Design of the Steelhead Autonomous Underwater Vehicle for the International RoboSub Competition 2026

Viktor Moreno*, Brenna Sullivan*, Joshua Han*, Artur Graefenstein*,
Gyan Zesiro*, Ryan Nashota*, Abhishek Raghuwanshi*, Yerassyl Abilkassym*, Hashaam Zafar*,
Anush Mutyala*, Clark Jeffrey*, Dorson Tang*, Fei Kuan*, Emma Greville*
**UBC Subbots*
University of British Columbia
Vancouver, Canada
Email: ubc.subbots@gmail.com

***Abstract*—UBC Subbots' submission to RoboSub 2026 is the Steelhead Autonomous Underwater Vehicle (AUV). The team's competition strategy is introduced, followed by a discussion of the in-house development and production of novel elements, including machined aluminum enclosures and lids, custom neoprene seals and internal mounting, actuation and torpedo systems, system control, navigation and acoustic localization procedures, and an object recognition pipeline. Methods of physical and analytical testing such as prototyping, experimentation and simulations are explored, with results analyzed in context.**

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

I. Competition Strategy

In preparation for RoboSub 2026, the team's strategic vision has evolved to align with the AUV's current stage of development. This year's strategy is characterized by the continual iteration of the design presented at competition in 2024 and 2025. With the current aim of completing as many tasks as possible, general functionalities such as pathfinding, computer vision, maneuverability, and object manipulation were prioritized. During the competition, the AUV is expected 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, which have conducted extensive physical and virtual testing throughout the year.

This shift is also driven by a desire to prioritize student learning. In choosing to develop each part of the AUV from scratch rather than relying on off-the-shelf components, we have 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 full-scale prototypes, the software team has expanded its simulation tools and localization algorithms, and the sound localization team has developed a working hydrophone-based prototype which is expected to be integrated in the near future.

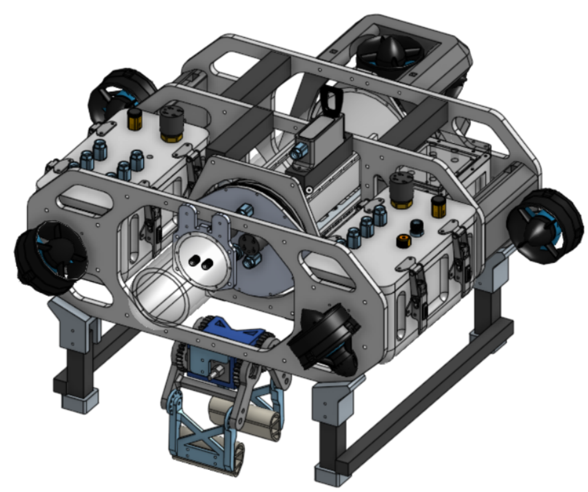


Fig. 1: Full Steelhead CAD Model

Designing and manufacturing all major subsystems in-house has increased the AUV's overall complexity, but has also provided valuable experience and an opportunity to gather extensive documentation that will inform our future work. This approach 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.

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

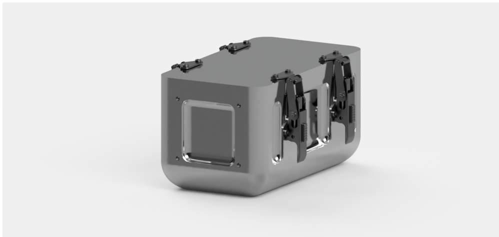


Fig. 2: Steelhead Auxiliary Hull

B. Revised Design

Steelhead's current design serves as a middle ground between previous AUV iterations and the original Steelhead concept. It was developed with competition strategy and adaptability in mind, enabling seamless integration of all subsystems. Over the past two years, Steelhead has undergone upgrades driven by the team's evolving vision, including a custom claw to replace the off-the-shelf device described in UBC Subbots' RoboSub 2024 Technical Report [7] and 2025 Technical Report [8], as well as a redesigned enclosure for actuators hardware following revised internal and external mounting schemes. After RoboSub 2026, Steelhead will be retired and repurposed as a test bench for software and mechanical development.

1) Design Goals: Steelhead prioritizes affordability, manufacturability, and adaptability. It is low-cost and rapidly manufactured, with most materials and components repurposed from previous years and aluminum sourced at scrap prices. The majority of materials and components were repurposed from previous years, 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: After virtual design and assembly in Onshape, HSMWorks generated and verified toolpaths. G-code was imported into Tormach's PathPilot CNC controller, with CAM completed over three days. Main and actuator enclosures were machined from 6061 aluminum blocks sourced from a local scrap depot. Two 5"×7"×11" blocks were cut from a four-foot stock using a horizontal band saw at the UBC Mechanical Engineering machine shop, then faced and machined on a Tormach 1100MX CNC mill at the UBC Integrated Engineering machine shop. Each enclosure took approximately 24 hours to machine, and can be seen in Figure 2.

3) Custom Neoprene Seals and Lids: The seal design adapts O-ring principles for rectangular enclosures, as off-the-shelf O-rings were unsuitable. A 1/8" face seal was selected to maximize internal volume. Each enclosure uses a 1/2" machined lid that compresses the seal into its groove, secured by four stainless steel latches for quick installation and removal as shown in Figure 2.

4) Internal Component Layout and Mounting: Two enclosures house all electronics except the battery and monitoring systems. One contains high-power components (ESCs and high-current cabling), and the other houses low-power sensitive electronics to reduce interference. Penetrators are mounted on enclosure lids, and internal components are fixed to machined lid faces for full access, as highlighted by the low power mounting scheme in Figure 3.

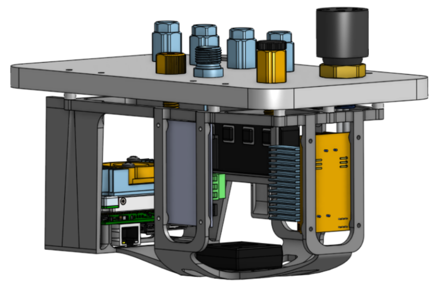


Fig. 3: Low Power Mounting System

C. Actuators

1) Claw: Steelhead's claw is powered by a single waterproof PWM servo motor connected to a gear train(Figure 4). This simple mechanism minimizes mechanical complexity to provide reliable performance. The gripper portion is made with cast Dragon Skin™ high-strength silicone that conforms to the object being picked up. This allows the claw to firmly grip a range of differently-shaped objects. The claw is mounted to the AUV using T-slot nuts that fasten the claw to the 30mm T-slot extrusions of the frame. The mount is also hinged, allowing the claw to be manually folded into the profile of the AUV for ease of transport.

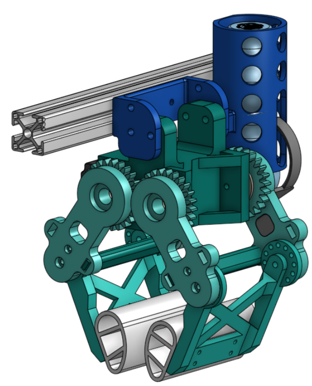


Fig. 4: Claw Design

2) Torpedo System: The torpedo system consists of two parallel tubes that each contain a torpedo projectile and a compression spring (Figure 5). When ready to fire, each torpedo is held in place with a latch. The latches can be released independently by a PWM servo motor. The close alignment of the tubes allows each torpedo to be fired sequentially while having very similar trajectories. Each tube is designed with perforations to prevent pressure buildup underwater, with the added effect of saving weight. Similarly to the claw, the torpedo system is mounted to the AUV using T-slot nuts and a specially designed mounting bracket.

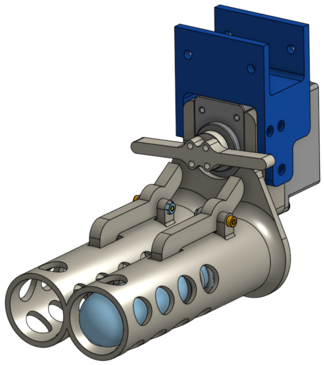


Fig. 5: Torpedo System Design

The torpedoes themselves follow a teardrop form factor (Figure 6), designed to minimize drag. The center of mass is closer to the front of the projectile in order to maximize stability during deployment. Both torpedoes were 3D printed from resin to achieve approximately neutral buoyancy, allowing them to maintain a straight trajectory underwater.

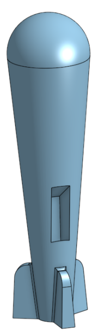


Fig. 6: Torpedo Design

3) Dropper System: In order to minimize cost and electronic complexity, the dropper mechanism was designed to integrate with the claw, using the same servo motor rather than using an independent one. The mechanism uses a vertical tube containing two marker objects (Figure 7). When the claw is closed, the markers are prevented from falling by small paddle-shaped components attached to the arms of the claw. When the claw opens, the lower marker falls from the tube. The curved section of the left paddle moves into the tube to block the second marker from doing the same. Once the claw is closed again, the second marker falls into position for deployment.

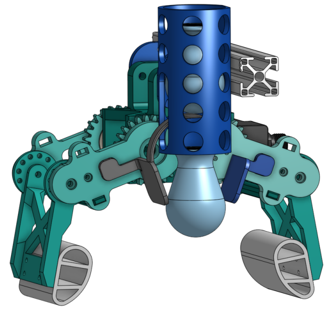


Fig. 7: Dropper Design

4) Actuator Controls: For the sake of simplicity and minimizing space requirements, both servos are controlled by a single Arduino Nano microcontroller kept in the AUV's low-power enclosure. The commands to engage each system are provided via a Serial signal from the AUV's central control system. In addition, the Nano's program includes a watchdog timer. This timer acts as a monitor to detect any interruptions in the system, allowing it to reset itself without needing manual intervention.

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. New to this year is the inclusion of a VINS-Mono state estimator. 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.

Our overall state estimator relies on an initial estimation of the robot's position in the pool, which is achieved through the VINS-Mono state estimator [9]. 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 all the degrees of freedom of Steelhead. The output is a vector of desired forces 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.

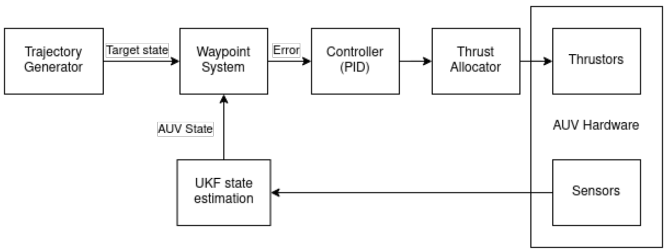


Fig. 8: Block diagram of the software control system used by the Steelhead AUV.

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 Radxa X4 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 ESCs, which then control the Blue Robotics T200 thrusters via 3 phase AC power.

3) Actuator Control: We use a dedicated ESP32 Nano microcontroller to control actuator servos. The board is soldered via pin headers to a breadboard-style through-hole PCB mounted inside the main enclosure. Servos and actuators are located externally and waterproofed, with wiring routed into the enclosure through rear end cap 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 function (e.g., firing one of two torpedoes) is assigned to a separate digitalRead enabled input pin. When the computer vision module determines an action should be triggered, it sends a high signal to the corresponding ESP32 input pin. This signal activates a preprogrammed servo procedure which initiates the relevant actuator operation.

4) Architecture: Our software architecture uses the ROS2 framework running on Ubuntu Linux as a Docker container on a Radxa X4. 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 Radxa X4 as Docker containers built using Dockerfiles. 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 (IaC) approach of using Dockerfile also enables version management of our software environment, so that changes are visible and rollbacks are easy.

Finally, our on board computer hosts Lichtblick and OliveTin web interfaces on the network, allowing easy operation of our robot including teleop, sensor visualization, launching pipelines, restart, and updates.

F. Computer Vision System

New to this year is the inclusion of the Intel Realsense D455. We utilize the RTAB-Map localization library to build a map of the course for navigation as we traverse [9]. Because of the limited range of the infared sensors used in the D455, the map is mostly used for redundancy and improving the initial estimation of VINS [8].

1) 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.

2) 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.

3) Sensors: This year, we continued using the Adafruit BNO055 IMU since the previous year gave us the knowledge needed for more precise tuning. This allowed for nine degrees of freedom (9-DOF) for enhanced state estimation. We utilize a custom plugin to translate the signals received by the BNO055 to ROS2 readable messages as this IMU is not natively supported by ROS2.

G. 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.

H. Sound Localization System

This year, the sound localization team continued development of a hydrophone-based acoustic localization system intended to determine the azimuthal bearing of underwater pinger chirps. The system uses a five-hydrophone array to measure the relative arrival time and phase of acoustic signals across spatially separated sensors. By comparing these signals, the system can estimate the direction of the acoustic source and provide bearing information to the AUV's navigation stack.

The system is divided into a custom analog front end and a digital signal processing back end. The analog front end is implemented using a set of modular PCBs designed to condition the weak hydrophone signals before digitization. Each channel includes filtering and amplification stages to improve signal quality and reject out-of-band noise before the signals are passed to the digital back end. This modular PCB approach was chosen to simplify testing, debugging, and future iteration of individual analog stages.

For digital processing, the team developed a DE1-SoC-based back end capable of handling high-bandwidth digitized hydrophone data. The processing pipeline uses IQ demodulation to translate the received chirp signals into baseband in-phase and quadrature components. From these components, the system extracts phase differences between hydrophone channels, which are then used to infer relative timing information across the array. This approach allows the system to preserve useful phase information while reducing the complexity of the downstream localization problem.

The extracted phase and timing information is passed to a software-based localization algorithm running on a Raspberry Pi. This algorithm is used to unit test the sound localization platform independently before full integration with the AUV. Testing the system in this staged manner allows the team to validate the hydrophone array, analog front end, FPGA-based processing, and software localization pipeline separately before combining them into the full autonomy stack.

III. Integration, Verification, Validation

During production, it was essential to verify the feasibility of designs in parallel. We sought to integrate different parts of the system in different stages, testing them in their respective environments.

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 10. 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. We also made use of open-source Gazebo plugins for robot localization and IMU emulation. Finally, the competition course was modeled and included in the simulation to-scale, allowing us to simulate a full course run virtually, as seen in Figure 9.

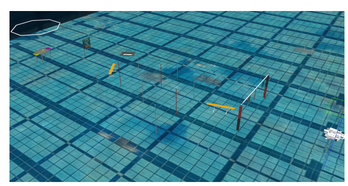


Fig. 9: Gazebo course simulation

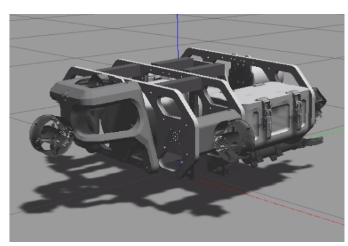


Fig. 10: Steelhead frame in Gazebo

Using models of the AUV and gate seen in Figure 11, 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 use the time we do have for real-world testing more efficiently.

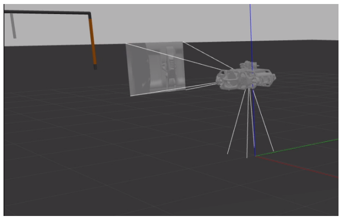


Fig. 11: Gazebo gate navigation test

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 12 indicated that the claw structure remains well within acceptable stress limits, using its truss structure to distribute load evenly.

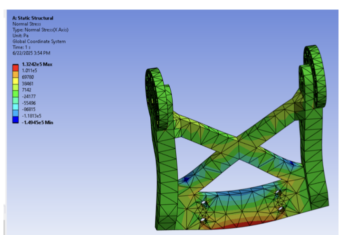


Fig. 12: FEA Analysis, Actuators

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 that 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 on the basis of competition requirements and to retain the same functionality as the original Steelhead. The stresses and deflection were 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 13, 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. 13: 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 each end was melted, we joined them. This was then placed into the Blue Robotics 8” Series enclosure in Figure 14.

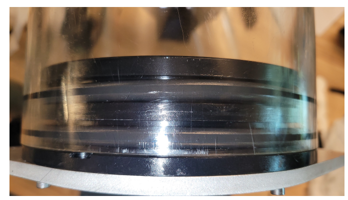


Fig. 14: The 3/16” Neoprene O-ring installed into the 8” Blue Robotics Flange

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 that reads 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 meters 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 2026: 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

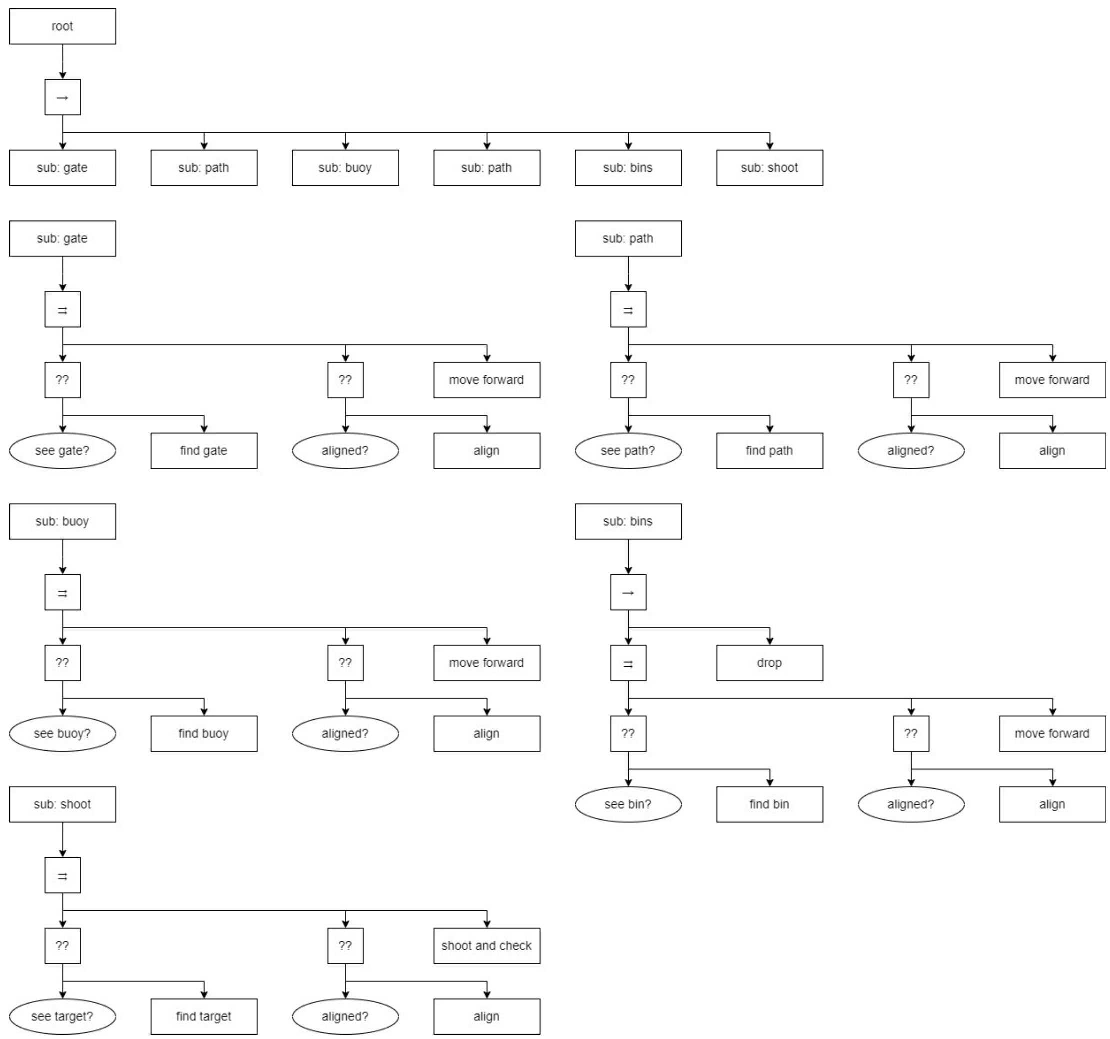
[8] V. Moreno, B. Sullivan, J. Wu, C. Jeffrey, G. Zesiro, R. Nashota, J. Han, J. Chan, H. Zafar, D. Chung, A. Raghuwanshi, D. Tang, F. Kuan, E. Greville. Design of the Steelhead Autonomous Underwater Vehicle for the International RoboSub Competition 2024. UBC Subbots University of British Columbia, 2025

[9] Qin, Tong, Peiliang Li, and Shaojie Shen. "Vins-mono: A robust and versatile monocular visual-inertial state estimator." IEEE transactions on robotics 34.4 (2018): 1004-1020.

[10] Labbé, Mathieu, and François Michaud. "RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation." Journal of field robotics 36.2 (2019): 416-446.

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	Radxa	Radxa X4	https://arace.tech/products/radxa-x4?variant=43482009338036	$115	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			$139	Installed
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			70h		
Testing Time: In-water			0h		



Appendix B: Mission Planner Behavior Tree Architecture