RoboBoat 2026: Technical Design Report

Autoboat, Cornell University
Ithaca, New York, United States

I. Abstract

For RoboBoat 2026, the team's main focus was codebase development and validation through water testing and simulations, to improve performance in autonomous navigational tasks. An emphasis was placed on simplicity and efficiency, both for technical aspects and team timelines and milestones. Each competition task has a clear logical strategy associated with it, and each component of the software system was designed from the ground up to integrate easily to accomplish these tasks. This year marks the debut of our new trimaran vessel, Scooby. The new platform provides increased accessibility to internal systems, enhanced stability, and more real estate for several key hardware upgrades, such as new sensors, electrical bay, robotics subsystems, and thrusters. Additionally, the electrical system has undergone a complete redesign, including reorganizing high and low power distribution across the vessel, increasing shielding against electromagnetic interference and integrating a custom microcontroller PCB.

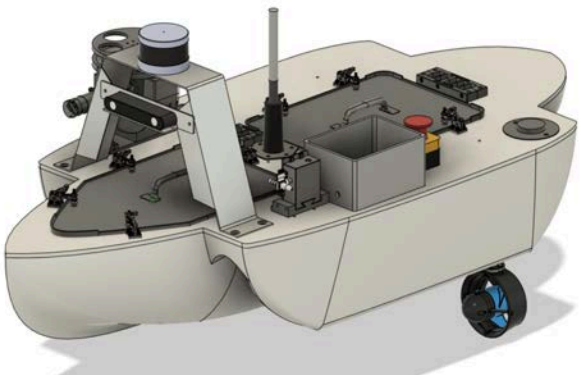


Figure 1: Top Level Vessel Assembly

II. Competition Strategy

A. Approach

Our primary strategy for the RoboBoat 2026 competition is to perform all navigation tasks (tasks 1-3, 5). We directed the majority of our focus towards these tasks as they make up a large part of point awards and serve as an essential capability for any intelligent ASV.

B. System Overview

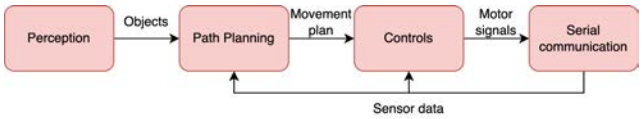


Figure 2: Overview of communication between software components. See Appendix D Figure 20 for a more comprehensive diagram of the ROS framework.

Autoboat's software system has three main components: Perception, Path Planning, and Controls. The Perception system gathers data on the vessel's surroundings, position, and orientation, combining it with sensor serial communication. Path Planning uses this information to determine the task and plan a path or sequence of movements. The Controls system then executes the plan by driving the motors to achieve the desired motion. All components are integrated through our ROS2 Humble [1] framework, enabling parallel operation and seamless communication.

B. Task Strategies

<u>Evacuation Route & Return (Entry & Exit Gates)</u>
For the Evacuation Route & Return task, the vessel uses computer vision and LIDAR to identify red and green buoy pairs forming gates. A waypoint [2] is set at the midpoint of the closest gate, and the vessel then searches for the next gate, moving forward if needed. Once found, a second midpoint waypoint is created to continue the route.

<u>Debris Clearance (Navigation Channel)</u>
To complete the Debris Clearance task, the vessel identifies red and green buoy pairs

forming gates, with the buoys sorted by distance. For each gate, a waypoint is created at the midpoint between the buoys. If only one buoy is visible, the vessel pivots to locate its pair; if unsuccessful, it searches within a viable radius for the closest buoy that could form a valid gate. If neither gate buoy is visible, the system selects the most viable pair using distance and angular information. Throughout execution, Perception continuously monitors for obstacles (black buoys) and recalculates a safe, collision-free path while maintaining progress toward the gate. The channel is considered ended when no gate buoys are visible ahead, and the vessel then backtracks along its waypoints to complete the task.

Emergency Response Sprint (Speed Challenge)

For the Speed Challenge task, the vessel scans its surroundings and detects all visible buoys using its onboard LIDAR system, then constructs a convex hull [3] to define the navigational boundary. As it progresses, the hull is continuously updated; if a yellow buoy is detected, the hull is rebuilt to incorporate the new information. Based on the beacon color, the vessel either preserves the hull's counter-clockwise ordering (for green) or reverses it to execute a clockwise loop (for red). The vessel then navigates around the yellow buoy in the required direction, maintaining safe offsets and real-time obstacle avoidance. After completing the loop, it departs the buoy region and returns to the bay to finish the task.

Navigate the Marina (Docking)

The LIDAR-based docking system employs a multi-stage waypoint navigation strategy. At task start, it computes the dock center as the centroid of all detected bays and an entrance center as the midpoint of the first bay pair, establishing the entry gateway. The vessel then navigates through three waypoints: current position → entrance center → dock center using the standard path pipeline. At the dock center, it selects the lowest-numbered open bay and executes a three-stage approach: lateral alignment, perpendicular entry, and a final 0.5-meter penetration to ensure secure docking.

III. Software Design Strategy

Perception

This year, our perception pipeline underwent a major architectural shift by integrating Light Detection and Ranging (LIDAR) [4] as a primary detection modality alongside our existing computer vision (CV) stack. Instead of relying exclusively on You Only Look Once (YOLO) [5] neural networks for both detection and classification, we decoupled these tasks across two specialized subsystems: LIDAR to handle object detection and localization, and CV models to classify pre-detected regions of interest.

Our CV model is a combination of 3 separate models: buoy classification, sign classification, and beacon classification. For the buoy classification model (YOLOv8 neural network), our data comprises images that we have gotten from previous competitions and current testing videos that we recorded through our ZED2i camera [6]. For the sign classification model, we have the black triangle and cross image data from last competition [7], as well as a public TMNIST dataset [8] for images of black font 1, 2, and 3 on a white background, analogous to the numerical signs that we will see. For classifying the beacons by color, we use OpenCV [9] to get the average color inside the passed 'bounding box' and sequentially classify it as red or green.

The LIDAR subsystem is responsible for developing and deploying advanced 3D perception algorithms using a Velodyne VLP-16 [10] sensor for autonomous navigation and object recognition. The perception pipeline follows a multi-stage architecture consisting of point cloud preprocessing, geometric feature extraction, and object classification.

Our detection framework employs conditional Euclidean clustering with distance-adaptive threshold parameters to segment point cloud data into discrete object candidates. These thresholds dynamically adjust based on point cloud density and range information, enabling robust segmentation across varying environmental conditions and object scales.

Object classification is achieved through custom RANSAC geometric fitting algorithms

[11] tailored to competition elements. We use specialized models for planar feature extraction in dock detection, as well as cylindrical and spherical primitive fitting for buoy identification. These model-based approaches provide reliable performance even in the presence of sensor noise and partial occlusions.

The LIDAR subsystem serves as a critical component of our multi-modal sensor fusion framework, working closely with the controls team to integrate perception data from sensors. We perform coordinate frame transformations and temporal synchronization between LIDAR point clouds, ZED2i stereo camera imagery, and IMU measurements to generate a unified environmental representation.

Our ROS infrastructure supports this multi-model system through a main_perception node that orchestrates the perception pipeline by launching and managing all perception-related nodes. Requests are routed between LIDAR, classifier nodes, and a buffer service, and with synchronized timestamps, LIDAR detections trigger classification services while LIDAR and computer vision reference the same frame.

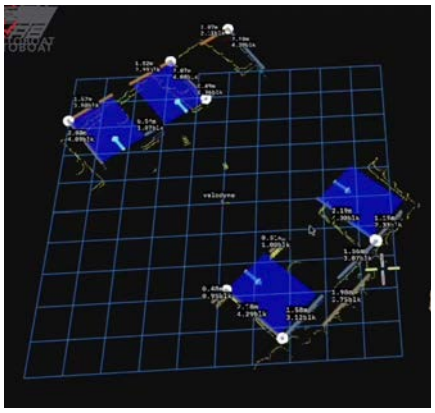


Figure 3: LIDAR Docking visualization

Controls

For accomplishing navigational tasks, the vessel explores the environment until it gains sufficient information regarding the task at hand. Then, it uses the information about its current state (GPS location, VN-300 heading) and environment consisting of objects detected and their locations relative to the vessel, to plan a path of GPS waypoints. After primary waypoints are selected we inject intermediate waypoints every meter or half meter along the path and smooth any harsh angles to produce more natural paths of movement.

To follow the waypoint path outlined by the path planning system, the control system employs a combination of Pure Pursuit tracking algorithm and PD control. Our implementation is inspired by the Purdue SIGBot's controller [12]. We found that controlling on velocity produces rough and shaky movements due to the sensitivity of our IMU sensor, thus we chose to monitor the heading variable from the VN-300. We then apply PD (proportional, derivative) control to reduce error, which outputs the offset which should be applied to the PWM signals sent to move the thrusters.

Our new multi-modal sensor fusion framework also plays a critical role in supporting the controls and autonomy of our system. A central fusion node takes in readings from multiple sensors aboard (ZED2i camera, LIDAR, GPS, IMU, etc.) and synchronizes its information to form a cohesive real-time understanding of the environment. For object localization, we rely on LIDAR readings for buoy localization and CV readings to fill in information on color and visual patterns of detected objects. In the docking task, the fusion node similarly relies on LIDAR readings for dock positioning while CV determines the sign values associated with specific dock bays. Euclidean distance uncertainty thresholds are used to match LIDAR and CV object readings and to filter out inaccurate readings.

IV. Hardware Design Strategy

Our past vessel, Clifford, proved to be reliable and stable, but struggled significantly with ergonomics and accessibility. Scooby has rectified these shortcomings while iterating on Clifford's strengths in unison. The trimaran configuration of Clifford was further optimized for a range of metrics. A high level of stability was achieved through innovations in the hull geometry and the adoption of shoulders on the main hull. Revised manufacturing procedures enabled the amas and main hull to be connected, maximizing internal volume as well as deck space. Ergonomics were improved by adopting

an alternative internal arrangement and revised deck layout, increasing accessibility to critical elements and simplifying assembly. The electrical system was completely overhauled, introducing multiple new PCBs able to support an increased current load from the upgraded thrusters.

A. Mechanical System

Hull Design Iteration

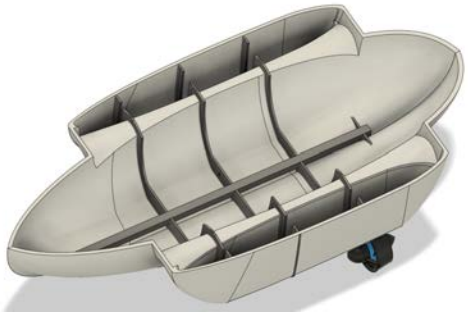


Figure 4: Internal View of Hull Geometry

The hull geometry and dimensions were determined by running multiple iterations of the Hydrostatics & Stability simulations on Orca3D, a software plug-in providing specialized features for vessel design and analysis. A key metric used to evaluate stability is the transverse metacentric height (GM_t). A similar GM_t value (within ± 5%) to Clifford was targeted, as it has provided a balanced level of stiffness and stability.

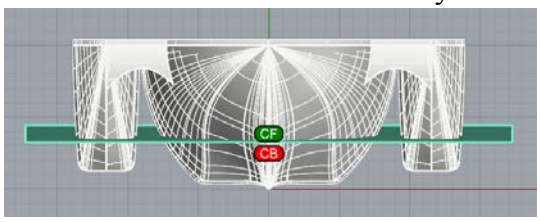


Figure 5: Boat front profile with sinkage indicated by water plane as well as labelled points for center of buoyancy (CB) and center of flotation (CF)

One of the main geometric changes we made to the main hull was adding shoulders to improve stability against heeling, acting like a small bilge keel. This feature greatly improves stability, increasing the waterplane area on the submerged side, and damping the motion of the vessel during movement. The static trim angle of the vessel was adjusted to account for increased thrust, ensuring the main hull remains flush to the waterline at full displacement and the vessel remains in line with the water plane.

Revised Manufacturing Methodology

Although the trimaran configuration was retained, a new combined hull and ama structure was employed, simplifying system integration and maximizing available deck space.

By using a single mold, we remove the need for the structural beam that was present to connect the hull and amas in past designs. This enables the use of a continuous deck, increasing the amount of available space for robotics sub-systems Additionally, previously external wiring for components such as the thrusters, can now be routed internally. See appendix B for details.

Improved Ergonomics and Accessibility

A major issue encountered during RoboBoat 2025 was the accessibility of the electrical components within the vessel. To address these concerns, broad changes were made to the deck and internal layout.

The arrangement of the deck hatches was adjusted to closely match the deck geometry, the internal ribs were also redesigned to eliminate the top bar that was blocking access between hull sections [17]. Additionally, the total number of ribs was decreased, improving accessibility while also saving weight. The final number of ribs was validated by structural simulations of the hull.

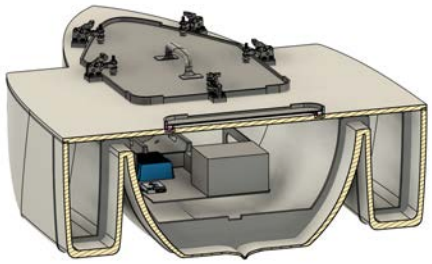


Figure 6: Cross-sectional view of the hull showing enlarged custom hatches and internal ribs.

Previous attempts at addressing clutter in the electrical bay used swappable 3D-printed bins, which frequently obstructed wire pathways. The increased internal volume of the new hull enables the use of two larger open sheet metal trays that provide mounting points for individual components. These trays are mounted directly

underneath the large deck hatches, allowing for easy access and viewing of wire configurations.

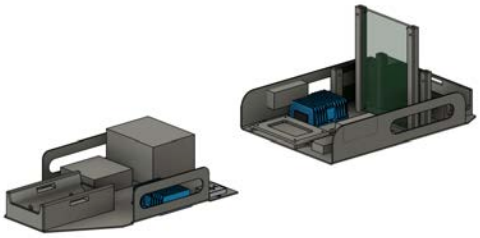


Figure 7: Electrical bay assembly

Water Pump Enclosure

During fall testing sessions, we identified EMI from the water pump as a key risk to system reliability. A faraday cage was subsequently integrated into a redesigned enclosure, whose effectiveness was validated with testing.

B. Electrical System

Upgraded Thrusters

A driving force of the electrical redesign was the integration of a pair of T500 thrusters, which increases the maximum current draw of the system by over 20 amps, which requires both a larger high power battery and significant changes to our high power kill switch design.

Power System

The core concept of isolating high and low power has been maintained from our previous design, protecting the more sensitive low power signals like the RP2040 based microcontroller from noise and interference coming from the higher power switching of our thrusters.

Our high-power system continues to run through our custom kill-switch board. This year, the board has two LED indicators to indicate the source of the kill signal. The low-power electronics pass through our power distribution board, which now includes a boost converter step up 14.8V to 24V to inject POE for the Jetson and Bullet.

A new robotics power distribution board was added to the system to increase the modularity of our low power system, and support the two 12V motors and two 5V servos. This board can be automatically disconnected from power via the robotics relay.

Sensor and Computer Hardware

The vessel's main computer is a Stereolabs Nvidia Jetson Orin NX, an upgrade from our Jetson Xavier last year. We use a ZED 2i stereo camera for object detection and depth sensing. A goal this year was to improve the accuracy of our localization, which was previously informed by a compass and GPS system using RTK. However, experimentation showed that this system did not provide consistent, accurate results. We have instead opted to rely on the VectorNav VN-300 inertial navigation system (INS), which implements an extended Kalman filter to couple information from its built-in GPS and IMU to precisely determine global position and heading.

PCB Design

The kill switch, power distribution, robotics power distribution, microcontroller and robotics relay are handled by compact PCBs, increasing modularity. One major addition to the system is the design of a custom microcontroller PCB with embedded crystal, SWD header, flash, and RP2040 IC for all on-boat digital controls. One major advantage of this board is the ability to ensure robust connections by replacing previous loose jumper cables with standardized molex microfit connectors.

V. Testing Strategy

A. Land & Simulation Testing

Mechanical Testing

Any component that was expected to be under significant load in the event of a crash was simulated in ANSYS Structural to determine the factor of safety. The simulations were verified with mesh independence studies as well as stress convergence tests. The stability of the vessel was simulated in Orca. The new hatch design, as well as all of the waterproofing on the vessel, was tested thoroughly by spraying water over the vessel and checking for leaks before electrical integration.

Electrical Testing

Thorough electrical testing was essential to confirming the functionality of each individual PCB system and the functionality of the system as a whole. Our primary concern is maintaining

required voltages over a significant duration of time at expected maximum load. To verify this, each PCB's design incorporates test points of all voltage nets that enables verification using power supplies, microcontrollers, oscilloscopes, multimeters, and various test loads. Verification was performed individually on each both before being connected to the rest of the system, to protect our sensitive components from unexpected voltage or current spikes.

Software Testing

We tested our computer vision model by deploying it onto the vessel's computer, staging buoys around the vessel's environment, and using a combination of command line output and visualization output to smoke-test our model.

Given constraints on water testing due to weather and a lack of access to pools, we continued to use our Unity-based simulation framework to test path-planning code. The simulation connects to ROS and runs the path planning node assuming perfect controls and perception. It then displays the vessel traveling the path in real time. The additional drag and drop map builder allows different testing scenarios to easily be created and shared.

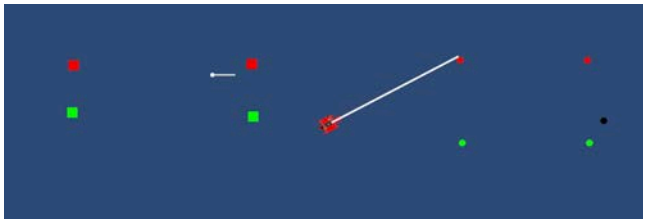


Figure 8: Unity simulation of a vessel running a "hit buoy"algorithm. See Appendix D Figure 21 for more detail on the Unity framework

Additionally, we constructed a Unity-based simulation framework to generate LIDAR data that would be representative of the competition environment, allowing us to test our LIDAR-based detection algorithms. This simulation is fully 3D and connects to ROS. Moving forward we plan to add compatibility with a camera, allowing us to generate data to test our computer vision, as well as add features to combine this with the 2D simulation to have a fully functional three-dimensional simulation.

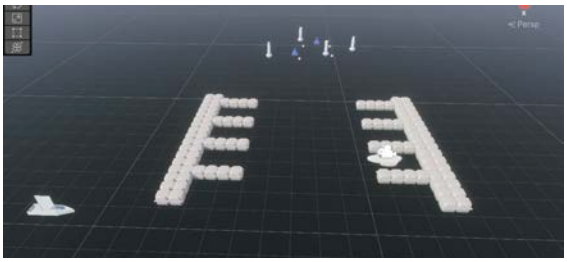


Figure 9: LIDAR Simulation Setup

B. Water Testing

Our water testing includes outdoor testing at a private pond and indoor testing at local pool facilities. This testing consists of four stages: 1) safety testing to replicate the checks done at the onset of competition, 2) remote controlled buoyancy, speed, and stability testing to examine the maneuverability of the hulls, 3) autonomous control algorithm tuning (e.g. PD gains, pure pursuit lookahead distance), and 4) autonomous navigation testing.

VI. Acknowledgements

We sincerely thank everyone who supported our team and enabled us to innovate and grow as engineers. We are grateful to the Cornell University College of Engineering for funding our travel and material expenses and providing access to facilities for vessel construction. Our corporate sponsors, Saronic Technologies and ASML, provided funding and invaluable guidance throughout the development of Scooby. We also thank Vectornav, Orca3D, and Dassault Systèmes for supplying industry-leading hardware and software.

We appreciate the support of our friends, family, and AutoBoat alumni whose generous contributions have greatly aided our efforts. Special thanks go to Dr. Hunter Adams, our faculty advisor, and past members Joanna Lin, Jeffrey Xiang, Kevin Cui, Ben Melcher, and Slava Koblov for their guidance. We are also grateful to UM::Autonomy, RoboBoat staff including Nicholas Cassasa and Bill Porter, and our fellow RoboBoat teams for their advice and support. Thank you to Hotel Ithaca for allowing us to use their pool facilities. Finally, we acknowledge the dedication of former AutoBoat team members whose contributions laid the foundation for our current achievements.

VII. References

[1] ROS 2 Documentation — ROS 2 Documentation: Humble documentation. (n.d.). https://docs.ros.org/en/humble/index.html

[2] N. Sariff and N. Buniyamin, "An Overview of Autonomous Mobile Robot Path Planning Algorithms," 2006 4th Student Conference on Research and Development, Shah Alam, Malaysia, 2006, pp. 183-188, doi: 10.1109/SCORED.2006.4339335.

[3] Jeff. (2004). Algorithms Lecture 14: Convex Hulls. In *Convex Hulls Algorithm*. https://jeffe.cs.illinois.edu/teaching/compgeom/notes/14-convexhull.pdf

[4] Ibm. (2025). *LiDAR*. https://www.ibm.com/think/topics/lidar

[5] *YOLOV8: State-of-the-Art Computer Vision Model*. (n.d.). https://yolov8.com/

[6] *ZED 2i | Stereo Camera | StereoLabs*. (n.d.). https://www.stereolabs.com/store/products/zed-2i

[7] https://app.roboflow.com/cornell-autoboat-fz9yv/sign-classification-model-2026-qwzrd/3/images

[8] https://www.kaggle.com/datasets/nimishmagre/tmnist-typeface-mnist

[9] https://opencv.org/

[10] *VLP 16 | Ouster*. (n.d.). Ouster. https://ouster.com/products/hardware/vlp-16

[11] Oreate. (2026, January 7). Detailed explanation of RANSAC plane fitting algorithm based on PCL - Oreate AI blog. *oreate.com*. https://www.oreateai.com/blog/detailed-explanation-of-ransac-plane-fitting-algorithm-based-on-pcl/cdb5bed2df58c08c2d90f4251187f458

[12] S, Xiang, “Basic Pure Pursuit.” Last Modified May, 2023. Purdue SIGBots Wiki. https://wiki.purduesigbots.com/software/control-algorithms/basic-pure-pursui

[13] https://pip-assets.raspberrypi.com/categories/610-raspberry-pi-pico/documents/RP-008307-DS-1-pico-datasheet.pdf?disposition=inline

[14] https://github.com/v923z/micropython-ulab?tab=readme-ov-file

[15] https://github.com/scipy/scipy/blob/v1.16.2/scipy/signal/_spectral_py.py#L1413-L1601

[15] “ORCA 6.0 Manual,” Faccts.de, 2024. https://www.faccts.de/docs/orca/6.0/manual/

[16] A. Biran and Rubén López-Pulido, Ship Hydrostatics and Stability. Elsevier, 2024.

[17] “Installing Hatches and Deck Plates,” Boatus.com, 2025. https://www.boatus.com/expert-advice/expert-advice-archive/2012/july/installing-hatches-and-deck-plates

VIII. Appendix A: Pre-Analysis and Simulation

Hydrostatics

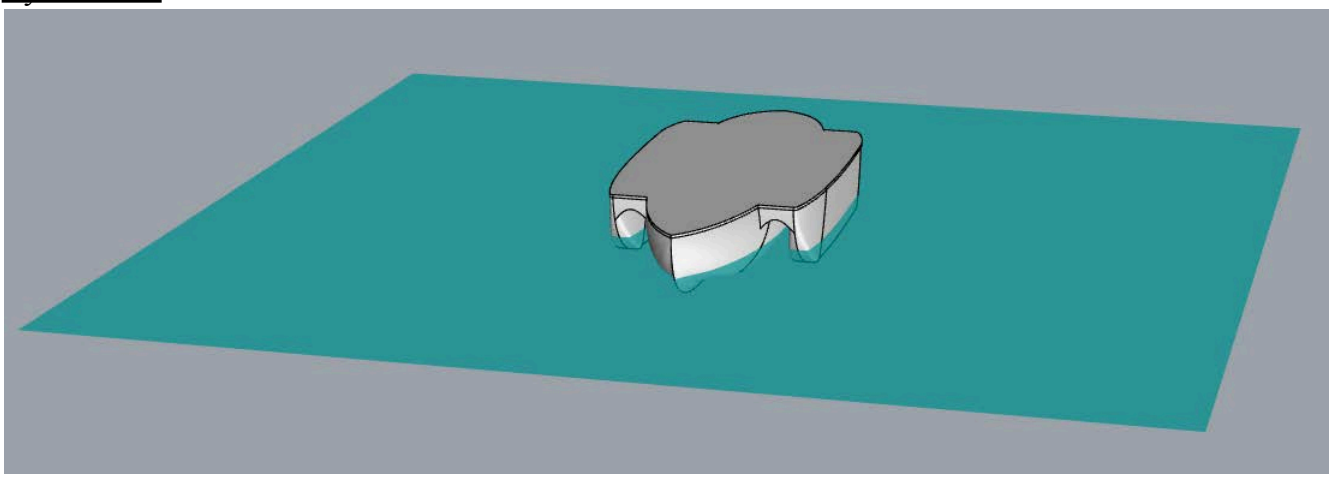


Figure 10: Render of Scooby with simulated waterline

The simulation results reported the draft of the vessel, in terms of its sinkage under the waterline, to ensure vertical stability. At the reported sinkage, the weight of the vessel hull and amas combined is counteracted by the vertical buoyancy force, as informed by Archimedes' principle. In analyzing the stability of the vessel, it was important for the vessel to be able to carry the predicted weight without excessive sinkage. Ideally, the buoyancy force can fully counteract the vessel's weight at a sinkage less than half of the vessel's overall height. A vessel of excessive draft can result in high resistance in the water or drag, which, in turn, reduces the maneuverability and speed at which the vessel is able to travel [16].

For hydrostatics, a mass budget for the entire vessel was completed. Components were then put into a Weight and Cost table in Orca, and the total center of mass was calculated. The final trim is -0.165 degrees and the heel is 0.004 degrees [15].

Stability

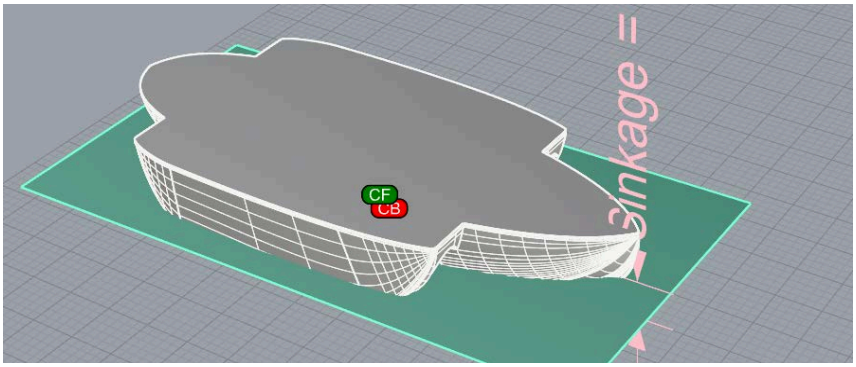


Figure 11: Boat angled perspective with numerical sinkage value as well as CB and CF points displayed

For the current design iteration, a target GM_t of approximately 13 inches was selected, matching last year's value, as it provided an effective balance between stability and stiffness. Stability analysis shows that the current design achieves nearly identical performance to the previous iteration, with only a 2.499% difference in GMt, confirming that the stability objective was met.

An increased GM_t produces a larger righting moment and reduces the likelihood of capsizing;

however, excessively high GM_t values can result in an overly stiff vessel that responds too abruptly to changes in heel angle. In general, increasing the ama dimensions enhances static stability and increases the vessel's GM_t by providing greater buoyant force to resist heeling. It was important for the transverse center of gravity (TCG) to stay zero at all time values to ensure that the vessel remains balanced laterally when it is not moving .

The righting arm reaches its peak at a heel angle of approximately 35°, which aligns well with expected operating conditions involving small to moderate environmental disturbances. While the righting-arm curve rises slightly more gradually than before, it reaches a comparable maximum value of approximately 4.25 inches. The vessel maintains strong stability across a wide range of heel angles.

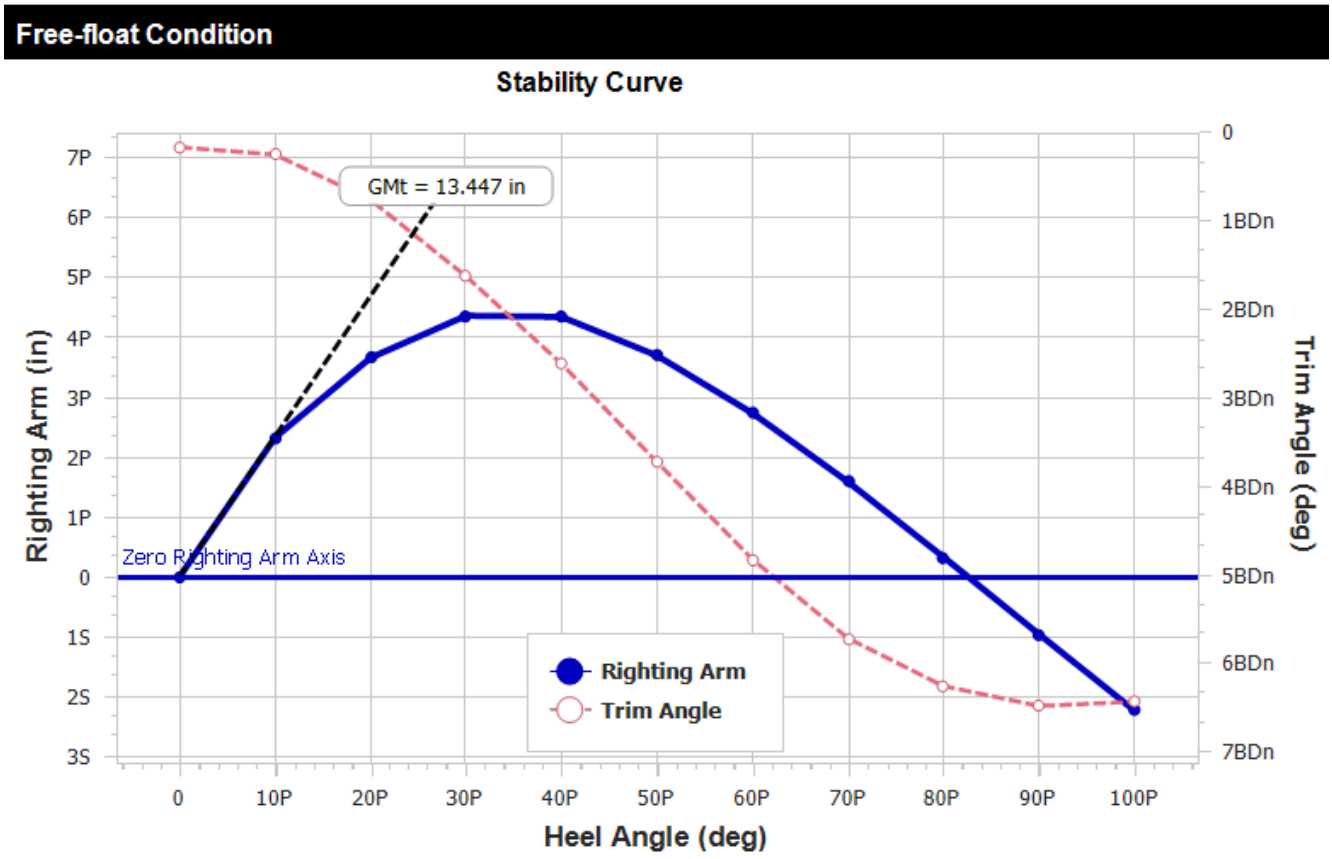


Figure 12: Updated stability curve

Hull Design Motivation

The main hull is designed to maximize the length of the waterline (LWL) to help maximize the hull speed and minimize the Froude number. The hull speed does not dictate the absolute maximum speed the vessel is able to travel at, but rather the speed at which the vessel will start to ride its own bow wave. This speed should not be exceeded to help minimize trim angle to ensure the camera stays as level as possible while navigating autonomously. With an overall length of 60.390 in and a LWL of 56.606 in, the LWL is 93.734 % of the overall length. This corresponds to the main hull acting as a displacement hull at operating speeds, as shown by a Froude number of $Fn = 0.399$ which is just within the boundary of a displacement hull ($Fn < 0.4$). As this Froude number was calculated using the hull speed, it is important to consider the expected speed during autonomous runs to be around $u_{hull} = 0.5m/s$. The Froude number at this speed is $Fn = 0.13$, which shows the design goal was achieved and the main hull will act as a displacement hull to maximize stability.

To achieve a sufficiently high measure of rotational stability, the width of the main hull as well as all three dimensions of the ama were concurrently iterated upon to identify an optimal balance between weight and stability. Additionally, the lateral spacing between the amas as well as the width of the main hull has a significant influence on stability, as a wider separation increases the generated righting arm.

Several geometric modifications were incorporated into the hull design to improve stability, including the addition of shoulders to increase resistance to heeling and an increase in the main hull width to accommodate these features. These geometric changes directly influenced the vessel's hydrostatic characteristics and stability behavior.

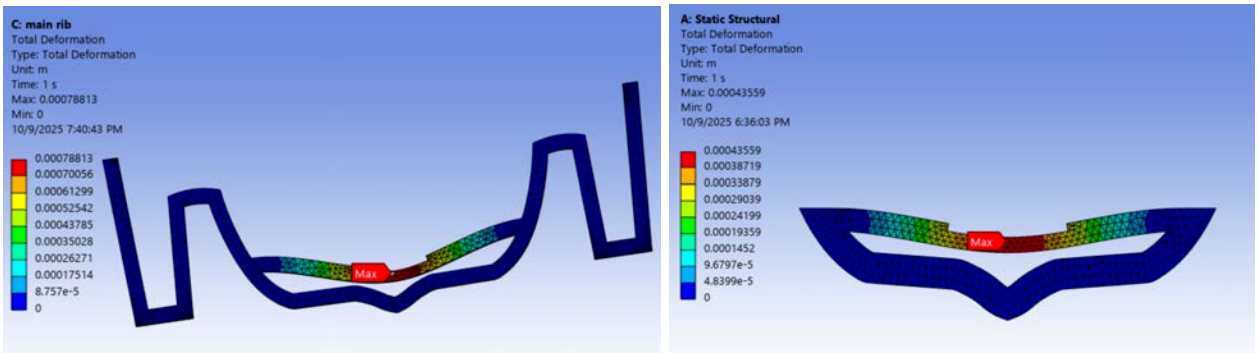
Hull speed: $u_{hull} = \sqrt{\frac{LWL \times g}{2\pi}} = 1.498\, m/s$

Froude number: $Fn = \frac{u_{hull}}{\sqrt{g \times LWL}} = 0.399$

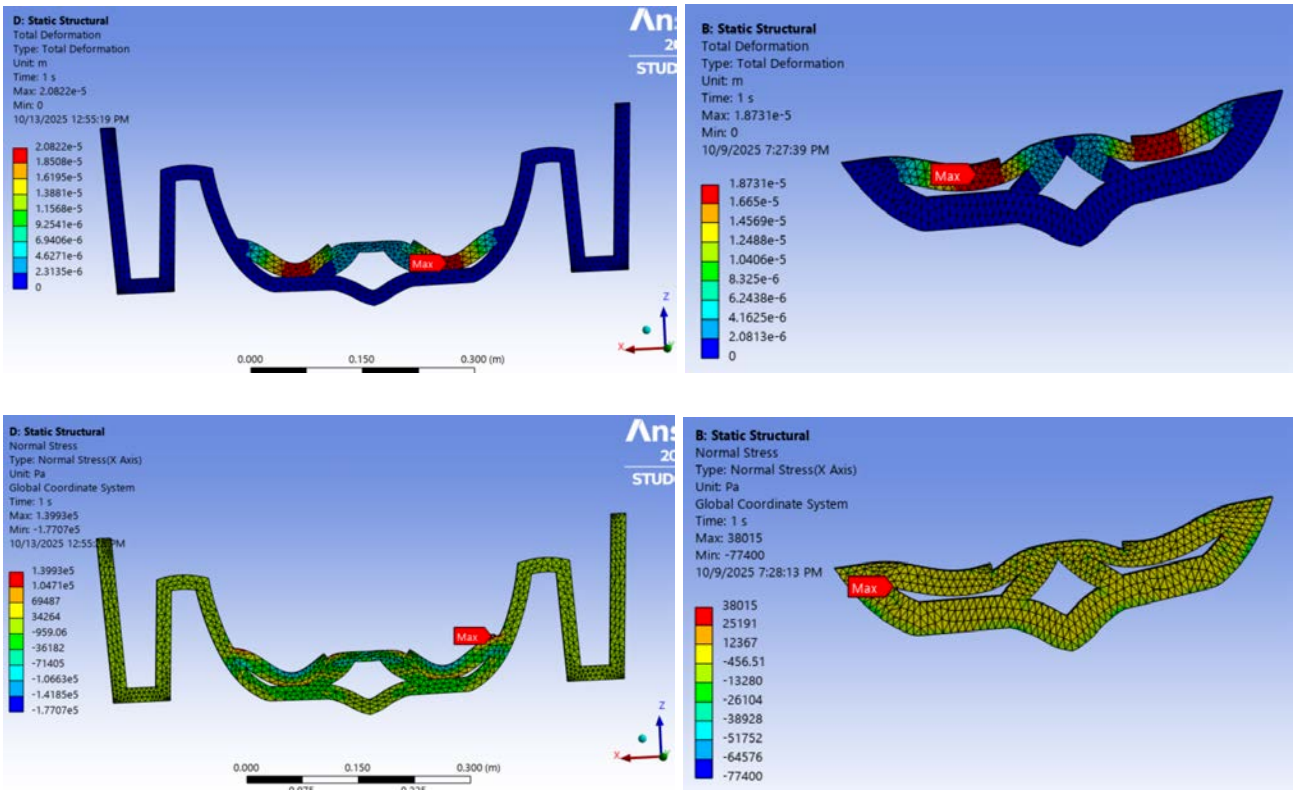
Finite element analysis for loaded components

The internal rib and structural system of the hull was analyzed using Ansys Static Structural to evaluate deflection and strength under operational loading. The ribs were modeled using polyethylene material properties, with fixed boundary conditions applied at the interfaces between the main hull and amas. A vertical load equivalent to a 36 N force, representing the weight of onboard components with an added safety margin, was applied downward on the structure.

Two configurations were evaluated: ribs without a truss and ribs reinforced with an internal truss. In the non-truss configuration, the maximum deflection occurred at the main rib and small ribs, with peak values of approximately 0.7 mm and 0.4 mm, respectively. When the truss was added, maximum deflections were significantly reduced to approximately 0.028 mm at the main rib and 0.018 mm at the small ribs. The truss-reinforced configuration exhibited sufficiently high high factors of safety.



Figures 13(a-b): *Finite element analysis of the internal rib and hull support structure without truss reinforcement, showing total deformation under applied loading and highlighting maximum deflection locations.*



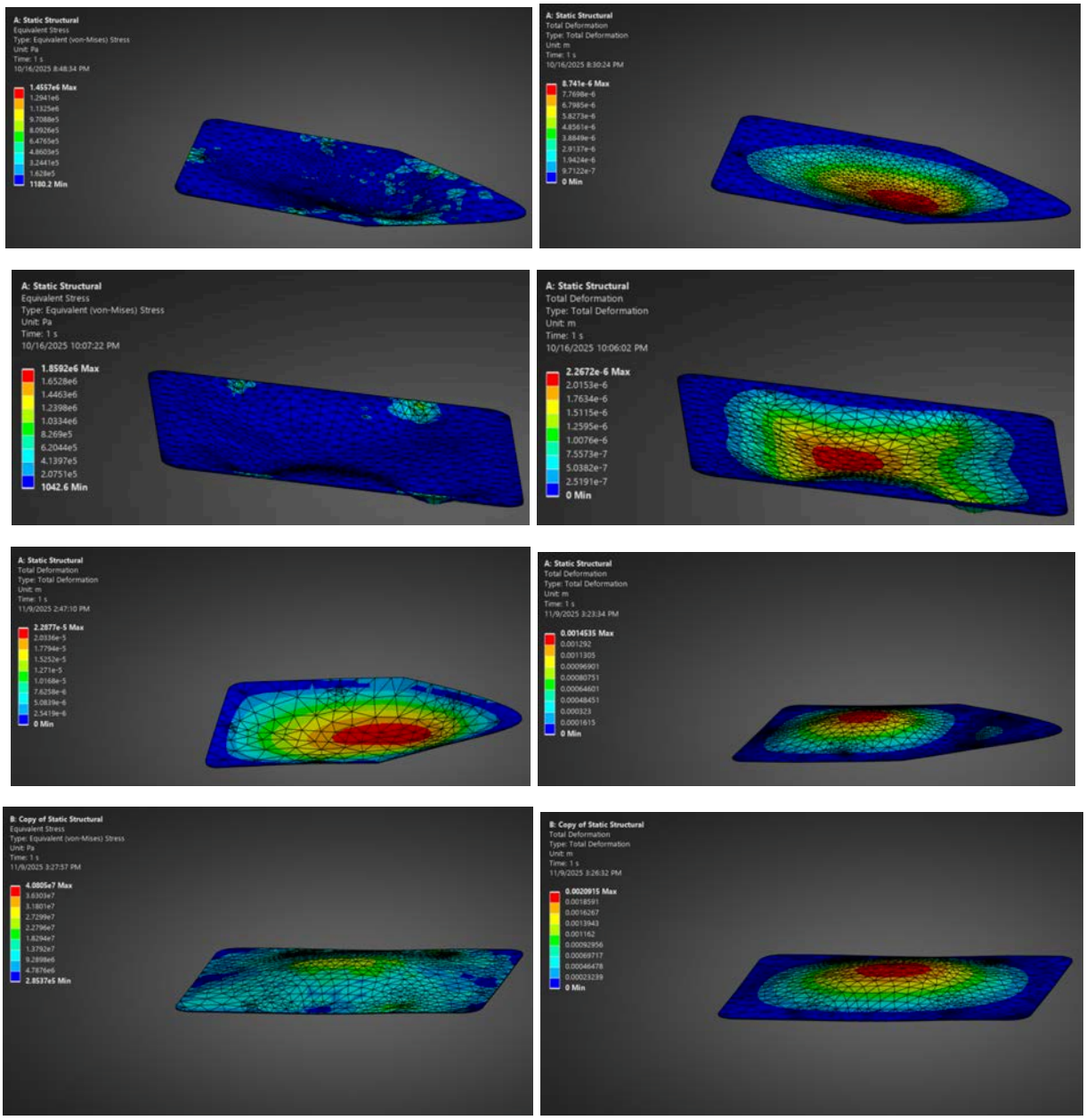
Figures 14(a-d): *Finite element analysis of the internal rib and hull support structure with truss reinforcement, showing reduced total deformation and normal stress, with an increased factor of safety under the same loading conditions.*

New Hatch Design

The custom sealing and U-channel were iterated and improved upon, with new shape matching the shape of the hull itself. The hatch is elevated from the deck creating a barrier for water entry, which was an issue at competition previously. Compact hold down toggle clamps were chosen to seal the hatch due to the clamp's small footprint on the deck and adequate force to maintain a seal.

The new hatch lid must have a high flexural strength to ensure that a proper seal is made with the weather strip while using the minimum number of clamps needed to apply a consistent pressure on the lid. The material chosen for the lid is 18 gauge 7075 T6 aluminum. The lid will be water jetted to fit the hatch shape from Xometry.

The main hatch was simulated to ensure that it doesn't fail under loading from the clamps that secure the cover down on top of the waterproofing gasket. The force required to adequately compress the gasket was determined based on o-ring material loading tables. This force was equally split between each gasket and applied in Ansys as a distributed force over a surface area equivalent to the surface area of each clamp. All of the simulations had the loading forces from the hatch clamps at 150 lbs of force and a static structure at the contact with the weather strips. The simulations for forces on the handle had a loading of pulling the handle up with 35 lbs of force with the clamps clamped. The FEA for lid deformation due to the clamps showed nominal displacement for both hatches with the yield convergence of 1.45 MPa for the front hatch and 1.86 MPa for the rear hatch (Figures 6-9). The FEA for the lid deformation from pulling on the handle was minimal (0.001m) [13].



Figures 15(a-h): FEA simulation for the hatch lid

IX. Appendix B: Manufacturing Methodology

<u>Center Hull, Frame & Deck</u>

In order to provide a higher level of surface smoothness, a female mold was integrated for the first time. The bottom of the hull mold was constructed from CNC-machined high-density foam, while the upper half was completed usings 3D printed compliments generated from the CAD model. Mold sections were aligned, glued, and clamped to ensure accuracy and stability during manufacturing. Once assembled, the surface of the mold was sanded to remove any scalloping from the CNC. The completed mold was then fully covered with Tyvek tape to serve as the release layer, with additional tape overlapped along seams and over the edge of the mold to help with demolding.

Figure 16: CAD model of the assembled female hull mold (left) and the physical mold setup with the CNC-machined high-density foam base and 3D-printed top section covered in Tyvek release tape prior to fiberglass layup (right).

The layup process began with a thin layer of marine epoxy brushed onto the Tyvek tape followed by layers of mat, weave, then mat fiberglass. Wooden blocks with holes for the thruster mounts were placed in the amas and were covered with a layer of mat fiberglass and epoxy to seal the wooden blocks into the amas. An additional fiberglass layer was applied as reinforcement to high stress areas, like the keel, bow, stern, and ama attachment points.The wooden ribs were then installed into the hull through fiberglass tabbing. The hull was then removed from the mold. The outermost layer of fiberglass was found to have not fully cured after over 48 hours so any uncured areas of the mold were removed and replaced with a new layer of fiberglass and marine epoxy. Any excess fiberglass that was over the edge of the mold was removed using a dremel and was then sanded to a smooth and flat finish. The deck was then sanded and installed into the hull through fiberglass tabbing.

Figure 17: Hull post-processing and structural integration stages, showing the demolded fiberglass hull prior to finishing (left), laser-cut Baltic birch ribs (center), and the assembled hull with installed ribs, middle beam, and deck openings before final surface finishing (right).

The hull was then sanded and a layer of fairing compound was applied to the exterior of the hull to smooth out the surface by filling small divots to reduce drag in the water. Two layers of barrier coat

were then applied followed by three layers of gel coat with the hull being sanded in between layers to increase adhesion, hull smoothness and waterproofing. Once fully cured, the hull and deck were spray painted.

Figure 18: Exterior surface finishing stages of the hull, showing the application of epoxy fairing compound to smooth surface irregularities (left) and the subsequent epoxy barrier coat applied to seal and protect the composite structure.

X. Appendix C: Robotics Systems Iteration

SkeeBall Launcher

The SkeeBall launcher went throughTesting of the SkeeBall shooter identified excessive vibration and launch inconsistency originating from drivetrain alignment sensitivity and fastening methods. In response, we implemented a mechanical redesign focused on improving robustness, alignment repeatability, and resistance to assembly-induced variation.

Key design changes included eliminating heat-set inserts, which introduced alignment uncertainty during installation, and replacing them with more reliable mechanical fastening strategies. The drivetrain geometry was refined to improve shaft alignment and reduce vibration transmission, while the rigid machined aluminum chassis was retained to counteract motor-induced loads.

In parallel, the feeder mechanism is being developed to ensure controlled and repeatable ball delivery into the launcher. Current work focuses on integrating the feeder with the redesigned drivetrain to enable consistent autonomous firing performance under competition conditions.

Figure 19: CAD model of Skeeball shooter.

Microphone Task Switching System

For task six, we rapidly developed and integrated a new sensing system. Utilizing a RP2040 and microphone, the system is able to identify target frequencies under operating noise conditions and interface with the ROS framework using a custom STFT firmware version of Micropython + uLab [13] [14] [15]

XI. Appendix D: Software System Diagrams

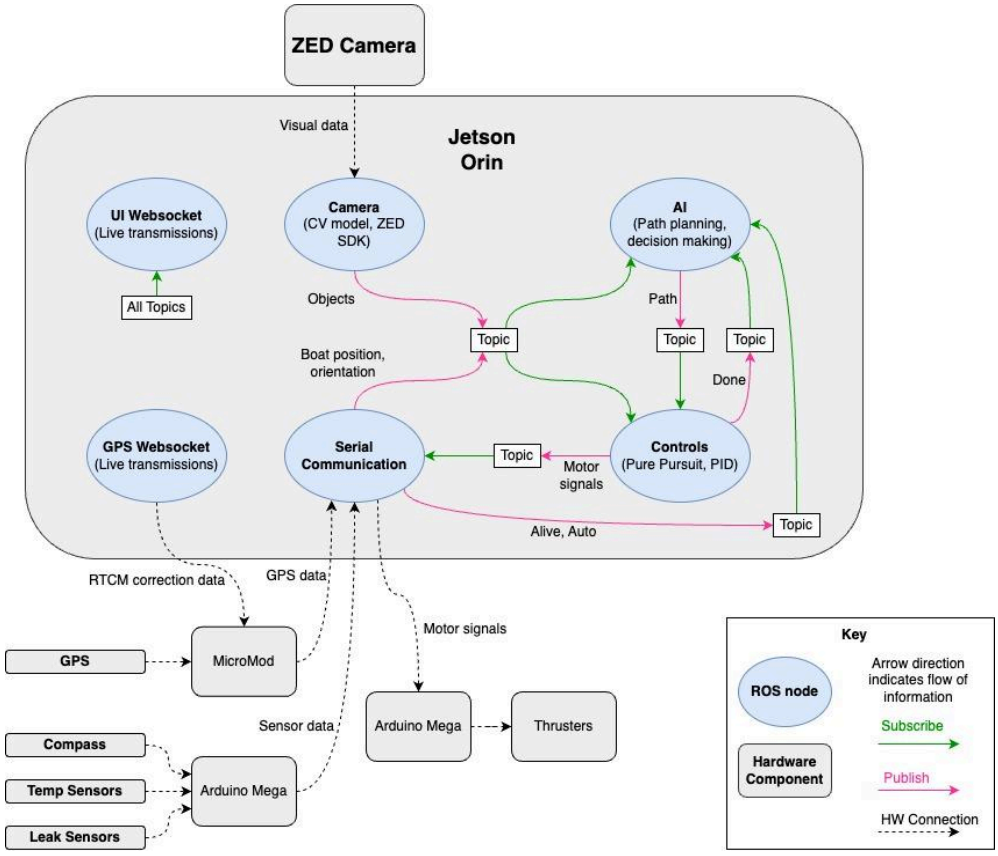


Figure 20: Publisher/subscriber node structure of the ROS framework, showing intercommunication between hardware and software systems, including sensors.

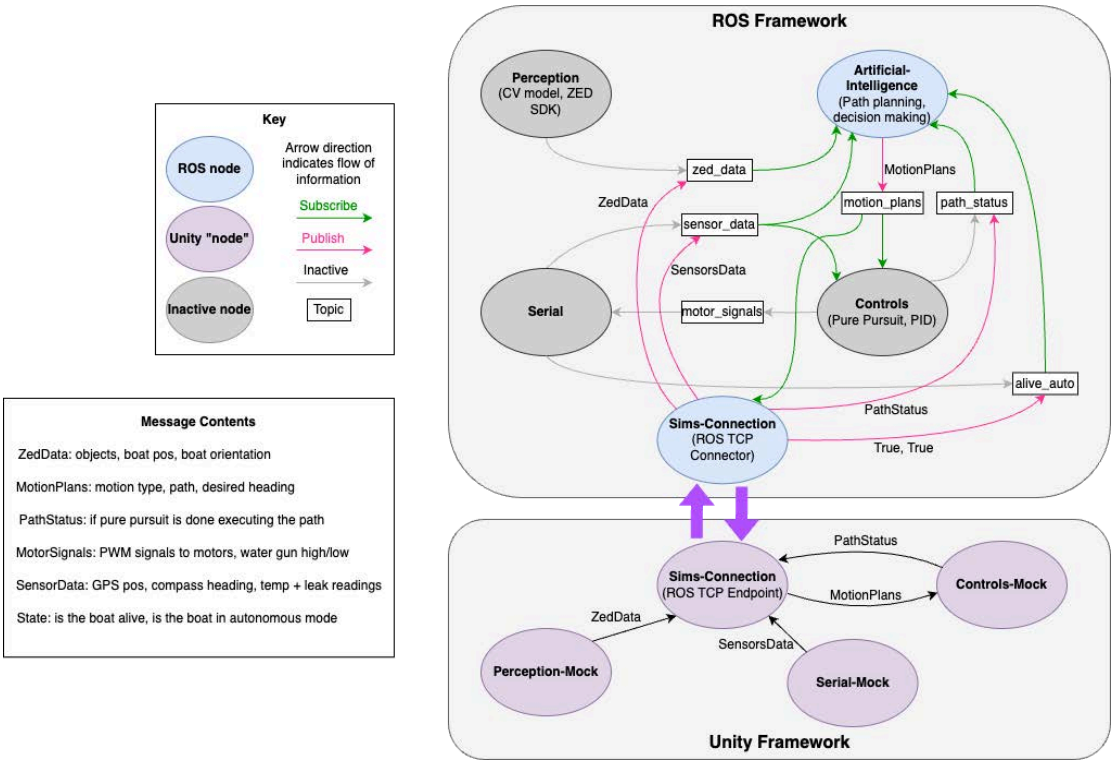


Figure 21: ROS framework contextualized by Unity simulations framework, with mocked message data sent across TCP.

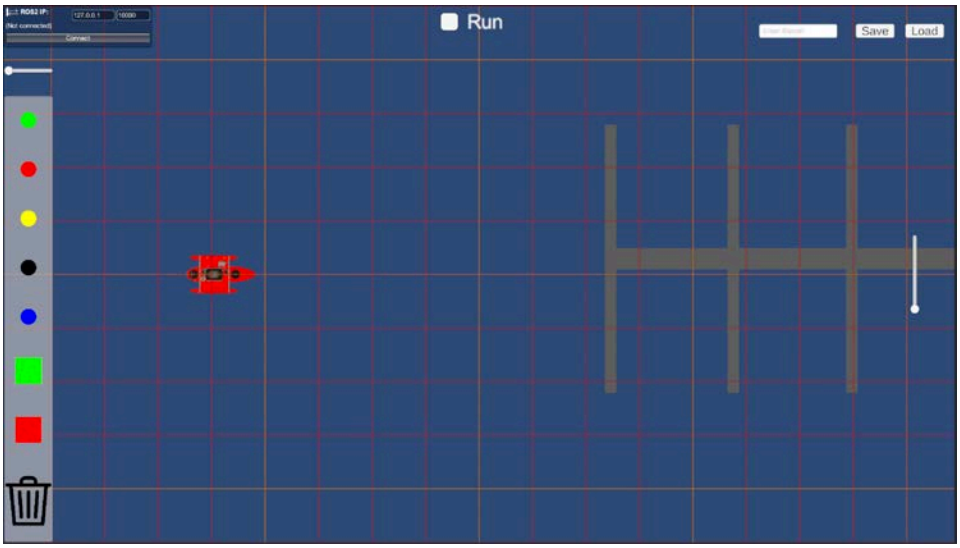


Figure 22: Simulations map builder interface.

XII. Appendix E: Electrical Schematics

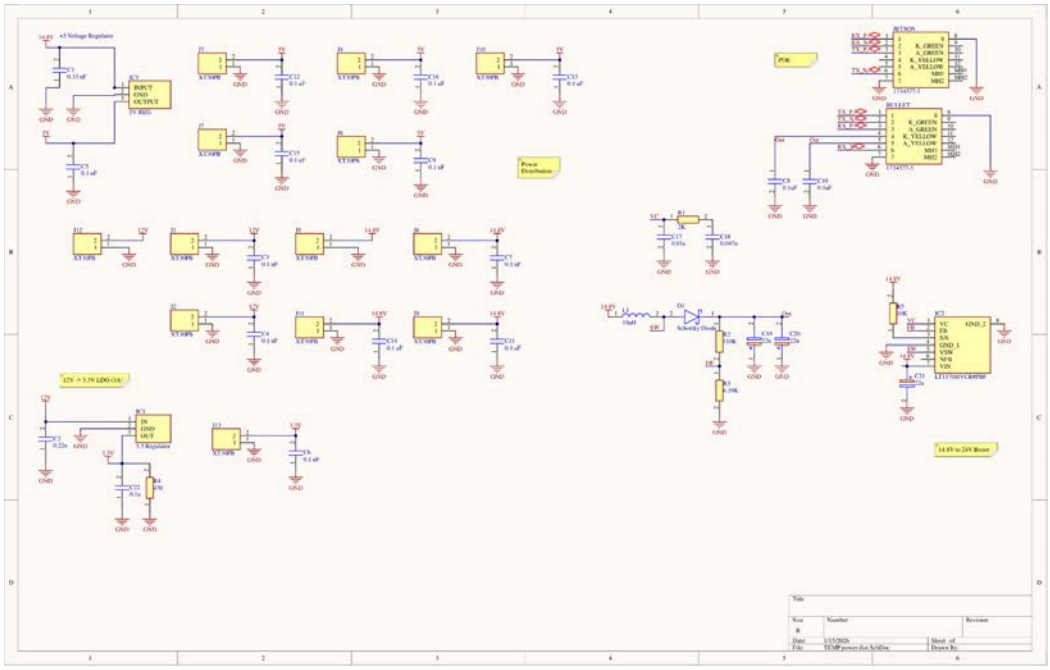


Figure 23: Power distribution board schematic

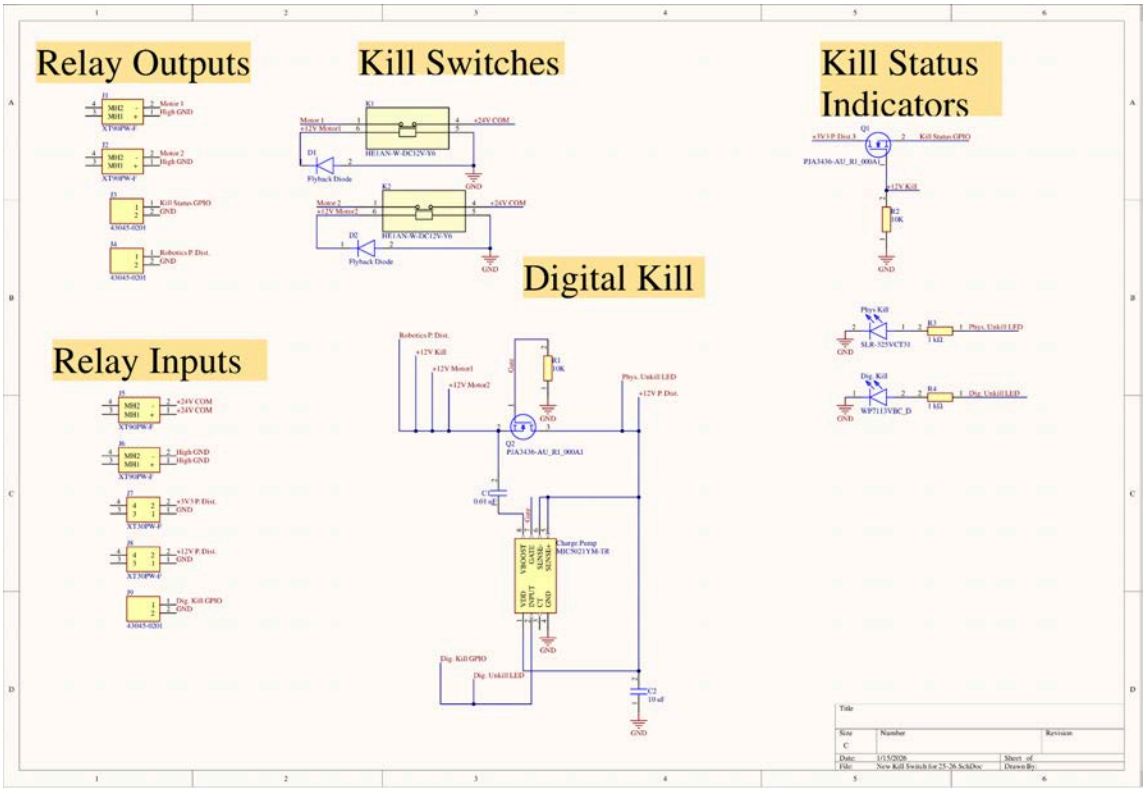


Figure 24: Kill Switch board schematic

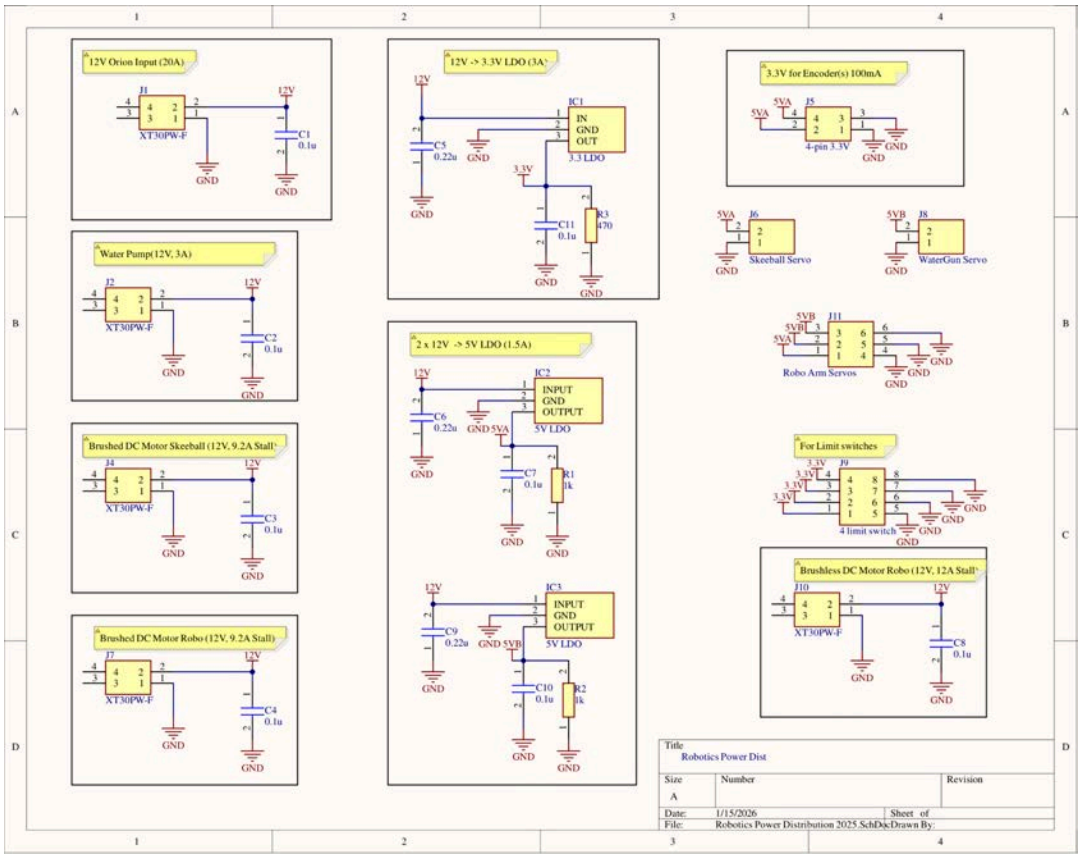


Figure 25: Robotics Power Distribution Schematic

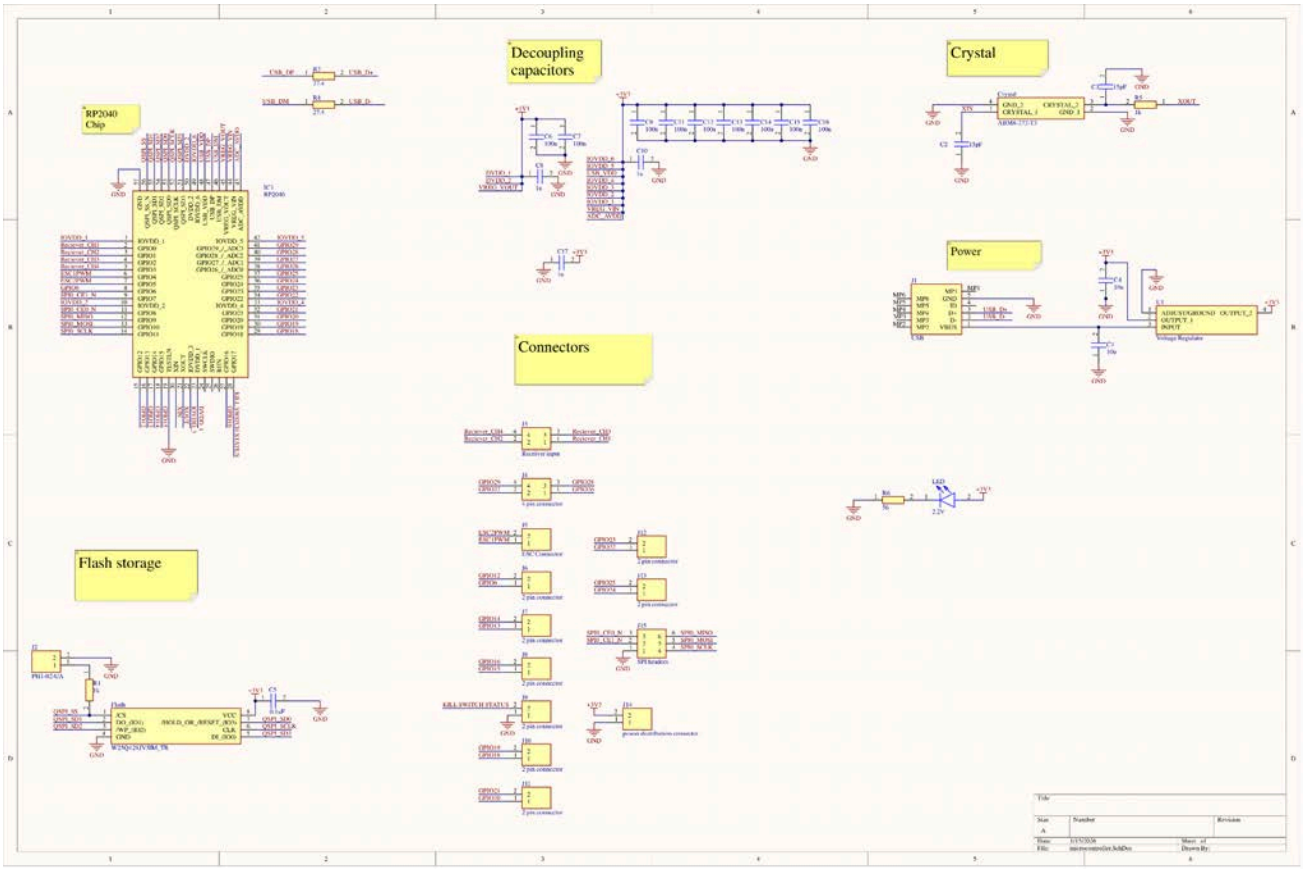


Figure 26: Microcontroller board schematic

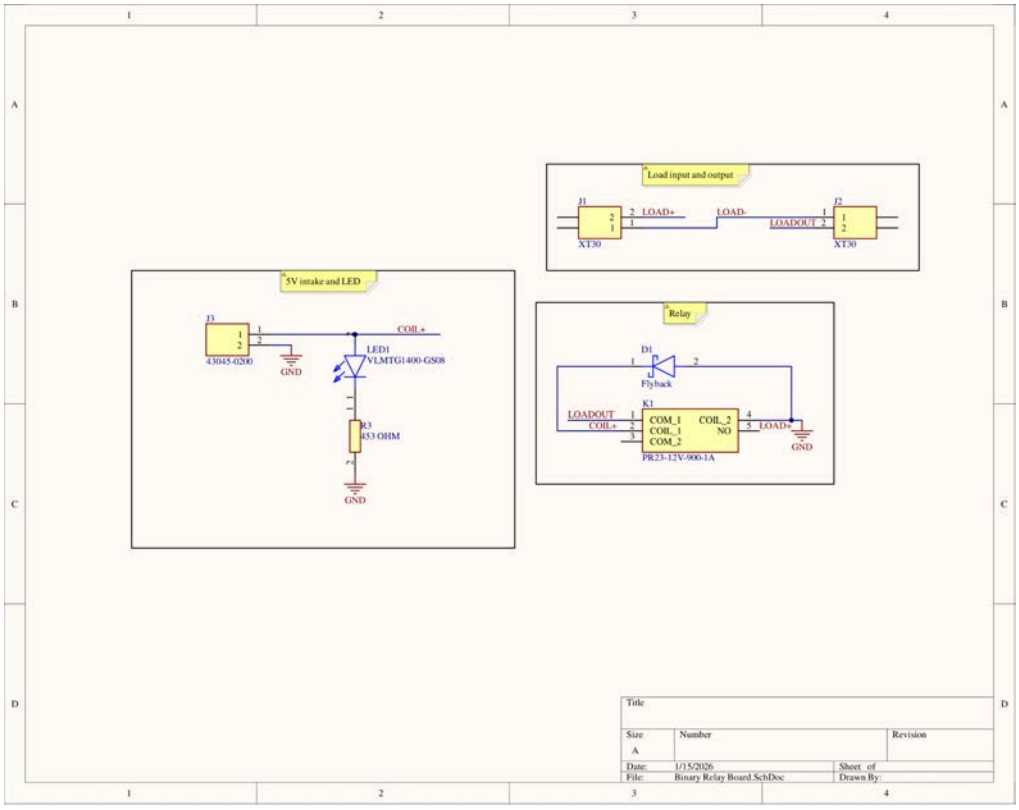


Figure 27: Robotics Relay board schematic