components were repurposed from previous years' AUVs, and suitable blocks of aluminum were purchased at scrap prices. Additionally, the design and manufacturing of nearly all mechanical components was completed in-house to align with the team's goal of providing students with valuable, hands-on experience. Lastly, due to its numerous mounting points and aluminum extrusions located around the perimeter, Steelhead serves as a highly adaptable platform where components can easily be added or replaced to better serve the team's needs.

2) Production Methods: Once the Steelhead concept was designed and virtually assembled using Onshape, HSMWorks was used to generate a toolpath and verify part geometry. G-code from the CAM software was imported to Tormach's PathPilot CNC controller. The CAM was completed over the course of three days. Next, the main enclosures as well as the actuators enclosure were machined from large 6061 aluminum blocks, which were purchased from a local scrap metal depot. Two blocks with dimensions of 5"x7"x11" were cut from a block four feet in length using a horizontal band saw with support from the UBC Mechanical Engineering machine shop. These blocks were placed on a Tormach 1100MX CNC mill to be faced and machined into the two main enclosures with support from the UBC Integrated Engineering machine shop. The actuators enclosure and all enclosure lids were similarly prepared. Each enclosure took approximately 24 hours to machine, and can be seen in Figure 2.

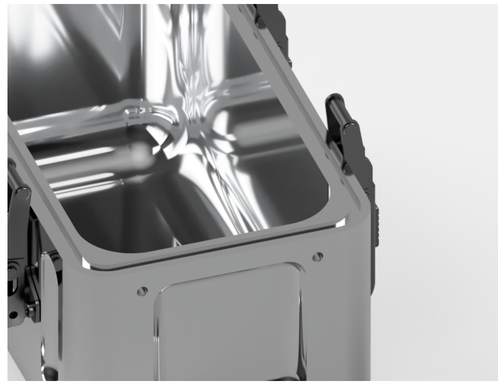


Fig. 3: 1/8" Face seal in the enclosure groove

3) Custom Neoprene Seals and Lids: The custom seal derived its design from O-rings around a cylindrical enclosure as in Figure 3. The rectangular shape of the enclosures prevented us from using off-the-shelf O-rings. For the size of the enclosure, we designed it to use one 1/8" face seal in order to maximize the internal volume for components. The lid is a machined 1/2" plate, which compresses the seals into the groove. Each lid is secured with four stainless steel latches for easy installation and removal. See Figure 2 for the latching system and Figure 3 for the machined grove.

4) Internal Component Layout and Mounting: The two machined enclosures contain all electronics with the exception of the battery and its monitoring system. One enclosure contains "high-powered" electronics such as electronic speed controls (ESCs) and high-current cabling, while the other contains "low-powered", sensitive electronics. The separation of these components reduces signal interference. Penetrators are located on the lids of each enclosure, and mounting schemes are secured to the machined inner face of each lid for easy access to all components as highlighted by the low power mounting scheme in Figure 4.

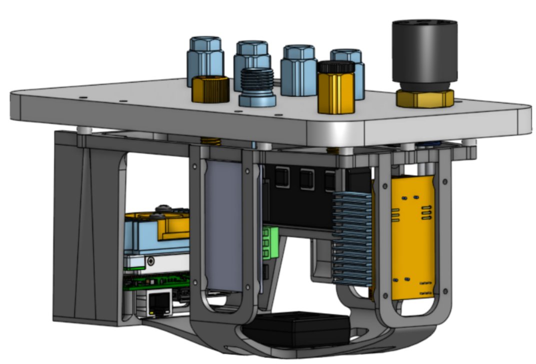


Fig. 4: Low Power Mounting System

C. Actuators

1) Claw: Steelhead's claw is actuated by a single servo motor, which drives a gear-based transmission system as shown in Figure 5. Using a single actuator minimizes mechanical complexity, streamlining both manufacturing and control. The claw assembly itself features adjustable arms with soft gripping pads. These design choices enable adaptability across a range of target object sizes and geometries. The arm angle can be manually tuned to fit specific use cases, while the gripping pads provide a secure hold by conforming to the object's surface. The claw is primarily composed of 3D-printed components, allowing for rapid prototyping and iterative design refinement. This fabrication approach significantly accelerates development cycles and reduces the cost of custom mechanical components.

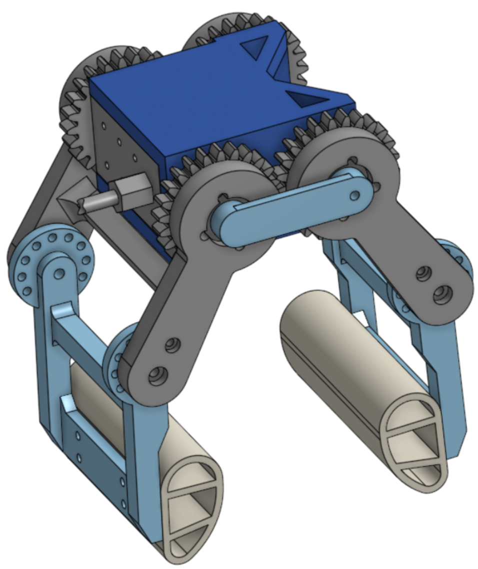


Fig. 5: Claw Design

2) Torpedo System: Each torpedo is powered by a compressed CO_2 cartridge encased in a streamlined, hydrodynamic torpedo housing (Figure 6). This housing is designed to minimize drag underwater while providing a secure enclosure for the cartridge. The torpedo assembly employs a threaded, three-piece construction, enabling quick and tool-free replacement of spent cartridges. The upper chamber is filled with weights to achieve neutral buoyancy. This neutral buoyancy is crucial to allow for a smooth torpedo trajectory. Torpedo launch is achieved via a spring-actuated firing mechanism. A manually compressed spring stores potential energy, which is released by a small servo motor upon activation. This release drives a puncture pin into the CO_2 cartridge's seal, triggering a rapid expulsion of gas that propels the torpedo forward. The separation of spring and puncture mechanisms ensures consistent puncture force and reliable ignition of the cartridge.

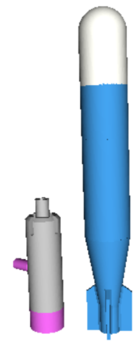


Fig. 6: Torpedo Design

3) Dropper System: The marker dropper system utilizes a passive retention and active release mechanism to deploy markers with precision and reliability. Each marker rests within a grooved cylindrical barrel and is held in place by fins integrated into the marker body (Figure 7). These fins are intentionally offset from the internal grooves of the barrel, preventing premature release during normal operation.

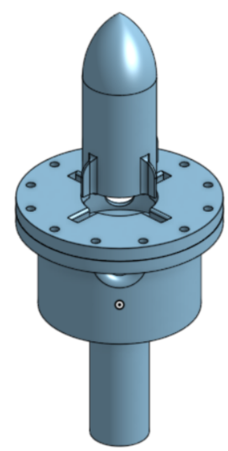


Fig. 7: Dropper Design

To deploy a marker, a compact servo motor is used to rotate the marker within its barrel. Once the fins are aligned with the grooves, the marker is no longer constrained and falls freely under gravity. This mechanism provides a simple yet effective method for controlled deployment, with minimal mechanical complexity and low power consumption.

4) Actuators Enclosure: A small aluminum enclosure houses the electronic components for our actuators. The guiding philosophy for this design was modularity. The intent was to make the actuator systems largely independent from the main structure of the robot, allowing for modular placement of devices while providing easy access to the wiring and waterproofing. The enclosure can be seen in Figure 8.

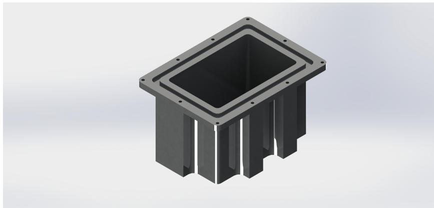


Fig. 8: Actuators Enclosure

D. Battery and Camera Enclosures

The battery is placed in the central acrylic cylindrical enclosure shown in Figure 1. All power management hardware is mounted to one end of the enclosure, allowing batteries and other components to quickly be swapped. MacArtney SubConn 4 contact connectors connect the battery enclosure to each aluminum enclosure. The camera enclosure is a 3” Blue Robotics Acrylic Enclosure, and hosts two low-light USB cameras.

E. Control System

1) Main Control System: The control system of Steelhead consists of five major components: an unscented Kalman filter (UKF) state estimator, a waypoint system, a PID controller, a trajectory generator, and a thrust allocator. The communications between these modules are handled by our new Mission Planner.

First, the UKF state estimator uses input from sensors like our IMU to estimate the current pose of the AUV. A UKF filter is chosen for its accurate estimation of non-linear systems and ability to fuse several sensors together.

With the positional information obtained from the state estimator, the waypoint system can then keep track of the state of the AUV and the target state, calculating the error between them for the PID controller. In addition, it notifies the trajectory generator when a waypoint is reached so that the AUV can move on to the next target.

Using the error information from the state estimator, the PID controller then controls four degrees of freedom of Steelhead: linear, horizontal straightness, vertical straightness, and yaw. The AUV relies on the positions of the center of mass and center of buoyancy to passively control roll and pitch. The output is a vector of desired forces on the four degrees of freedom that is passed to the thrust allocator.

The trajectory generator receives target poses from the computer vision system. Based on the type of target, it generates a series of waypoints for the waypoint system. This series of waypoints allows the AUV to move in an optimal way, like keeping the target in sight, and taking advantage of the AUV’s higher maneuverability in the x- and yaw-axis.

The thrust allocator takes into account the surge, heave, and sway contributions of each thruster, as well as its position relative to the center of mass. This allows it to adequately allocate force output for each thruster to achieve the desired movement. The configuration of thruster allocation is highly configurable, and it allows for up to six thrusters for enhanced maneuverability.

2) Thruster Control: We have chosen to keep the actual thruster control fairly simple, using serial communication by sending strings of encoded instructions from the Nvidia Jetson computer to a Teensy 4.1 microcontroller via USB. The Teensy then decodes the input and sends PWM signals of appropriate frequency to the Blue Robotics Basic ESC’s, which then control the Blue Robotics T200 thrusters via 3 phase AC power.

3) Actuator Control: We employ a dedicated ESP32 nano microcontroller for control of our actuator servos. This board is soldered via pin headers to a breadboard-style through-hole PCB mounted within the main enclosure. The servos and accompanying actuators are situated outside of any enclosures and were made waterproof. The servo wiring enters the main enclosure through the rear end cap via waterproof cable penetrators. These wires are then soldered to the aforementioned PCB such that they align with any of the ESP32’s digitalWrite enabled output pins. Each actuator operation (e.g., the firing of one of our two torpedoes at a time) is mapped to a different digitalRead enabled input pin. Once the computer vision module determines it is the appropriate time to activate an actuator operation, it sends a high signal to the corresponding input pin on the ESP32. This activates the corresponding pre-programmed servo procedure, thereby initiating the relevant actuator operation.

4) Architecture: Our software architecture uses the ROS2 framework running on Ubuntu Linux as a Docker container on a Nvidia Jetson TX2. ROS2 provides various common robotics tools, allowing us to focus on developing the custom behaviors of our AUV. Extensive logging capabilities of ROS2 also allow for easy debugging and diagnostics. In addition, ROS2 is language-agnostic, so different parts of the system can be written in different languages. For applications requiring low-latency processing, we use C++, while Python is used primarily as a high-level interface for managing our pipeline and launch sequences. Our custom pipeline manager can be configured to execute arbitrary sequences of actions, starting and stopping nodes based on published feedback according to criteria we define.

The software architecture runs on the TX2 as a Docker container built using Dockerfile. Using Docker increases overall reliability as it takes only a few minutes to deploy on a spare SBC if any hardware issues arise. The infrastructure as code

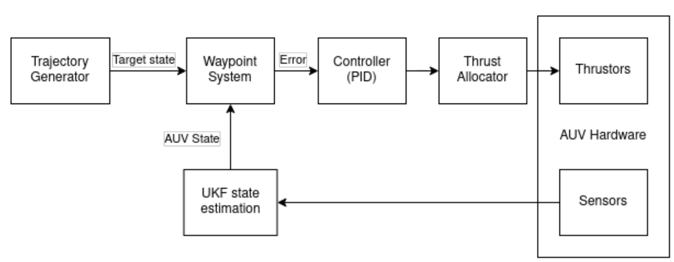


Fig. 9: Block diagram of the software control system used by the Steelhead AUV.

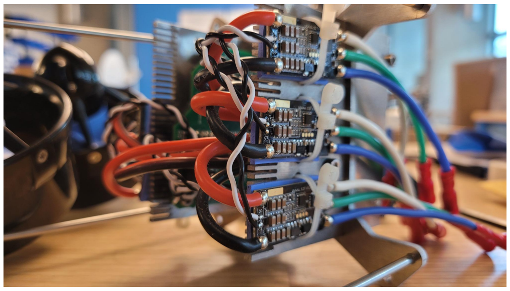


Fig. 10: ESC's mounted in Steelhead

(IaC) approach of using Dockerfile also enables versioning of our software environment, making changes to it visible and allowing easy rollbacks.

5) Mission Planner: The mission planner coordinates the AUV's ROS2 nodes using the BehaviorTree (BT) library. It is comprised of a 'controller' ROS2 node, and several BT nodes, organised into trees and subtrees – one for each task. The 'controller' node subscribes to the other packages' topics, and interprets their data to determine which nodes to call, and with what instructions. For example, once the object recognition package detects the start gate, the mission planner sends an instruction to the trajectory generator to approach it. The tree architecture is designed to be pseudo-parallel, so that if at any point a condition is no longer met, the associated action can be terminated and re-tried. A diagram of the tree architecture can be seen in Appendix B.

6) Object Recognition: The gate and marker tasks require the detection of the objects for navigation. For these tasks, we segment the image in the HSV colour space, which models perceptual changes in colour better than RGB, especially in the underwater environment. We then apply a convex hull algorithm to detect the gate and markers based on the filtered image. After detecting these objects, we perform simple pose estimation relative to the AUV, providing targets for the control system. Before any image processing, we also correct the distortion of images due to the lens or enclosure.

To detect objects, we upgraded to the YOLOv8 object detection model which we fine-tuned on the Roboflow platform. The model was fine-tuned using a dataset generated on Unity of images of gates and markers underwater from various angles. This should reflect the actual environment of our AUV.

7) Sensors: This year, a new IMU was installed into Steelhead, the Adafruit BNO055 IMU. This allowed for nine degrees of freedom (9-DOF) for enhanced state estimation. As this IMU is not natively supported by ROS2, a custom plugin was created to translate the signals received by the BNO055 to ROS2 readable messages.

F. Power System

1) Battery Monitoring: The BMS uses an Arduino nano on a custom voltage divider circuit to monitor real-time battery level and current throughput. This ensures the battery is above the minimum required level to prevent device damage and there are no overcurrents. The system measures the overall voltage and current of the battery. Our system contains two different shutdown switches: 1. Emergency shutdown switch; 2. Fault detection switch. The emergency switch is a magnetic switch key that can be pulled out manually to cut off the main power from the robot in an emergency. The fault detection switch is triggered by the BMS when the voltage or current is outside of acceptable range, which sends a signal to an external relay circuit and cuts off power from the other systems.

2) Various Output: Our main system has four different output levels: 14.8V, 12V, 9V, and 5V. The system contains two output per voltage level. Each voltage level is created using a buck converter except for the 14.8V. The 14.8V comes from the BMS through a voltage follower to provide a more stable ADC reading.

III. Integration, Verification, Validation

During production, it was essential to verify the feasibility of designs in parallel. With out team's resources, we sought to integrate different parts of the system in different stages, testing them in their respective environments, verifying and validating that they do indeed meet the design requirements.

1) Software Simulation: Our testing and verification focus was on our simulation environment, as it provides a cheaper, safer, and faster way to test Steelhead and collect ample synthetic data. The simulation environment we deployed was developed using the open-source simulation tool Gazebo, which allowed us to create a simulation description format (SDF) file representing our robot, shown in Figure 11. The SDF description imported an STL-format model of our robot from SOLIDWORKS and applied mechanical properties such as inertia and damping to generate realistic restoring forces on the vehicle. ROS2 was used to tie the mission planning control logic to the physical thrusters. Using this environment, we developed camera, position, gyroscope and depth sensor emulators, as well as thruster driver emulators in the form of plugins that interact with our control pipeline. We implemented buoyancy and hydrodynamic force plugins that use the second-order equations of motion for the AUV, including position, velocity and acceleration values at each iteration of the simulator's update loop. These calculate the environment forces acting on the AUV at any given time. Other than in-house plugins, we also made use of open-source Gazebo plugins for robot localization and IMU emulation.

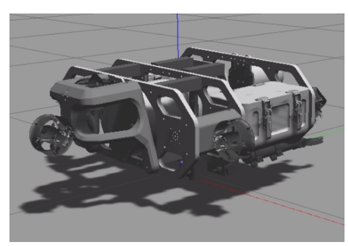


Fig. 11: Steelhead frame in Gazebo

Using models of the AUV and gate seen in Figure 12, we were able to test our control and gate detection systems.

During simulation, the AUV was able to reliably detect the gate, calculate the forces required to move to the gate, and apply those forces using thrusters. With limited pool access, the simulation allows us to be more efficient during pool testing in the real world.

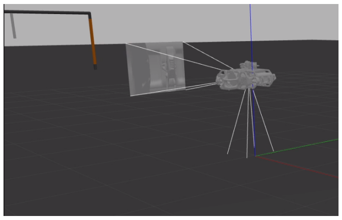


Fig. 12: Gazebo gate navigation test

2) Mechanical Testing: To validate the structural integrity and performance of the claw design, Finite Element Analysis (FEA) was conducted on critical load-bearing components. The analysis focused on stress distribution, deformation under operational loads, and safety factor evaluation. Simulations were carried out using ANSYS, applying boundary conditions representative of the maximum expected weight of a target object. Figure 13 indicated that the claw structure remains well within acceptable stress limits, using its truss structure to distribute load evenly.

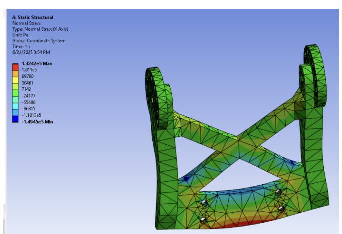


Fig. 13: FEA Analysis, Actuators

For the torpedo and dropper designs we similarly utilized ANSYS Fluent to analyze the performance of our design. Our analysis tries to minimize the drag coefficient of the objects to ensure we can clear our objectives faster and with more precision. Various other metrics were also considered, such as force distributions and exit velocity.

3) Enclosure Testing: The two aluminum enclosures were also tested with FEA in Fusion 360 at depths of 30, 50, and 100 meters. These depths were chosen based on competition requirements and to retain the same functionality as the original Steelhead. The stresses and deflection was minimal. Pressure testing using a vacuum pump was used to test the custom-made O-rings and verify the watertight properties of the rectangular enclosures. As shown in Figure 14, we also manufactured a model enclosure to optimize our approach for working with our chosen material to produce the desired tolerances. This included testing a new tool for creating the channels for a custom seal. We used a keyseat cutter on another scale model resulting in a wall-face channel for the Neoprene seal.

Fig. 14: Scale model test enclosure

4) Seal Testing: A test using Blue Robotics enclosures was devised as a benchmark for proper o-ring enclosure design. We tested 2 seal-joining methods: adhesive bonding and vulcanization. We utilized an instant contact adhesive from Weicon made for Neoprene o-rings. For the vulcanization method, we heated a thin hobby knife to a temperature that would melt the o-ring ends, then once melting each end, joined them. This was then placed into the Blue Robotics 8” Series enclosure in Figure 15.

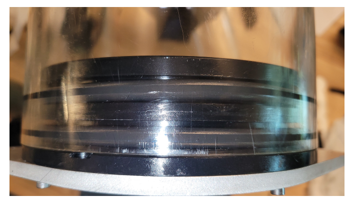


Fig. 15: The 3/16” Neoprene O-ring installed into the 8” Blue Robotics Flange

Due to limited access to a pool or a similar large body of water, we opted to perform the test with the vacuum method, where a difference in pressure between the inside and outer atmosphere would simulate a certain depth of water. This was done using a hand-operated pump with a dial indicator reading the pressure inside the sealed hull. We first calibrated our pump to measure drop in pressure, then we installed a vacuum vent plug on each enclosure and pulled the vacuum to be equivalent of being 6 metres underwater. The result was a success, as the only drop in vacuum pressure was the calibration that we performed on the pump itself.

Acknowledgments

UBC Subbots is supported as a student design team by the Faculty of Applied Science at the University of British Columbia. We would like to thank the UBC Applied Science Professional Development Team for supporting us. For supporting our production and prototyping activities, we give thanks to the UBC Electrical and Computer Engineering, Mechanical Engineering, and Integrated Engineering Departments for allowing members to utilize various tools and facilities to complete our projects. We sincerely thank our faculty advisors Ioan (Miti) Isbasescu and Dr. Adrien Desjardins. Finally, we would like to acknowledge our sponsors for 2025: UBC Applied Science, UBC Engineering Departments, Altium, Onshape, Matlab and Pressing Media.

REFERENCES

[1] T. Marques, "Design and validation of pressure vessels for optimized payload in subsea applications," Thesis for: Master's, 2021, doi: 10.13140/RG.2.2.21870.13121.

[2] T. Ueda, K. Yamada and Y. Tanaka, "Underwater Image Synthesis from RGB-D Images and its Application to Deep Underwater Image Restoration," 2019 IEEE International Conference on Image Processing (ICIP), 2019, pp. 2115-2119, doi: 10.1109/ICIP.2019.8803195.

[3] T. I. Fossen, Handbook of marine craft hydrodynamics and motion control. Hoboken N.J.: Wiley, 2021.

[4] J. H. A. M. Vervoort, Modeling and Control of an Unmanned Underwater Vehicle. Christchurch, New Zealand: University of Canterbury, 2008.

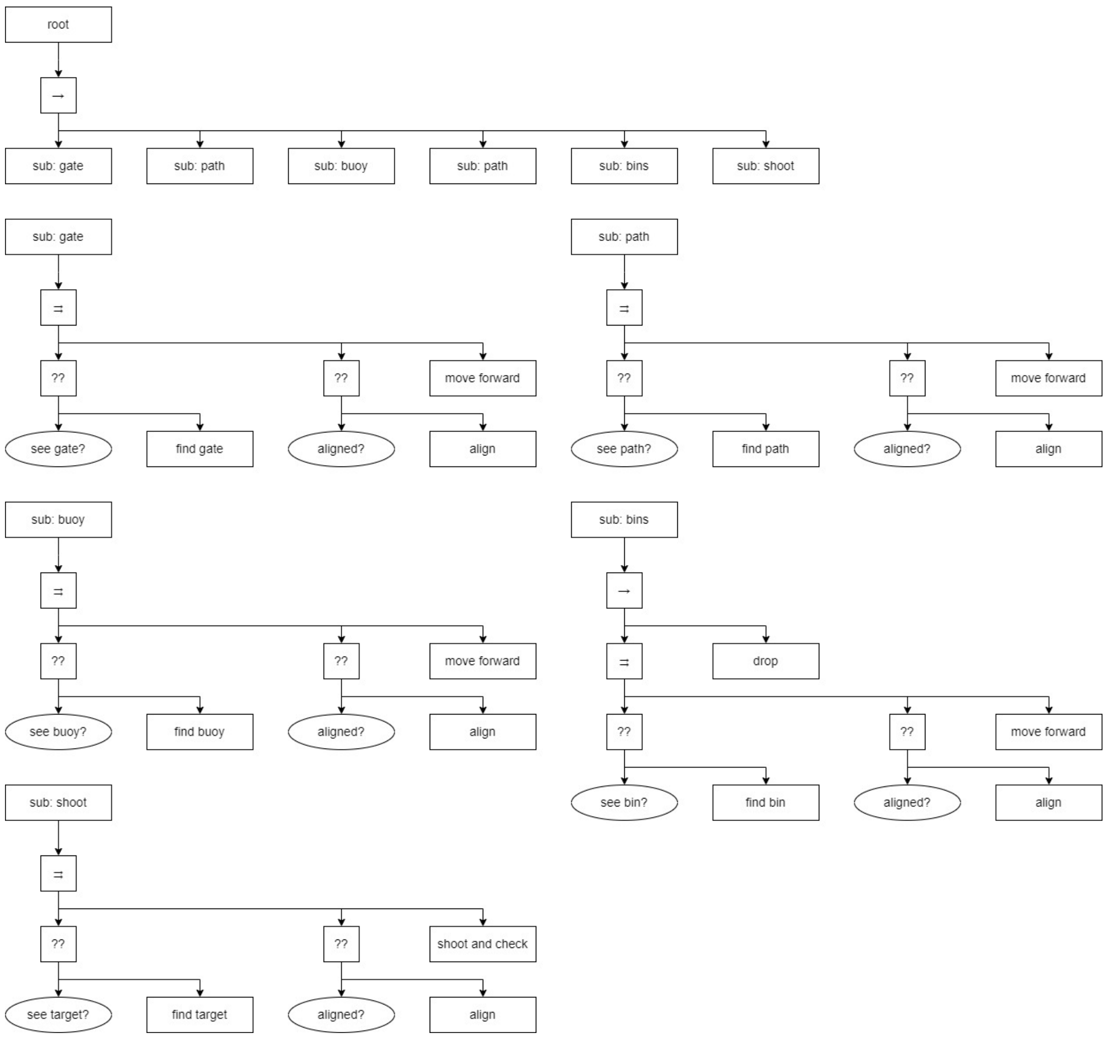
[5] L. A. Gonzalez, DESIGN, MODELLING AND CONTROL OF AN AUTONOMOUS UNDERWATER VEHICLE. The University of Western Australia, 2004.

[6] V. Moreno, C. Fletcher, I. Su, L. Gallant, J. Wu, A. Pinto, D. Qaimari, C. Polloway, R. Remo, F. Kuan, Design of the Steelhead Autonomous Underwater Vehicle for the International RoboSub Competition. UBC Subbots University of British Columbia, 2023

[7] V. Moreno, B. Sullivan, J. Wu, C. Jeffrey, Z. Yun, G. Zesiro, R. Nashota, J. Chan, H. Zafar, J. Han, D. Chung. Design of the Steelhead Autonomous Underwater Vehicle for the International RoboSub Competition 2024. UBC Subbots University of British Columbia, 2024

Appendix A: Steelhead AUV Component Specification

Component	Vendor	Model/Type	Spec	Cost (USD)	Status
Buoyancy Control	Blue Robotics	Subsea Buoyancy Foam: R-3318	https://bluerobotics.com/store/watertight-enclosures/buoyancy/float-r3318-r1/	$216	Installed
Frame	McMaster-Carr/Metal Supermarket/In-house	20x20 Aluminum Extrusion / 3/8” 6061 Aluminum Plate	https://www.mcmaster.com/5537T101-5537T504/ https://www.metalsupermarkets.com/product/aluminum-plate-6061/	$506.49	Installed
Waterproof Housing	In-house/Blue Robotics	Aluminum blocks / Watertight Enclosure for ROV/AUV (8” Series)	https://bluerobotics.com/store/watertight-enclosures/non-locking-series/wte8-asm-r1/	$1,180.34	Installed
Waterproof Connectors	Blue Robotics/ MacArtney/Blue Trail Engineering	WetLink Penetrator / SubConn Circular Series / Cobalt 14 Bulkhead Connectors	https://bluerobotics.com/store/cables-connectors/penetrators/wlp-vp/ https://www.macartney.com/connectivity/subconn/subconn-circular-series/subconn-circular-2-3-4-and-5-contacts/ https://www.bluetrailengineering.com/product-page/cobalt-14-bulkhead-connector	$350 (est.)	Installed
Thrusters	Blue Robotics	T200 Thruster	https://bluerobotics.com/store/thrusters/t100-t200-thrusters/t200-thruster-r2-rp/	6 x $220	Installed
Motor Control	Blue Robotics	Basic ESC	https://bluerobotics.com/store/thrusters/speed-controllers/besc30-r3/	6 x $38	Installed
High Level Control	Teensy	Teensy 4.0	https://www.pjrc.com/store/teensy40.html	$39.90	Installed
Actuators	In-house	Custom	N/A	$139	In Progress
Propellers	Blue Robotics	T200 Thruster Propellers	Included with T200 Thrusters	Included	Installed
Battery	Blue Robotics	Lithium Polymer Battery (14.8V, 10Ah)	https://bluerobotics.com/store/comm-control-power/powersupplies-batteries/battery-lp-4s-10ah/	$240	Installed
Converter	In-house	Custom	N/A	Free	Installed
Regulator	Texas Instruments/In-house	Linear Regulators	https://www.ti.com/lit/ds/symlink/lm340.pdf	4 x $1.96	Installed
CPU	NVIDIA	Jetson TX2	https://developer.nvidia.com/embedded/jetson-tx2	$199	Installed
Internal Comm Network	In-house	USB wires (for serial communication)	Custom	Free	Installed
External Comm Interface	Blue Robotics	Fathom-X Tether Interface Board	https://bluerobotics.com/store/comm-control-power/tether-interface/fathom-x-tether-interface-board-set-copy/	$132	Installed
Compass	N/A	N/A	N/A	N/A	N/A
Inertial Measurement Unit (IMU)	Adafruit	BNO085 IMU Breakout	https://www.adafruit.com/product/4754#technical-details	$25	Installed
Doppler Velocity Log (DVL)	N/A	N/A	N/A	N/A	N/A
Manipulator	In-house/ROVMaker	ROVMaker Underwater Servo	https://rovmaker.org/product/underwater-digital-servo/	$139	In Progress
Vision	Blue Robotics	Low-Light HD USB Camera	https://bluerobotics.com/store/sensors-cameras/cameras/cam-usb-low-light-r1/	2 x $110	Installed
Acoustics	In-house	FIR, Cross Correlation, Beamforming, etc.	Custom	Free	In Progress
Localization	In-house	Mapping Subsystems	Custom	Free	In Progress
Autonomy	In-house	Navigation	Custom	Free	
Open Source Software	N/A	ROS2, Gazebo, OpenCV	Custom	Free	
Inter-Vehicle Communication	N/A	N/A	N/A	N/A	
Programming Language(s)	N/A	Python/C++	N/A	Free	
Algorithms: Vision	In-house	Underwater Image Synthesis, Gate/Marker Detection		Free	Installed
Algorithms: Acoustics	In-house	Bandpass FIR Filter, Cross Correlation, Time Difference of Arrival, Multilateration, Beamforming		Free	Selected
Algorithms: Navigation and Control	In-house	PID controller		Free	Installed
Algorithms: Localization	Charles River Analytics, Inc.	Unscented Kalman filter	https://github.com/cra-ros-pkg/robot_localization	Free	Installed
Battery Management System	In-house		circuit designed from scratch from basic components	$100 (PCB + components, for both batteries)	Installed
Team Size			51		
Testing Time: Simulation			30h		
Testing Time: In-water			UBC Aquatic Center, about 2 hours	Free	



Appendix B: Mission Planner Behavior Tree Architecture