Troy High School NJROTC
RoboSub Technical Design Report

Thomas Nguyen-Ta, Derek Peng, Yireh Ban, Kaleb Lee, Aidan Chen, Dylan Xiang, Joshua Kim, Bruce Deng, Yifei Zhang, Daniel Tran, Elvina Liou, Gavin Gibson, Jason Pan, Kaileo Truong, Shri Krishna Sivakumar, Landis Tien, Mateus Noronha, Ryan Zhou, Yongjing Li, Yun Long, Humza Shahzad

Abstract—The Troy High School NJROTC RoboSub team's Autonomous Underwater Vehicle (AUV), Fishtank, was designed to consistently complete missions for the 2024 RoboSub competition and beyond. Built completely new this year, our sub was designed to conduct the basics of autonomous movement and to be modular, for easier possible future improvements. Our team leveraged resources such as Simscale and Onshape, and rapidly went through hydrodynamic designs for our torpedoes. Furthermore, improvements were made upon our previous design and software, including replacing our IMU with a DVL, and using YOLO v8 software in place of YOLO v4. In addition, designing Fishtank allowed our team to learn how to use these elements and various others such as ROS1 Noetic, a proportional integral derivative (PID) controller, and power distribution boards (PDBs). Lastly, due to a limited amount of time and resources, our team decided to use a single AUV and focus on its design and reliability.

I. Competition Strategy

This year's competition, Logarithmic Spiral, is updated with a couple unique twists on previous years tasks. Despite this, the course remains to consist of 6 components:

(i) Rough Seas – Coin Flip
(ii) Enter the Pacific – Gate
(iii) Hydrothermal Vent – Buoy
(iv) Ocean Temperatures – Bin
(v) Mapping – Torpedoes
(vi) Collect Samples – Octagon

As a new team competing for the second year with an AUV, our overall approach to this competition was to tackle the most fundamental components of the course first, such as the gate and surfacing task, and then move onto tasks that were more complex in order to maximize the number of points that we could score consistently within the time allocated.

A. Number of AUVs

As per rule 4.3.2 of the RoboSub 2024 Mission and Rules, each team is allowed to enter up to two vehicles. [1] While this option was considered by our team as it would decrease the time in the pool and improve task specialization, we ultimately decided against it due to the increased cost and complexity. Our team determined that investing our time and resources into one submarine would not only allow us to reduce complexity in our setup but also allow us to allocate more time towards perfecting advanced systems that would enable us to complete tasks with greater precision. As Fishtank runs more extensive trials, our team will modify our design according to the results.

B. Task Prioritization

Giving each task an equal amount of time would result in our AUV not being able to properly complete any of the tasks. Therefore, during the planning process, we opted to allocate each of the tasks a different amount of time and prioritized the tasks in the following order: Enter the Pacific (Gate), Rough Seas (Coin Flip), Hydrothermal Vent (Buoy), Mapping (Torpedoes), Collecting Samples (Octagon). This was mainly determined via the complexity of the tasks which our AUV must complete, and the time we have to design and implement solutions for these tasks.

The order was also selected based on requirements (navigating through the gate is a required task and thus was our top priority), point values, and ease. It was also extremely efficient, as the AUV wouldn't need to repeat certain paths. The prioritization can be seen in the submarine's design and software.

II. Design Strategy

A. Overall Design (Mechatronics)

The main goal for Fishtank was to create an AUV that is reliable and modular, given our lack of manufacturing equipment. With this, our mechatronics subteam settled on using a custom frame, with high support for off-the-shelf components, and a durable frame. Our design process was also heavily driven by the complexity of our electrical and software subteams having to implement their solutions.

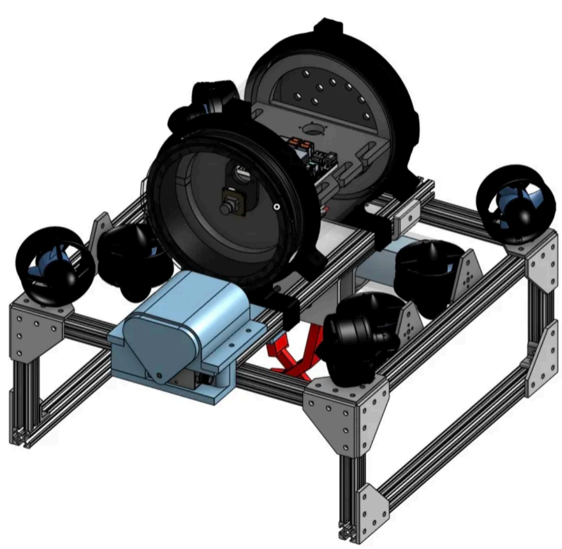

Fig. 1. CAD model of the sub in OnShape

(i) Cameras

Fishtank utilizes 2 mounted low light 1080p cameras with continuous video capture as its primary vision input. Both cameras are suited to balance between detail and performance, helping our AUV be capable of completing tasks during the competition in an effective manner. They have been placed perpendicular to each other to ensure Fishtank's efficient scanning of its environment as seen in Fig. 1. The front-facing camera is utilized for odometry and localizing the AUV's position relative to the tasks. The bottom camera is mounted to better complete the tasks throughout the competition and allow the AUV to better orient itself. This setup was chosen for its cost-efficiency, reliability, and simplicity.

(ii) DVL

After consideration of the complexity of the use of sonars in Fishtank's systems from last year and interaction with other teams, a DVL was implemented in place. Due to an inertial measurement unit having mediocre outcomes as its accelerometer derives results with errors that are not tolerable after a short duration of movement, we superseded the IMU with a Doppler Velocity Log (DVL). It soon became an essential component of the localization process as a result of its great underwater accuracy.

(iii) Kill switch

Upon analysis of how our kill switch from last season burnt due to high voltage, this year we decided to put an SSR (Solid State Relay) in the main enclosure of our AUV in order to control the main current that is sent to the robot.

The kill switch outside this enclosure, that is attached to the AUV on the rear of the left side of Fishtank, runs a control current using a 9V battery. This lower current is much safer than the high voltages that the AUV is accessing. The external kill switch communicates to the SSR when to stop sending current to the rest of the robot. This results in power to the motors, as well as any other systems aboard Fishtank, being cut. Furthermore, the placement of this SSR in the outspread of voltage, protects the rest of the robot from harm, in the case of high voltage being sent to the SSR. In this instance, only the SSR is damaged, and not the other components of the robot which receive voltage from the SSR. Due to this structure, the SSR acts as a safeguard for the body of the AUV.

(iv) Upgraded Claw

Last year, though our team did not get a chance to use our claws, we still developed it this year in order to give us an opportunity to earn more points and do better during the competition.

For this season, both our final model and prototypes are 3D printed in PLA for added rigidity in the system. During the process of development of our claw, experimentation with interlocking and non interlocking fingers occurred for our manipulator. Initially, non-interlocking claws that were designed to grab all the objects in this year's competition were prototyped. Yet, upon consideration of the tube worm seacreature's structure, we switched our design to an interlocking structure in order to be able to grab the tubeworm and the rest of the objects during the competition.

(v) Dropper System

After much consideration, we chose to make our dropper utilize a barrel mechanism in order to keep the dropping point as consistent as possible relative to the robot. Furthermore, this mechanism allows the system to store multiple droppers. In regards to the item being dropped, we chose to design with a raindrop shape. This design was inspired by AVBotz's item due to its geometry allowing it to drop more consistently.

(vi) Torpedo System

With the addition of torpedoes came the need for many iterations and tests to create the most optimal system. These tests were effectively carried out using SimScale, a cloud-based computer-aided engineering software, which allowed us to perform fluid dynamics simulations on any computer to optimize the shape of the torpedoes [2] as seen in Fig. 2. For propulsion, we opted for springs over last year's CO_2 canisters due to the reusability aspects which allowed us to test more often but also allowed us to simplify calculating the motion of the projectiles.

Furthermore, we concluded that Version 3 of our torpedo was the best compromise between V1 and V2 because it retains V1's ability to cut through the water but due to the curved hull, it has more stability and can guide through the water more streamlined compared to a cylindrical hull, shown in Fig. 2.

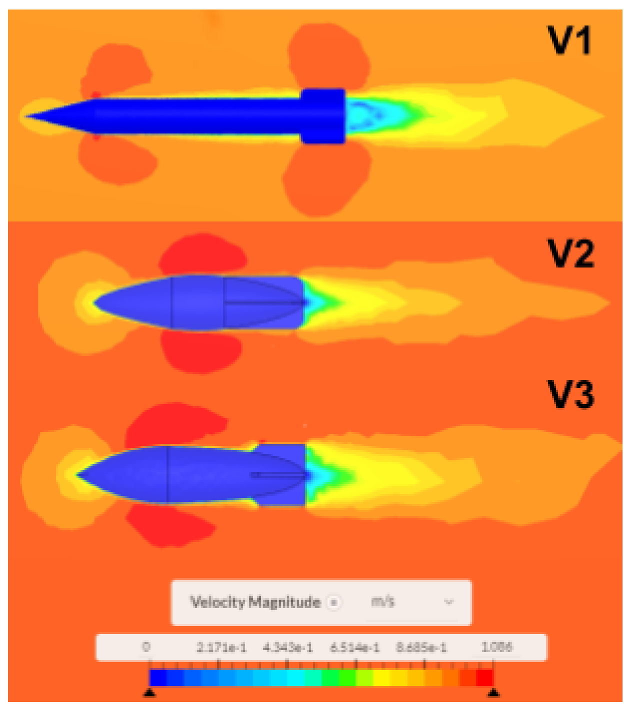

Fig. 2. Different torpedoes under testing in SimScale's CFD (computational fluid dynamics) simulation tool

(vii) Computer

Fishtank's onboard computer is a Jetson

Orin Nano. It runs most of Fishtank's software including the object detection algorithm, mission planning, and more. Our team chose to switch to a Jetson Orin Nano over a NVIDIA Jetson Nano, which was used last season, due to its faster processor and improved compatibility with our software. Additionally, it allowed us to implement newer versions of ROS, namely, ROS1 Noetic.

B. Software Overview

(i) Mission Planning

Our team narrowed down our options to either a Finite State Machine (FSM) or using a behavior tree. We ultimately chose to utilize a behavior tree, using BehaviorTree CPP with ROS for high-level decision making, due to its simplicity and flexibility. As a first-time team, its capabilities in abstraction and easy management of nodes made it a suitable choice for use. A FSM would overcomplicate our design and make scaling and further development exponentially more difficult. Additionally, its specific integration with ROS made it the optimal choice. Its ability to be configured and altered during runtime makes it easy to test and fix in a competition or testing environment.

(ii) Computer Vision

Computer vision was used as a basis for navigation and decision making, as hydrophones were not utilized in Fishtank. The AUV runs YOLO v8 and its resources in using Path Aggregation Networks and Cross Stage Partial Networks in conjunction with OpenCV and Tensorflow for optimization and accuracy. [3] The core computer vision was switched from YOLOv4 in the previous season to YOLOv8 because of its increased speed and improved accuracy in diverse environments.

Though Fishtank utilizes the BlueRobotics Low-Light HD USB Camera, which is calibrated for underwater low-light conditions, tint and discoloration from water depth had the ability to impact the computer vision performance. To combat this, Sea-Thru, an algorithm designed by Derya Akkaynak, shown in Fig. 3, is to be used, allowing for improved computer vision accuracy and more flexibility in obtaining training data. [4] With color correction, it is possible to use training data exclusively from on land and in the simulator without the need to replicate the effect of discoloration.

Fig. 3. Non-color corrected image on the left and image with Sea-Thru algorithm applied on the right

(iii) Architecture and Navigation

To operate efficiently and autonomously in an underwater environment, we rely on a series of processes. These processes work together in a coordinated and consistent manner, utilizing different algorithms to navigate and accomplish tasks as seen in Fig. 4. For communication between different algorithms and parts of the system, we used the publisher/subscriber model provided by ROS, and then MAVROS to communicate over MAVLink to our Pixhawk PX4.

For navigation and localization, we used a Doppler Velocity Log (DVL) due to its ability to track our velocity. Furthermore, with the use of proportional integral derivative (PID), Fishtank can adjust the PWM of the motor to get to the location more accurately with feedback from the velocity collected by the DVL. In addition, we can determine the distance our AUV has traveled over a period of time, helping us understand the AUV's environment at any point of time. To aid in localization, the built-in IMU and compass of the Pixhawk FCU was also utilized.

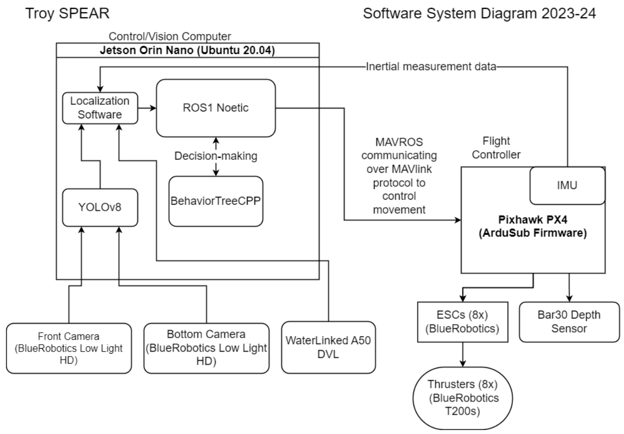

Fig. 4. Software Diagram illustrating our system organization

The steps taken by Fishtank after an object is detected are split into 3 stages:

Initial Detection - Fishtank moves around the area of the pool it is in until it recognizes an object significant to the current task.

Repositioning - Once the AUV finds the object it is searching for, it repositions itself such that the object is centered in the frame of the forward-facing camera.

Distance detection - Using localization with the known height and width of the object, Fishtank calculates its relative position to the task with the front camera.

III. Testing Strategy

Our testing plans consist of three stages, based around the philosophy of gradual development based on the importance of the particular function tested. Our three phases are:

1) Testing and preparing the individual sensors themselves, before any of the sub was built. Our focus for this step was computer vision as it is a key part of our localization strategy. As a first phase, this also made it easier for us to integrate our cameras, DVL, IMU, and other sensors into our final submarine design as we understood the necessary measures for their use.

2) Simulation: In order to ensure that all subsystems would be able to integrate together on the final submarine and the sub would be able to properly move, we simulated the submarine in an aquatic environment after designing it.

3) Pool Testing: As the competition is held in a pool environment this year, we will move on to testing the movement and successful sensor output in a community pool. We will test the functionality of the sub in respect to each task separately, and then move on to integrating each part of the competition together afterwards.

A. Vision

There are two cameras on the submarine, one facing forward, centered on the front of the main enclosure, and one facing down, located at the bottom of the electronics bay. Both cameras are essential to localization of the sub and the completion of tasks. The first aspect of Fishtank that needed to be tested was the vision algorithm. This was arguably the most integral aspect of Fishtank if it were to succeed in the competition. We needed to test if it was able to correctly identify objects and images based on the database we provided. Fishtank's vision algorithm utilizes a framework that allows for us to switch between databases in a single line of code. Thus, we are able to use different databases instantaneously to identify numerous images and objects.

Upon completion of our vision algorithm, our testing phase shall begin. The first steps in this process will include manually inputting images into our algorithm. Some of these images include an underwater shot of a pool to test as a baseline, and homemade orange markers and other pictures similar to the one used for the competition to see if the vision algorithm could successfully identify it. Once our algorithm is able to accurately identify and classify these images, the same process will be used for the cameras on the AUV.

B. Simulated Testing

For testing movement without having to use the AUV, we used a variety of software tools, such as BlueROV2's simulator [5], powered by Gazebo, and RViz, a powerful 3D visualization tool in the ROS framework [6]. For the environment within Gazebo, an underwater environment was emulated through the use of Project DAVE. This environment simulated water and the surface of the ocean, allowing us to test the design of our AUV. Furthermore, by using an emulated flight controller, Ardupilot's SITL, we were able to send commands, sending feedback from the simulator to the virtual AUV in a loop. In regards to RViz, we loaded a URDF file that described the specifications of a 3D model of our AUV, shown in Fig. 5. We visualized the transformation frames (TF) to see how different parts of the robot were oriented and positioned relative to each other and monitored the position and movement of the robot's joints.

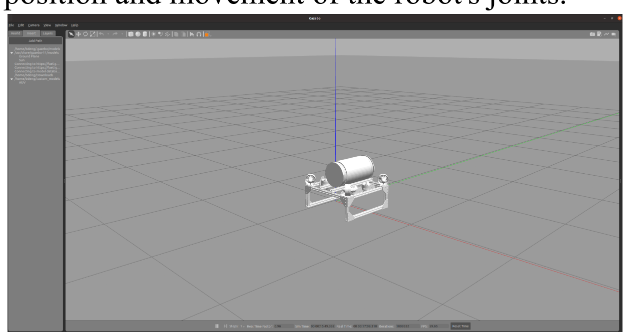

Fig. 5. AUV in a customizable environment in Gazebo

C. Physical Testing

To ensure a watertight environment for our electronics, we used a vacuum pump to test whether there would be any chances of leakage in our main enclosure and battery enclosure.

After ensuring safety, we moved forward to testing motor function. Something we struggled with last year was veering after steering the AUV for extended periods of time, and so something we wanted to test at the very beginning was our PID controller and staying fixed to a compass heading. To further ensure stability, we tested in both calm and turbulent water (generated through causing waves in a pool).

To test the ability of the submarine autonomously, we mainly focused on the submarine being able to complete the gate task, in order to pass the qualification round and also for the first part of the course. As a result, we used PVC to construct a similar frame to test it in a community pool. Due to ROS's data logging tools, we were able to record data that was sent to topics in ROS by the sensors, and evaluate the functionality of the sensors while testing the submarine.

This year we modified our Blue Robotics tether and used BlueTrail Engineering Cobalt Bulkhead connectors to add a quick disconnect feature to the AUV. This allowed us to have a more reliable method of transferring code into our AUV compared to relying on the performance of an external WiFi router. This also lets us modify the PID while the sub is submerged during testing.

IV. Acknowledgements

Our team could not have functioned without guidance from our mentor, Lt. Roger Fronek, and support from our generous sponsors. We would like to thank the following organizations for sponsoring our team: Troy High School NJROTC Booster Club. We would also like to thank the following organizations for providing our team with software and materials for discounted prices, saving our team a total of around $7,000 in costs: Blue Trail, Blue Robotics, WaterLinked, Onshape, and Simscale. Additional thanks to Team Inspiration and CSULA Robosub for helping us better understand the competition and how to set up our AUV. We would also like to thank RoboNation for helping us get access to data and resources to perform better in the competition.

V. References

[1] "Resources." *RoboSub*, Available: robosub.org/resources/ (2023/06/16).

[2] K. Gore, A. Gote, A. Govale, A. Kanawade, and S. Humane, "Aerodynamic Analysis of Aircraft Wings Using CFD," *International Research Journal of Engineering and Technology (IRJET)*, vol. 5, no. 6, pp. 639-644, June 2018.

[3] Supeshala, Chamidu. "YOLO v4 or YOLO v5 or PP-YOLO?" *Chamidu Supeshala*, Available at: https://towardsdatascience.com/yolo-v4-or-yolo-v5-or-pp-yolo-dad8e40f7109 (2023/06/16).

[4] Akkaynak, Derya. "Sea-Thru." *Derya Akkaynak*, Available: www.deryaakkaynak.com/sea-thru (2023/06/16).

[5] "BlueROV2 ROS Simulation" *UUVControl*, Available at: https://github.com/UUVControl/bluerov2 (2023/06/16).

[6] "rviz" *ROS*, Available at: https://wiki.ros.org/rviz

Appendix A: Component Specifications

Component	**Vendor**	**Model/Type**	**Specs**	**Qty**	**Total Cost**
Electronics Kit	BlueRobotics	Electronics Kit (terminal blocks)	-	1	Reusing from last year sub
Battery	Turnigy	Turnigy Graphene Panther 5000mAh 4S 75C	-	1	$81.74
CPU	Nvidia	Jetson Orin Nano	GPU and 8 GB of RAM	1	$499.99
Camera	BlueRobotics	Low-Light HD USB Camera	1080p30	2	$99.00
Pinger Localization	BlueRobotics	Ping Sonar Altimeter and Echosounder	BLUART USB to TTL Serial and RS485 Adapter	2	$558.00
3D Printer Filament	Elegoo	PLA	1.75mm 1.0kg	2	$37.98

Battery Bags	Amazon	Tenergy 2 Pack, Fire Retardant Lipo Bags	-	1	$11.99
Kill Switch	Amazon	Sensata-Crydom EL100D20-05 SSR	Input: 4~8 VDC Output: 3V~100VDC Load Current 20A	1	$65.44
Kill Switch	Amazon	Hmknana IP67 Waterproof Inline Cord Switch	IP67 12V-24V 20A	1	$14.99
Tether Spool	BlueRobotics	Fathom Spool	-	1	$750.00
Thrusters	BlueRobotics	T200 Thruster (w/ESCs)	-	8	$1808.00
Controller	Amazon	Logitech F310 Wired Gamepad Controller	-	1	$15.19
Torpedo Propulsion	Amazon	SP 9706 Spring Mechanical Compression Spring Stainless Steel Extension Spring, 1/2-inch by 1-1/2-inch (6 Pieces)	Stainless Steel Max. Load: 6.84 lbs. Max Deflection: 0.4in	1	$6.79
Algorithms: vision	-	-	Sea-Thru, YOLO v8 Object Detection	-	-
DVL	WaterLinked	A50	Min. distance: 5cm	1	$2500.00
Algorithms: localization and mapping	-	-	-	-	-
Open source software	-	-	ROS Noetic, OpenCV, YoloV8, Tensorflow	-	-
Team size (number of people)	-	-	20 persons	-	-
Expertise ratio (hardware vs. software)	-	-	10 mech to 10 software	-	-
Testing time: simulation	-	-	14 hours	-	-
Test time: in-water	-	-	7 hours	-	-
Programming languages	-	-	Java, Python, C++	-	-