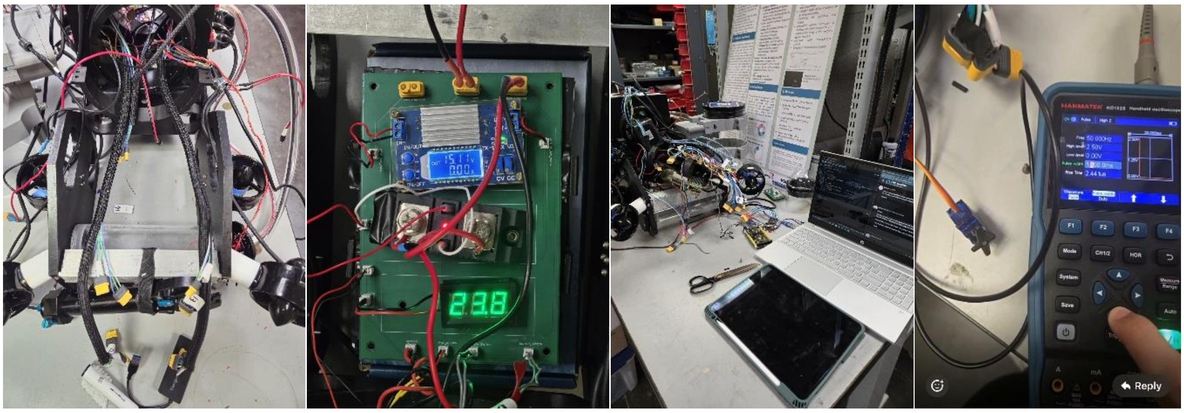
RUMarino End of Year Report 2025

Pre RoboSub:

Electrical

- Kept working on optimizing cable harness on Hydrus, this involved doing repairs, maintenance, and taking approaches for cable management.
- The electrical components on the sub were put on a scale to determine the total weight of the devices used on Hydrus.
  - Battery → 5.6 lb
  - PCB and PCB mount →0.30 lb
  - Stack with Jetson, Keystudio, ESC, and accessories → 2 lb
- Run test of each of the motors and servos using the Keystudio microcontroller on Hydrus power
- Motor identification and pin matching – each motor was identified using a label maker to its corresponding cable harness and pin connection on the Keystudio to facilitate assembly and disassembly.
- Installation of a brand-new camera cable and monitor



Management

- Continue developing fundraising initiatives to strengthen and maintain a contingency budget.
- Prepare and finalize the proposals for the RoboSub trip in California.
- Develop a new website on a different platform to integrate new tools and simplify future editing and maintenance.
- Maintain consistent communication between divisions to ensure alignment of goals, coordination of tasks, and efficient progress across all team operations.
- Organize multiple raffles to support team fundraising efforts while increasing community outreach and engagement.
- Record and edit the team video for the RoboSub competition.

Mechanical

- Ensure vacuum seals are stable during testing.
- Ensure the structural integrity of the submarine.
- Manufacture spare parts for the AUV in case of a break
- Take inventory of every item necessary to assemble and disassemble the AUV during the competition period.
- Train team members to assemble and disassemble the AUV quickly in preparation for tight competition deadlines
- Modify AUV design to account for larger cable camera penetrations.

Software

- Developed a FastAPI-based API for video streaming and real-time detection results, along with a React frontend for web-based visualization.
- Implemented Docker and Gazebo fixes, including Mesa flag support, updated drivers (kisak-mesa PPA), and rendering improvements to resolve GPU compatibility and high memory usage issues.
- Configured camera settings and conducted DVL simulation testing using ROS publishers and listeners to validate sensor functionality in simulation environments.
- Enhanced Arduino integration by refining thruster logic, separating ROS-to-PWM control, handling thruster orientation in firmware, and improving documentation and hardware dependencies.
- Strengthened mission planning and testing by creating testers for gate, tagging, and slalom missions, implementing scoring systems, and improving detection logic and reliability.
- Improved Hydrus system integration by stabilizing Arduino communication, removing broken packages, adding direct connection scripts, and introducing tmux-based multi-process monitoring tools.
- Updated control logic and thruster behavior, including consistent PWM ranges, natural movement adjustments, and loop-independent thruster execution for more stable operation.
- Expanded RViz integration to visualize detections, markers, and CPU-based perception outputs for debugging and system monitoring.
- Automated testing through GitHub Actions, including execution of unit tests, mission testers, and refactored test suites within the Docker workflow.
- Refactored testing infrastructure by adding unit tests, debouncing features, benchmarking tools for ROS nodes, and performance monitoring for time and memory usage.
- Introduced “Hocker” and hydrus-cli tools to simplify Docker interaction, streamline development workflows, and improve container-based testing and deployment.
- Added comprehensive documentation across autonomy, Docker, and embedded systems folders, including development guidelines and repository usage instructions.

During RoboSub:

Electrical

- Manage electrical systems of the AUV
- Ensure correct assembly of the AUV before and during the competition
- Ensure battery safety protocols were followed at all times
- Conduct and monitor testing of the AUV.

Management

- Document and share our team’s experiences with the community.
- Maintain consistent internal coordination to support planning, outreach initiatives, and overall team operations.
- Conduct outreach to companies and fellow teams to exchange knowledge and learn from their experiences.
- Initiate the process of the recruitment strategy for the semester.
- Engge in team bonding activities to strengthen collaboration, morale, and overall team cohesion.

Mechanical

- Manage mechanical systems of the AUV.
- Ensure correct assembly and disassembly of the AUV before, during, and after the competition.
- Ensure vaccum seal remained intact during operation of the AUV.
- Conduct and monitor testing of the AUV.

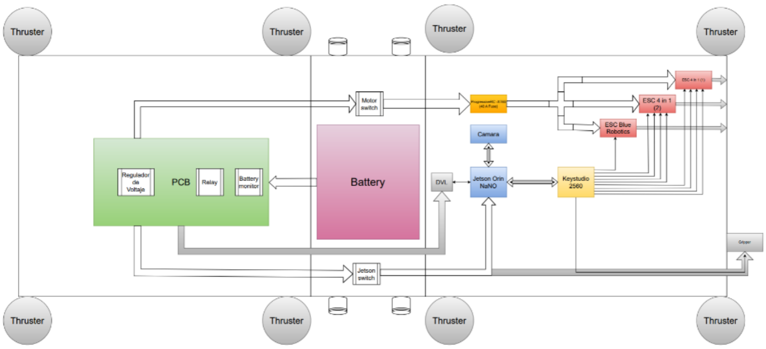
Software

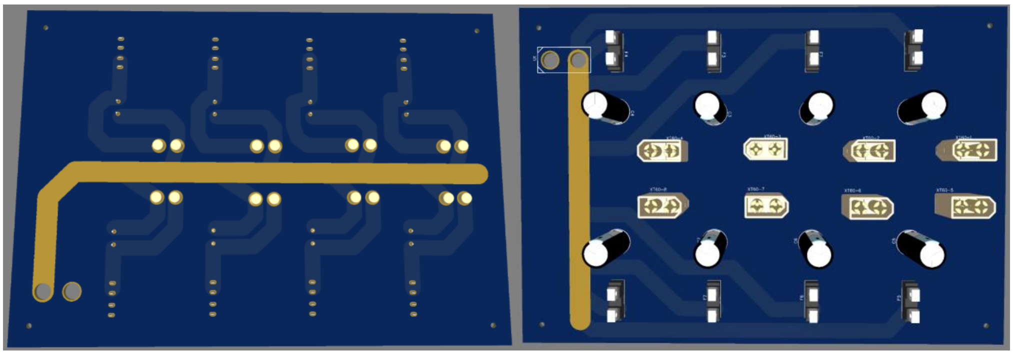
- Conduct and monitor testing of the AUV.
- Optimize autonomy between tests of the AUV.

Post RoboSub:

Electrical

- Improving Hydrus PCB based on Robosub feedback to improve PCB reliability.
- Capacitors were added at ESC terminals on Hydrus to avoid voltage spikes and ensure smooth power delivery.
- Developed an updated block diagram for Hydrus AUV.
- The prototype design of PDB for ESC was developed, taking into consideration what we learned from the previous designs.
- Assure a parallel battery system is used for future AUV designs
- Create a list of new electrical components for the new AUV.
- Research on Fathom Tether for communication with the new AUV.





Management

- Present documented evidence of expenses incurred during the competition period.
- Maintain communication with fellow competitors to foster continued learning and create growth opportunities for student members.

- Apply the judges’ feedback to improve the structure and organization of the website.
- Completed full website deployment and domain transfer to ensure the new website is accessible online to the public.
- Recognize members who participated in the competition on social media, highlighting their contributions, dedication, and achievements to acknowledge their efforts and strengthen team morale.
- Continue recruitment and outreach efforts to attract new members, expand community engagement, and support sustained team growth.
- Highlight the goals achieved during the trip, recognizing our successes while reflecting on areas for improvement.
- Update the team logo and establish official brand colors and fonts to ensure consistent and cohesive branding across all materials.

Mechanical

Post Robosub competition, the mechanical structures team spearhead the design of a next generation AUV ensuring the following core design principles are always followed:

- Modularity
- Maneuverability
- Cost effectiveness

To ensure these principles are followed, team members were encouraged to come up with ideas for designs to present to the team. Once ideas were selected to move forward, area members were separated into four different designs. As time went on, each design was optimized to follow the 3 core principles using prototype 3D printed mockups and FEA analysis. Team members selected the designs that they felt best followed by the core design principles to move forward during each selection phase, ensuring all division concerns were addressed.

Currently the final design has been selected and has been separated into 4 areas to optimize:

- Final Structure design
- Underside carriage 3D design
- Main Hull Design
- Outer frame design

Software / Autonomy

This year, the team achieved a number of important milestones. We exceeded much of our original plan and obtained results that we are excited to share.

Our team focused on Autonomous Underwater Vehicles (AUVs). Much of our work consists of understanding how to make autonomous vehicles by researching and prototyping solutions. We know that the field of autonomous vehicles is a field with significant room for growth, both in underwater robotics and across other domains. We dedicate our efforts to building solutions for visual perception, control systems and simulation of underwaters vehicles. Our end goal is to construct a solution that is fully programmable for an autonomous AUV.

RUMarino has a long history as a team, but until recently, we had not developed a complete autonomy solution for competition. Most previous efforts focused on the mechanical design, integration, and testing of the vehicle. Over the years, the software team implemented intermediate control solutions and supported technical maintenance, including limited pseudo-autonomous behaviors. Many members of the software team have worked not only on building functionality, but also on understanding the design principles behind autonomous vehicles. Based on this experience, we decided to design our own autonomy stack instead of relying on general purpose solutions that are not well adapted to our specific problem.

Our design strategy is to first build the simplest autonomous system capable of completing the Robosub prequalification mission. This mission consists of navigating through a gate, circling a buoy, and returning through the gate.

One of our core principles is simplicity: systems should be easy to test, modify, and extend. This philosophy required significant refactoring and reengineering of our previous software.

From this process, we identified the core technical problems we needed to solve. As a result, we defined six major projects at the start of the semester. I am proud to say that we have made substantial progress in each of them.

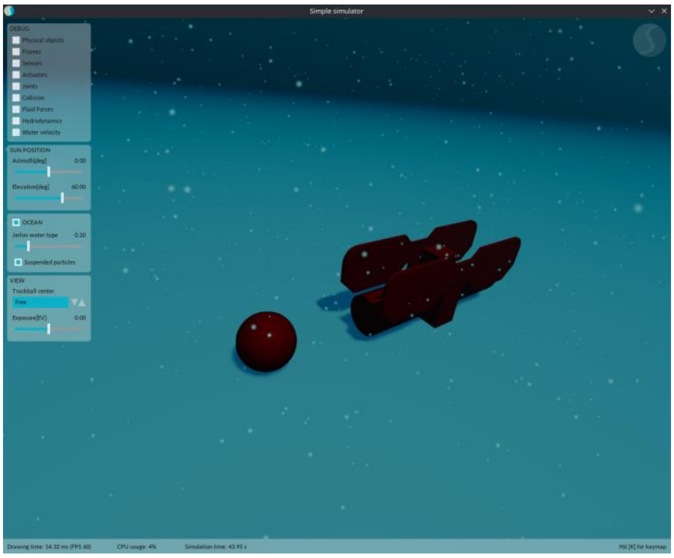
- Simulator
- Data Engine
- Visual Odometry
- Perception
- Controllers
- Mission Planner

Simulator:

One of our main challenges was the lack of testability and slow iteration cycles in our software. To address this, we adopted a simulation-first development approach. After evaluating multiple underwater simulation environments, we selected **Stonefish**, an underwater simulator written in C++ using Bullet and OpenGL.

The simplicity and performance of Stonefish aligned well with our design philosophy and allowed us to develop software more quickly and in a more structured manner.

During the first month, our primary effort was integrating our existing AUV, **Hydrus**, into the simulator.

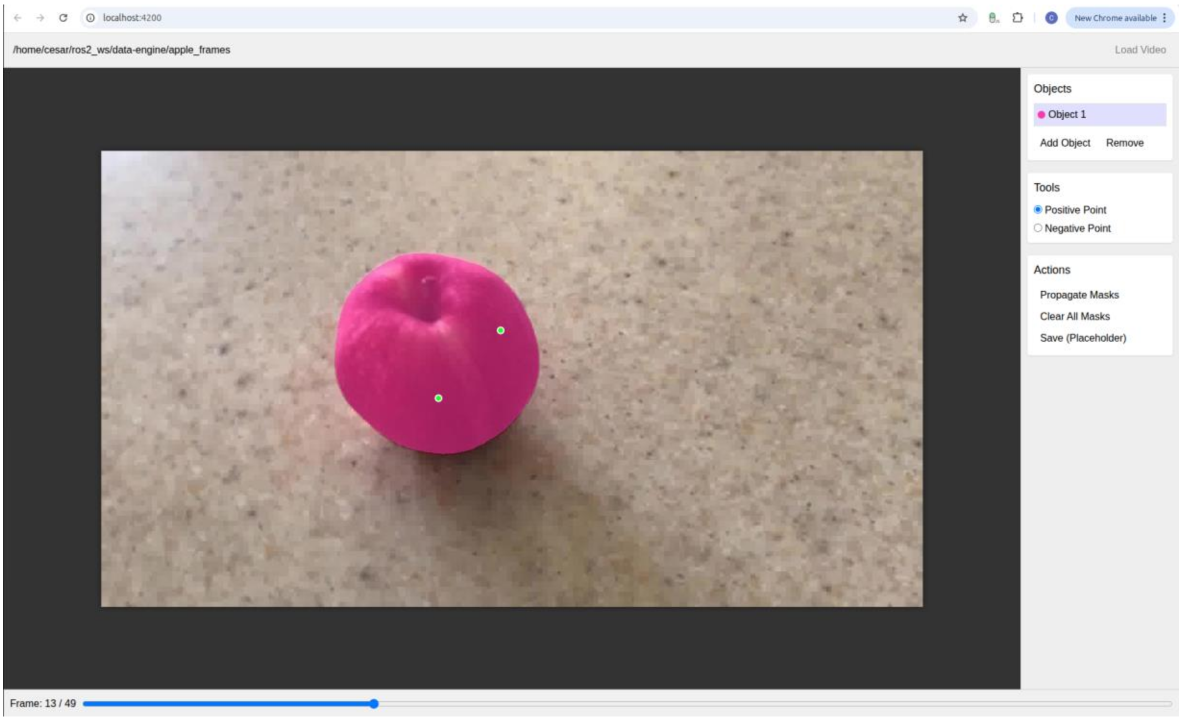


The image below shows early progress in the simulation. We adjusted geometries, materials, and force configurations to make the simulated vehicle closely resemble its real-world behavior.

Data Engine:

Following our design principles, we recognized that rapid iteration required a more efficient way to generate annotated data in a semi-automatic way. To achieve this, we leveraged recent advances in image and video segmentation, including **SAM2**, a deep learning model developed by Meta.

We built a web application with a backend system designed to generate YOLO-style annotations, that is the standard for Object Classification in robotics environments. This system is powered by SAM2 and allows us to annotate competition objects with significantly less time and human effort than traditional manual labeling.



/home/cesar/ros2_ws/data-engine/apple_frames

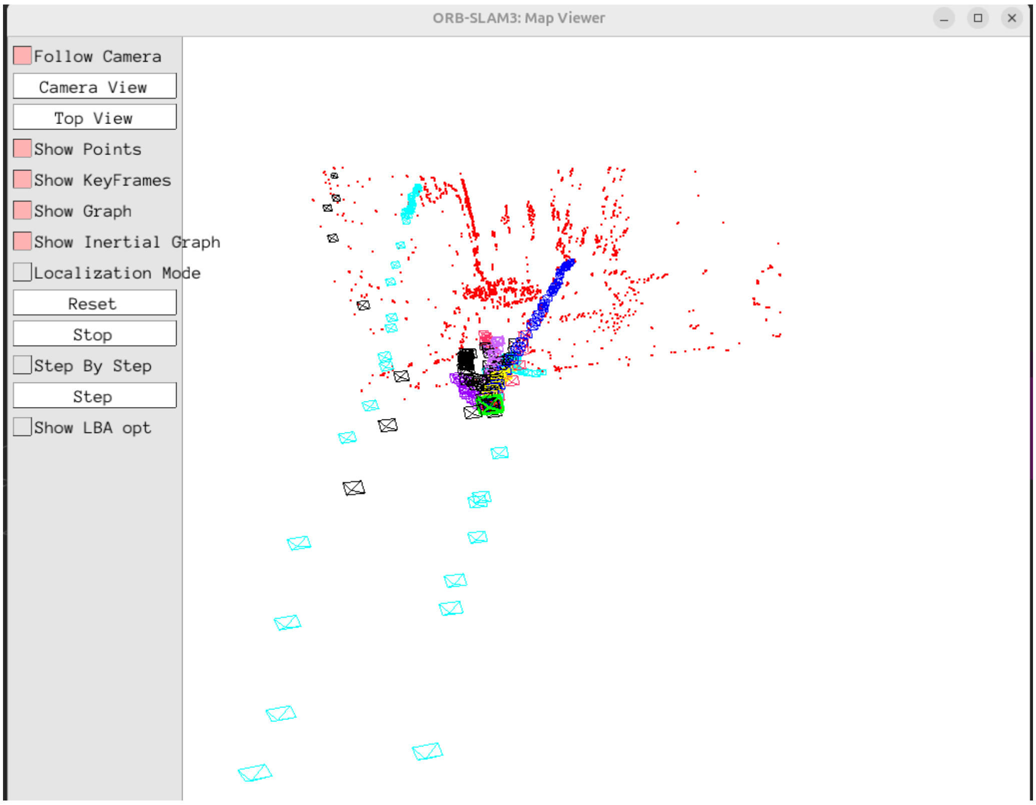
Frame: 13 / 49

Visual Odometry:

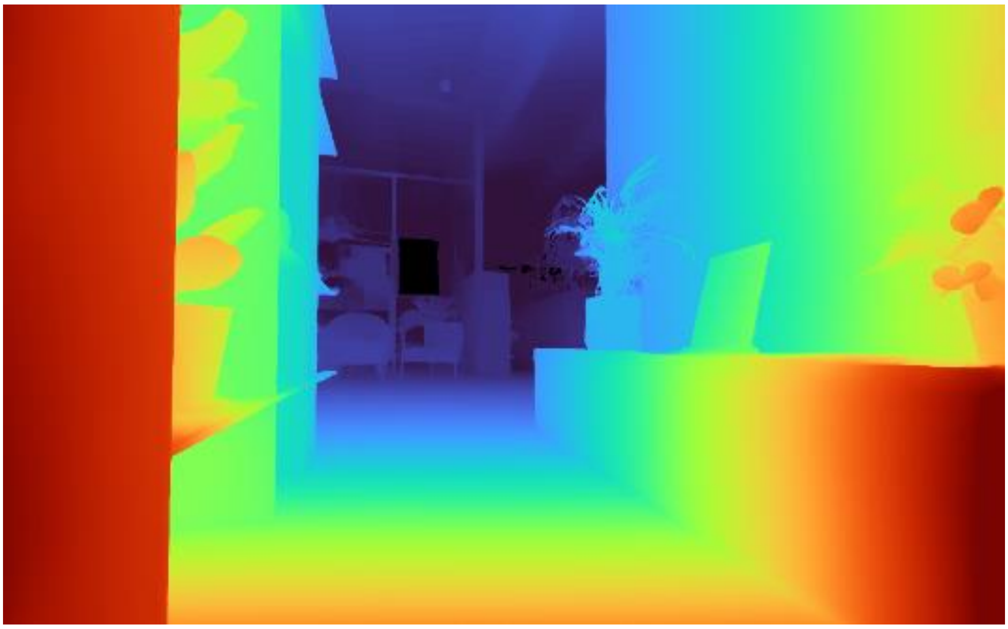
Visual odometry is one of the most exciting areas of development within RUMarino. Traditionally, odometry for AUVs relies on a combination of sensors such as cameras, IMUs, DVLs, and sonar. Our goal is to create a more efficient and cost-effective solution by relying primarily on cameras.

By focusing on vision-based approaches, we aim to reduce the dependency on expensive sensors such as DVLs and sonar. This could reduce the overall cost of future AUV designs by up to 50%. Our approach uses deep learning methods to estimate position and orientation in world coordinates using visual data, similar to how GPS provides localization, but relying solely on vision much like human perception.

In our experiments we have concluded that the ZED SDK solution outperforms by far the Open-Source alternative, but we truly believe that if we are well tuned and with a better understanding of the system, we can make it work.



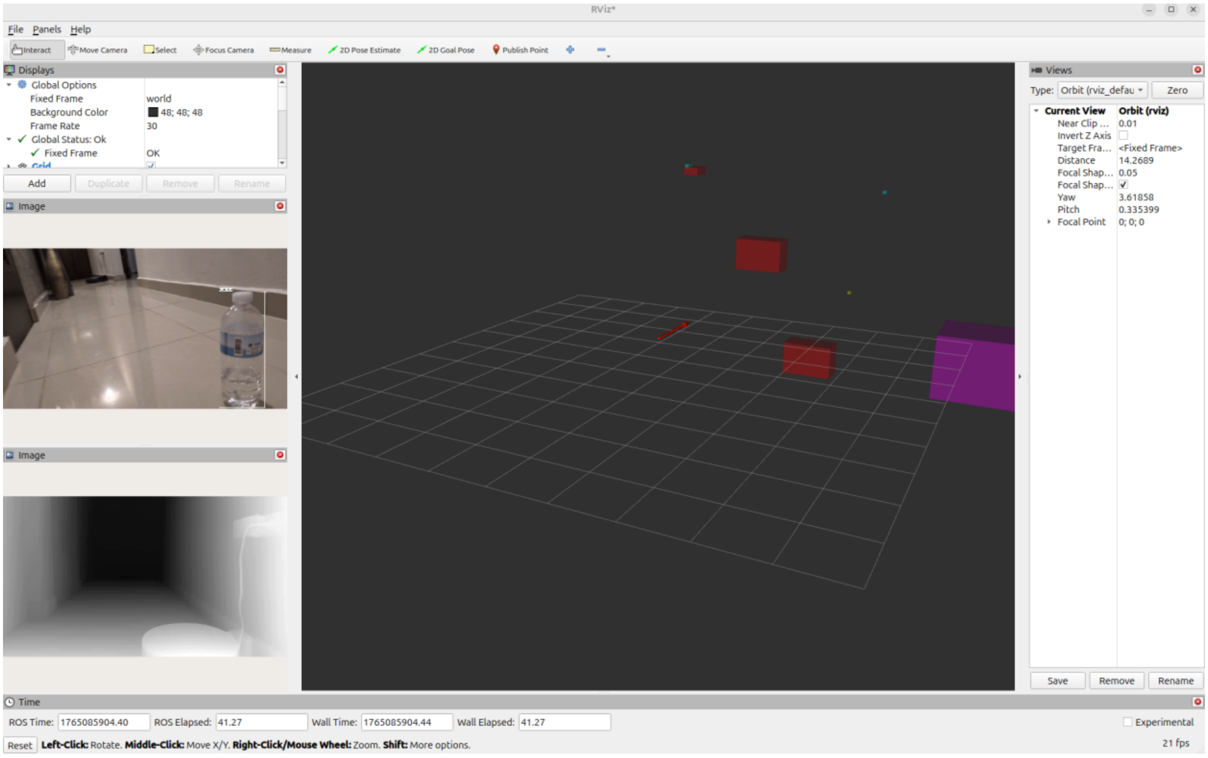
☑ Follow Camera
☑ Show Points
☑ Show KeyFrames
☑ Show Graph
☑ Show Inertial Graph
☐ Localization Mode
☐ Step By Step
☐ Show LBA opt



Perception:

In autonomous robotics, high-definition (HD) maps are a core component for enabling data-driven decision-making. This year, we implemented our own HD mapping system tailored specifically to the Robosub competition. For that, we develop a mapping system that detects objects in a world coordinate space where each object occupies a 3d bounding box, with position, orientation and a tracking class and id.

Our approach relies on machine learning models such as YOLO (Ultralytics YOLOv8) for object detection and transformer based models fine-tuned for depth prediction (DepthAnythingV2). Using calibrated pinhole camera models, we convert 2D detections into 3D information. Object tracking is performed using frequency-based updates and Kalman filtering.



Global Options
Fixed Frame: world
Background Color: 48; 48; 48
Frame Rate: 30
Global Status: Ok
Fixed Frame: OK
Grid: ☑

Type: Orbit (rviz_defau
Current View: Orbit (rviz)
Near Clip ...: 0.01
Invert Z Axis: ☐
Target Fra...: <Fixed Frame>
Distance: 14.2689
Focal Shap...: 0.05
Focal Shap...: ☑
Yaw: 3.61858
Pitch: 0.335399
Focal Point: 0; 0; 0

ROS Time: 1765085904.40
ROS Elapsed: 41.27
Wall Time: 1765085904.44
Wall Elapsed: 41.27

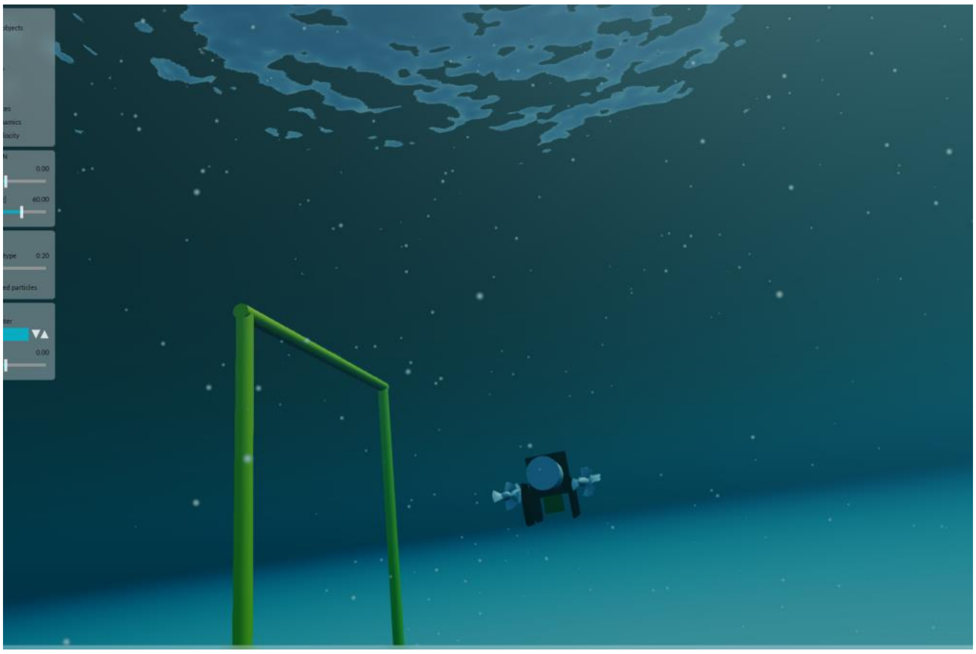
Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:** Zoom. **Shift:** More options.

21 fps

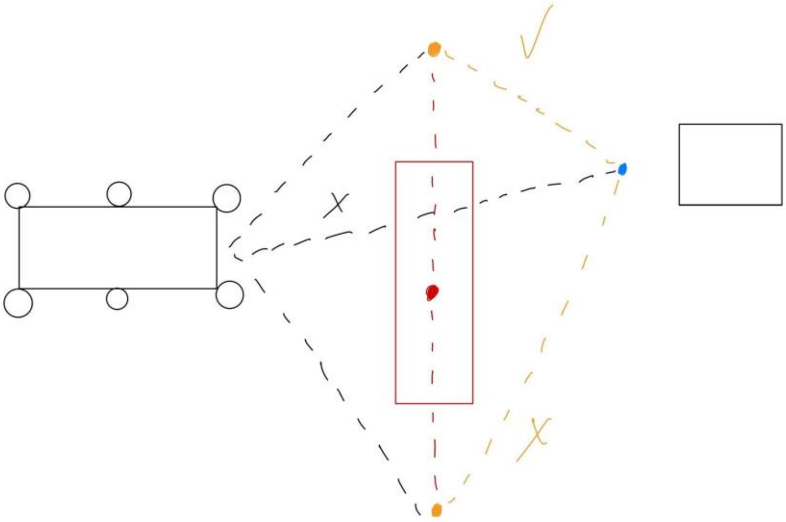
Controllers:

Controlling a 6 degree of freedom submarine is not an easy task. There are an infinite number of ways you control an AUV and it could be tempting to try to implement the hardest control algorithms, the reality is that tuning controllers is an art because it always depends on the number of variables you're taking into consideration. in practice when we were talking about AUV controllers, we talked about controlling the position, rotation velocity and acceleration of the robot at each point in time.

Our solution, although simple in terms of controllers' theory, requires creating our own mathematical model that explains the forward, yaw, and depth movement of our submarine. We simplify the 6 degrees of freedom problem into one of 4 degress by limiting the roll and pitch rotation movement. We used PID feedback loops with error correction and tuned them in the simulator to achieve competent accuracy.



In addition to low level control, we implemented a path-planning module to avoid collisions while navigating toward target points. Our approach evaluates candidate waypoints by projecting points from the center of detected object bounding boxes toward their surfaces, adding a safety margin that accounts for the submarine's size. The waypoint with the shortest distance to the target is selected. Although illustrated in 2D for simplicity, the actual implementation operates in 3D space. During testing, this method proved to be reliable and computationally efficient in simulated environments.



Mission Planner:

The mission planner is the highest-level control module of our AUV. In robotics, missions are often implemented using behavior trees or state machines. While many existing libraries provide abstractions for these approaches, we found them too restrictive for the level of control and customization we required.

As a result, we designed and implemented our own mission planner executor and scheduler. This allowed us to structure mission logic in a way that better matches our vehicle architecture and the specific behaviors required for competition.

We want to express our gratitude for reading this report and for being part of the process of this project. In the case of further questions, please visit our website https://rumarino.org/ where you can find our contact information.