Design of the Triton Autonomous Underwater Vehicle for the International RoboSub Competition

Dvir Hilu (Team Captain)*, Angie Pinto*, Kobe Ng*, Kevin Huang*,
Luke Gallant*, Evelyn McGregor*
**UBC Subbots*
University of British Columbia
Vancouver, Canada
Email: ubc.subbots@gmail.com

***Abstract*—UBC Subbots's submission to RoboSub 2022 is the Triton Autonomous Underwater Vehicle (AUV). Novel elements designed in-house include mechanical components, such as our enclosure and pull-out mounting plate, and electronics, such as our battery management system. Our software pipeline, running on a Jetson TX2, takes advantage of ROS2's modular design, introspection tools, ease of integration. This report outlines Subbots' competition strategy, Triton's novel design elements, and the experimental results throughout the designing, building, and testing phases.**

***Index Terms*—robotics, navigation, autonomous, controls**

I. Competition Strategy

Our competition strategy comes from prioritizing adaptability and reliability. As a fairly young team with limited resources, we focused on ensuring that the robot can complete tasks that are earlier in the competition, while being mindful of all the design changes that will be required for future iterations. Although the specific competition tasks are unknown every year, there are consistencies such as path finding, recognizing objects, manipulating objects, etc. With this in mind, we prioritized general functionalities such as object classification underwater through computer vision, general sensor systems, and a propulsion system with 5 degrees of freedom. We expect our robot to pass the gate and follow the path markers to the first task. Our adaptable design approach will let us improve our robot for future competitions to tackle more sophisticated competition tasks.

At competition, we plan to use 4 degrees of freedom and the six-thruster configuration to maneuver the vehicle and propel it through the gate. Before the competition begins, we will test our robot at the competition pool and calibrate it to adapt to the competition environment. Testing over multiple hours of the day will prepare the robot to handle different lighting and visibility conditions during competition.

II. Novel Design Elements

A. Main Enclosure

1) Design Goals: The goals of the main enclosure are to protect critical operational components, allow for easy maintenance and to organize and secure cabling, while providing a significant buoyant force to the autonomous underwater vehicle (AUV).

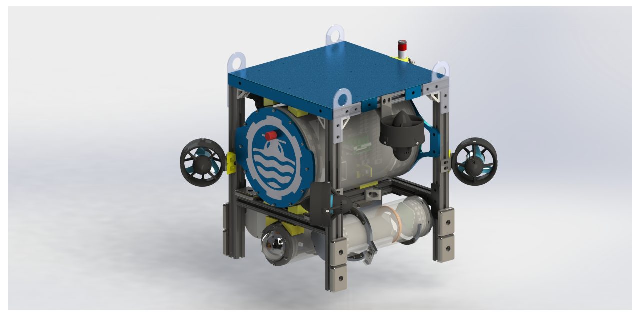


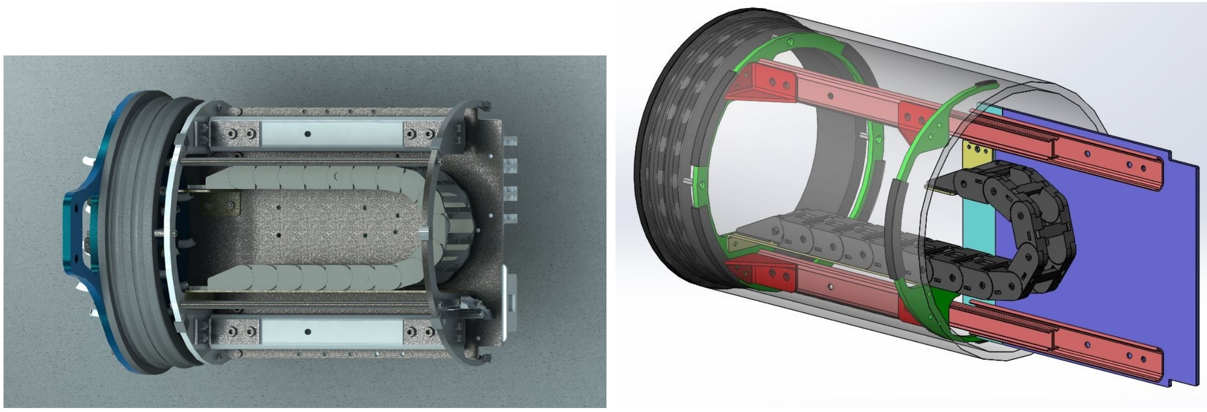
Fig. 1: Full robot CAD render

The enclosure measures 8.5" in diameter 2.5" thick with aluminum end caps. Each end has a double seal to prevent leaks. Incoming and outgoing cable penetrations are routed through one end of the enclosure. These are secured with compression seals and fitted with O-rings that provide strain relief and waterproofing. Handles extending about 2" from the enclosure were included for easier manipulation when installing the enclosure. The other end can be opened to access the inside electronics. It includes a pressure valve to equalize air pressure when at the surface.

2) Mounting Plate: The mounting plate mechanism was designed to reduce strain from waterproof connectors and increase ease of access during maintenance. Dual stainless steel rails allow the aluminium mounting plate to extend from the enclosure, giving convenient access to components. The vertical orientation allows for better loading on the slide rails with minimized deflection and sturdy component mounting. The plate can also be removed easily, if needed.

Located directly behind the mounting plate is a cable carrier. Proper strain relief was implemented on both the moving and stationary ends of the cable carrier to ensure that connectors and cables were not unnecessarily strained when moving the plate, as shown in Fig. 2b.

This also reduces the chance for failure of the waterproof connectors on the front of the enclosure during both in operation and during maintenance. Securing this assembly are aluminum brackets attached on aluminum rings pressed against the acrylic enclosure, cushioned with rubber. At the access end of the enclosure is a small latch to secure the rails



(a) Pull-out mounting plate in its stowed position.

(b) Pull-out plate when deployed.

Fig. 2: CAD renders of the pull-out mounting plate.

during operation.

3) Component Layout: Heat and signal noise were both considered when laying out the Electronic Speed Controllers (ESCs) (top left of Fig. 3), motor drivers (top middle of Fig. 3), and main computer (green block on right of Fig. 3).

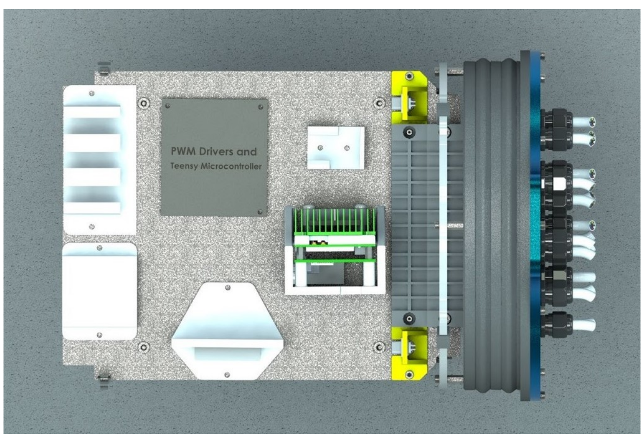


Fig. 3: Overhead view of the mounting plate and component mounts.

To minimize noise, we first ensured that sensitive components such as the Inertial Measurement Unit (IMU) (bottom left of Fig. 3) and the surface communications module (bottom middle of Fig. 3) use data cables that are less susceptible to electronic noise. We also made sure that incoming high power wires are located further away from the sensitive wires which reduces the potential for interference in signal, increasing overall reliability. Incoming signals from other enclosures would also be able to pass underneath the main computer to avoid high power cabling.

The volume of cabling was another consideration when configuring the layout. Wide gaps between components and relatively high clearance above the plate allows for flexible cable routing. A separate cable connection plate is used to secure cabling when the rail system is being moved. The center of the plate also provides additional space for USB hubs to be added.

Mounting hardware was 3D printed with Acrylonitrile Butadiene Styrene (ABS) on an Fused Deposition Modelling (FDM) printer to custom fit components. Since the mounting plate was vertically positioned, mounting hardware had to allow for components to be cantilevered. The main computer was the largest component by volume and mass, requiring a longer protrusion from the plate. It required access to multiple connector ports, leading to its current design. The ESC mounting hardware is also taller than necessary to facilitate sufficient heat dissipation. Other components were not as critical in heat dissipation nor had as much mass, so conservative cantilevered designs were sufficient.

B. Thruster Configuration and Buoyancy

1) Thruster Geometry: Triton uses 6 Blue Robotics T200 Thrusters to attain 5 degrees of freedom (DOFs) (Fig. 4). Four thrusters are positioned in plane with the centre of mass at $45°$ to the diagonals, and two are positioned vertically. This provides control over the surge, sway, heave, yaw, and roll. The vertical thrusters were placed alongside the main enclosure to allow easy access. The degree of freedom for controlling pitch was determined to not be critical to the function of the AUV for the competition tasks, and was removed to reduce cost. Additionally, to simplify the control system, we will only control the robot for stability in roll.

2) Buoyancy: To increase stability in the pitch and roll axis, it is important to position the center of buoyancy above the center of mass. Enclosures create a large buoyant force while the frame and other heavy components (i.e. batteries) significantly contribute to centre of mass. As such, the large enclosure is positioned higher up on the AUV frame, while the batteries are positioned near the bottom. Control over the placement of these components are limited by other design and space considerations. This results in a centre of mass (COM) and centre of buoyancy (COB) that are not perfectly positioned to maintaining level operation. Ballast weights are placed along the aluminium extrusion near the bottom of the robot and foam blocks are placed along the top plate of the AUV to adjust the COM and COB and maintain near neutral buoyant forces.

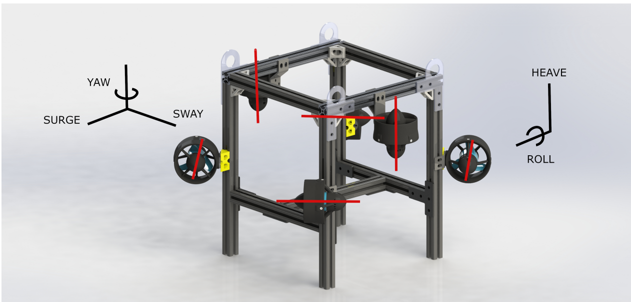


Fig. 4: AUV thruster layout illustrating degrees of freedom

Due to uncertainty in the SOLIDWORKS model, the COM of the robot will ultimately be determined using tension scales attached to four points on the AUV for multiple faces. The weights placed at the bottom of the robot will be adjusted to change the location of the COM. In turn, the thrusters will be adjusted to align with the COM to prevent unwanted pitch and roll. Although some water will be carried in the robot during motion, we neglect the head generated as the effect will be negligible at the low operational velocity.

C. Control System

1) Control Model: To control the robot, we use multiple PD controllers (pictured in Fig. 5), each designated for a single degree of freedom. The outputs from each controller are then summed together to determine the net output of the propulsion system.

Our roll controller attempts to keep the robot upright using IMU measurements. We also maintain a steady heave and sway unless required by the task. Objects detected by our computer vision system are converted to an angle relative to the AUV after correcting for distortion; this angle is used to determine the yaw error, which is the main degree of freedom we control. The AUV moves with a fixed surge speed, which we set as a constant signal added to the output of the controllers.

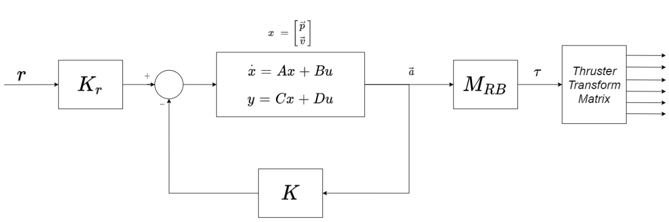


Fig. 5: Block diagram of the PD controller used by the Triton AUV.

2) Architecture: Our software architecture was designed using the open-source ROS2 framework, which allows us to implement our AUV's necessary functions as modular nodes that can run concurrently on our AUV's Jetson TX2. Telemetry and sensor data are passed between nodes as messages, which can be easily monitored for introspection and debugging. ROS2 is also language-agnostic, so we can pass messages between nodes 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. 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.

3) Testing and Verification: With little pool access due to the COVID pandemic, our team made the decision to shift our focus to developing our simulation environment. Simulation provides us with a cheaper and safer way to test our AUV, as well as 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. The SDF description allowed us to import an STL-format model of our robot from SOLIDWORKS and apply mechanical properties such as inertia and damping to generate realistic restoring forces as the vehicle moves through space. 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, as well as 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 created a variety of scripts to run individual PD controllers for each degree of freedom, allowing us to tune each one independently before summing the signals as the input into the propulsion systems. These can be used both on the real world and in the simulation environment, allowing us to tune parameters in simulation and refine in-water.

D. Computer Vision System

1) Object Recognition: The gate and marker tasks require detection of orange objects. For these tasks, we segment the image in the HSV colour space, which better models perceptual changes in colour than RGB. We then apply a convex hull algorithm to detect the gate and markers.

Many of the tasks involve recognition of printed pictures. For this, we decided to use a YOLOv4 object detection model, which can not only detect multiple classes of objects, but give their bounding boxes as well, allowing us to localize the object relative to our AUV. We trained our YOLOv4 model on synthetic data generated in our underwater simulation environment.

E. Power Distribution

Our focus in power management was toward battery and system safety. By focusing on these, we could mitigate battery failure, giving us more potential to succeed in the water. Due to the power requirements of the thrusters and the noise they generate, we utilize two voltage domains in our AUV, one for our thrusters and another for our more sensitive sensors and the main computer.

We use two, 4S LiPo batteries (14.8V) to generate our two voltage domains. Given that LiPo batteries are susceptible to damages once their voltage drops below a certain threshold, we designed an in-house system to monitor their voltage and cut-off power to the robot in the event it approaches that threshold.

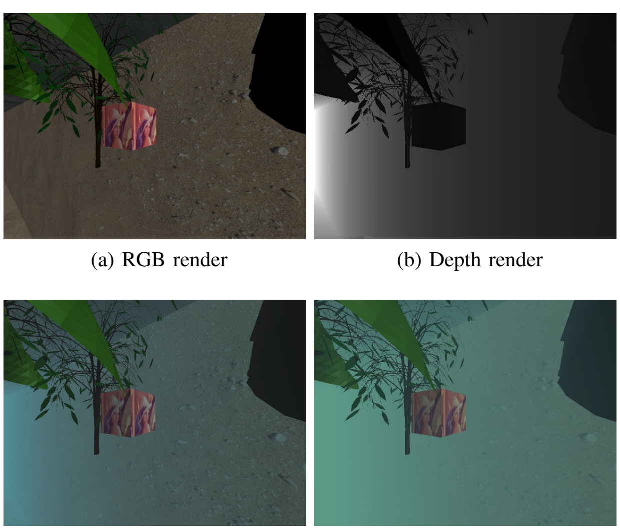
III. Experimental Results

A. Computer Vision

Our underwater synthesis module takes an RGBD image (colour and depth) rendered in Gazebo and generates an RGB image of an underwater scene. Our implementation, based on the work of Ueda et al. [1], allows us to simulate a variety of underwater environments to verify the robustness of our model. Fig. 6 demonstrates the synthesis of two different water types from the same RGB and depth images.

We generated a dataset of 800 underwater images using our pipeline, with 10% reserved as a validation dataset and the rest used for training. On our dataset, we trained a YOLO model to recognize instances of the common test image Lenna.

The score we use to judge the robustness of the model is mean average precision (mAP), which accounts for the precision (proportion of detected positives that are true positives) and the recall (proportion of positives detected as positives) of the classifier, as well as intersection-over-union of the bounding boxes. Training over 2000 iterations resulted in a mAP score of 95% on our validation dataset, so we are confident about the model's performance. Once pool access is readily available, we plan on collecting in-water data for use as a validation dataset, as real-world performance will be the true test for our model.



(a) RGB render

(b) Depth render

(c) Synthesized image (clear water)

(d) Synthesized image (murky water)

Fig. 6: Results of underwater synthesis from a) RGB and b) depth renders, using parameters of c) clear and d) murky water.

Acknowledgment

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 and allowing the team to continue functioning throughout the COVID-19 Pandemic. A special thanks to our faculty advisors: Ioan (Miti) Isbasescu and Professor Professor Lutz Lampe. Finally, we would like to acknowledge our gold level sponsors for 2022: UBC Applied Science and SOLIDWORKS.

References

[1] 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.