Roboboat 2025: SEALs Technical Design Report

Liam Bray, Mitchell Hulme, Alec Jenson, Kai Jerkins, Joel Lee, Tammy Lin, Karen Nguyen, Kris Nguyen, Melanie Nguyen, Michael Nguyen, Prisha Poudyal, Bennet Rodriguez

I. ABSTRACT

This report details the development of two Autonomous Surface Vehicles (ASVs), *Blastoise* and *Squirtle*, for the RoboBoat 2026 competition. The competition strategy emphasizes autonomy and reliability, striking a balance between complexity and optimization to enhance performance. *Blastoise*, designed for precision tasks, uses omnidirectional thrusters and sensor fusion for fine maneuvers. *Squirtle*, optimized for speed, features a hydrodynamic hull design that enables high-speed performance. The systems integrate localization through LiDAR, stereo cameras, and RTK GPS for adaptive navigation. Custom PCBs streamline power distribution and sensor integration. A modular software framework built on ROS 2 coordinates perception, planning, and control subsystems.

II. COMPETITION STRATEGY

2.1.0 Strategy Overview

The competition strategy focuses on completing a diverse set of autonomous tasks reliably within the strict time constraints, achieved by breaking down each task into fundamental components, simplifying execution logic, and ensuring reliability. Tasks were selected to maximize scoring potential and enable consistent performance. The approach leverages each ASV's strengths, with *Squirtle* utilizing its speed to rapidly clear the entry gate, navigation channel, and speed challenge, while *Blastoise* utilizes its maneuverability and comprehensive sensor stack to perceive the environment, effectively performing docking and object delivery. Several alternative competition strategies were evaluated during system design. A single-ASV approach was initially considered to reduce coordination overhead; however, this significantly increased mission risk, as failure of a single subsystem would result in complete task loss. Additionally, the strict time limitations imposed by harbor alert events reduce the opportunity for a single vehicle to sufficiently observe and characterize multiple operational zones, particularly the emergency response sprint and the marina. Deploying multiple ASVs enables parallel data collection and task execution across these regions, improving situational awareness and increasing the likelihood of successful task completion within the allotted time.

The system adopts a closed-loop, map-aware competition strategy based on SLAM, providing a continuous estimate of the ASV's pose in the map frame as the course is explored. Sensor data are fused using an Extended Kalman Filter (EKF), favoring robust state estimation over simpler filter-only or single-sensor localization approaches that degrade under dynamic conditions. Rather than committing to a fixed task order or waypoint-based strategy, object detections are tracked over time using deep learning and the Hungarian Algorithm, enabling task locations to be refined and re-evaluated as new information becomes available. Adaptive state machines and decision trees dynamically select actions, enabling the ASV to pursue the most advantageous tasks at any given moment while ensuring that each task is fully attempted and minimizing overall mission time. To ensure the ASV efficiently progresses to subsequent tasks, the competition strategy was carefully crafted with sensor limitations and time constraints in mind, as outlined below.

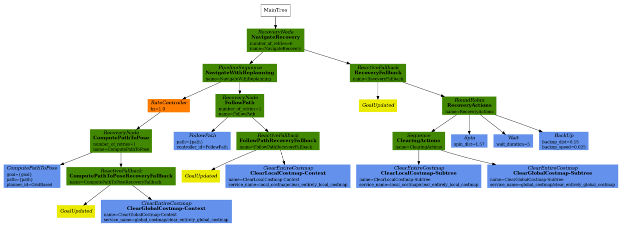


Figure 1 - ASV Control State Handler System

2.1.1 Entry/Exit Gates and Navigation Channel (Debris Clearance)

Buoy following and obstacle avoidance are achieved by identifying sequential pairs of static buoys that define the channel. Each pair generates a midpoint goal pose that incrementally guides the ASV along the channel centerline, allowing the path to be continuously updated as new buoys are detected. This approach is robust to partial observability and sensor dropouts, since navigation decisions rely only on locally confirmed structure; an alternative strategy that directly plans to individual buoys or fixes a goal far ahead in the channel is more sensitive to missed detections and can lead to oscillatory or overconstrained behavior as the perceived channel geometry changes.

Buoys and light beacons are detected using a YOLOv11-based vision model and are spatially associated with LiDAR clusters produced via DBSCAN. Comparing detections across modalities improves map accuracy by combining vision-based semantic classification with LiDAR-derived spatial consistency, reducing false positives and maintaining a stable estimate of each object's position over time.

Path planning is performed using a hybrid A* approach that respects vehicle kinematics and the local costmap. Trajectories are executed using an omnidirectional Model Predictive Controller with integral action, which enables constraint-aware prediction while eliminating steady-state tracking error from persistent disturbances such as wind and current, resulting in reliable and precise path tracking under competition conditions.

2.1.2 Speed Challenge (Emergency Response Sprint)

The system detects both the buoy pair defining the traversal gate and the light beacon to be circled, including its color. The buoy pair is used to compute a gate midpoint that defines the nominal path, while the beacon is used to determine the ASV's heading. After initial detection, the ASV leverages its omnidirectional mobility to face outward while maneuvering around the beacon, continuously collecting sensor data to refine the local map and observe nearby objects. A tuned navigation sequence enforces a fixed standoff radius, and the local costmap, which encodes buoy boundaries and static obstacles, ensures safe clearance throughout the maneuver.

2.1.3 Water and Object Delivery (Supply Drop)

The pose of each of the six vessels moored throughout the course is detected using a specialized computer vision model and tracked in the local map shared between the ASVs. After completing any previous tasks, Blastoise's state-handling system greedily selects the nearest available task and, when appropriate, navigates toward the corresponding delivery vessel using the stored positions. To activate the water gun or ball-dropping mechanism, a predefined navigation and firing sequence is executed based on the ASV's position, orientation, and nearby obstacles.

2.1.4 Docking (Navigate the Marina)

The docking task involves identifying the most desirable open dock, marked by a green indicator with the lowest number. To ensure the correct dock is selected and avoid accidental docking into the wrong location, the ASV performs a full lateral sweep of the marina while tracking the lowest observed number. Once the sweep is complete, the ASV's navigation system generates a goal pose at the center of the chosen dock and computes an entry path using costmaps built from vision and LiDAR data. LiDAR provides a precise representation of the dock's boundaries, allowing the ASV to avoid collisions and maintain safe clearance during approach.

2.1.5 Sound Signal (Harbor Alert)

Throughout each run, specific noise frequencies act as dynamic cues indicating that the ASV should override other mission objectives and navigate to an emergency response location. To isolate these frequencies, the microphone input is processed using a Fast Fourier Transform to extract frequency components and identify peaks that trigger action overrides. Upon detecting the sound signal, the ASV recalls the mapped locations of each emergency response zone and navigates to the appropriate one. After reaching the emergency response zone, any tasks not marked as complete are resumed, ensuring continuous scoring. The approach equips the ASV for immediate, precise, and low-overhead detection of emergencies, to quickly override other objectives and return to normal tasks after its response.

III. DESIGN STRATEGY

3.0.0 Design Overview

After evaluating last year's ASV, Bruce, the engineering department identified several key points that needed to be addressed in this year's design. One of the biggest concerns was flooding of the hulls, which limited the amount of time we could safely operate on the water. Alongside this, we wanted to reinforce each ASV's resilience for improved operation and improve modularity by allowing structures to be mounted directly inside the pontoons.

To address these concerns, the new design incorporates carbon fiber composites, polyurethane foam, and revised geometry to boost stability and structural robustness, in addition to support for omnidirectional thrusters.

The introduction of Rule 5.3.2 allowed the use of a second ASV. While the original ASV, *Blastoise*, remains optimized for precise, independent motion, the second ASV, *Squirtle*, was designed specifically for the speed challenge, emphasizing a high thrust-to-weight ratio and straight-line performance.

3.1.1 Hull and Frame Design

Blastoise features a double-hull design with a rounded profile, which reduces flow separation and drag compared with sharper hull designs, improving overall hydrodynamic efficiency while providing a stable platform for payloads. Two thruster mounts positioned at the base of each hull allow for a flexible omnidirectional movement configuration. To improve transportability while maintaining buoyancy and hydrodynamics, the hulls were shortened and widened by 5.5% and fitted to a custom transportation mount. Carbon fiber composite was selected as the primary hull material for its durability, high stiffness, and low mass. Aluminum crossbars and 3D-printed mounts support the superstructure and electrical box.

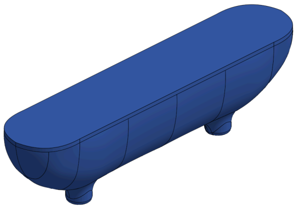


Figure 2 - Blastoise Hull Design.

Squirtle uses a trimaran hull configuration to maximize straight-line speed while providing sufficient buoyancy for the onboard electrical systems.

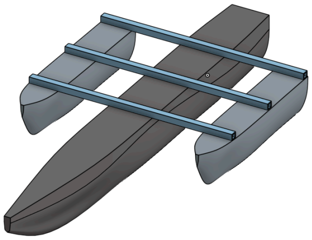


Figure 3 - Squirtle Hull Design.

Simulations were conducted to predict water flow around the hull and identify areas of high pressure under load. Pressure distribution across an early version of Blastoise is shown in Figure 4, highlighting regions that could contribute to increased drag or structural stress. These results were used to guide hull shape adjustments and optimize weight distribution, improving overall stability and maneuverability.

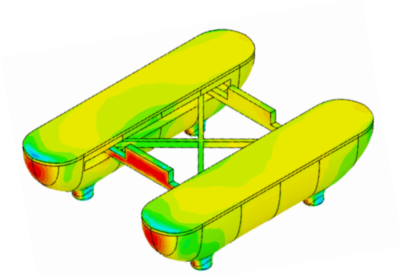


Figure 4 - Pressure Distribution, Blastoise

The carbon fiber hulls were manufactured using gel-based tooling molds. In collaboration with the University of Texas at Arlington's Formula SAE team, two independent pontoons were constructed in their composite lab. To mitigate the risk of flooding from small leaks, expandable polyurethane foam was incorporated as a core inside the hulls, serving both as structural backing and buoyancy. This design enabled components to be drilled and mounted directly into the hulls without compromising watertight integrity, simplifying installation of the electrical mounting box and supporting reliable execution of object delivery tasks. These design choices maximize durability while maintaining proper buoyancy in waves.

3.1.2 Propulsion System

Both ASVs use Blue Robotics T200 thrusters. *Blastoise* employs four thrusters, each mounted at a 45-degree angle relative to the forward axis, allowing for forward, lateral, and rotational forces.

Independent thruster control enables omnidirectional motion, including strafing, diagonal movement, and in-place rotation. This improves maneuverability, especially during docking, obstacle navigation, and precision tasks. The angled layout allows fine positional adjustments to counter drift from wind or current. See Appendix E and F for full thruster breakdowns and calculations.

3.1.3 Water Pump

Water flow is delivered by a 12-volt diaphragm pump with tubing running to the bow. This pump was selected over a custom-built design due to its straightforward implementation and ability to generate approximately twice the pressure. Alternative approaches, such as custom pumps, submersible pumps, or more complex multi-stage systems, introduced greater design complexity, higher risk of failure, and longer development time. In contrast, the diaphragm pump offers a reliable, low-complexity solution that directly addresses issues encountered during previous testing, including insufficient pressure and inconsistent flow rates.

3.1.4 Racquetball Delivery System

The racquetball delivery system originally designed for the 2025 competition was employed to target the desired stationary vessels.

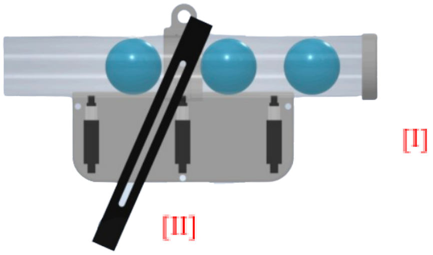


Figure 5 - Racquetball System With Servos

The mechanism shown in Figure 5 consists of a pipe [I] with three linear actuators [II] capable of storing up to three balls. The actuators retract sequentially, releasing one ball at a time while retaining the others. This design was selected over alternatives due to its simplicity, ensuring reliable and consistent operation during competition.

3.2.0 Electrical System

The updated electrical system addresses previous shortcomings, focusing on efficiency, reliability, and repairability to ensure smooth operation during testing and competition.

3.2.1 Battery Management System

Battery reliability was improved through advanced monitoring and management. A custom PCB measures the voltage of each cell and automatically balances them, preventing undervoltages that could compromise the ASV. This extends battery life and improves system efficiency. The PCBs also feature indicators for real-time diagnostics, allowing the team to identify issues before they escalate.

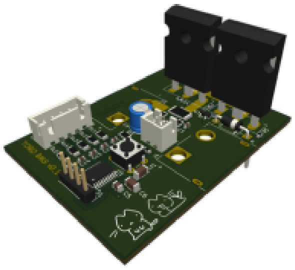


Figure 6 - Custom Battery Management PCB

3.2.2 Efficiency

To improve operational efficiency, a battery management PCB (*Figure 7*) manages the redundant power supply systems, allowing seamless switching between wall and battery power without requiring the shutdown of onboard electronics, reducing downtime and preventing data loss. Redundant power increases operational flexibility and simplifies troubleshooting and maintenance.

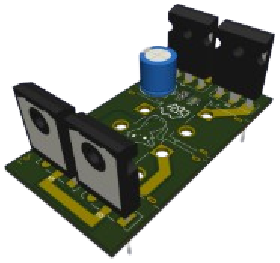


Figure 7 - Hotswapping Power System PCB

3.2.3 Repairability

A 3D-printed frame (*Figure 8)* houses all electrical components to reduce clutter, minimize weight, and improve accessibility. The compartmentalized design enables cleaner wiring, easier repairs, and modular upgrades. This structure improves organization and efficiency while increasing the overall power-to-weight ratio. The modular design of the frame allows for easy access to each component, streamlining repairs and upgrades.

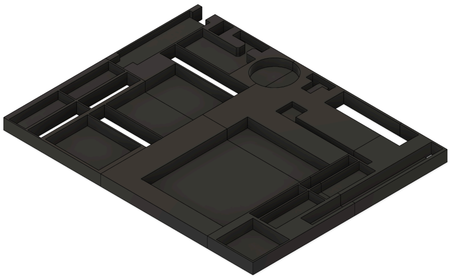


Figure 8 - Wiring Organization Frame

3.3.0 Software Infrastructure Overview

The software architecture is layered to process and fuse data for precise thruster control. Visual data from a ZED 2i stereo camera feeds into ORB-SLAM3 for visual odometry. LiDAR data is processed with GTSAM to generate LiDAR-based odometry. Both streams, along with filtered IMU and real-time kinematic (RTK) GPS data, are fused through an EKF to produce a continuous, high-accuracy position estimate. Static landmarks are detected and incorporated into the map to improve localization.

This map data feeds into the ROS 2 Nav2 stack, where plugins handle navigation, task selection, and dynamic replanning, ensuring precise positioning, obstacle avoidance, and adaptive navigation in real time.

3.3.1 Computer Vision

The computer vision pipeline uses a custom dataset and transfer learning to fine-tune YOLOv11 Small. Over 14,000 hand-labeled images were used to train the model, resulting in robust detection of buoys and other objects. Transfer learning allows efficient adaptation to new objects like light beacons without retraining from scratch.

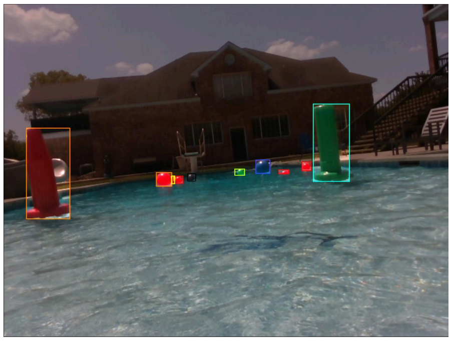


Figure 9 - YOLOv11 model detection outputs

Onboard, the model runs via the ZED SDK to generate 3D bounding boxes for detected objects. Deep learning-based tracking maintains consistent object identities over time, enabling the system to estimate task locations and interpret the surrounding environment accurately.

3.3.2 Localization and Mapping

Localization and mapping use a multi-sensor SLAM pipeline built around RTAB-Map. Stereo depth and visual features from the ZED 2i provide visual odometry and loop-closure constraints, while LiDAR and filtered IMU measurements provide robustness against visual degradation and surface disturbances. These sources, combined with RTK GPS, are fused through an EKF to produce a globally consistent state estimate. Detected static landmarks are incrementally integrated to maintain a persistent map and accurate localization relative to previously observed tasks and structures.

3.3.3 Navigation and Path-planning

Navigation is handled through ROS 2 Nav2 with mission logic implemented in behavior tree plugins. Global and local costmaps are constructed from fused SLAM output, LiDAR obstacles, and mapped task boundaries to enable dynamic obstacle avoidance. Behavior trees coordinate high-level decisions such as gate traversal, beacon interaction, and docking. This modular architecture allows navigation behaviors to be composed, interrupted, or resumed in real time, ensuring adaptive and efficient mission execution.

3.3.4 Motor Control

The ASV's motion is actuated through ArduPilot, which controls the individual thrusters. Each thruster's position and orientation are modeled in a URDF, allowing the system to map forces into linear velocities along the X and Y axes and angular rotation about Z.

Using thruster positions and orientations, the control system represents thrust control allocation as a Moore-Penrose pseudo-inverse optimization problem, constrained by tuned parameters to determine the necessary thrust commands to attain the desired body-frame velocities (v_x, v_y) and yaw rate (ω_z). Using a closed-loop approach with tuned PID controllers, the system minimizes error and ensures accurate motion even under environmental disturbances. This approach is preferred because simpler methods struggle to simultaneously satisfy all three motion requirements given the ASV's complex kinematic arrangement.

To interface with ArduPilot, ROS 2 Control outputs are sent as RC override messages via MAVROS. This allows the navigation engine, whether following buoys, avoiding obstacles, or interacting with light beacons, to issue high-level velocity requests that are translated into coordinated omnidirectional thrusts, enabling precise execution of complex maneuvers.

IV. TESTING STRATEGY

4.1.0 Testing Overview

Both simulated and in-water testing were critical for refining *Blastoise* and *Squirtle*. Approximately 150 hours of simulation and 25 hours of in-water testing provided the iterative feedback needed to validate system design against specific goals. Simulation allowed verification of logic, sensor fusion, and control algorithms in a low-risk environment, confirming, for example, that the boat would not get stuck on tasks. In-water tests validated performance under real conditions, confirming PID tuning, reliable path following, and the accuracy of 3D buoy tracking and return-to-location maneuvers, capturing environmental effects and finalizing parameters for autonomous operation.

4.1.1 Simulation

The CRANE (Cross-domain Robotics Autonomous Navigation Engine) Simulator uses Unity to provide realistic physics, sensors, and environmental interactions. Unity allows importing nearly any robot and adding custom physics or sensors, making the simulation highly flexible. Integrated with the ArduPilot SITL system and a source-agnostic ROS navigation stack, CRANE enables rapid testing of control, sensor fusion, path planning, and thruster allocation. Simulation validated that the boat would not get stuck on tasks and that the navigation system correctly identified each task based on detected arrangements, allowing the team to iteratively refine behavior and confirm design assumptions before in-water trials.

4.1.2 Data Visualization and Analysis

ROS bags record sensor outputs, commands, and system states for post-run analysis, allowing detailed replays that revealed inconsistencies in navigation parameters and highlighting where prediction, control horizons, or weight factors needed adjustment. This approach provided a clear link between observed trajectory deviations and specific parameter settings, guiding targeted refinements. A real-time dashboard visualizes errors, trends, and deviations, while a parameter-tuning GUI enables quick iteration to refine system performance.

4.1.3 In-Water Testing

After simulation, in-water testing refines performance under real-world conditions not fully captured virtually, such as hydrodynamic effects, sensor noise, and environmental variability. During these trials, T200 thruster outputs were characterized and PID parameters tuned to match observed velocities against ideal kinematics. GPS and IMU systems were calibrated to improve localization, and camera calibration and image processing were adjusted to account for water-induced distortions. Path-following was optimized through prediction and control horizons, as well as weight factoring, enabling tighter trajectory adherence.

V. ACKNOWLEDGEMENTS

We would like to thank our mentors, **Sara Harris** and **Osama Almasri**, for their help and expertise in carbon fiber composite manufacturing. We also extend our appreciation to **Blue Robotics** for their thruster discount and to the **University of Texas at Arlington** for providing access to their composite laboratories and donating carbon fiber for hull construction. Finally, we express our sincere gratitude to our teachers and sponsors, James Hovey, Jason Forsythe, Joshua Ogg, Ben Taylor, Brandon Searcy, and Laura Ebanks, for their time, guidance, and continued support throughout this project.

References

[1] B. Doyon, S. Gopalakrishnan, Frederik Møller, Jörg Schmiedmayer, and R. Vasseur, "Generalized Hydrodynamics: A Perspective," *Physical Review X*, vol. 15, no. 1, Jan. 2025, doi: https://doi.org/10.1103/physrevx.15.010501.

[2] T. Fossen, "Fossen's Marine Craft Model," *Fossen.biz*, 2021. https://fossen.biz/html/marineCraftModel.html

[3] "Nav2 — Nav2 1.0.0 documentation," *docs.nav2.org*. https://docs.nav2.org/

[4] "ZED SDK - Stereolabs," *@Stereolabs3D*, 2026. https://www.stereolabs.com/docs/development/zed-sdk (accessed Jan. 08, 2026).

[5] "RTAB-Map," *Github.com*, 2026. https://github.com/introlab/rtabmap/wiki (accessed Jan. 08, 2026).

[6] M. S. Ahmad, "Deep Learning 101: Lesson 23: The Basics of Audio Signal Processing with FFT," *Medium*, Sep. 02, 2024. https://muneebsa.medium.com/deep-learning-101-lesson-23-the-basics-of-audio-signal-processing-with-fft-ffef65689c1d

[7] ros-drivers, "GitHub - ros-drivers/velodyne: ROS support for Velodyne 3D LIDARs," *GitHub*, 2024. https://github.com/ros-drivers/velodyne

[8] "MAVROS," *GitHub*, Apr. 19, 2023. https://github.com/mavlink/mavros

[9] F. M. Mims, *Getting started in electronics*. Editorial: Niles, Il: Master Publishing, Inc, 2016.

Appendix A: Component List

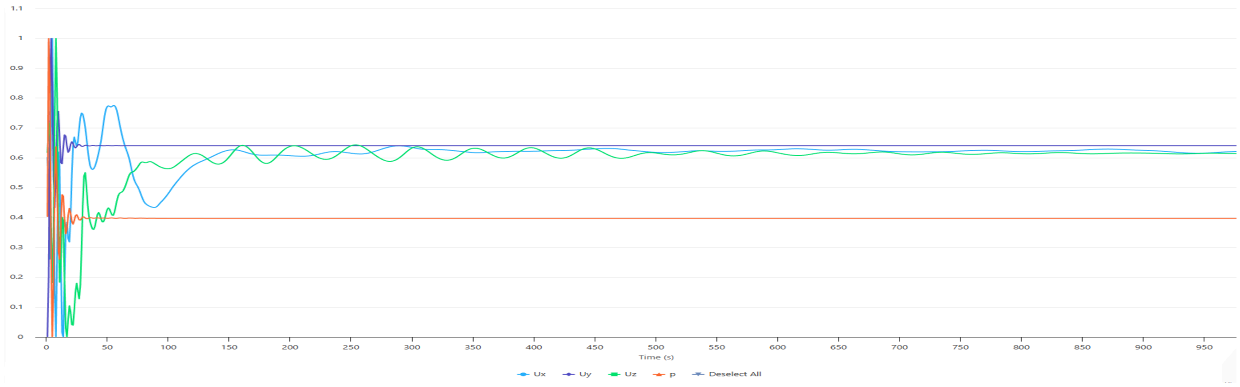
Component	Vendor	Model/Type	Custom/ Purchased	Cost	Year of Purchase
Electrical Systems Box	Polycase	WH-32 Hinged NEMA Enclosure	Purchased	$90	2025
ASV Hull Form/Platform	N/A	Carbon Fiber and Fiberglass	Custom	$175	2025
Propulsion	BlueRobotics	T200	Purchased	$230	2022
Battery	HobbyKing	Lithium-polymer battery	Purchased	$90	2022
Battery	BlueRobotics	Lithium-polymer battery	Purchased	$425	2025
Motor Controls	BlueRobotics	Basic ESC	Purchased	$40	2022
Primary GPU	Amazon	Jetson Xavier NX	Purchased	$700	2023
Primary CPU	Amazon	Odroid M1	Purchased	$135	2023
Teleoperation	Amazon	Taranis Q X7	Purchased	$137	2022
GPS	CubePilot	Here4	Purchased	$230	2025
Camera	Stereolabs	ZED 2i Stereo Camera	Purchased	$520	2025
Microphone	Amazon	NIGNYA Conference Microphone	Purchased	$12	2025
Compass/IMU	CubePilot	Cube Orange	Purchased	$485	2022
Battery Management System	JLCPCB/Mouser	N/A	Custom	$25	2025
Hotswapping Power System	JLCPCB/Mouser	N/A	Custom	$20	2025

Appendix B: Software Components

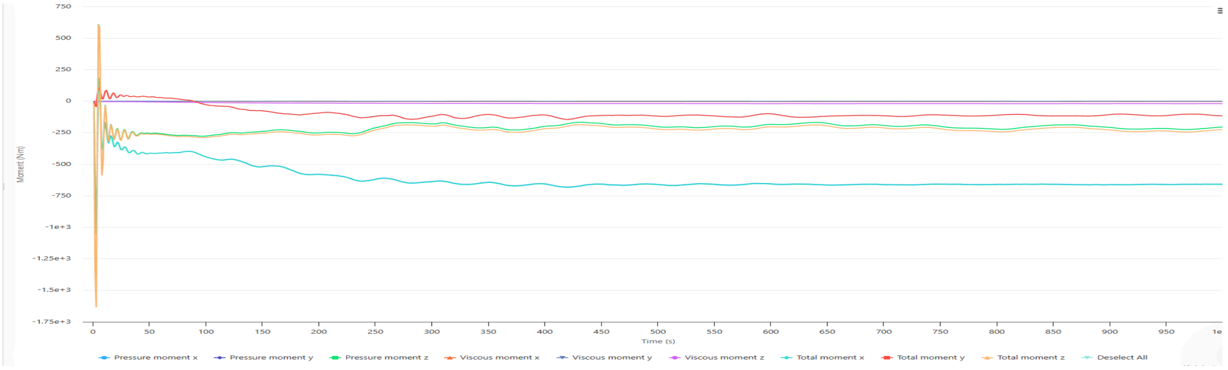
Component	Software	Publisher/Developer
Thruster Control and Navigation Sensor Data Collection	Ardupilot	Jordi Munoz and Chris Anderson (3DRobotics)
Mavlink-ROS Bridge	Mavros	DroneCode
Navigation Algorithms	Nav2	Steve Macenski, Open Navigation LLC
Robot Control Algorithms	ROS 2 Control	Open Source Robotics Alliance
Physics Simulation	Unity	Unity Software
Communication Protocol and Base Robotics Framework	ROS 2	Open Robotics
Portable Development Environment/Containerization	Docker	Docker Inc.
Teleoperation	Mission Planner	Michael Oborne
Mapping and Odometry (ORB-SLAM3)	Real-Time Appearance-Based Mapping (RTAB-Map)	Mathieu Labbe
Computer Vision and Camera Input	ZED SDK	Stereolabs
Object Detection	YOLO	Ultralytics
Object Tracking and Mapping	Find Object 2D	Introlab

Appendix C: Simulation Plots

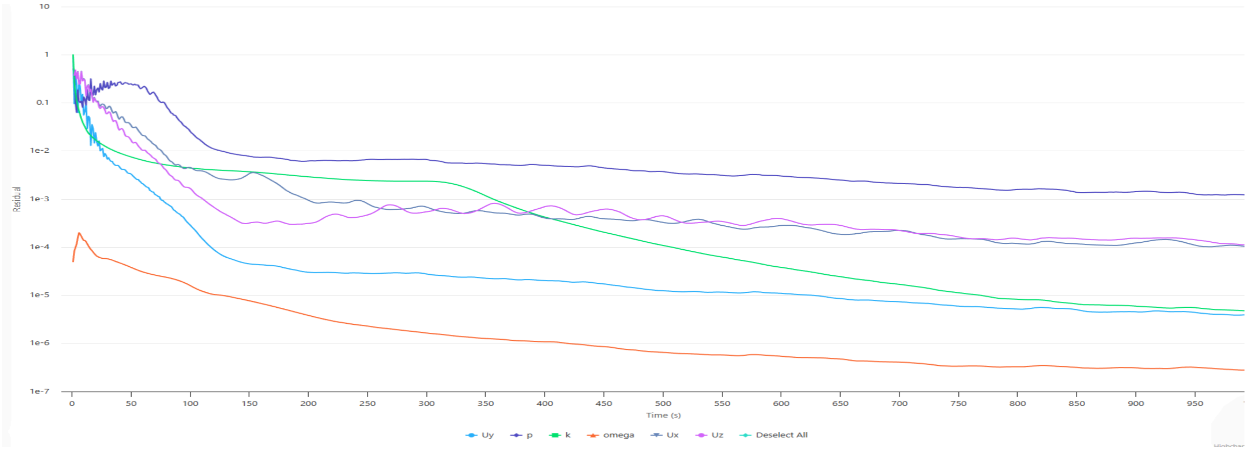
Domain Convergence Plots:



Forces and Moment Plots:



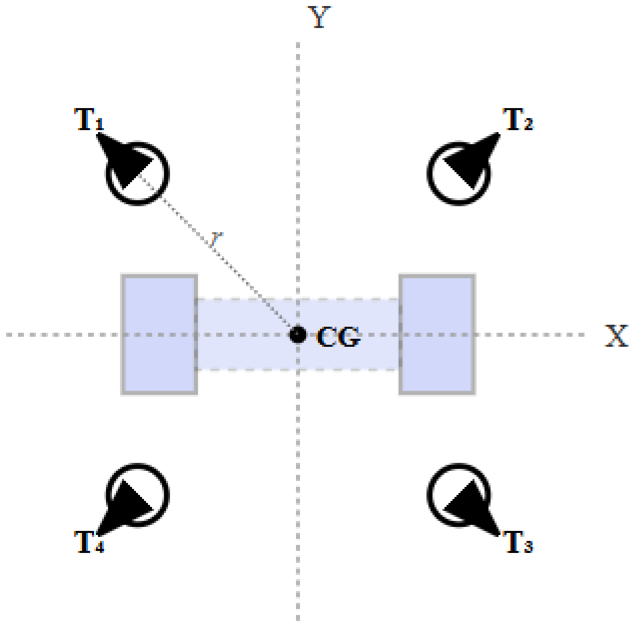
Residual Convergence Plots:



Appendix D: Thruster Configuration Analysis, Blastoise

1. Thruster Configuration:

The autonomous surface vehicle, *Blastoise*, uses an X-configuration thruster arrangement with four thrusters positioned at even intervals, angled at 45° relative to the primary body axes. This configuration provides full universal control, enabling control of forward/backward, lateral, and rotational motions.



X-Configuration thruster layout. T_1 through T_4 represent individual thrusters positioned at 45°, 135°, 225°, and 315° from the positive X-axis, respectively. The parameter r denotes the distance from the center of gravity to each thruster.

2. Forward Kinematics

The allocation matrix represents the relationship between individual thruster forces and the resulting body forces. Each thruster can be represented by a radius and a thrust vector, respectively:

$$\mathbf{r}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad \hat{\mathbf{t}}_i = \begin{bmatrix} \cos\left(\frac{\pi}{4}\right) \\ \sin\left(\frac{\pi}{4}\right) \end{bmatrix} = \frac{\sqrt{2}}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

The force generated by each thruster can be calculated by directly multiplying the thrust vector by a force scalar. The torque is the resultant of the cross product of the radius and thrust vectors.

$$\mathbf{F}_i = f_i \hat{\mathbf{t}}_i = f_i \begin{bmatrix} \cos\left(\frac{\pi}{4}\right) \\ \sin\left(\frac{\pi}{4}\right) \end{bmatrix}$$

$$\begin{aligned}\tau_i &= \mathbf{r}_i \times \mathbf{F}_i \\ &= \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ x_i & y_i & 0 \\ F_{x,i} & F_{y,i} & 0 \end{vmatrix} \\ &= x_i F_{y,i} - y_i F_{x,i} \\ &= f_i \left(x_i \sin\left(\frac{\pi}{4}\right) - y_i \cos\left(\frac{\pi}{4}\right) \right)\end{aligned}$$

Thus, the kinematics of can be represented by:

$$\mathbf{w}_i = \begin{bmatrix} F_{x,i} \\ F_{y,i} \\ \tau_i \end{bmatrix} = f_i \begin{bmatrix} \cos\left(\frac{\pi}{4}\right) \\ \sin\left(\frac{\pi}{4}\right) \\ x_i \sin\left(\frac{\pi}{4}\right) - y_i \cos\left(\frac{\pi}{4}\right) \end{bmatrix}$$

$$\begin{bmatrix} F_x \\ F_y \\ \tau \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\pi}{4}\right) & \cos\left(\frac{\pi}{4}\right) & \cdots \\ \sin\left(\frac{\pi}{4}\right) & \sin\left(\frac{\pi}{4}\right) & \cdots \\ x_1 \sin\left(\frac{\pi}{4}\right) - y_1 \cos\left(\frac{\pi}{4}\right) & x_2 \sin\left(\frac{\pi}{4}\right) - y_2 \cos\left(\frac{\pi}{4}\right) & \cdots \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ \vdots \end{bmatrix}$$

3. Inverse Kinematics (Control Allocation)

We model the relationship between thruster forces and the resulting body force and torque as a linear system.

Because the system usually has more thrusters than controlled degrees of freedom, this matrix is not square and cannot be inverted directly. Instead, we use the Moore–Penrose pseudoinverse, which provides the least-squares solution that minimizes the total thrust magnitude while best achieving the desired force and torque.

The pseudoinverse gives the thruster forces that most closely produce the requested body wrench in a minimum-norm sense.

$$\mathbf{w} = \begin{bmatrix} F_x \\ F_y \\ \tau \end{bmatrix} = \mathbf{Bf}, \qquad \mathbf{f} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix}$$

$$\begin{aligned} \mathbf{f} &= \mathbf{B}^{\dagger}\mathbf{w} \\ &= (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{w} \end{aligned}$$

$$\mathbf{B} = \begin{bmatrix} \cos\left(\frac{\pi}{4}\right) & \cos\left(\frac{\pi}{4}\right) & \cdots \\ \sin\left(\frac{\pi}{4}\right) & \sin\left(\frac{\pi}{4}\right) & \cdots \\ x_1 \sin\left(\frac{\pi}{4}\right) - y_1 \cos\left(\frac{\pi}{4}\right) & x_2 \sin\left(\frac{\pi}{4}\right) - y_2 \cos\left(\frac{\pi}{4}\right) & \cdots \end{bmatrix}$$

Combined with constraints, it is possible to derive the necessary outputs from each thruster to achieve each of the desired velocities.

4. Motion Capabilities

The symmetrical design of the X-configuration offers several operational benefits. Pure translation in any direction is achievable by activating specific thruster pairs. Pure rotation is produced through differential thrust; T_2 and T_4 generate clockwise torque, while T_1 and T_3 produce counterclockwise torque. The system allows for simultaneous translation and rotation, enabling complex maneuvers like arc trajectories and point turns. Its balanced geometry ensures equal maximum acceleration in all horizontal directions, simplifying path planning and enhancing maneuverability in confined spaces such as competition courses.

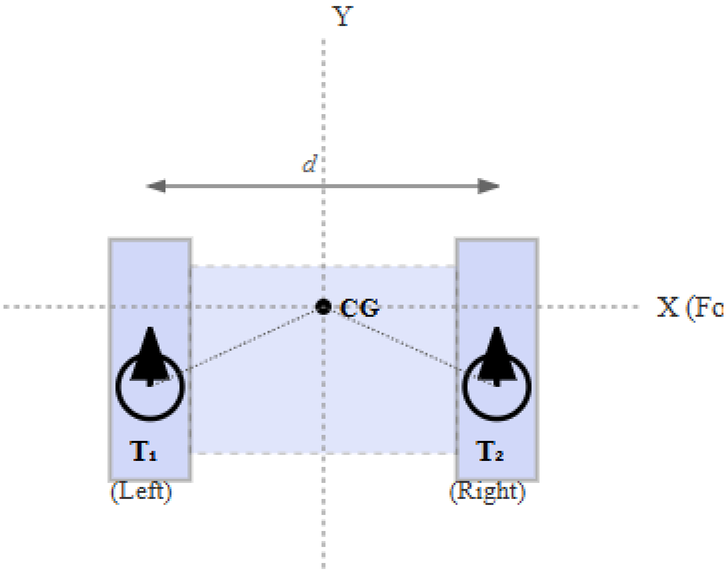
5. Considerations

In real-world applications, control allocation must handle thruster limits. When thrust surpasses the maximum, the control system scales commands proportionally to preserve the desired direction while reducing magnitude.

Appendix E: Thruster Configuration Analysis, Squirtle

1. Thruster Configuration

In a differential drive, two thrusters are aligned parallel to the vessel's longitudinal axis and positioned side-by-side. By applying differential thrust, the vehicle can move forward, backward, and rotate independently. Compared to omnidirectional configurations, this configuration provides reliable control with reduced complexity.



Differential drive thruster layout. T₁ and T₂ represent left and right thrusters, respectively, positioned symmetrically about the centerline. The parameter d denotes the lateral separation distance between thrusters, measured perpendicular to the vessel's longitudinal axis.

2. Forward Kinematics

The differential drive configuration enables two degrees of freedom: forward/backward translation and rotation about the vertical axis. Lateral motion is not directly achievable. The relationship between individual thruster forces and resultant body forces is expressed as:

$$\begin{bmatrix} F_x \\ \tau_z \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ -\frac{d}{2} & \frac{d}{2} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}$$

F_x represents the net forward force, τ_z is the net torque about the vertical axis, T_i denotes the thrust magnitude from thruster *i*, and *d* is the lateral separation between thrusters (in meters). The torque contribution arises from the moment arms ±d/2, where the left thruster creates clockwise torque (negative) and the right thruster creates counterclockwise torque (positive) when producing forward thrust.

3. Inverse Kinematics (Control Allocation)

The control system must calculate the necessary thruster commands from the desired body motion in order to navigate autonomously. Matrix inversion yields the inverse relationship:

$$\begin{bmatrix} T_1 \\ T_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & -\frac{1}{d} \\ \frac{1}{2} & \frac{1}{d} \end{bmatrix} \begin{bmatrix} F_x \\ \tau_z \end{bmatrix}$$

Given the desired forward force and rotational torque, this inverse allocation matrix calculates the thrust needed from each motor directly. High-level motion commands are produced by the control system, transformed into forces and torques using dynamic models, and then mapped to specific thruster outputs using this equation.

4. Motion Capabilities

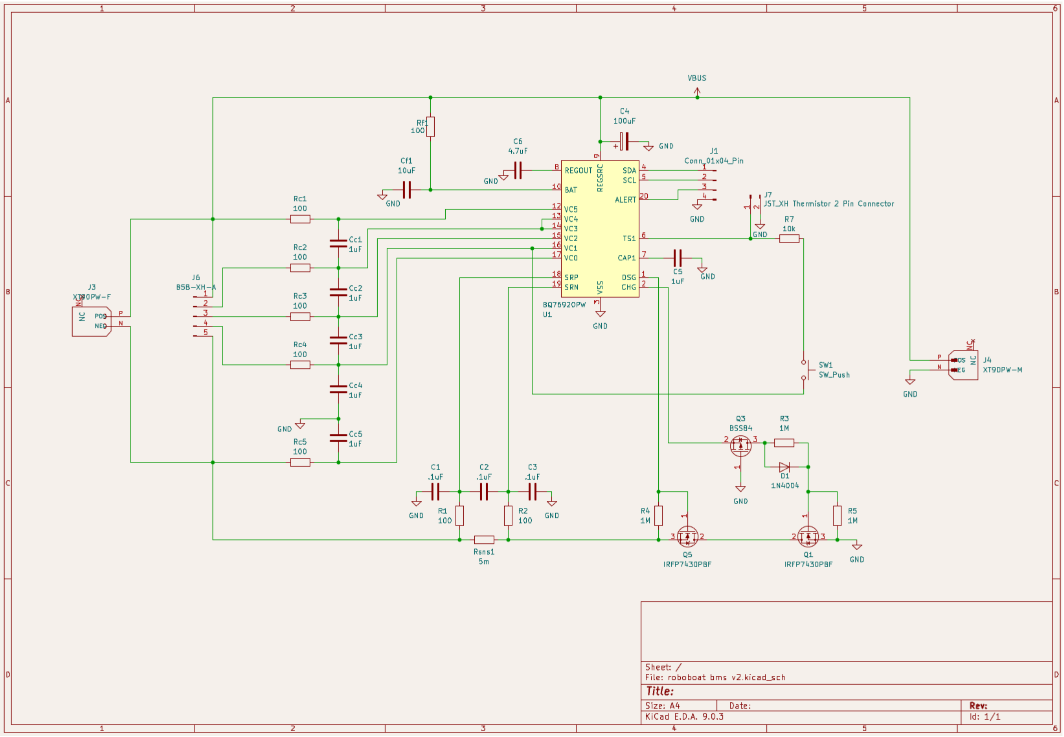
The differential drive enables several fundamental maneuvers. Pure forward motion is achieved when $T_1 = T_2 > 0$, producing maximum forward velocity with zero rotational component. Pure backward motion occurs when $T_1 = T_2 < 0$. Point-turn rotation, where the vessel rotates in place without translational motion, is accomplished by setting $T_1 = -T_2$, creating pure torque with $F_x = 0$. Arc trajectories result from unequal but same-sign thrust values (e.g., $T_2 > T_1 > 0$ produces a leftward arc). The minimum turning radius at a given forward speed v is $R_{min} = vd\cdot(2T_{max})^{-1}$, where T_{max} represents maximum available thrust per motor. Unlike omnidirectional configurations, lateral translation (sideways motion) cannot be performed without first rotating the vessel to align the thrust axis with the desired direction of travel.

5. Implementation Considerations

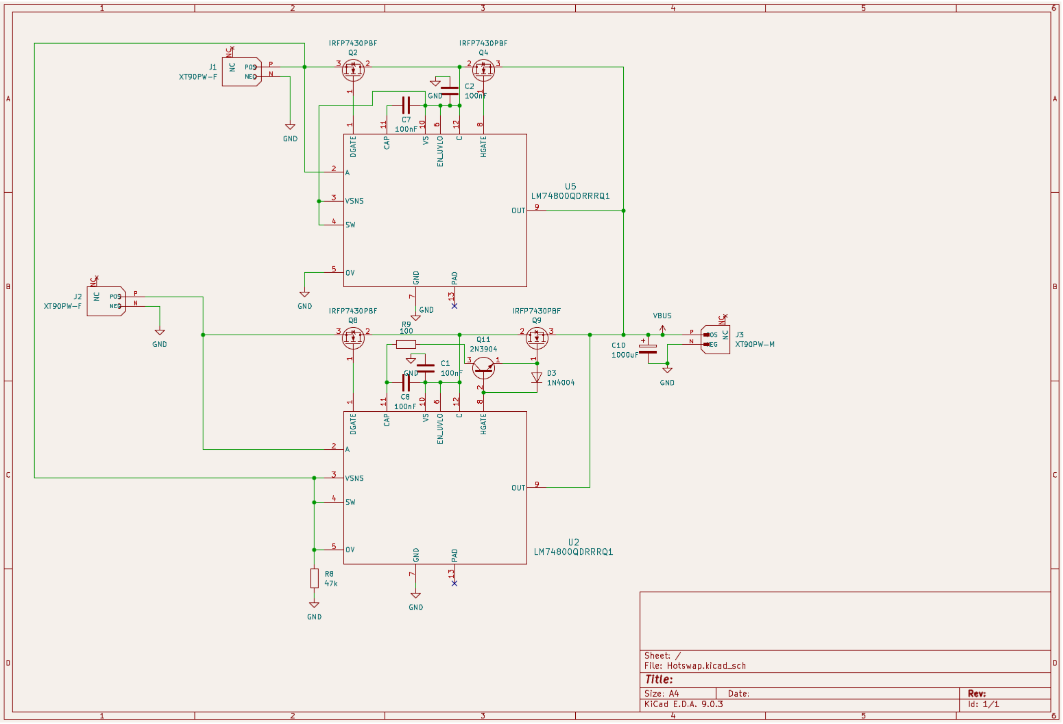
The differential drive configuration offers several practical advantages, including reduced component count, simplified mechanical integration, lower power consumption, and increased reliability due to fewer actuators. However, the lack of lateral control authority introduces constraints in mission planning. Complex maneuvers such as docking or precise lateral positioning require sequential operations: rotate to the desired heading, translate forward/backward, then rotate to the final orientation. This multi-step approach increases time to complete tasks compared to holonomic systems but benefits from simpler control algorithms and more robust performance in the presence of disturbances. Thruster saturation limits must be enforced in the control allocation, and when commanded thrust exceeds T_{max}, proportional scaling maintains the desired thrust ratio to preserve trajectory characteristics while reducing magnitude.

Appendix F: Electrical Schematics

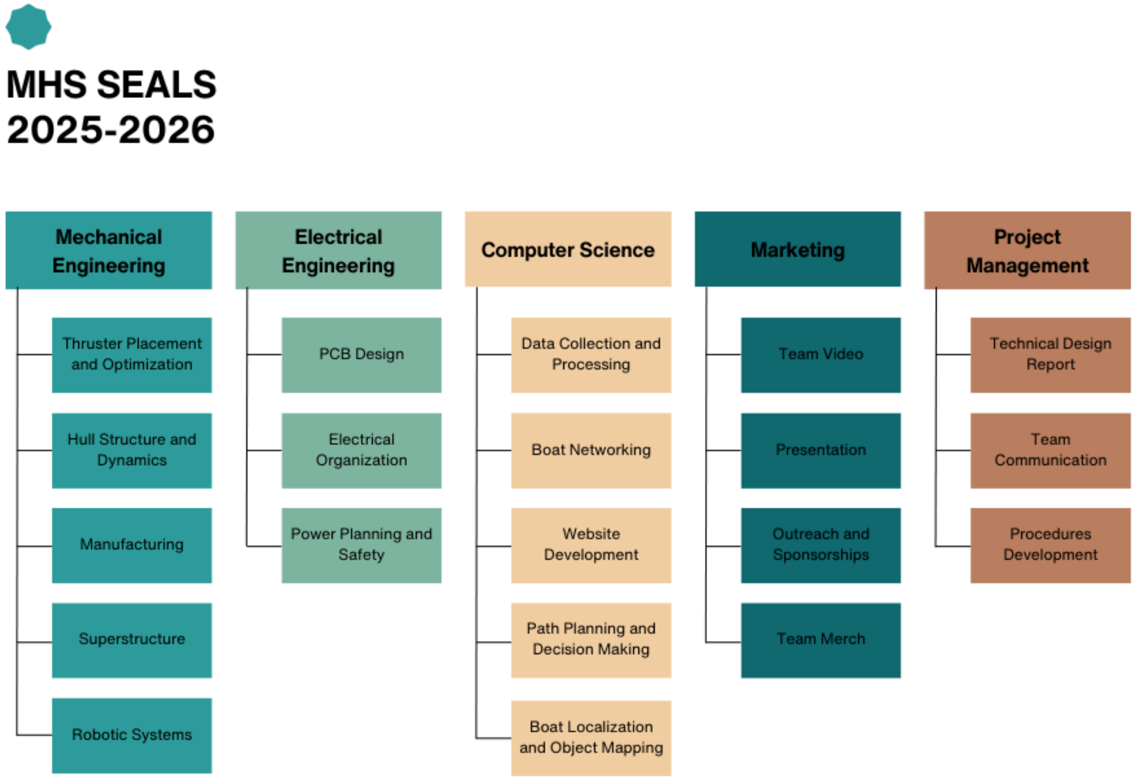
A. Battery Management System:



B. Hotswapping Power System:



Appendix G: Responsibility Hierarchy



MHS SEALS
2025-2026

- Mechanical Engineering
  - Thruster Placement and Optimization
  - Hull Structure and Dynamics
  - Manufacturing
  - Superstructure
  - Robotic Systems
- Electrical Engineering
  - PCB Design
  - Electrical Organization
  - Power Planning and Safety
- Computer Science
  - Data Collection and Processing
  - Boat Networking
  - Website Development
  - Path Planning and Decision Making
  - Boat Localization and Object Mapping
- Marketing
  - Team Video
  - Presentation
  - Outreach and Sponsorships
  - Team Merch
- Project Management
  - Technical Design Report
  - Team Communication
  - Procedures Development