Cyclone RoboSub

at UC DAVIS

Technical Design Report
2024-2025

2025 Technical Design Report

Leadership: Peter Webster, Jason Daniel Pieck, Anjani Gadkari, Nidhi Vadulas, Dilraj Gill, Luna Park, Jacob Marinas, Nico Holasek, Allen Cooke, Mohammad Ayub Hanif Saleh, Zenyn Ethridge, Luci Masselink, David Kou, Aditi Anand, Aubri Lee

Advisors: Shima Nazari PhD, Abhigyan Majumdar

***Abstract*—Cyclone RoboSub is an interdisciplinary student design team from the University of California, Davis, which has dedicated the past two years to designing, manufacturing, and programming an autonomous underwater vehicle (AUV) to compete in the international RoboSub Competition. The vehicle is capable of traversing the competition course and completing a series of tasks using eight thrusters for propulsion and an array of on-board sensors for navigation. Thoroughly tested feedback control and decision-making algorithms guide the vehicle through the course while a watertight enclosure protects sensitive electrical systems. Students were guided in applying methodical approaches to both competition and design strategy that prioritized consistent performance as a first year team and maximized our impact on student learning and success. Beyond the competition, Cyclone RoboSub is collaborating with UC Davis faculty by outfitting our vehicle for upcoming field deployments and environmental data collection.**

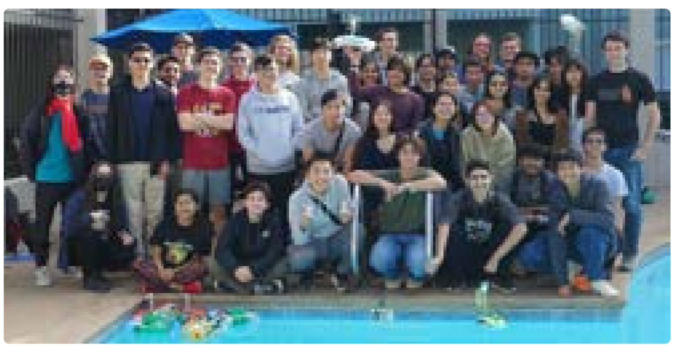

Fig. 1. 2024–2025 Team Photo

I. Competition Strategy

A. 2023-2024 Competition Year

Our first conversations regarding competition strategy began with the team's founding in 2023. As a new team, we understood the importance of setting achievable goals, so we set out to cultivate a solid understanding of the competition challenges to inform our team's expected abilities.

Our first team meeting involved breaking down the 2023 Team Handbook [1] and assessing strategy. We cataloged the maximum available points associated with each task, along with the number of teams able to successfully attempt each task as seen in Section V.A. We measured success rate as a team's ability to achieve the maximum points possible within a given task. Furthermore, we gave extra weight to tasks that succinctly demonstrated intended vehicle functionality. Doing so allowed us to identify trends and pinpoint tasks that yielded the most points while falling within most teams' predicted abilities.

During our analysis, we noted that the majority of teams' points were earned by completing navigation-based tasks such as surfacing in the *Octagon* and passing through the *Gate*. When compared with precise manipulation tasks, we found that successful attempts were less common. Based on this research, we prioritized navigation-based tasks while deprioritizing manipulation-based.

Originally, Cyclone RoboSub planned on competing in the 2024 RoboSub Competition. However, after lengthy deliberation, we chose to postpone competing until 2025 with the intent to use the extra time to build a solid foundation for our vehicle and team. Despite this, we sent representatives to the 2024 RoboSub competition to ask questions and witness different strategies firsthand. During their time at the venue, our representatives spoke with other teams and took detailed notes on the pitfalls and effective strategies teams encountered. Most teams advised against relying on vision systems as a primary means of navigation and instead recommended the use of a Doppler Velocity Log (DVL). Testing early and often was a common theme along with the use of fail-safes spread across all systems.

This new intel, combined with our previous year's assessment, culminated in our current competition strategy for the 2025 RoboSub Competition.

B. 2025 Competition Strategy

For the 2025 RoboSub Competition, we chose to prioritize the *Gate*, *Octagon*, and *Bin* tasks with a plan to add functionality for the *Return Home* and *Coinflip* tasks with time permitting. Manatee was designed to be small and maneuverable, allowing it to accumulate style points by rolling as it passed through the gate. The vehicle primarily relies on its DVL from Nortek [2], and Inertial Measurement Unit (IMU) from InertialSense [3], to navigate the pool.

To facilitate less complex vision-based tasks, Manatee is equipped with two cameras, one forward-facing and one downward-facing. The forward-facing camera recognizes which side of the *Gate* the vehicle passes through at the start of the run to inform our dropper location at the *Bin* task. The downward-facing camera allows Manatee to accurately position itself above the correct side of the *Bin* before releasing both droppers in succession.

Beyond the dropper, our design focus centered around keeping the vehicle small and nimble enabling it to efficiently complete navigation-based tasks such as the *Gate* and *Octagon*. This research regarding competition strategy formed the basis of our design strategy.

II. Design Strategy

A. Guiding Design Philosophies

At the core of our organization is a cycle of research, learning, doing, and teaching. This philosophy informs our system's design. Our mechanical and electrical systems are built with modularity, simplicity, and rugged functionality in mind. As a first-year team, time, budget, and experience constraints require us to focus on making a few key systems work well, meaning we combine careful theoretical research with consistent, hands-on testing. High fidelity CAD models made using Onshape [4] were used to plan the 3D layout and inform assembly procedures. We use 3D printing for rapid iteration, favor off-the-shelf components when possible to avoid unnecessary development costs, and design with practicality and durability in mind.

B. Hull

a) *Chassis:* Manatee's chassis is a ¼-inch aluminum base plate machined with a grid pattern of mounting holes. The hole-grid layout enables modularity and an iterative platform allowing us to easily mount, adjust, and swap out components.

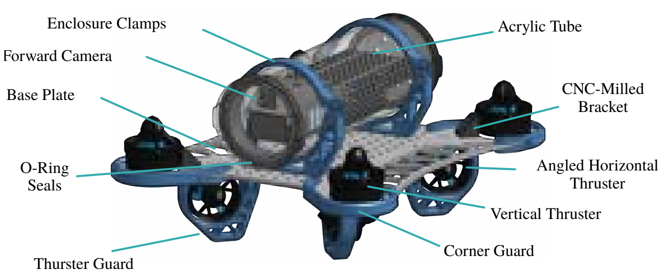

Fig. 2. Structural CAD of Robot (designed in Onshape)

As seen in Fig. 2, the propulsion system follows a standard symmetrical layout [5], with two thrusters in each corner: one angled horizontally beneath the chassis and the other mounted vertically in line with the base plate. The angled horizontal thrusters allow Manatee to propel itself forward, sideways, or control yaw independently from roll and pitch. Meanwhile, the vertical thrusters allow independent control of the pitch and roll of the vehicle.

The need to improve the structural integrity of Manatee was made apparent during pool testing, where one of the thrusters cracked due to a collision. Adding thruster guards for the angled horizontal thrusters was done to prevent further damage to both the frame and the thrusters. To improve ease of handling for Manatee, we also added handles that made transportation safe and easier for team members.

Design Upgrades: One of our most meaningful lessons came from an early fault mounting system for our main waterproof tube. The original design relied on a two-part clamp that required careful alignment and over-tightening of bolts. This led to cracking the 3D-printed clamp, proving unreliable. Eventually, this concern became an issue when the clamp snapped in half one day at our workbench.

To correct this, we embedded a threaded aluminum insert into the 3D-printed mount and allowing the bolt to lock cleanly with a sliding mechanism, we created a far more secure and user-friendly solution. Now, the print significantly easier to handle with no loose parts alongside additional rigidity as one component complementing a physical preventative to overtightening the bolt. It was a small change that represented a larger shift toward engineering that is thoughtful, resilient, and practical.

b) *Waterproof Enclosures:* Our robot includes two primary waterproof enclosures: a main cylindrical tube that houses most of the electronics, and a smaller waterproof box mounted beneath the chassis to contain the downward-facing camera.

The main tube is mounted using the hinge-and-clasp system previously described and is sealed using a combination of penetrators and O-rings on the end cap. The secondary enclosure, located underneath the chassis, was designed to house the downward-facing camera. This enclosure is an off-the-shelf waterproof box modified with a 3D-printed internal support structure to match the geometry of the box, holding both the downwards facing camera and IMU.

However, through testing and research, we encountered a critical limitation: the IMU required a USB-C connection, and routing that cable through a waterproof penetrator would either require splicing or re-crimping the connector — both of which posed a risk to signal quality and long-term performance. Given the importance of the IMU to our control system and position estimation, we made the decision to relocate it into the main tube, where it could connect directly to the processor without modification or potential signal degradation.

In order to validate successful water-proofing, strict testing procedures were undertaken to check and preserve the air-tight seals. Main tube penetrators and O-rings are regularly inspected. The team also follows the guideline of vacuum pressure testing the enclosure before every water test, detailed in Section V.B.

C. Manipulation

a) *Dropper:* We designed a servo-actuated mechanism to individually release two small markers for the Bin task during competition runs. Manatee centers itself over the bin using the downward-facing camera. Once the vehicle is positioned over the correct image, markers are released using a servo-controlled camshaft that interfaces with the markers' fins. Each component in this system can be quickly swapped out allowing for rapid, modular experimentation with different cam profiles, perfecting the release timing and repeatability.

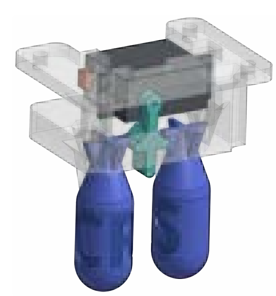

Fig. 3. Dropper Assembly

b) *Torpedo:* While not yet implemented on Manatee, development on a torpedo system is currently underway. The system uses elastic bands to store energy for individually launching two torpedoes. These bands are loaded and cocked by topside members, and are released by a servo linked to the vehicle's control system. The main roadblock towards implementation is the high shock load generated by the rapid release of the torpedoes. This caused the FDM 3D-printed parts to quickly fatigue and fail. To address this, the team is currently experimenting with stronger materials such as solid plastics and metals.

D. Electrical System

a) *Internal Electronics Mounting:* The main design goal for the electronics mounting structure is maximizing the efficiency of space usage in the waterproof enclosures. When our original design strayed from this idea, we designed an upgrade based on what we learned. Our initial design featured a U-shaped layout which cupped the battery. A practical starting point, less frequently adjusted components could be placed within the U while more active development areas would remain outside for easy access. However, we quickly ran into several limitations: the battery was not adequately secured. Maintenance access also proved challenging, regardless of where components were mounted, and as our onboard systems grew in complexity, management became chaotic – wires and components haphazardly zip-tied where they fit. The board no longer reflected our principles of clean, robust, and serviceable design. To address these issues, we launched a collaborative redesign process. Subteams explored a wide range of concepts: X-shaped boards, stacked layers, and even hot-swappable modular prints tailored for specific systems. After rounds of prototyping and discussion, we arrived at a T-shaped layout – an evolution of the U that preserved visibility and accessibility while significantly improving surface area and wire organization. The decision to cluster all high-power components into one section of the board was key, allowing for easier isolation and testing of the power subsystem while freeing up valuable real estate for the rest of our electronics. This redesign significantly improved maintainability, visibility, and adherence to our design philosophies.

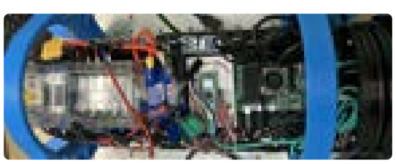

Fig. 4. Internal Electrical Systems

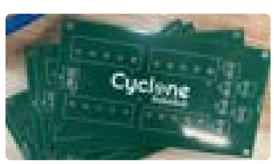

Fig. 5. 5V PDB

b) *Power Distribution:* The details of the power distribution system are described in detail in Section V.G. The key challenges faced by the electrical system were thermal management and space constraints.

Thermal performance issues stem from the Electronic Speed Controllers (ESCs) and processors overheating during extended operation. For better cooling we added a "heat bank" coupling the ESCs to a copper plate. This plate absorbs and stores heat until the AUV can be recovered post-run.

Our power system has grown, space constraints are a pressing issue. Looking ahead, we are exploring improved cooling solutions, including strategies like water cooling and heat pipes.

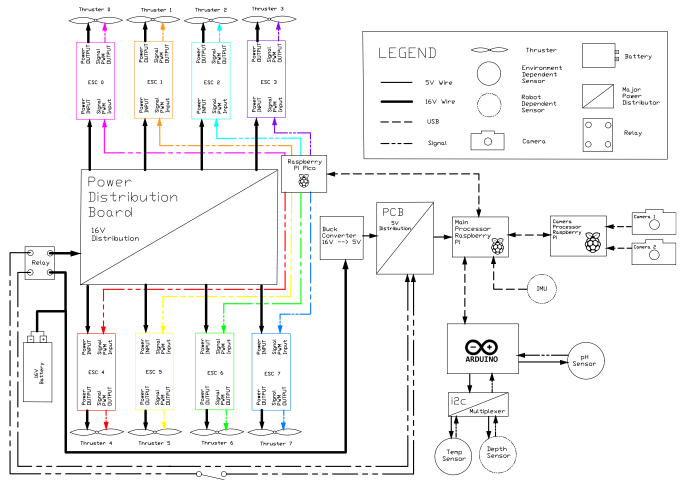

Fig. 6. Electrical Diagram

E. Control System

The design strategy for the control system is motivated by the selected tasks. These tasks require Manatee to perform dead-reckoning navigation between pre-programmed waypoints, "hover" at one location, and make fine adjustments with aid of the downward facing camera. The control system for Manatee is broken into two sections: executive control and low-level control. The role of the executive controller is to identify the current objective of the vehicle and select the necessary low-level controller mode to accomplish that objective, while the low-level controller performs the thruster and actuator control appropriate for the current task.

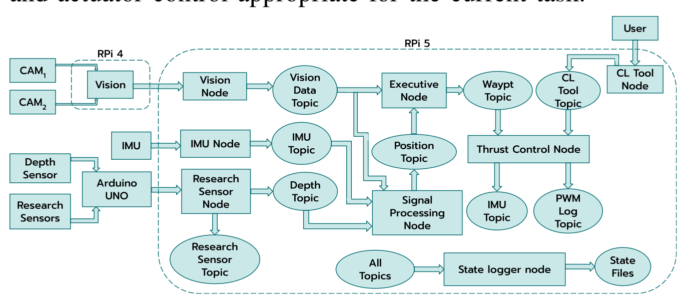

Fig. 7. Hardware-Process Flowchart

a) *Matlab Implementation:* Wherever possible, control systems were designed and tested in MATLAB and Simulink. Mathematical models for the vehicle and control system were selected from [6]. Once tests, we created generated code files for embedded hardware using MATLAB's C-code generation capabilities. This allowed us to take advantage of MATLAB's many built-in mathematical models necessary for predicting vehicle dynamics and control.

Auto-PID tuning [7] for the controller was conducted on a digital twin of the robot which simulated the robots 6-degree-of-freedom dynamics, hydrostatic and hydrodynamic interactions, thrusters, and sensor-noise models. The fidelity of the digital twin was improved empirically using data gathered during pool-testing to perform least squares parameter estimation [8] of drag and inertial coefficients.

The separation of the controller into two subsystems along with use of MATLAB's advanced toolboxes allowed for rapid prototyping of the control system components in parallel with the development of the executive and embedded C-code, allowing different engineers on the team to play to their own strengths while maintaining the capability to combine the disparate pieces into an effective whole.

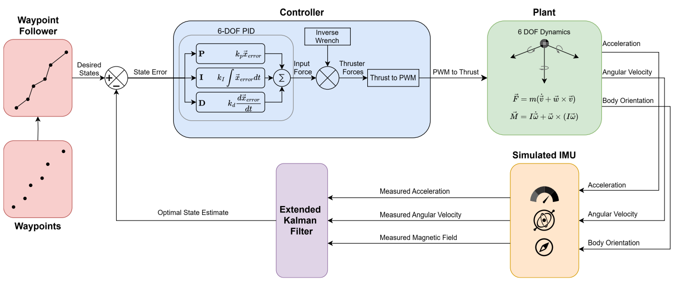

Fig. 8. Controller Flow Chart

b) *Software Integration:* High level integration relies on ROS2 [9] for inter-process communication and modularity. ROS2 provides an abstraction for multi-threading, simplifies the development process, and enables cross-language development.

Most of Manatee's sensors and peripherals have dedicated ROS nodes which interface with the rest of the system. This decouples the code base from the low level implementation, enabling system modularity and improved debugging workflows. These edge nodes are a mix of manufacturer provided SDKs and custom written software.

Manatee interfaces with its DVL and IMU using ROS nodes provided by Nortek and InertialSense respectively. Most other devices as custom software. For example, Manatee's thrusters are controlled using the PWM peripheral of a Raspberry Pi Pico. The Pico uses USB to communicate with a node running on a Raspberry Pi 5. The control system can send PWM commands to this node by publishing to a topic, rather than directly interfacing with the Pico over USB.

While most of Manatee is implemented in C++, Python is used in areas where performance is less critical. During testing, Manatee can be controlled with a command-line tool written in Python. This sped up development through the use of a higher-level language.

c) *Vision System:* Manatee's two cameras both feed into an Yolo V8 [10] vision model running on a dedicated Raspberry Pi. The vehicle only uses its vision systems for two main tasks, recognizing which side of the gate it passes through and centering itself over the dropper. This cut the need for any sort of complex depth perception, but left the option to expand the complexity and use of our cameras and vision model in the future. In the coming year, we plan on implementing position estimation using the video streams in order to confirm and correct our location over the course of a competition run.

In future competition years we hope to improve our vision model to aid in navigation with optical flow and state estimation, as well as more complicated manipulation tasks.

F. *Conclusions*

The design strategy of the Cyclone RoboSub team is a research and testing driven approach to create a system that is simple, reliable, and adaptable. Lessons learned from attending previous competitions are distilled into Manatee's requirements that optimize for competition score under the constraints of team resources and time.

Intrinsic to this goal is the construction of a strong foundation to build upon in subsequent competition years – a foundation not only of hardware and software but also of community. We build friendships, skills, and a love for science and engineering that we hope our team members will carry with them throughout their journey. Our competition vehicle is designed to evolve alongside the team.

III. Testing Strategy

A. *Component Level Testing*

Each component of our electrical system was tested before being implemented. After each new addition, we performed a full test of all systems to determine if any new issues had been introduced. This included methodically and redundantly checking all electrical connections and data feeds under stressful conditions.

B. *System Level Testing*

Our team employs a test-early and often strategy by performing bi-weekly pool-tests with minimal delays. Consistent weekly goals guided the team to prioritize the next viable product of the work, checking regularly that individual members' and subteams' work was compatible with the larger design. Routine pool testing often identified issues at the interface between subsystems that were difficult to identify in subsystem level testing. In addition to testing the robot, preparation for the system tests also revealed the need to design well-documented standard operating procedures (SOPs). These SOPs cover a range of critical topics including software configuration, battery and electrical safety and continuity, hull leak testing, and data logging.

C. *Standard Operating Procedure*

Our SOPs started small with a few important steps to take before in-pool testing, as we gained experience, we expanded this list into a decision-based case tree, allowing us to quickly respond when something goes wrong as seen in Section V.B.

One example of where this problem-solving structure was used was troubleshooting inconsistent thruster behavior ultimately tracing it back to a hidden solder bridge and misconfigured software. This structured, methodical approach saved time and minimized guesswork.

Beyond the procedure documents themselves, the real strength of our testing lies in the way our team executes it: with clear communication, calm problem-solving, and shared knowledge. These habits – built through repetition and reflection – are what allow Manatee to operate safely and effectively in uncertain conditions.

Fig. 9. AUV “Manatee” Before a Pool Test

Acknowledgements

We would like to extent a huge thank you to our sponsors, including Citris and the Banatao Institute, Nortek, InfoSec Decoded, and RIX Industries. In addition, we would like to thank our parents, and the Engineering Student Design Center for the tools and workspace.

In addition, we would like to thank our advisors Dr. Shima Nazari and Abhigyan Majumdar for their continued support and insight during the past year.

We would also like to thank all of our highly dedicated team members, including but not limited to Aishwarya Tawade, Albert Zheng, Alice Roy, Alex Zamora, Andrew Li, Brandon Santamaria, Danny Kwong, Hannah Kench, Hunter Ward, Josh Cook, Kaitlyn Hahn, Kekoa Olive, Kory Haydon, Lemar Argand, Naia Dalal, Nathaniel Fregoso, Ruby Stanton, Ryan Mathur, Sawyer Morgan, Sherry Lei, Tanishq Dwivedi, and William Barber.

V. Appendix

A. Task Breakdown

Name	# of Teams	Point Value
Gate: Maintain Fixed Heading	24	150
Gate: Pass Through	24	100
Gate: Style	18	800
Start Dialing: any, Correct side	13	600
Gate: Flip Coin	13	300
Chevrons: Surface in area	6	1000
Goa'uld (Torpedo): Any	4	500
Location: Any bin/Correct bin	3	1000
Time Bonus	2	100
Inter-Vehicle Com	2	1000
Random Pinger: Task 2	2	1500
Random Pinger: Task1	2	500
Goa'uld (Torpedo): Correct	2	500
Location: Remove Lid	1	500
Chevrons: Surface w/object	0	400
Chevrons: Drop Object	0	200
Chevrons: Object on table	0	500
Chevrons: Correct Sequence	0	300

B. Pool Test Procedure

The team has developed a procedure to check the submarine for preparedness, before submerging the robot in the water.

Check First

Make sure pressure holds:

- 15mmHg for 10 minutes

If not:

- Visually inspect o-rings for proper seal.
- Ensure tube is completely closed
- Inspect pressurizer to make sure it has a good seal
- Tug cables running into Wetlink penetrators

WiggleTest

Give all components a gentle wiggle to see if they move or come loose (be EXTRA careful with the wires)

Does something move?

- Alert Hull lead if it's serious (use best judgement)
- Bolt? Tighten it back into place.
- Wetlink? Take it apart and put it back together by reseating the wires.
- Wire? Plug it back in or tighten the connector.

Make sure the battery is at operational voltage (16.8V)

- Use the battery charger (or multimeter) and check that the voltage is at 16.8V

Not at correct voltage?

- Plug in the battery until it reaches the desired voltage

Electrical

- Battery plugged in
- Components light up/make sounds
  - ESCs
  - Raspberry Pi
  - Sensors
  - Cameras
- Thruster wires are out of the way of the thrusters
- Ethernet cable is organized and will not get caught with movement.
- Kill switch is off

Mechanical

- Tube clasps are secured down
- Pressure plug is all the way in
- Ensure bumpers (and all other 3D printed guards) are secured

Clean up.

Done with testing? Time to clean up!

- Battery is plugged in and on "Discharge" to get battery to storage voltage (14.8V)
- All wires are plugged in and out of the way of the tube closing
- Tube is closed
- Plug is in
- Ensure all tools are put away in the toolbox
  - Make sure all sets of tools are complete

C. *MATLAB Simulink Implementation*

To determine the robot's current objective, the executive controller is pre-programmed with information about the location of tasks and obstacles within the pool alongside the desired order to accomplish these tasks. Once autonomous operation begins, the executive controller leverages sensor data to determine if the success criteria for the task has been met by the low-level controller and updates the control strategy accordingly. The executive is also responsible for monitoring telemetry data including time and battery capacity to adjust task priority and ensure safe operation.

The low-level controller inputs measurements from the camera system, DVL, and IMU, as well as context from the executive controller to generate thruster commands. The DVL plus the IMU allows for much more stable navigation than an IMU alone due to accumulated integral error. State estimation is performed using the sensors to estimate the vehicle's attitude and position in the world frame with position measured relative to the starting point. Sensor fusion of the DVL and IMU (which includes a magnetometer) is used for roll and pitch stability as well as heading control. The low-level controller can be configured by the executive to hold a constant position, navigate to a pre-programmed waypoint, perform a barrel-roll for style points, or center an object in the downward facing camera's field of view. In the future, an improved vision system could aid navigation by allowing for vision based pose estimation and optical flow to account for sensor drift.

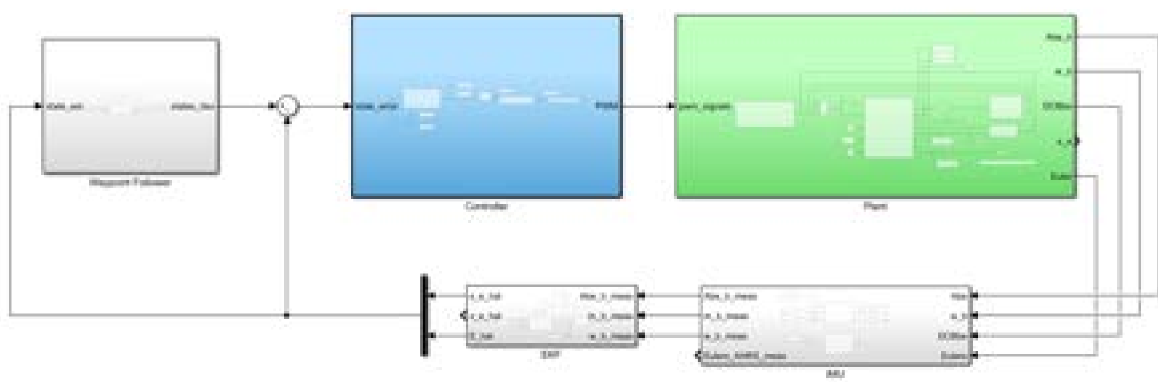

Fig. 10. MATLAB Simulink Model

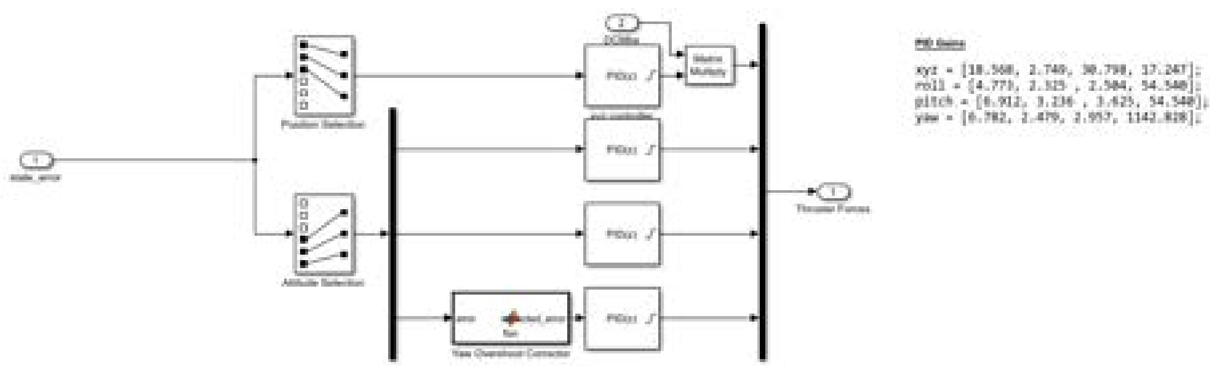

Fig. 11. The 6-Axis degree of freedom PID Controller. We implemented a Yaw Overshoot corrector to stop the yaw error increasing spontaneously in situations where Manatee needed to turn around 180 degrees.

D. Simulink Parameter Estimation

To have a higher fidelity simulation of Manatee, we utilized MATLAB's parameter estimation capabilities to refine our drag and inertial coefficients. Thruster commands are used as the model input while sensor data gathered during the pool tests were used as the model outputs. The mapping between the true thruster inputs and measured sensor data is a function of the various drag parameters, which are estimated from the data. Implementing these empirically informed coefficients into our MATLAB simulation then allowed us to iterate our PID coefficients with the auto-PID tuner.

Model Setup

- Create a Simulink model of the system whose parameters you must estimate.
    - Configure it as an open loop system with the input and output blocks in the locations of the model containing unknown parameters corresponding with the measurement data.
    - In this example, the input is torques and the output is the Euler angles.

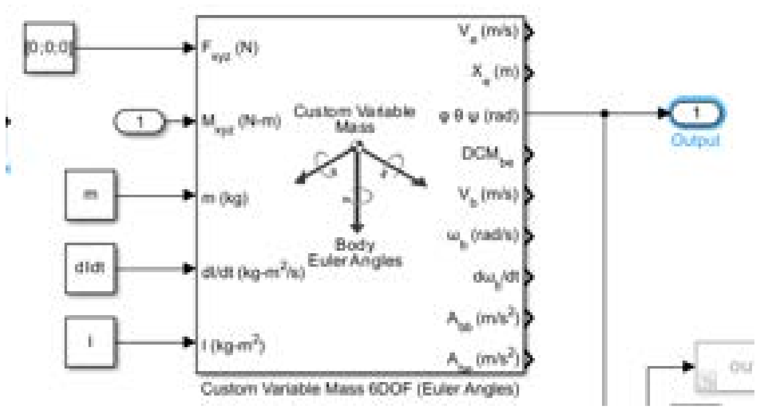

Fig. 12.

- load the data files output with the input and output into the workspace

Parameter Estimation

- Open the parameter estimation app
- New experiment

Fig. 13.

- Make sure the input and output are selected that correspond to the data to be used to estimate the parameter

Fig. 14.

- Enter the input and output data using the variable name in the workspace.

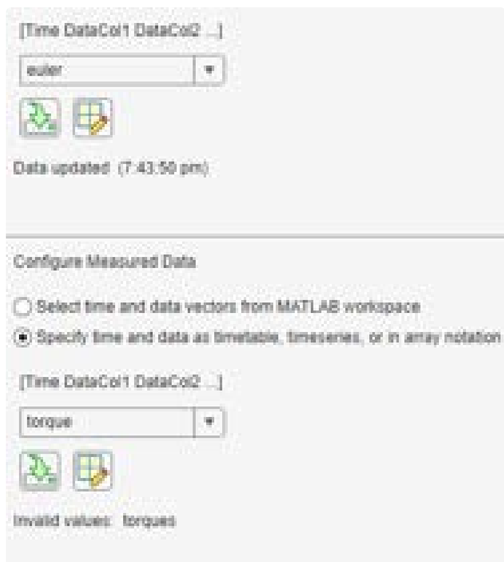

Fig. 15.

- Hit plot, then close the experiment window. You should see a plot of the input and output data.

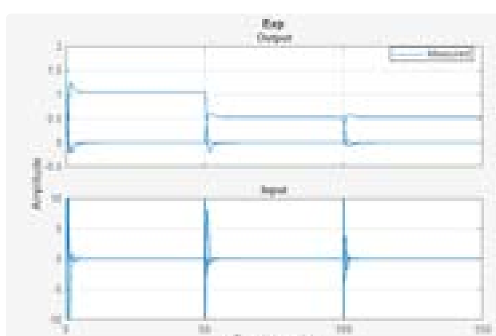

Fig. 16.

- Return to the parameter estimation tab and click “select parameters”.
- Click select parameters again, then click the continuous or discrete check box next to the parameter you wish to estimate. In this that is the inertia matrix I.

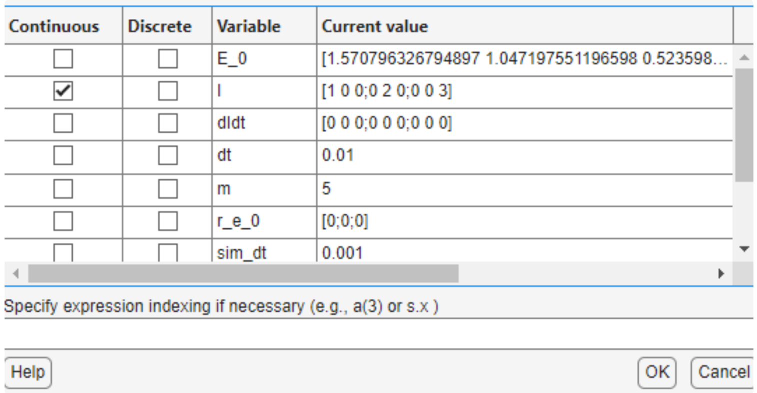

Fig. 17.

- Click “OK”. Back on the parameter estimation page, set an initial guess (in my case the identity matrix) and make sure estimate is checked.

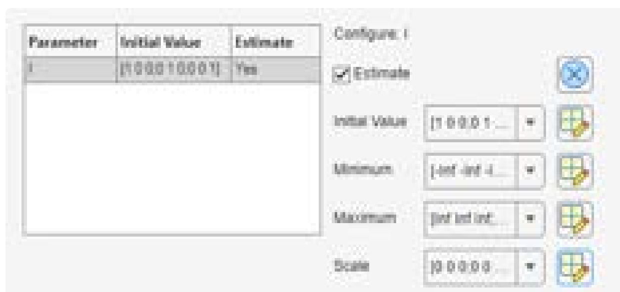

Fig. 18.

- Hit date model, close, then estimate
- The app will vary the parameter I until the input and output in simulation match the data. For high frequency data sets this could take a very long time

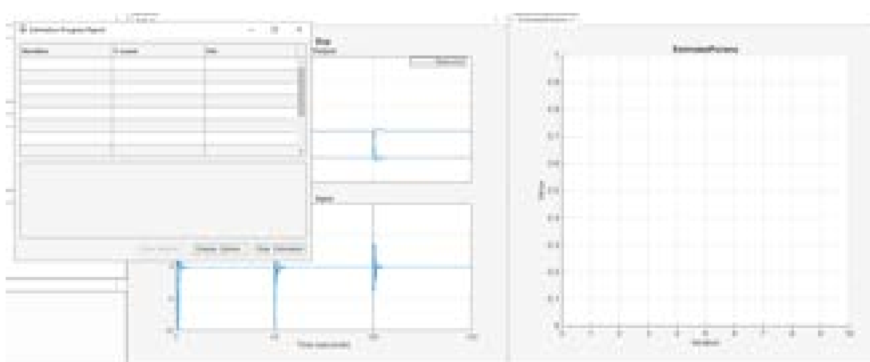

Fig. 19.

E. Vision Data Collection

To train our vision model for competition, we conducted numerous data collection sessions in order to gather data inputs for training and testing our YOLO vision model. We made an effort to emulate real game elements and game conditions as close as possible. Robust datasets were taken in order to generalize the vision model which could adapt to varied competition environments. Detailed below are the measures we took to replicate real competition conditions.

- Accurate game elements - All game elements are constructed to match their real competition counterparts. This includes using accurate colors, materials, and geometries based on previous year's courses and available information on current tasks.
- Multiple angles - We recorded video footage from a wide range of angles, especially on critical game elements including the *Gate* and *Paths*. We ensured that we collected ample footage in close range of the image targets on the gates to model the vehicle passing beneath them.
- Multiple light levels - We use two Blue Robotics Low Light cameras which help to filter out blue light underwater. We were unsure of the exact weather conditions at competition. To account for this we recorded footage on both sunny and overcast conditions.

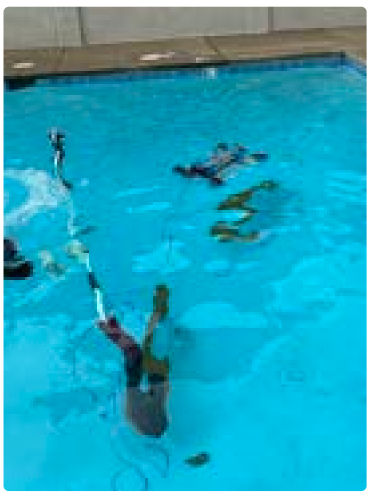

Fig. 20. AUV Testing

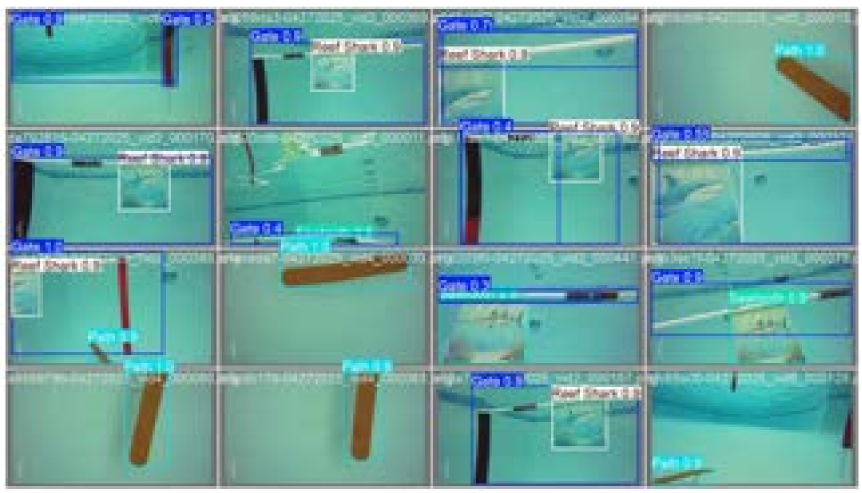

Fig. 21. Vision Model Testing Dataset Sample

Fig. 22. View of AUV Front Camera During Testing

Fig. 23. Sample Dropper Bin for Vision Training

F. Software Unit Testing

To ensure reliability, code is assumed to be nonfunctional until testing is complete. We follow the protocol that all code on the main branch must be tested and fully functional. Members create working branches for in-progress and untested code, which is then pulled to main once testing is complete. All unit tests must pass before code can be pulled to main.

We use the GTest framework for our C++ unit tests, and pytest for our Python unit tests. Every function should be rigorously tested to ensure that it behaves correctly under all input, including edge cases. Testing can be run with a flag for real hardware, or for simulated hardware. Simulated hardware tests confirm that the correct commands are sent to the Raspberry Pi Pico that runs the thrusters, while tests on real hardware confirm that the Pico received the correct messages and didn't return any error messages, through echoing received commands back over the serial connection that it received the messages. Any code changes that impact the transfer of messages to the Pico should be verified on real hardware before being pulled to main.

The final step in validating correct behavior is regular pool testing, where we ensure that the code actually moves the robot in the ways that we expect it to. We create extensive logs for later examination, troubleshooting, and verification. These pool tests are performed about once every one to two weeks, on an as-needed basis.

G. Electrical System

Our power distribution system is split into two distinct pathways: high power (16V) and low power (5V). This separation helps isolate sensitive components from high-current loads, improving system safety, organization, and troubleshooting. For a full overview, refer to the electrical schematic included above.

Power begins at the 16V battery, which connects to both the High Power Distribution Board (HPDB) and a relay-based kill switch (discussed in a later section). The HPDB supplies current directly to the Electronic Speed Controllers (ESCs) and thrusters, while also feeding into a buck converter that steps the voltage down to 5V. From there, power is passed to our Low Power Distribution Board (LPDB) — a custom-designed PCB that distributes power to our processors.

Our control system — what we call the "Brains"—includes the Main Raspberry Pi, a Camera Pi, Raspberry Pi Pico, and an Arduino. Each of these components handles a critical piece of the AUV's functionality:

- The Main Pi manages system-level integration and sensor fusion (e.g., IMU data),
- The Camera Pi handles real-time image processing from both onboard cameras,
- The Pico generates PWM signals to drive the thrusters,

And the Arduino governs servo control for the Manipulation subsystem and handles environmental sensor inputs like temperature, pH, and depth.

One of the key features of our system is the kill switch, which safely cuts power to the high-power lines (thrusters) while maintaining power to the Brains. This enables safer debugging and rebooting procedures without a full power-down of the system.

From the Camera Pi, there are two cameras onboard the AUV: forward-facing and downwards-facing.

H. Vehicle Transportation

Background

We fabricated a custom bicycle trailer that allows us to transport our robot, tools, Ethernet tether, and other equipment between our main workshop and the pools where we test.

Reasoning

- Walking and pulling a wagon
  - One way trip duration: 20 minutes
  - Pros:
    - Inexpensive (a decent wagon is about $80)
  - Cons:
    - Slowest
- Driving
  - One-way trip duration: 8 minutes
  - Pros:
    - Can bring the most equipment
  - Cons:
    - The nearest parking lot is 0.4 miles from our workshop
    - Parking is $17
- Bicycling with a trailer
  - One-way trip duration: 5 minutes
  - Pros:
    - Fastest
    - Davis is one of the most bike-friendly cities in the United States
    - Almost everyone on the team has a bike
  - Cons:
    - More expensive than a wagon (400 for the entire build)

Design Requirements

Must:

- Be able to securely hold the robot (22”x25”x14”)
- Be able to transport/store all of the required equipment
  - Extension cord
  - Powerstrip
  - Ethernet tether
  - Robot batteries and chargers
  - Two spare parts boxes (14”x7”x4.5”)
  - Toolbox
  - Soldering iron
  - Bike pump
  - Basic first aid kit
- Serve as a mobile workstation
- Be freestanding when not attached to a bike
- Be able to fit through an ADA compliant door
- Be able to fit in the trunk of a car

Nice to have:

- Lighting
- Umbrella holder

Design Implementation

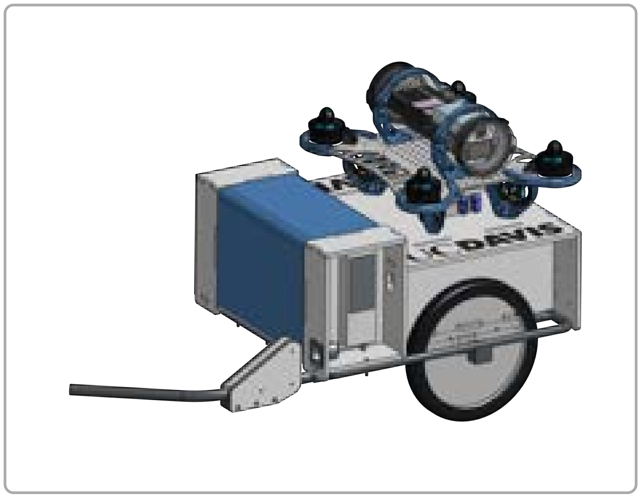

Fig. 24. Render of Cart

Fig. 25. Bare frame

The core of the trailer is a bicycle trailer that we purchased from Amazon. We removed the plastic flooring and side walls of the trailer, including the upwards protrusions that are used to attach the side walls to the frame.

The majority of the structure that we added to the trailer is laser cut ½” birch plywood. We chose birch plywood because it is relatively affordable, lightweight, easy to laser cut (150W CO_2), and sufficiently durable for this application. The plywood panels are aligned with box joints and glued and screwed together. After the plywood parts were assembled, we bolted the structure to the frame of the cart.

Fig. 26. Panel Assembly

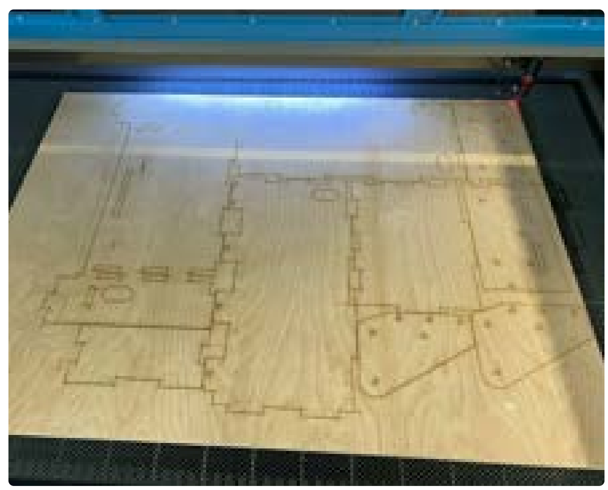

Fig. 27. Laser Cutting Panels

We then bolted the toolbox to the trailer and attached the power strip. Because we did not want the cord of the power strip to flop around while the trailer is in motion, we cut off the standard three prong plug and wired the power strip into a surface mount male receptacle.

Fig. 28. Power Supply

Fig. 29. Cart Storage

To keep the cart stable, we added a folding front kickstand and a bolt-on rear kickstand that also acts as the tailgate of the trailer. To secure the robot to the trailer for transport, we use 6” ¼-20 bolts that go up from the underside of the tabletop and through the chassis of the robot.

Completed Cart

Fig. 30. Completed Cart

Fig. 31. Cart on Display

I. *Tether Management*

Our team decided to design our own spooling system to manage our tether. We chose this solution to save costs and add flexibility in how we managed our 50 meters of heavy duty Ethernet cable.

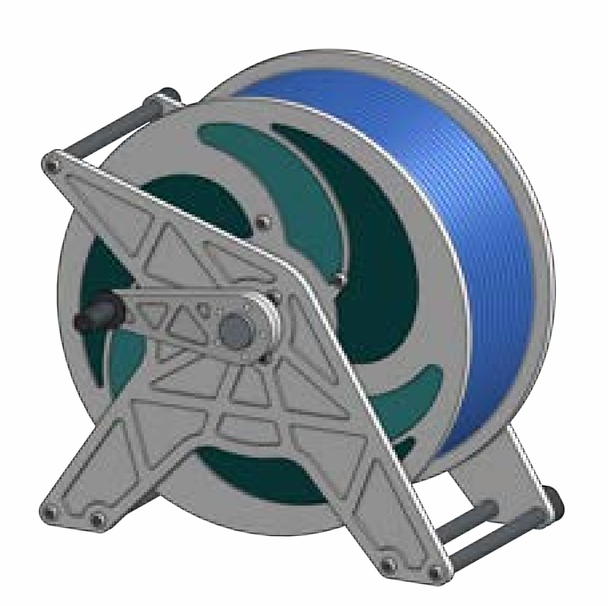

Fig. 32. Front Trimetric View of Tether Spool

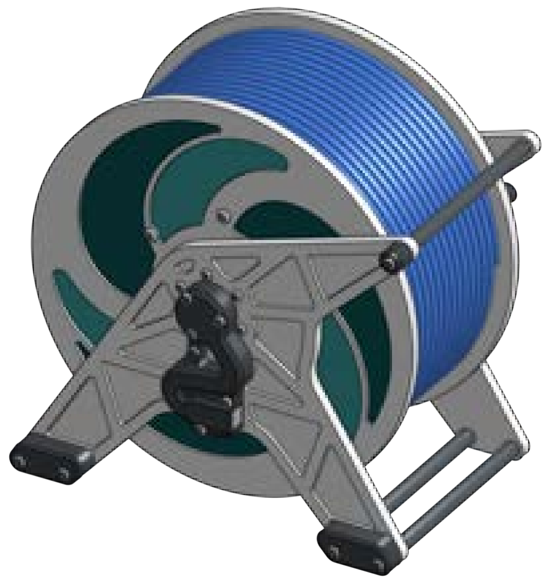

Fig. 33. Back Trimetric View of Tether Spool

Fig. 34. Side View of Tether Spool

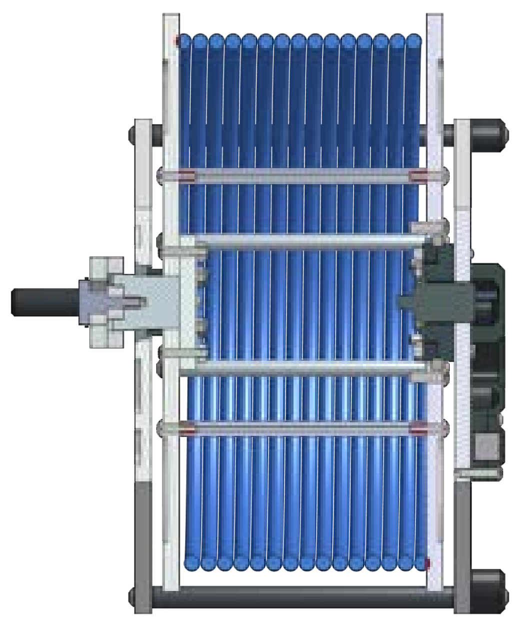

Fig. 35. Section View of Tether Spool

J. Leadership Structure

The Cyclone RoboSub is organized into two groups: Sub-Teams and Divisions. There are six subteams (Navigation, Propulsion, Hull, Manipulation, Research, Public Relations), each led by student leaders who are primarily responsible for project execution. Conversely, there are three divisions (Mechanical, Electrical, Software), each with multiple leaders who are subject matter experts and primarily responsible knowledge-sharing among the team. This structure has enabled the team to support a wide verity of student and projects.

Leadership Selection Process

- Applications open to all team members whether they are freshmen who just joined or seniors looking for hands on experience before they graduate.
- Applications are reviewed by current leadership members.
- Applicants are given 15 minute informal interviews where they are provided an overview of what they can expect as a leadership member and responsibilities are openly discussed.
- The current leadership team conducts a final round of reviews
- New leadership is announced and on boarded

K. New Member Training

Introducing new members to the team and ensuring they have a meaningful experience is one of the fundamental missions of our team. We leverage accessibility in a number of ways including the decision to omit membership dues and open leadership applications to anyone with an interest regardless of experience.

Battle Boats:

2024 saw the introduction of a new team member training activity titled battle boats! This group based, competition-style technical training served as a fun way to provide new members with an onramp into marine robotics. Details of the program are listed below. This past year we saw over 50 students participate in our training program and it set the fast paced, community oriented tone that we would maintain throughout the rest of the year!

- Premise - Team members would randomly be assigned to groups of 5 and tasked with the design and build of a small remote controlled boat for use in a capture the flag style competition. Trainings would be held to teach members the basic skills needed to build the vehicle and members could decide who would focus on which area: software, electrical, or mechanical.
- Competition - Each group would be paired with another team and face off against two opposing teams. Each side would be given a structure which held a loop of yarn which would serve as the team's flag and could be snagged by an opposing battle boat. Teams could win the game by "capturing" the opposing team's flag or by popping small balloons attached to the backs of each vehicle.
- Results - 8 different teams competed with over 50 students participating in the challenge. The competition style trainings encouraged personal investment and forced teams to think critically about how they designed their vehicles. Each battle boat was completely unique and teams were introduced to concepts of motor control, propulsion, and buoyancy.

Fig. 36. Team Members working on their battle boat

Fig. 37. Workshop for Electrical.

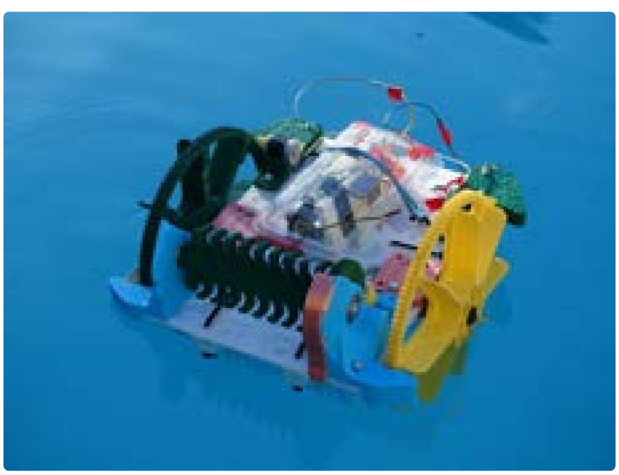

Fig. 38. Battle Boat Example

Fig. 39. Two Battle Boats in a Face-Off

L. Outreach

Over the past year, Cyclone RoboSub has engaged in a series of community outreach events on the UC Davis campus. Outreach gives us the opportunity to connect with new students and share the work our team is doing with the wider community. Events like Picnic Day, a UC Davis tradition, give our members the chance to educate the public about marine robotics and the engineering behind Manatee.

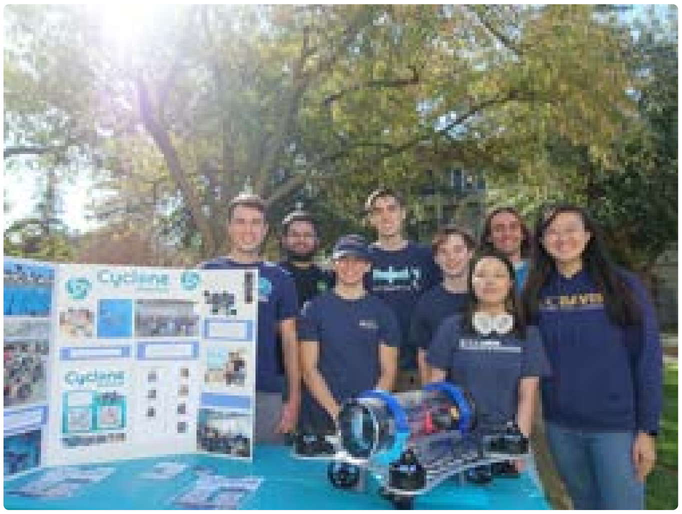

Fig. 40. Fall Engineering Club Fair

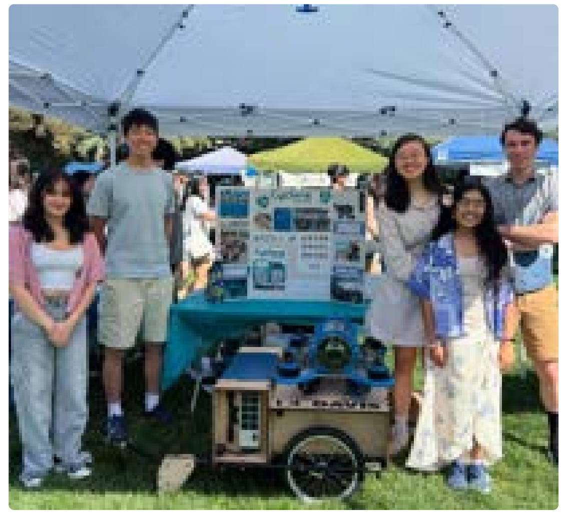

Fig. 41. UC Davis Picnic Day

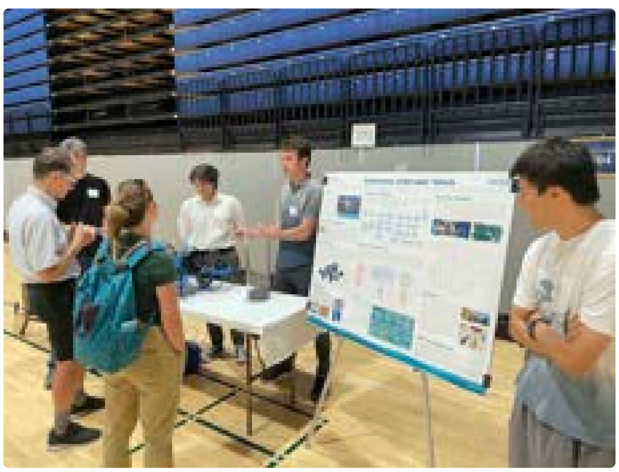

Fig. 42. Student Design Showcase

Fig. 43. Sponsor Showcase

M. Team Branding

As with everything on our team, we chose to be extremely deliberate with our branding which was developed during the team's formation in 2024. The strategies behind our branding elements are listed below:

- Team Name - Cyclone RoboSub was selected as the team name because we felt that the word "Cyclone" would be associated with rotation while staying unique from other RoboSub teams. Within our university the "RoboSub" part of "Cyclone RoboSub" quickly and clearly communicates what the team does: submarine robotics. Instead of being Cyclone AUV who's acronym is not widely recognized.
- Team Colors - We chose teal and white as it felt clean and tied in well with our Cyclone theming while also distinguishing ourselves from the traditional blue and gold color scheme employed by the University of California System. Three shades of teal were selected and are prevalent across all of our media and branding
- Team Logo - The Cyclone RoboSub logo was designed in Inkscape with simplicity at heart. We wanted a logo that was easily recognizable and conveyed the rotational nature of a cyclone. Alongside our colored logo, we also have a version in black and white along with inverted black and white for official documents.
- Wave Patterning – Across our media you can find a distinct wave pattern using three shades of teal. The regular use of this patterning creates a cohesive brand and is not overly distracting to the viewer.
- Fonts - We have an official set of font guidelines which are used for all official competition materials. Fonts used are Righteous for titles and Prompt for body text.

All branding resources are maintained on our GitHub for easy access!

https://github.com/Cyclone-Robosub/Cyclone-Logos/

Fig. 44. Cyclone Propeller Logo

Fig. 45. Cyclone RoboSub Title Card

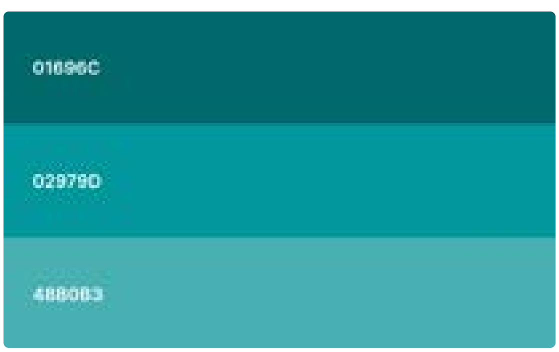

Fig. 46. Cyclone RoboSub Color Pallet

Fig. 47. Cyclone RoboSub Title Card (B&W)

N. Environmental Research

Beyond the competition, Cyclone RoboSub is contributing to environmental research efforts through field deployments and interdepartmental collaborations. Equipped with sensors to measure temperature, depth, pH, and dissolved oxygen, the vehicle can collect environmental data and is scheduled to take two transects along the UC Davis Arboretum. The team is also exploring opportunities to contribute to marine science research at the Bodega Marine Lab.

Beyond the competition, Cyclone RoboSub is contributing to environmental research efforts through field deployments and interdepartmental collaborations. Our team is unique in that we have had a subteam dedicated to research since day one. The goal of the research team is to coordinate and implement opportunities to get our team involved in marine and environmental research. AUVs are a versatile technology and our hope is to provide students within the environmental and marine sciences the chance to get involved and engaged with high end technology. This past year, our Research Subteam has worked to implement a new suite of sensors and discuss data collection opportunities with local researchers at UC Davis.

Sensors for Data Collection:

- Temperature sensor - collects live temperature readings useful for understanding water properties.
- Depth Sensor - Used in combination with other data sets during vertical transects
- Doppler Velocity Logger & IMU – used for geospatial records as a reference for other data sets
- pH Sensor - Records live water pH data
- Dissolved Oxygen Logger - Records dissolved oxygen content every minute
- Cameras - Can record imagery of aquatic life and algal populations

Planned Research Deployments

- UC Davis Arboretum Transect – The UC Davis Arboretum is a stretch of natural land within the campus which follows Putah Creek. Sections of the Arboretum were remodeled over the past year to improve wildlife habitats and wetlands. Our vehicle, Manatee, will be deployed to measure the current pH, temperature, and dissolved oxygen content of the Arboretum at two locations through vertical transects. The vehicle will be deployed at the site, manually navigate to the estimated deepest location and slowly descend while taking readings. These readings will be compared with those taken before the renovation to assess impact.
- UC Davis Tahoe Environmental Research Center - Multiple members of our team have spent time conducting research for the UC Davis Tahoe Environmental Research Center and our plan is to use it as our first serious field testing site. The clarity and low current conditions of Lake Tahoe make it an ideal location to test Manatee's abilities. Our hope is to collect imagery of algal populations and record pH and temperature data in collaboration with their researchers in the Summer of 2025
- UC Davis Bodega Marine Laboratory - This year we have actively discussed collaboration efforts between Cyclone RoboSub and the Stachowicz Marine Research Lab. We presented to their researcher on the capabilities of Manatee and they proposed cases in which they would benefit from the use of an underwater vehicle. Use cases included taking temperature samples of soil at low tides, recording imagery of fish species, and collecting light level data at various depths to evaluate seagrass photosynthesis. After further testing of Manatee, we plan to revisit these proposals and select which seem to be feasible.

Fig. 48. pH Sensor Calibration

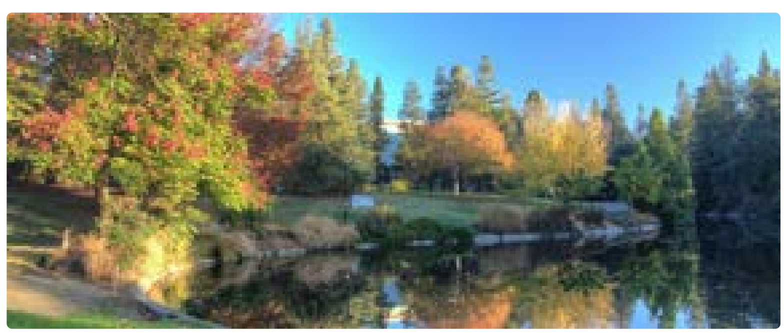

Fig. 49. UC Davis Arboretum

References

[1] [Online]. Available: https://robonation.org/app/uploads/sites/4/2023/06/2023-RoboSub_Team-Handbook_v2.0.pdf
[2] [Online]. Available: https://www.nortekgroup.com/products/nucleus1000
[3] [Online]. Available: https://docs.inertialsense.com/
[4] [Online]. Available: https://www.onshape.com/en/resource-center/
[5] [Online]. Available: https://bluerobotics.com/learn/thruster-usage-guide
[6] A. H. De Ruiter, C. Damaren, and J. R. Forbes, *Spacecraft dynamics and control: an introduction*. John Wiley & Sons, 2012.
[7] [Online]. Available: https://www.mathworks.com/help/control/ref/pidtuner-app.html
[8] [Online]. Available: https://www.mathworks.com/discovery/parameter-estimation.html
[9] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot Operating System 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm6074, 2022, doi: 10.1126/scirobotics.abm6074.
[10] [Online]. Available: https://yolov8.com/