RoboSub 2024 Technical Design Report

Recinto Universitario de Mayagüez (RUMarino Autonomous Underwater Vehicle)

Ordep E. Agustin, Kevin Y. Díaz Vélez, Victor O. Riefkohl, César A. Ruiz Navarro, Willie Baez Justidiano, Anibal E. Rosado Reyes, Andrea P. Sepúlveda Vargas, Claudia Vélez Tosado, Diego Quiñones, Edyan A. Cruz Vélez

***Abstract*—For RoboSub 2024, the RUMarino team comes with a new constructed AUV, Hydrus. This AUV being more advanced and modular than the previous AUV, Proteus, the team's strategy involves deploying Hydrus to efficiently complete all tasks with an optimized space, weight, and structural design. This new structural design includes a more robust platform, a dual hull design, vectorial motors, and torpedo launchers. Electrical work has been centered on maintaining current systems while researching and applying new technologies and development processes. In terms of the software, the team is working all tasks independently by creating different modules from scratch utilizing tools such as ROS to manually manage software alongside the integrations of a DVL (Doppler Velocity Log) system to keep track of the current state of Hydrus.**

I. Introduction

The purpose of the RoboSub competition is to enhance the community of innovators capable of substantive contributions to the domain of autonomous unmanned underwater vehicles (UUVs). This enhancement is achieved by providing a venue and mechanism whereby the practitioners of the autonomous UUV community may form new connections and collaborations, increase their proficiency and inventiveness, and foster their passion for robotics and the underwater domain. While RoboSub has been an *autonomous* vehicle competition since its inception in 1998, there will be a continually increasing emphasis on autonomy going forward.

II. Competition Strategy

This year the team will be competing with Hydrus, an upgrade to our previous AUV, Proteus. Designing a software solution for a submarine is a complicated process with all the details, data processing and calculations it needs to perform to be able to control and efficiently synchronize every part of its hardware to be able to complete the necessary tasks assigned to it; in a simple case, traverse efficiently through a three-dimensional space. The decision our team made to design and develop the software was splitting the different functions that the software will have to perform and solving each requirement or task independently of each other by creating different modules. Some examples needed for this type of system are the controller modules that manage the device's different components that control the position of the AUV, vision modules that can capture and process what the submarine is seeing, and the sensor module that manages the different sensors like depth and pressure and the logic module where the decisions are made based on the information gathered from all the modules mentioned before.

One of the disadvantages of developing software based on modularity is that every module used needs to be able to efficiently function and communicate with each other [1]. Using the competition as an example [2], badly implemented modules could lead to slower data processing making the submarine performance extremely poor and slow which is detrimental when participating in a competition. To help us solve this problem of managing the different modules efficiently we are using the Robot Operating System (ROS). With this framework we can easily implement independent modules where our team can focus on the desired module behavior and the heavy lifting of communication and synchronization is taken care of by ROS. One of the great advantages of using ROS is that any information processed, or any information needed from other modules is made accessible to every module in the system by ROS making it a great option [3].

By maintaining the current design process the submarine will be able to use its modules concurrently providing the advantage of making logical decisions while gathering data and processing it at the same time to be able to alter its decisions while at the same time performing them. A very clear example where the ability to use all its modules concurrently is selecting the gate which is the first task the submarine will need to perform. The submarine can keep track of its current position with the Doppler Velocity Log (DVL) while at the same time find and locate the gate and its current position relative to the submarine, then proceed to pass the corresponding gate and make any necessary adjustments if for some reason the position of the submarine or gate where to change due to external factors and apply them accordingly, instead of performing all these actions one at a time. This type of task, tracking an object while positioning itself according to it, can be seen as simple, but failing to achieve it reliably and efficiently can cost the whole competition.

The control systems team built a Pyrunner library (code generator for python based on block programming software like Simulink and LabVIEW). In addition, a system independent communication architecture for the control system was designed and implemented using ROS. Currently, the code is being tested using a simpler system where the dynamics and controller were already designed beforehand.

A. Course Strategy

As we approach the competition, our strategy has been one of our most discussed topics. During the course all types of failures can happen and that's why we have prepared redundancies and backup plans. First, we have the original plan of using all our sensors in tandem with our camera to identify the location of the AUV in its environment and moving accordingly keeping the objectives in view. If some of our sensors begin to give erroneous data during the run, we can shift to the built in sensors within our camera and base the environment and point of view from our camera's location in the front lower cabin. With this we can ensure that no matter what may occur, we can always pivot and continue our run without having to lose time troubleshooting.

This system allows us to not only have safety nets making our AUV's data as reliable as possible. We also must prioritize task points so a careful analysis of our AUV's capabilities and testing until the competition can help us narrow down the strategy to accurately complete tasks while gaining the most points possible. Not only do we have to prioritize points but also the safety of the team and everyone involved. To cover this, the team has integrated two kill switches on top of the AUV to shut off either the motors or the electronics in case of a failure. Similarly, a valve system has been implemented to secure the AUV's seal, preventing any leaks from occurring for 2 hours and 10 minutes, as of our last tests. This valve system also helps the team quickly undo the seal if we must inspect the interior of the vehicle or perform any type of repairs.

II. Design Creativity

A. Mechanical Systems

1) Design of Main Hull

Hydrus' main hull is composed of a central cabin made of aluminum that serves as the main hub for connections and wiring for the engines or other components, and the valve to depressurize the submarine. Two cabins made of acrylic, where the electrical components are stored, are attached to both sides of the central cabin. This is also composed of a mechanical claw and a camera inside an acrylic cabin; both positioned in the front of the submarine.

2) Design of Battery/Camera Hull

As stated previously, the main haul in combination of the two acrylic cabins serves as the location for all the electronics of the AUV. Similarly, in the middle of the AUV the battery has been placed to serve as a counterweight of the electronics on both sides, the torpedo system and our third cabin, in the front of the submarine. In the front most cabin, the eyes of our vehicle, a ZED 2i camera, is attached to two rotating axes allowing for 180° movements via a servo motor, improving our overall visibility.

3) Design of Actuation Systems (Torpedo / Gripper)

Tasked with the challenge of lifting different objects, a gripper was implemented and is being tested at different angles to provide the best opportunities to succeed.

Our gripper motor accepts standard servo-style PWM signals which resembles how the thrusters are controlled, simplifying the use of the gripper. Similarly, the torpedo system utilizes two similar motors to launch our torpedoes within the safety parameters accurately.

B. Electrical Systems

1) Power System-
   Our power system is made up of two voltage phases: 22V and 12V. A 22.8V, 23,000mAh battery is utilized to provide power to the PCB. A ZK-12KX voltage regulator is used to lower the voltage to enable the 12V phase. We have a system of fuses and relays set up to protect important components such as thrusters and micro-computers.

2) Board Design – Optimizations
   The team designed one central board from which the whole AUV draws power. Our previous PCB design did not have the capabilities to add the gripper and torpedo system. Faced with this problem, we redesigned our PCB to add these key components to our AUV. Also, due to limited knowledge about PCB's there were 4 tests batches before arriving on a design that fulfilled the needs of our AUV.

C. Software Systems

1) Mission Planner

The mission planner of the AUV submarine is built using the State Machine Library (Smach), which allows for the creation of flexible and reusable state machines. The submarine is equipped with a variety of sensors, including cameras, Inertial Measurement Units (IMUs), Doppler Velocity Logs (DVLs), and passive sonars. These sensors provide a wealth of data that must be processed and utilized to achieve mission objectives.

To handle the data and execute tasks in an organized manner, the mission planner uses a hierarchical task structure. Basic tasks, such as sensor data acquisition and movement commands, are abstracted into higher-level tasks that form complex mission behaviors. This modular approach allows for easy configuration and extension of mission plans.

An example of this hierarchical task structure is the movement state machine, which controls the submarine's navigation and positioning based on sensor inputs and mission goals. An example of this is the movement of State Machines.

2) Computer Vision

The AI and computer vision stack of the AUV submarine is crucial for its autonomous operation, enabling it to detect and interact with underwater objects in real-time.

The core component of this system is the YOLO-World model, an innovative approach that enhances the traditional YOLO (You Only Look Once) object detection algorithm with open-vocabulary detection capabilities. YOLO-World uses vision-language modeling and pre-training on large-scale datasets to overcome the limitations of predefined object categories.

Camera Integration

ZED Stereo Camera:

- Resolution: The ZED camera captures high-definition video, providing detailed images necessary for precise object detection and recognition.
- Depth Perception: Using stereo vision, the ZED camera generates depth maps, which are crucial for understanding the spatial arrangement of objects and the environment.
- Field of View: The wide field of view ensures that a large area is monitored, reducing the chance of missing critical objects or obstacles.

Object Detection and Tracking

The YOLO-World model processes the images and depth data from the ZED camera in real-time. Key features of the system include:

Open-Vocabulary Detection:

- The YOLO-World model utilizes a new Re-parameterizable Vision-Language Path Aggregation Network (RepVL-PAN) and region-text contrastive loss, facilitating the interaction between visual and linguistic information. This allows the AI system to detect a wide range of objects in a zero-shot manner, meaning it can recognize objects it has never seen before without additional training.

Depth-Based Localization:

- By combining depth information with object detection, the AI system accurately determines the position of objects relative to the submarine, which is essential for navigation and interaction tasks.

Environmental Mapping:

- The combination of high-resolution imaging and depth perception enables the creation of detailed 3D maps of the underwater environment, assisting in obstacle avoidance and path planning.

Overall, the integration of advanced AI and computer vision capabilities ensures that the AUV submarine can perform its missions autonomously with high precision and reliability. The use of the ZED stereo camera and the YOLO-World model represents a state-of-the-art approach to underwater exploration and object detection.

3) Control System

The control system for the AUV submarine is designed to manage its movement and navigation autonomously. The core idea of the control system is to translate high-level navigation commands into precise motor actions to achieve desired positions and orientations in the underwater environment.

Key Components and Functions

Thruster Control:

- The submarine is equipped with eight thrusters that control its depth, rotation, and linear movement. The control system calculates the required thruster values based on the difference between the current pose and the target pose.
- Depth Control: Adjusts the submarine's vertical position using specific thrusters.
- Rotation Control: Manages the submarine's orientation by applying differential thrust to the rotational thrusters.
- Linear Movement: Controls forward and backward motion by adjusting the linear thrusters.

Movement Logic:

- The movement logic ensures that the submarine moves towards the target pose in a step-by-step manner. It first addresses vertical movement (depth), then rotational alignment, and finally linear movement towards the target.
- This hierarchical approach ensures stability and precision in navigating towards the target waypoint.

Real-Time Adjustment:

- The control loop runs at a set frequency (e.g., 10 Hz) to continuously adjust thruster outputs based on the latest sensor data and the current state of movement. This allows for real-time corrections and smooth navigation.

III. TEST STRATEGY

For our tests, we mainly focus on the AUV’s functionality as well as safety and reliability. The team rigorously tests all the electronics before implementing them into the AUV as well as test the seal weekly to ensure there are no leaks present at the time of testing. For our software test, we run Docker containers and create synthetic data to train our models and verify their functionality as well as cooperate with the other divisions to run multiple pool testing at our campus pools. With the competition closing in, these tests become more frequent and stricter to ensure all the safety precautions and requirements are met while maintaining the AUV’s functionality.

ACKNOWLEDGEMENTS

We want to thank all our sponsors for believing in us and for their continued support throughout the years. We also want to thank our advisor, Dr. Raúl Torres, for helping us get back on track by providing support and guidance to the RUMarino team. Similarly, we want to thank the IEEE UPRM Student Chapter, specifically the IEEE RAS and IEEE CS for helping the project stay relevant and present in the university by helping promote the project as an opportunity for students to grow and cultivate professional experience in the STEM fields. We also want to thank the University of Puerto Rico – Mayagüez and its staff for helping us make the trip planning for the competition possible. Special thanks and appreciation to the RoboNation and RoboSub teams for welcoming us back and staying in touch with us during our hiatus. More special thanks to the RoboBoat UPRM team for giving us the first steps into preparing for the competition and trip plannings. Lastly but most definitely not least, huge thanks to all the members of the RUMarino team over the years, if we list them all this would never end, without everyone's help and contributions none of this could have been possible and we hope to live to your expectations and more as well as keeping the team going for more years to come.

REFERENCES

[1] University of Lucknow, Software Design Principles. Accessed on: Jun. 4, 2021. [Online]. Available: https://www.lkouniv.ac.in/site/writeread data/siteContent/20200329162359626020200329162359626 0rohit_yadav_engg_Software %20Engg_2.pdf

[2] RoboSub, 27th Annual International RoboSub Competition Team Handbook_v1.0 , June. 8, 2024. Accessed on: June. 8, 2024. [Online]. Available: https://robonation.org/app/uploads/sites/4/2021/04/ RoboSub-2021-Mission-and-Rules__V1.pdf

[3] Open Source Robotics Foundation, Inc, Is ROS for Me?, Accessed on: Mar. 15, 2021. [Online]. Available: https://www.ros.org/is-ros-for-me/

[4] "Brick by Brick". Accessed on: May.9, 2021. [Online]. Available: https://static1.squarespace.com/static/55 b7729ce4b00871cfc18b4a/t/57d0737f6a496397948 89141/1473278847679/Rook_Chapter_Website.pdf

[5] Google LLC, Protocol Buffers. Accessed on: Feb. 20, 2021. [Online] Available: https://developers.google.com/p rotocol-buffers

[6] Open Source Robotics Foundation, Inc. Gazebo Simulation, Open Robotics Foundation, 2014. Accessed on: Jan. 27, 2021. [Online]. Available: http://gazebosim.org/

[7] The SWARMs Project, Unmanned Underwater Vehicle Simulator, Accessed on: Jan. 27, 2021. [Online]. Available: https://uuvsimulator.github.io/packages/uuv_simula tor/intro/

[8] The SWARMs Project, Unmanned Underwater Vehicle Simulator, Accessed on: Jan. 27, 2021. [Online]. Available: https://uuvsimulator.github.io/packages/uuv_simula tor/intro/

[9] "Stereolabs Docs: API reference, tutorials, and Integration," Stereolabs. [Online]. Available: https://www.stereolabs.com/docs. [Accessed: 01-Jul-2024]

[10] Ultralytics, "Ultralytics solutions: Harness yolov8 to solve real-world problems," Ultralytics YOLO Docs, 02-Jun-2024. [Online]. Available: https://docs.ultralytics.com/solutions/. [Accessed: 01-Jul-2024]

[11] AILab-CVC, "AILab-CVC/Yolo-world: [CVPR 2024] real-time open-vocabulary object detection," GitHub. [Online]. Available: https://github.com/AILab-CVC/YOLO-World. [Accessed: 01-Jul-2024]

APPENDIX A

Some of our models of the AUV and equipment.

Figure 1: Hydrus dismantled, compoment view

Figure 2: Hydrus CAD Model

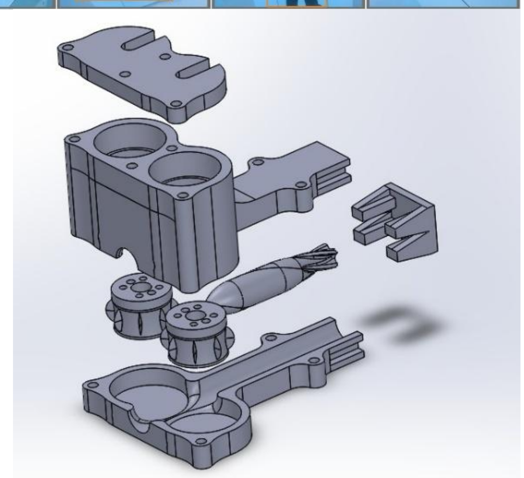
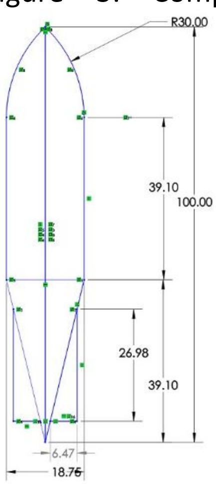
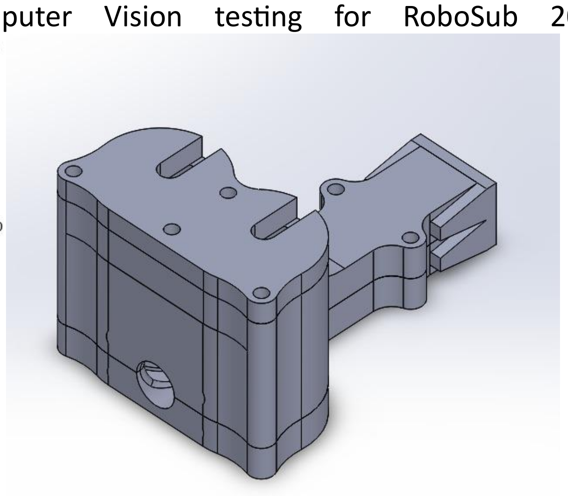


Figure 3: Computer Vision testing for RoboSub 2022





Figures 4-6: Torpedo System models

APPENDIX B AUV COMPONENT LIST

Component	Vendor	Model / Type	Specs	Custom / Purchased	Cost	Year of Purchase
ASV Hull Form / Platform	BlueRobotics	6" sereies Cabin, 3" series Cabin		Purchased	$317, $180	2019, 2023
Waterproof Connectors	Blue Robotics, McCartnet	M10 Penetratos Subocnn Wet Pluggables	M10 Thread 6pin connector, 10 pin connector	Purchased	$828.05	2019,2020
Propulsion	Blue Robotics	T200	**Forward Thrust:** 5.1 kgf **Reverse Thrust:** 4.1kgf (16V) **Forward Thrust:** 3.55kgf **Reverse Thrust:** 3.0kgf (12V)	Purchased	$36.00	2019-2020 and 2023-2024
Power System						
Motor Controls		DYS F30A 4-in-1 ESC BLHeli_S Dshot	**Continous Current:** 30A **Input Voltage :** 2s-6s Lipo	Purchased	$45.00	2022, 2024
CPU	NVDIA	Jetson TX2	**GPU:** 256-core NVDIA Pascal **CPU:** Dual 64-bit NVDIA Denver 2 and quad-core Arm Cortex-A57 processors **Memory:** 8GB LPDDR4 **Storage:** 32GB eMMC **Connectivity:** 802.11ac WLAN and Bluetooth	Purchased	$479.00	2020

Teleoperation						
Compass						
Intertial Measurement Unit (IMU)	VectorNav	VN-100	Gyro In-Run Bias: 5°/hr Accelerometer Range: ±16 g Power: 200 mW Gyroscope Range: ±2,000˚/sec Extended Kalman Filter Update Rate: 400 Hz IMU Data: 800 Hz Pitch/Roll: 0.5° Accel In-Run Bias < 0.04 mg	Purchased	$1,075	2020
Doppler Velocity Logger (DVL)	Teledyne	Wayfinder DVL	**Max Altitude:** 60m **Min Altitude:** 0.5m **Long-Term Accuracy:** ±1.15% **Velocity Range:** 10 m/s **Ping Rate:** 16 Hz **Beam Angle:** 30® **Depth Rating:** 200m **Communications:** RS-232 **Average Power Conssumption:** 3W	Donated	$7,500.00	2020
Camera(s)	Stereo Labs	ZED 2i Camera	**Stereo Baseline:**120mm **Resolution:** 1080p @30fps **Built-in Sensors:** IMU,	Purchased	$579.00	2023

			barometer & magnetometer **IP:** IP66			
Hydrophones						
Algorithms	Stereo Labs, Open CV, ROS, Docker, Google	ZED SDK, Pyrunner, Docker containers , Google notebooks				
Vision	Stereo Labs	ZED 2i Camera	**Stereo Baseline:**120mm **Resolution:** 1080p @30fps **Built-in Sensors:** IMU, barometer & magnetometer **IP:** IP66	Purchased	$579.00	2023
Localization and Mapping						
Autonomy	ROS, Mision Planner, Pyrunner	ROS, Pyrunner				
Open-Source Software	ROS Docker, ZED SDK, Ubuntu					