Robosub 2026 Technical Design Report

Cornell University Autonomous Underwater Vehicle Project Team
Alex Song, Angel Sumimoto, Angela Liu, Ani Velaga, Anthony Fei, Bobby Henkelmann, Catherine Seo, Claire Moon, Coby Lai, Colby Bittner, Connor Lynaugh, Daniel Zlotnick, Dhruv Mandalik, Eli Pate, Emily Hinge, Emma Camacho, Grace Young, Hannah Jansen, Henry Yoon, Isabella Frank, Isabelle Schwartz, Jayden Lim, Jonathan Song, Joseph Cobb, Joshua Kwakye Minott, Junayed Ridwan, Kurato Inaba, Kyle Du, LeAnn Zhu, Leila Boesch-Powers, Leo Zhi, Linda Gu, Michelle Chiu, Namitha Devulapalli, Nathaniel Banjo, Rayanu Adam, Richie Xue, Rostyslav Dashkiiev, Samantha Muller, Samuel Jin, Sonja Wong, Swati Sriram, Vaishnavi Vednere, Wali Afridi, Wei Ouyang, Yvonne Bao

***Abstract*—CUAUV presents Scylla and Orion 2.0 for RoboSub 2026, a dual-vehicle system designed to increase task reliability through specialization and parallel execution. Orion 2.0 builds on a validated platform through incremental upgrades. Scylla introduces a new cylindrical, flat-profiled architecture to address previous limitations in enclosure complexity, vehicle balance, and precision manipulation. The electrical system then strengthens robustness through upgraded telemetry, battery management, power distribution, and STM32-based hydrophones and transmit systems. These acoustic systems enable pinger localization and inter-vehicle communication, allowing task allocation to adapt based on mission progress. Software addressed task robustness and testing-time limitations with an upgraded simulator, an in-house telemetry pipeline for post-run analysis, depth aware perception, and optimized controls. By combining a proven vehicle with a new precision-focused platform, our team aims to maximize task attempts while maintaining reliable autonomous operation under competition conditions.**

I. Competition Strategy

A. Core Strategy

Our competition strategy is built around specialization and parallel task-execution. The team operates two specialized vehicles in tandem: Scylla is optimized for precision-oriented, near-field tasks that require stable positioning and manipulation, whereas Orion is optimized for high mobility tasks, supported by extensive pool testing and proven dynamic control. We will be pre-allocating tasks with respect to these strengths to maximize task reliability, while duplicating attempts whenever sensible to maximize success. We further leverage the dual-vehicle strategy via Inter-Vehicle Communication (IVC), made possible with improvements to acoustic hardware. Task order can now be modified at runtime based on pinger localization, task completion messages, and timeout fallbacks. In particular, we use IVC to coordinate the random-pinger branch, allowing one vehicle to begin the second pinger task after the first vehicle reports completion or timeout.

This combination allows for parallel task execution, redundant attempts at critical tasks, and flexible task allocation catering to each vehicle's strengths, enabling the team to improve task reliability while minimizing total runtime.

B. Course Strategy

1) Begin Assessment (Gate): Orion and Scylla begin their missions by passing through the gate consistent with the coin-flip role assignment. Each vehicle first identifies the desired image class using YOLO, made robust with software's caustic-removal algorithm. After centering the target in image space, both vehicles drive through the corresponding side of the gate. In this process, Orion 2.0 attempts to earn style points by performing two 360-degree roll maneuvers, stabilized via control stack improvements and Orion's dynamic maneuverability.

Fig. 1: Orion AUV.

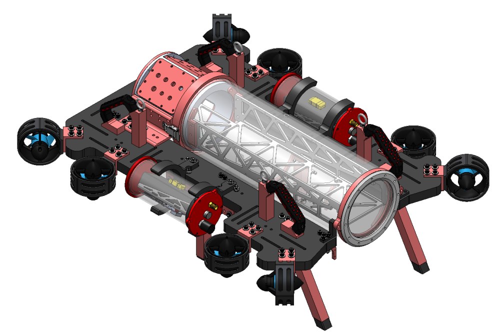


Fig. 2: Scylla AUV.

2) Avoid Debris (Slalom): Only Orion will attempt Slaloms, as Scylla's long frame poses a risk in navigating tight spaces. The vehicle will utilize our 7-D Yolo model's RGB and depth channels to geometrically differentiate pipes at different distances. Once every target red pipe is in view and identified, Orion performs the TARZAN maneuver, centering on the closest pipe and swinging around each successive red pipe with a fixed radius; the direction of swing is matched to the side of the gate the vehicles entered through.

3) Recon (Bins): This task will be attempted by both Orion and Scylla, albeit in varying manners. Scylla will perform a precision approach to take advantage of its dropper-dedicated camera, performing the established YOLO detection-and-centering scheme to steadily align its drops and maximize success probability. Orion, on the other hand, will perform a flyby operation, performing overhead drops while moving past the bins. Given Orion's limited hardware, this compromise offers a quick and low-risk attempt at scoring extra drops to count towards the 2-markers cap, buffering any potential perception or actuation failures that could occur on Scylla's end.

4) Deploy (Torpedoes): This will be attempted by Orion, given its success last year and its larger camera, which provides more accurate and reliable normal readings. Once the board is in range, we use normals to estimate its orientation and rotate the vehicle until it is perpendicular to the target plane, improving shot accuracy. After the first fire, the vehicle repeats the alignment and firing sequence. This sequence allows us to fire torpedoes through the correct sequence.

5) Resupply (Octagon): This task is assigned to Scylla as it is optimized for high-precision tasks. We use YOLO to target and align with both the objects and bins. The mission planner also includes a fallback behavior, where if the manipulator is unavailable or the pickup sequence fails, the vehicle will skip directly to surfacing within the octagon.

6) Return Home: The Return Home task may be performed by either vehicle and is dynamically assigned based on pinger branch using IVC. For this task, we use a hybrid dead-reckoning and vision-based approach for efficiency and reliability respectively. After dead-reckoning to the approximate region (location saved from mission-state history), it performs a 360-degree search to reacquire the gate visually before driving straight through.

C. Inter-Vehicle Communication

The random pinger task is conventionally performed by a single vehicle and the initial location of the random pinger is a dynamic environment variable, both of which are factors that could hinder our dual-vehicle coordination strategy. In response, we present an IVC based approach that enables us to split the random pinger task between two vehicles and dynamically replan the mission to maximize efficiency (Appendix P).

D. Trade Offs Between Complexity and Reliability

Consistent with CUAUV's long-standing design philosophy, the team prioritized reliability across both fully capable vehicles. Our team size enables parallel development: Orion 2.0 continues to serve as a proven competition platform with incremental updates, while Scylla introduces a new mechanical architecture intended to address limitations from previous design iterations. This added complexity creates risk by departing from a platform the team has already validated, but it also offers long-term reliability and performance benefits by addressing recurring mechanical limitations, including difficulty tuning buoyancy and control challenges.

To balance this tradeoff, the team preserved proven subsystems wherever possible. The torpedoes and droppers were retained from a successful previous design, the manipulator was designed to be swappable, and most boards remain compatible with both submarines. This approach allows Scylla to validate a new mechanical design without requiring every subsystem to be redesigned.

The team also managed the risk of developing a new sub on a short timeline through structured design reviews. We incorporated design for manufacturing and assembly (DFMA) principles, allowing us to improve access and maintainability, especially since new members often represent the team at the competition. Before fabrication, major design decisions were evaluated using failure mode and effects analysis (FMEA) to identify potential failure modes, particularly leakage. These reviews were especially important because consolidating most boards into a single pressure vessel increases the consequence of a single leak.

As a result, Scylla accepts calculated short-term design complexity in exchange for long-term mechanical improvement, while preserving Orion as a dependable backup with additional software testing time. Together, both vehicles remain versatile platforms designed to attempt as many tasks as possible.

II. Design Strategy

A. Mechanical Systems

1) Main Pressure Vessel: Reflecting the competition strategy, Scylla was designed with manufacturability and reliability as key priorities. To meet these goals, the mechanical team focused on further reducing the number of enclosures on the submarine. On Orion, each board and enclosure had different placement constraints, contributing to the vehicle's asymmetry. Scylla directly addresses this challenge by housing all electronics in a single waterproof pressure hull, except the battery pods (See Appendix J for electrical boards housing layout). This is the first year our team has attempted this level of enclosure consolidation, and the resulting symmetries (Table III) are promising for improvements to control efficiency.

This choice of mechanical design also reduces potential failure points and cost, with the stock cost of Orion's enclosures (over $2000) eclipsing the entire cost of the MPV (Appendix B). Because each additional enclosure introduces another component that must be leak tested and manufactured, reducing the number of enclosures shortens the integration and validation process. Therefore, instead of using separate enclosures for the transmit and hydrophone boards, Scylla's main cylindrical pressure hull was designed to house both systems along with the other electronics boards.

2) Frame: Scylla highlights an innovative structure with a main focus on positive buoyancy and weight optimization, diverging from Orion's lunch-box design developed over the last two years. High-density polyethylene (HDPE) is a durable, lightweight, and corrosion-resistant material, which justifies its selection for Scylla's main board [6]. Importantly, the material is much closer to neutral buoyancy than Orion's aluminum frame (Appendix K), yet maintains enough structural integrity to support all mounted components and operating conditions (Appendix L). The sheet is also easily manufacturable with a waterjet, avoiding outsourcing complex parts that previously introduced bottlenecks in Orion's development. As a result, Scylla is 20N positively buoyant—matching our calculated expectations (Appendix N)—with slots on the sheet for adjustable weight placement. By reducing the need to increase the vehicle's buoyancy using foams that are voluminous and easily deformable, the design improves maintainability.

Given Scylla's role as the precision-task platform, most of Scylla's weight was placed along the horizontal plane, intended to stabilize depth fluctuations via high heave drag while allowing low-friction locomotion in surge and sway. The design reduces the vehicle's drag by around 20% in the mission-critical horizontal plane (TABLE II). Not only are all enclosures that contribute to most of Scylla's weight mounted on the same plane, but also the thrusters are on the same plane. These thruster positions are intended to reduce any unwanted rotational torque as the vehicle translates.

TABLE I: Lumped drag coefficients from SolidWorks Flow Simulation at 0.3 m/s.

Axis	Orion	Scylla	Change
Surge (X)	0.1034	0.0762	−26.3%
Sway (Y)	0.1144	0.0955	−16.5%
Heave (Z)	0.1878	0.3288	+75.1%

$C_D A$ values are reported in m^2 at 0.3 m/s.

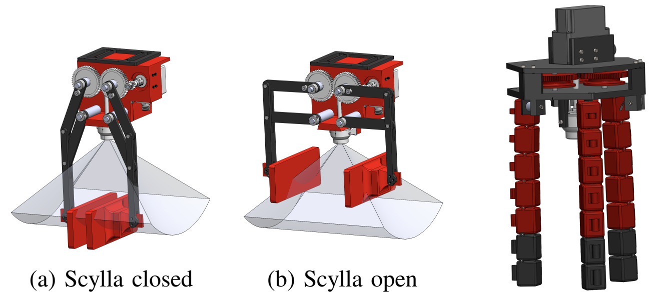
TABLE II: Cross-axis drag coupling coefficients from SolidWorks Flow Simulation at 0.3 m/s.

Coupling	Orion	Scylla	Reduction
Surge → lateral (F_y)	0.00228	0.0000258	88.3×
Sway → surge (F_x)	0.00164	0.00119	1.4×
Heave → sway (F_y)	0.01259	0.00572	2.2×

$C_D A$ values are reported in m^2 at 0.3 m/s.

Below the main sheet, Scylla features six leg pieces that provide stable ground contact. This design helps prevent the sub from tipping over while eliminating wire-pinching issues seen in the previous design and preserving unobstructed space for the grabber during object manipulation tasks. These fewer, yet rigid supports, in contrast to Orion, also enhance access to the mechanisms beneath the main sheet during maintenance, reducing time constraints when issues arise at competition. This focus on accessibility also guided the design of the battery pod mounts. Since we aim battery pods to be easily hot-swappable, they are secured with 3D-printed elastic clamps.

3) Foldable Thrusters: Alongside Scylla's novel mechanical architecture, Orion 2.0 was also improved through the addition of foldable thrusters. In previous years, vehicles' depth thrusters were constrained to a horizontal spacing of 25 inches to fit within the testing environment of the team. However, increasing the lateral distance between the depth thrusters provides larger moment arms, allowing the vehicle to generate greater roll torque when Orion barrel rolls for style points. To balance these competing constraints, Orion 2.0 uses an adjustable foldable thruster mount that increases roll torque by a margin (Appendix M).



(a) Scylla closed (b) Scylla open (c) Orion manipulator

Fig. 3: Manipulator designs for Scylla and Orion.

4) Manipulators: Orion and Scylla are both designed with a unified torpedoes/droppers mechanism attached below the front of the vehicle and a manipulator at the rear. This mechanical layout choice is necessary for convenient torpedo alignment using frontal vision and complete the task with a minimal number of servos. However, this layout presented a challenge as its single downward-facing camera offered a significantly limited field of view for Orion. The camera's failure to properly aid the operation of both the dropper and manipulator mechanisms was a major barrier to the completion of the Octagon task last season. In response, we have decided to include two downward-facing cameras positioned directly above each mechanism for Scylla. To prevent overcrowding in this process, we transitioned from the space-exhaustive camera housing system to modular waterproofed cameras. The shift added the benefit of reducing the number of enclosures needed for the sub, decreasing the manufacturing time associated with enclosures.

To support the fail-safe approach, Orion features a three-leg manipulator inspired by the motion of octopus arms. Each leg retracts through constant-force springs, while cables routed from gear spools drive segment extension. This mechanism addresses the gripping difficulty observed in last year's design by reducing the need for continuous actuator force once an object is captured.

Meanwhile, Scylla's manipulator prioritizes adaptability through a grabber with a wide actuation range of 1.3 to 6.1 inches, providing flexibility for game objects that are typically released late. For finer adjustment, the modular grabber allows different claw geometries to be 3D-printed and tested for specific objects without re-manufacturing the entire mechanism. These diversities in manipulator designs allow quick adaptation during competition according to the game elements the vehicle needs to grab.

B. Scylla Electrical Systems

The team continued developing the electrical architecture of previous vehicle iterations while preserving infrastructure that had proven reliable. Rather than completely redesigning the vehicle's electrical stack, this year's effort focused on improving reliability, reducing unnecessary subsystem redundancy and enclosure complexity, and expanding telemetry on our custom PCBs.

1) Maintaining the Modular Backplane Architecture: The backplane infrastructure and PCIe board connection system has been retained from previous vehicle generations due to their proven modularity and maintainability during pool testing. This architecture continues to allow subsystem boards to be quickly swapped and independently debugged without major rewiring or vehicle disassembly (Appendix C).

2) Battery Management Board Reliability and Telemetry Improvements: The Battery Management Board (BMB) was updated to improve reliability and onboard telemetry capabilities. In earlier iterations of the board, onboard components often failed due to mishandling and age. Through the addition of humidity sensing, the replacement of several ESD-sensitive components, and the removal of redundant circuitry, these issues have been resolved.

3) Hydrophones System Redesign: The Hydrophones system was completed this year using a new STM32-based architecture rather than the ATxmega32A4U microcontrollers traditionally used throughout the vehicle (Appendix D). The board integrates bandpass filters, synchronized ADC sampling, and onboard signal processing to estimate acoustic pinger direction through phase difference calculations across four hydrophones mounted in an array on the submarine[9]. This improvement may also indicate a transition toward STM32-based PCBs, which could potentially streamline the stack in future vehicle iterations.

4) Power Distribution Board Simplification: The Power Distribution board was redesigned to remove outdated circuitry and redundant voltage output channels from previous revisions. This improved architecture has current monitoring on individual channels, and the subsystem toggle-ability, which advanced the system to be more reliable.

5) Actuation Systems Improvements: The Actuators Board was updated this year by integrating current sensing, enable logic, and protected actuator channels to improve robustness during manipulator and torpedo/dropper tasks. This was a needed upgrade, as the previous revision of the Actuators Board had displayed inconsistent behavior. This board has been much simpler to debug and has exhibited much more consistent behavior.

C. Orion Electrical Systems

Orion 2.0 continues to use the modular electrical architecture developed over previous competition seasons while focusing on reliability and acoustic system compatibility.

1) DC-DC Board: One major redesign effort centered on the DC-DC board, which had previously exhibited over-current issues during pool testing and extended operation. This year's revision implements updated onboard components and a more conservative layout, improving the robustness and stability of its output voltages.

2) Hydrophones Integration: As previously mentioned, the Hydrophones board has proven to be a challenge over the years. This year's design, which was fully redesigned around a new STM32-based architecture, was not only functional but significantly more modular than previous iterations. Earlier acoustic systems had limited integration capability, but the new Hydrophones board was able to connect directly to Scylla's backplane while also integrating into Orion's hydrophones enclosure.

D. Software Systems

The 2025–2026 software stack was developed with the main goal of improving robustness under real-world conditions, while increasing the speed and safety of development and deployment. Rather than broad rewrites of existing systems, we prioritized targeted upgrades driven by recurring issues seen during pool tests and competition, such as lighting-related perception failures, limited camera throughput, nonlinear vehicle dynamics and slow debugging cycles.

This approach resulted in a tightly integrated set of improvements across vision, controls, infrastructure, simulation and developer workflows. These systems were designed to reinforce one another, where improved camera synchronization enables multi-modal perception, improved state estimation and controls increase mission robustness, and improved infrastructure accelerates iteration and deployment.

1) Perception: Our vision system improvements this year focused on improving robustness in visually unstable underwater environments. Specifically, caustics, the high-frequency brightness patterns formed by sunlight reflecting through the water surface, were a significant source of false positives and detection instability. To address this, we developed a caustic suppression pipeline using exponential moving average temporal filtering, depth-aware alignment, and motion-based optical flow algorithms inspired by prior work on motion-aware deflickering [1]. As temporal filtering on a moving vehicle introduces motion blur, we introduced motion-aware filtering that dynamically changes the moving average filter based on vehicle velocity calculated by the Kalman state estimator.

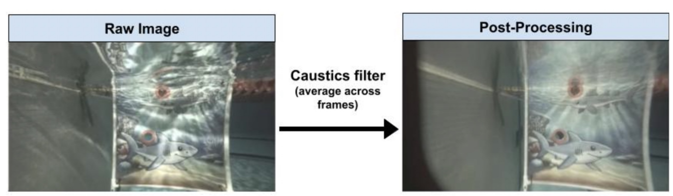


Fig. 4: Example of removing caustics from the torpedo board.

Building on this, we developed YOLO-7D, a custom 7-channel Oriented Bounding Box detection pipeline that incorporates depth and surface normal data on top of regular RGB input channels. This allows the model to use geometric cues that are less sensitive to lighting variation, improving detection accuracy by approximately 20% over RGB-only detection [2], [3]. To support this workflow, we transitioned from RoboFlow to Label Studio as our main annotation software, giving us more control over the annotation of synced frames, dataset management and training iteration.

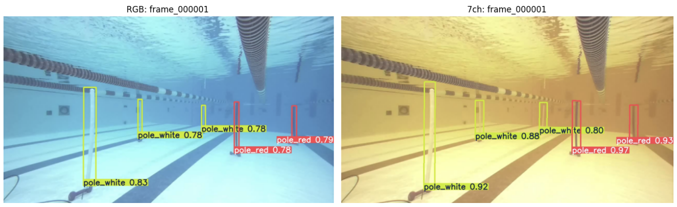


Fig. 5: Comparison of regular RGB YOLO output and YOLO 7D (RGB+Normals+Depth) output, with over 10% increase in accuracy for all detections.

2) State Estimation and Control: The control stack was significantly rewritten to improve trajectory smoothness and stability under nonlinear vehicle dynamics. We used System Identification to model underwater vehicle dynamics, capturing nonlinear effects that are difficult for feedback controllers to compensate. Using in-water sensor data, reconstructed thruster forces and a Fossen dynamics model [4], we used gradient-based optimization to refine estimates for key physical parameters such as drag, added mass, buoyancy and restoring forces. To fully leverage the improved dynamics model, we transitioned from a PID-based feedback architecture to an LQR-based feedforward/feedback controller. Lastly, we developed a new path planner that specifies all six degrees of freedom at each waypoint and generates a smooth spline-interpolated trajectory optimized for collision avoidance and curvature minimization, enabling more stable and robust movement between mission states. Additional details on the control architecture are provided in Appendix E.

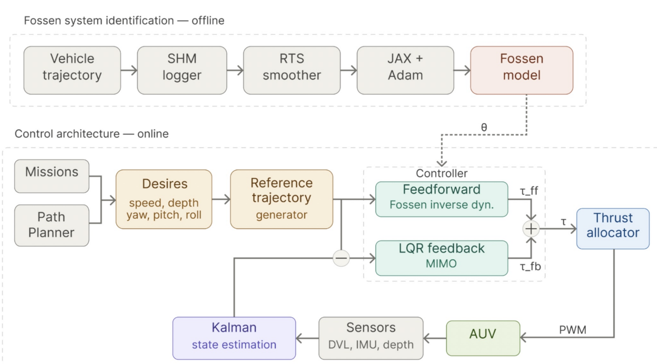


Fig. 6: Controls architecture showing system identification, feedback and feedforward control, and state estimation.

The state estimation pipeline was updated alongside the control stack to improve robustness under nonlinear vehicle dynamics and asynchronous sensing. We restructured the Kalman filter architecture, adopting a hybrid approach consisting of a multiplicative unscented Kalman filter [5] for orientation estimation and a separate linear Kalman filter for translational state. To address low-frequency and unreliable measurements, we introduced innovation-based gating for DVL and depth sensors, ensuring that only fresh and consistent measurements were incorporated into the state update. Further discussion of Kalman architecture and sensor consistency checks is included in Appendix F.

3) Infrastructure, Simulation, and Telemetry: Infrastructure work this year focused on increasing the speed and safety of experimentation, motivated by compute and memory constraints of the Jetson Orin Nano. Limited VRAM and CPU capacity can directly reduce YOLO frame rate, delay sensor fusion, and slow mission response, so we prioritized reducing compute overhead in core processes while preserving compatibility with the existing software stack.

To free compute for perception, we optimized the Kalman pipeline by moving performance-critical filtering operations from Python into C++, while maintaining interoperability with the Python stack through foreign function interfaces. Using Eigen for vectorized linear algebra reduced Kalman CPU usage by nearly an order of magnitude, allowing the estimator to run at a higher frequency while freeing compute and memory for YOLO inference.

We also overhauled the ZED camera pipeline to reduce image-processing overhead. The new C++ pipeline reads frames directly from the GPU, preprocesses them with CUDA kernels, and stages them in shared memory for YOLO inference. This reduced unnecessary CPU transfer and preprocessing bottlenecks, doubling the effective vision frame rate.

III. Testing Strategy

To translate the design strategy into reliable competition performance, CUAUV validated its systems through extensive verification. The following section outlines the team's testing strategy, focusing on how each subsystem was evaluated within the constrained development timeline.

A. Mechanical Testing

A comprehensive pressure testing protocol was developed to validate the watertight integrity of the main pressure vessel (MPV) prior to any electrical integration. We developed a three-phase testing protocol to ensure that any enclosure stays watertight during its operation.

The first phase employed SolidWorks FEA static simulations to assess the mechanical response of the MPV under 20 feet of hydrostatic pressure. These simulations validated the structural integrity of the MVP before any physical manufacturing and testing were done (Appendix O).

The second phase consisted of controlled negative air pressure testing using an analog pressure gauge mounted internally. For the 300-series O-rings [7] used on the main sealing surfaces of both Orion and Scylla, the applied vacuum helps pull the O-rings into their grooves, improving seal engagement. A vacuum reduced the internal pressure to 0.7 atm and was carefully monitored for 24 hours. Sustained vacuum at this level confirms o-ring engagement and air-tightness. If pressure loss was detected, soapy water was applied to all sealing surfaces to identify the source of the leak. Bubbling at any location indicated a breach in the seal, allowing for targeted corrective action.

The third phase consisted of an overnight pool submersion test conducted at a depth of approximately 15 feet. This served as the final validation checkpoint before electrical integration, exposing the MPV to realistic hydrostatic loading over an extended duration. The enclosure was inspected for any moisture ingress upon retrieval. A dry interior confirmed that the sealing system performed reliably under sustained submersion conditions and that the MPV was ready for full system assembly.

Smaller enclosures (such as Orion's Hydrophones enclosure) were similarly tested in FEA static simulations and subsequently tested for 24 hours in a pressure chamber at 15 PSI.

The strategy described how the team verifies that the sub is leak-free during initial validation. For subsequent pool tests throughout the semester, the second phase is repeated as a weekly maintenance procedure, using the onboard pressure sensor to monitor hull pressure through shared memory before the sub enters water.

B. Electrical Testing

CUAUV's modular approach to electrical design allows for robust testing of each of our PCBs before they are connected to the larger electrical system. At a minimum, this involves running the board off of bench power, verifying that it is heartbeating, and making sure all voltage lines are at the right level. Most boards also have additional requirements that are described in depth in their documentation.

A core component of testing for all boards that communicate with the Jetson through our serial protocol is the serial debugger. This piece of software communicates directly with the ATxMega or STM32 microcontrollers on our boards over RS232, allowing us to read and write each of the variables that software has access to over shm when the board is plugged into the sub. It also allows us to view the board's behavior when the submarine has been killed. Passing a serial debugger test verifies correct firmware configurations. Board to serial debugger connection issues indicate physical wiring or integration issues.

Power boards are all tested using a DC load based on the measured demands of our electrical system. In particular, our battery management and merge boards underwent load testing of 60A for 10 seconds and 30A continuously without overheating, which in past years has been a source of major issues.

Finally, a major component of testing this year has been our acoustic boards, hydrophones and transmit. At the time of writing, we have successfully tested hydrophones on signal generator inputs, verifying frequency magnitude detection and relative phase measurements between channels, both critical for pinger localization.

Our transmit board is at a similar stage, successfully producing the high-voltage signal required to drive the transducer on bench. Current testing focuses on validating performance in the pool, including SNR measurements at varying distances and verification of reliable bit transmission through our encoding scheme.

C. Software Testing

Our testing strategy prioritizes fast and safe iteration across simulation, offline playback and in-water validation. Across Orion and Scylla, we accumulated over 50 hours of in-water testing to validate controls, perception, and mission behavior under real conditions.

Initial development occurs in an in-house simulator (Fishbowl) of our pool environment, where we verify task sequencing, state transitions and other high-level mission logic. This year, Fishbowl was significantly improved to minimize differences between simulation and the real vehicle. New physical models for angular collisions, pool boundaries, water currents and hydrophone communication enables more realistic testing of the vehicle in underwater environments. The addition of real Kalman mode further strengthens simulation by allowing the real Kalman estimator to run from simulated noisy sensors instead of relying only on ground truth state.

As simulated vision is idealized, we also rely heavily on offline playback. Recorded pool footage is evaluated through CAVE, our custom playback application, which tests perception algorithms under realistic lighting and camera motion without additional pool time. This is especially important for caustic suppression and YOLO-7D development since both depend on representative visual data.

After simulation and offline validation, testing transitions to underwater trials to evaluate real-world performance and uncover edge cases. Competition strategy is tested bottom up. Individual mission tasks are first run independently assuming the relevant object is already visible. This isolates perception failures, control failures and mission integration issues before tasks are chained together. Once individual tasks are reliable, they are combined into a master mission that includes search logic, edge case handling and recovery behavior.

During all pool tests and mission runs, synchronized sensor and camera footage is recorded for post-run debugging, model training and performance evaluation. This creates a continuous improvement loop where the team tests in water, collects data, analyzes failures, retrains or modifies software, redeploys and tests again. Iteration continues until missions consistently succeed across varying environmental conditions.

Acknowledgements

CUAUV would like to thank all individuals and organizations who supported us over the past year. We especially thank Cornell University, including the MAE, ECE, and CS departments, for providing facilities and laboratory resources throughout our vehicle development process. We would also like to thank Professor Hunter Adams for being our advisor, as well as Cornell Aquatics and other supporting staff for their continued assistance with pool testing and operations. We also thank all CUAUV members and alumni for their technical expertise and contributions this year. Finally, the team extends its deepest gratitude to our sponsors listed in Appendix A, whose support made this work possible.

References

[1] Y. Swirski and Y. Y. Schechner, "3Deflicker from motion," in *Proc. IEEE International Conference on Computational Photography (ICCP)*, Cambridge, MA, USA, 2013, pp. 1–9, doi: 10.1109/ICCPhot.2013.6528294.

[2] T. Ophoff, K. Van Beeck, and T. Goedemé, "Exploring RGB+Depth fusion for real-time object detection," *Sensors*, vol. 19, no. 4, p. 866, 2019, doi: 10.3390/s19040866.

[3] Q. Wang, H. Yue, Z. Li, G. Huang, R. Liu, and J. Wang, "YOLO-depth: leveraging depth information for improved instance segmentation in construction scenes," *Advanced Engineering Informatics*, vol. 54, art. no. 104048, 2025, doi: 10.1016/j.aei.2025.104048.

[4] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*, 2nd ed. Hoboken, NJ, USA: Wiley, 2021.

[5] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[6] A. N. Biswas, N. Mahesh, and P. S. Rama Sreekanth, "Aluminium 6061 & HDPE incorporated study on gradient honeycomb-hybrid auxetic fused structure for energy absorption," *Materials Today: Proceedings*, vol. 56, part 3, pp. 1035–1042, 2022, doi: 10.1016/j.matpr.2021.09.105.

[7] Parker Hannifin Corporation, *Parker O-Ring Handbook*, ORD 5700. Lexington, KY, USA: Parker Hannifin Corporation, 2021.

[8] RoboNation, *RoboSub 2026 Competition Handbook*. RoboNation, 2026. Accessed: 2026.

[9] B. Benson, Y. Li, B. Faunce, K. Domond, D. Kimball, C. Schurgers, and R. Kastner, "Design of a low-cost underwater acoustic modem," *IEEE Embedded Systems Letters*, vol. 2, no. 3, pp. 58–61, Sept. 2010, doi: 10.1109/LES.2010.2050191.

Appendix A
Sponsors

A. Platinum Sponsors

Rigol — For providing electrical testing equipment used for subsystem validation, debugging, and vehicle development.

Hakko — For providing soldering and electrical integration resources used in board assembly, repair, and manufacturing.

Ansys — For providing software resources and simulation tools that supported vehicle analysis, design validation, and engineering development.

B. Gold Sponsors

Bonsai — For providing financial support toward vehicle development, testing resources, and competition preparation.

C. Bronze Sponsors

Polymer Plastics Company, LLC — For providing approximately $350 worth of cast acrylic tubing used for the main pressure vessel (MPV).

Appendix B
Component Specifications

Component	Vendor	Model/Type	Specs	Source	Cost	Year
Frame	CUAUV	Frame	HDPE	Custom	N/A	2026
Orion Frame	Datron + Kontrast 3D	Custom Aluminum Waterjet	Custom	Custom	Sponsored	2025
Outsourced Machining	Xometry	Custom Aluminum CNC	Custom	Custom	$3,100	2025
Aluminum Stock	Midwest Steel	Various	Various	Purchased	$800	2025
Aluminum Finish Anodization	Surface Finish Technologies	N/A	N/A	Purchased	$1,100	2025
Acrylic Tubing	Polymer Plastics	N/A	Custom	Custom	Sponsored	2025
Waterproof Housing	CUAUV	Main Pressure Vessel	Cylindrical pressure vessel to house electronics	Custom	N/A	2026
Waterproof Connectors	SEACON HUMMER/WET-CON	Dry/Wet connectors	Custom	Custom	$1,675	2018
Thrusters	BlueRobotics	T200	Brushless thruster	Purchased	$2,312	2018
Motor Control	CUAUV	Thruster PCBs	Four Blue Robotics Basic ESCs	Custom	$200/board	2026
Actuators	Blue Trail	SER-2020	230° underwater servo	Purchased	$495	2024
Battery	Turnigy	Graphene	10Ah 4S LiPo	Purchased	$85/battery	2022/2023
Regulator	Cincon	CHB75-12S12	Voltage regulation	Purchased	$76.28	2024
CPU/GPU	NVIDIA	Jetson Orin Nano	Six 2GHz, Arm8 cores	Purchased	$500	2025
Compass and IMU	LORD Microstrain	3DM-GX3 / 3DM-GX5	AHRS	Purchased	Sponsored	2018
Doppler Velocity Log (DVL)	WaterLinked	A50	DVL	Purchased	Sponsored	2024

Continued on next page

Component	Vendor	Model/Type	Specs	Source	Cost	Year
Manipulator	CUAUV	Manipulator	N/A	Custom	Free	N/A
Camera	FLIR	BFS-PGE-16S2C-CS Blackfly S GigE	Cameras	Purchased	$175	2024
ZED Camera	StereoLabs	Zed 2i	Cameras	Purchased	$499.99	2024
Hydrophones Processor	NXP	FRDM-K64F	Hydrophones	Purchased	$52.35	2024
Hydrophones	RESON	Acoustic transducers	Hydrophones	Purchased	N/A	2018
High-Level Control	CUAUV	6-DOF pose	Model-based LQR	Custom	N/A	2026
Vision Algorithm	OpenCV; YOLO	OpenCV4; YOLOv8	Transparent GPU support	Purchased	Free	2024
Depth Sensing	Blue Robotics	Bar02	N/A	Purchased	$80	2026
Acoustics	CUAUV	Custom DSP	N/A	Custom	$500	2026
Localization and Mapping	Mur-Artal	ORB-SLAM2	Modified	Purchased	Free	2024
Autonomy	CUAUV	Mission Planning System	N/A	Custom	Free	2026
Open Source Software	CUAUV/FSF	GNU	N/A	Custom	Free	2024
Battery Management Board	CUAUV	PCB	Cell balancing and protection	Custom	$80	2026
Power Distribution Board	CUAUV	PCB	Generates 5V and 12V low level board voltages	Custom	$60	2026
Serial Board	CUAUV	PCB	RS232 to USB conversion	Custom	$150	2026
Transmit Board	CUAUV	PCB	Drives transducer	Custom	$70	2026
Sensor Board	CUAUV	PCB	Depth, pressure, temperature, leak sensors	Custom	$80	2026

Continued on next page

Component	Vendor	Model/Type	Specs	Source	Cost	Year
LED Board	CUAUV	PCB	Status LEDs	Custom	$60	2026
Actuators Board	CUAUV	PCB	Drives the manipulator, torpedos, and droppers	Custom	$60	2026
Backplane	CUAUV	PCB	Accomodates 10 boards	Custom	$150	2026
DC-DC Board	CUAUV	PCB	5V, 12V, 7.4V	Custom	$80	2026

Appendix C
Electrical Architecture Diagrams

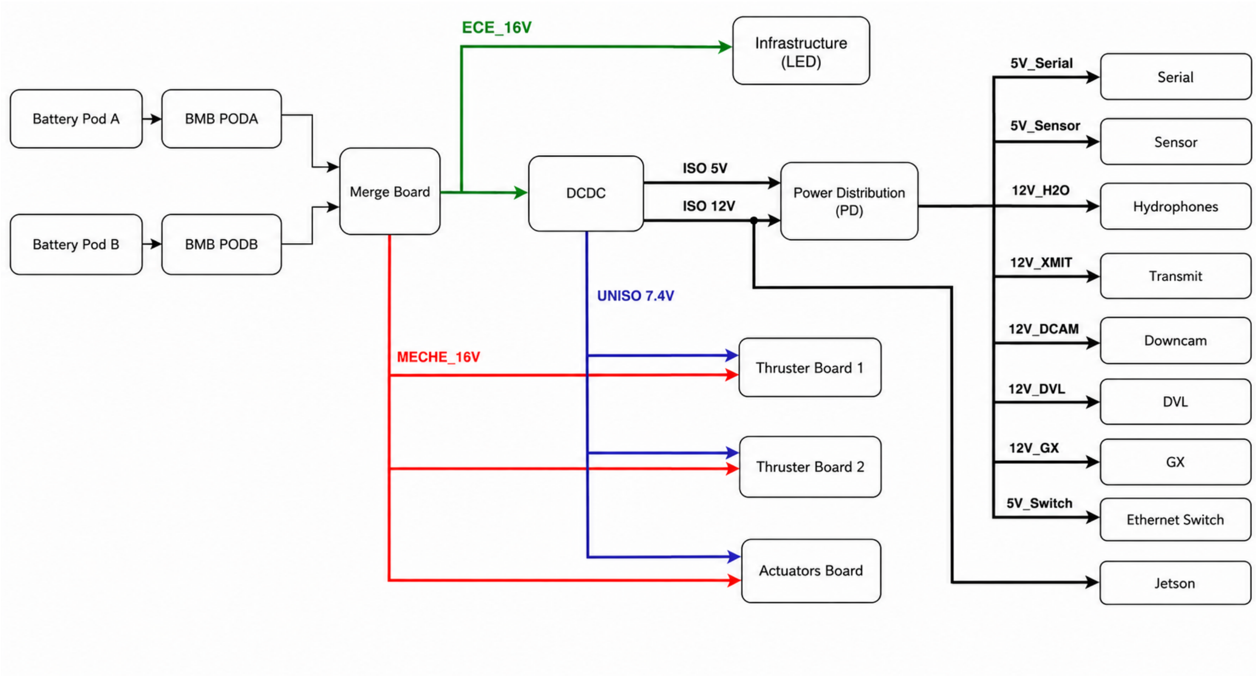


Fig. 7: Vehicle Power Architecture Diagram

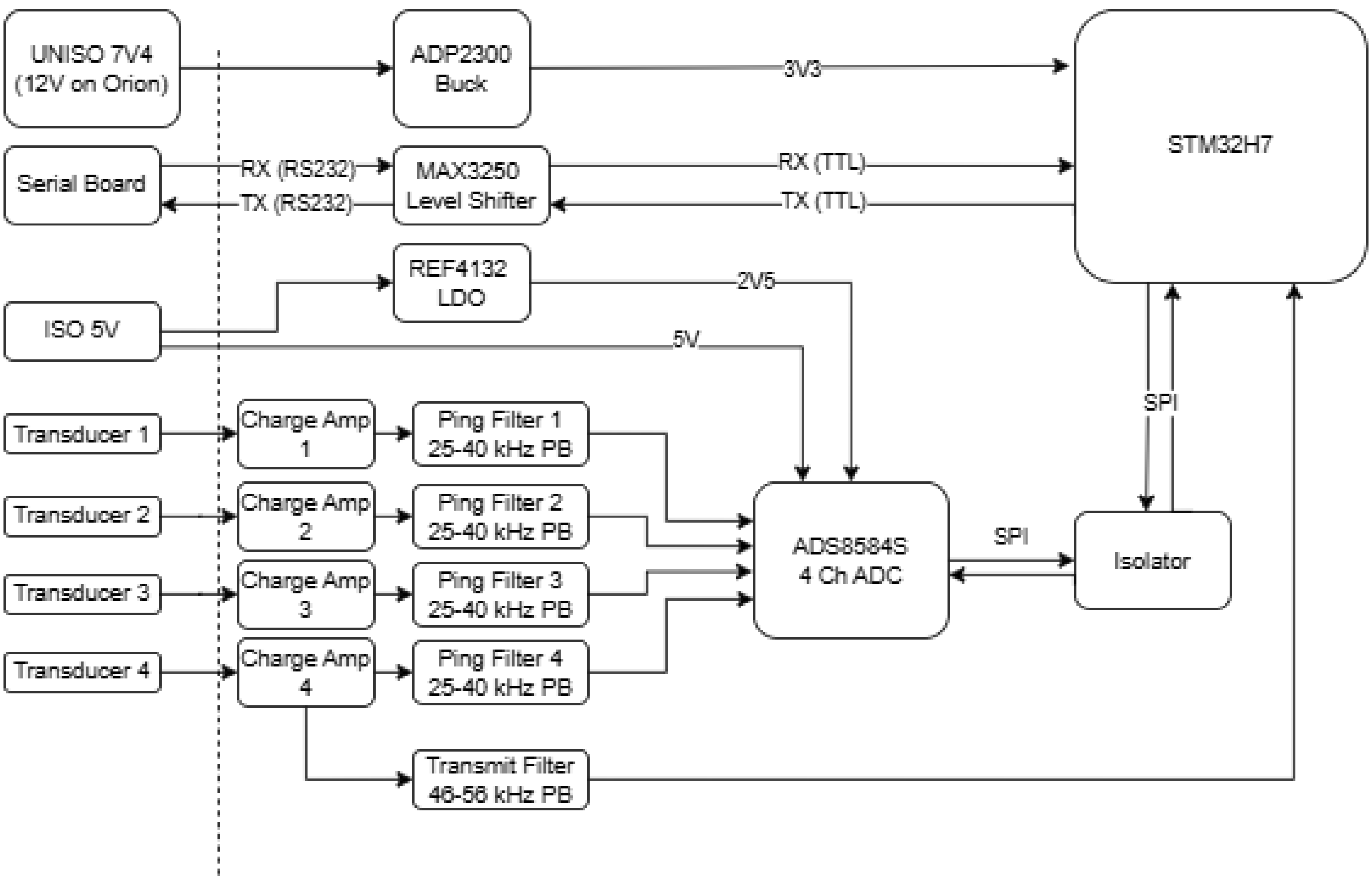


Fig. 8: Hydrophones High Level Block Diagram

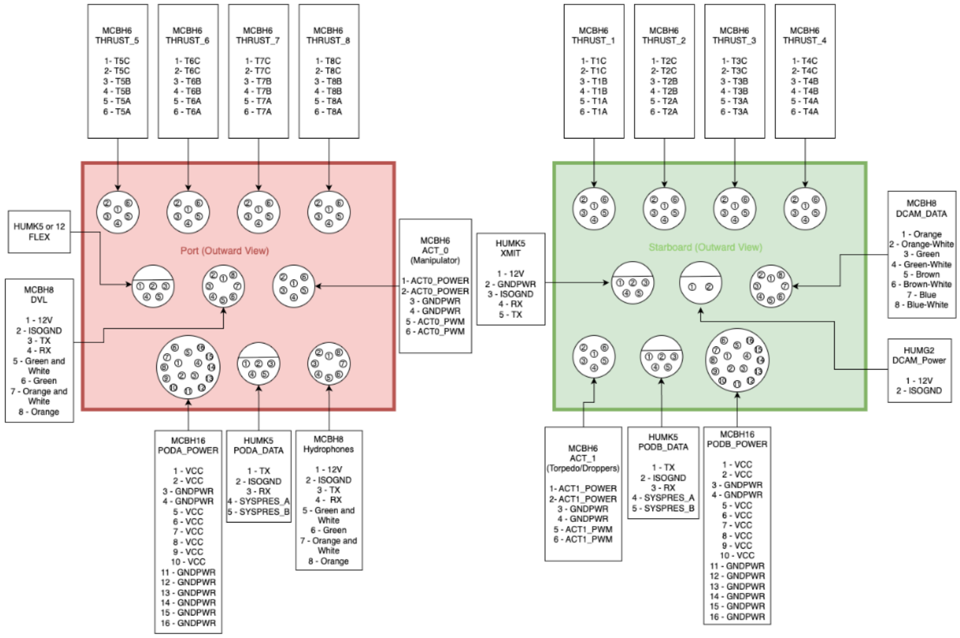


Fig. 9: Orion Seacon Layout

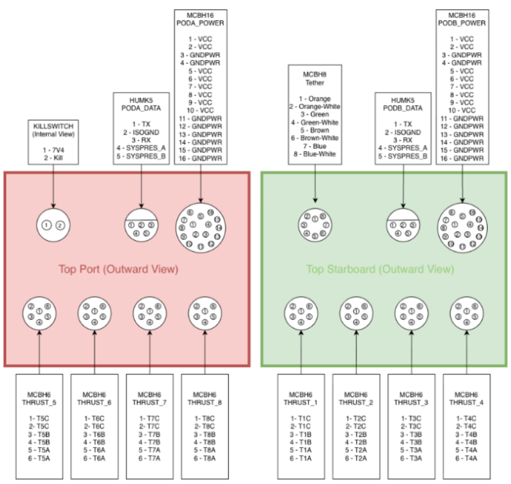


Fig. 10: Scylla Top Seacon Layout

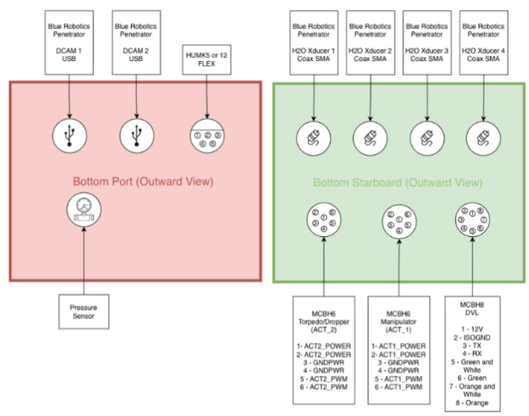


Fig. 11: Scylla Bottom Seacon Layout

Appendix D
Scylla Electrical Boards

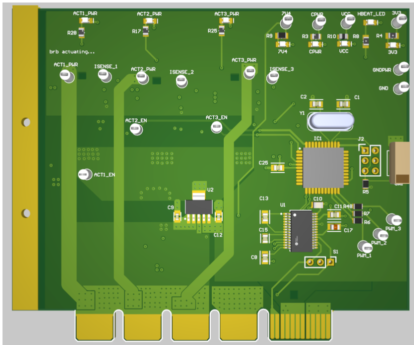


Fig. 12: Actuators Board

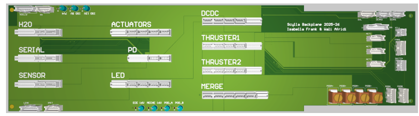


Fig. 13: Backplane Board

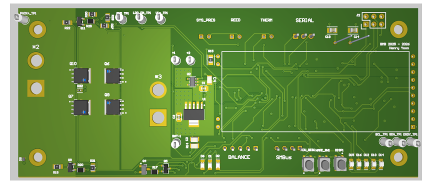


Fig. 14: Battery Management Board

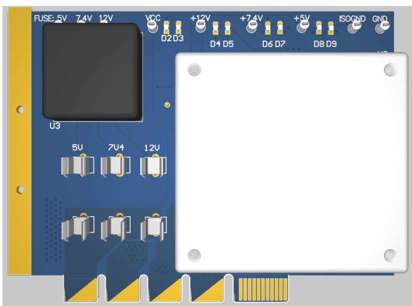


Fig. 15: DC-DC Board

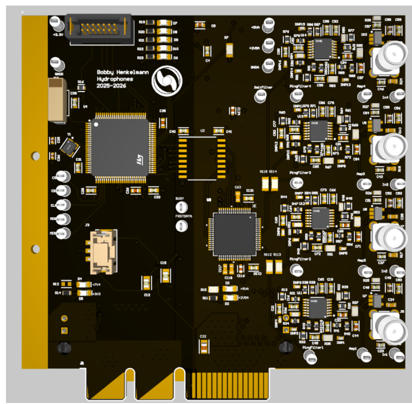


Fig. 16: Hydrophones Board

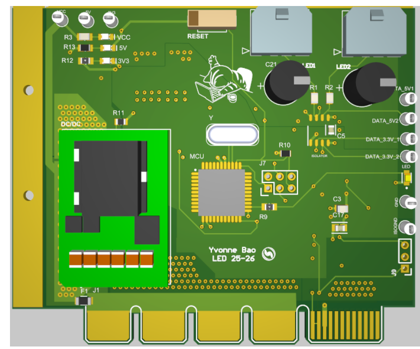


Fig. 17: LED Board

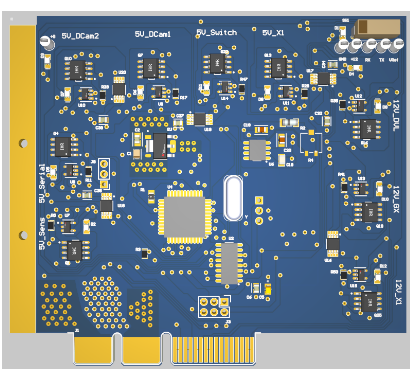


Fig. 18: Power Distribution Board

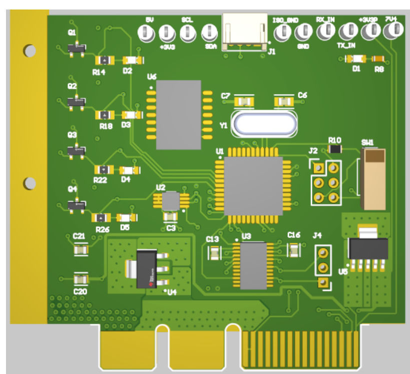


Fig. 19: Sensor Board

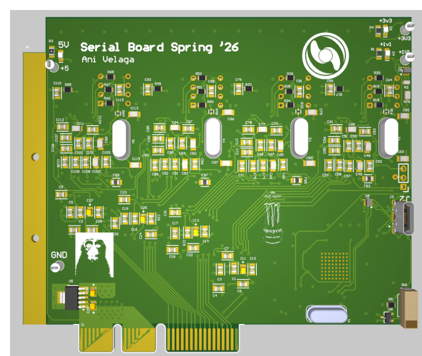


Fig. 20: Serial Board

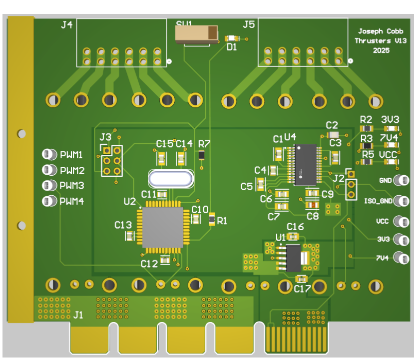


Fig. 21: Thrusters Board

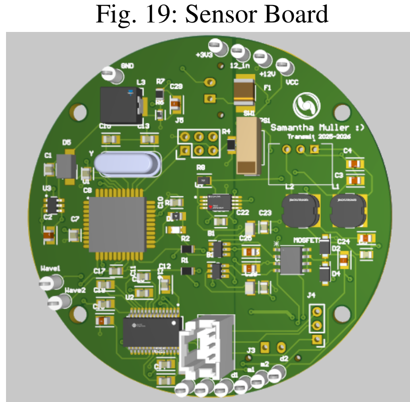


Fig. 22: Transmit Board

Appendix E
State Estimation, System Identification, and Controls

To support the transition from PID to our new LQR control stack, we developed a system identification pipeline using in-water sensor logs, actuator commands, and a Fossen dynamics model. Rauch–Tung–Striebel smoothing fuses sensor streams into dynamically consistent trajectories, allowing interpolation of low-frequency DVL measurements without phase lag.

We also reconstructed applied vehicle forces and torques from logged PWM commands, published thruster performance curves, and vehicle geometry. These reconstructed forces were used to optimize drag, added mass, buoyancy, and restoring-force parameters in the Fossen model using JAX-based autodifferentiation and Adam optimization. This reduced one-step prediction error by an order of magnitude and improved long-horizon trajectory prediction by several orders of magnitude.

Finally, we introduced a 6-DOF path planner that specifies full pose targets at each waypoint and generates smooth spline trajectories. This improves physical consistency between mission states and improves alignment with task elements.

Appendix F
Kalman Filter Architecture and Sensor Consistency

The state estimation pipeline was updated alongside the control stack to improve robustness and consistency under nonlinear vehicle dynamics and asynchronous sensing. We restructured the Kalman filter architecture, adopting a hybrid approach consisting of a multiplicative unscented filter for orientation estimates and a separate linear filter for translational state, velocity, and position. This decoupling allowed us to better handle nonlinear attitude dynamics while maintaining a stable and efficient estimator for translational motion.

To address the challenges of low-frequency and unreliable measurements, we introduced innovation-based gating for DVL and depth sensors. These sensors operate at lower rates and are more susceptible to dropouts and noise, so gating ensured that only fresh and consistent measurements were incorporated into the state update, preventing stale or corrupted data from degrading the estimate.

Additional improvements focused on sensor consistency, including alignment of coordinate frames, unit consistency, and explicit handling of IMU measurements. Together, these changes improved the stability and reliability of orientation and velocity estimates, supporting the overall control pipeline as it evolved toward more precise model-based methods.

Appendix G
Kalman and ZED Pipeline Optimization

Infrastructure work this year focused on increasing the speed and safety of experimentation. One major effort was optimizing our Kalman filters by moving them from Python into C++ to lower CPU compute. The `auv-kalman3d` process was taking up 3 cores, 50% of our Jetson's compute. Therefore, we deliberately built the new Kalman infrastructure in C++, interoperating with our existing Python stack using foreign function interfaces and taking advantage of the Eigen linear algebra library's vectorized computation. This change brought down the compute used by our Kalman process to less than 1 core, enabling our Kalman process to run at a higher frequency, increasing sensor estimation accuracy for movement-intensive tasks.

Another foundational improvement to our infrastructure was the major overhaul and optimization of the ZED camera pipeline, where we rewrote our image acquisition and preprocessing code from Python to C++. The prior Python implementation bottlenecked on processing by running each function sequentially on the CPU and sending different image channels at different speeds. The C++ version reads camera images straight from the GPU, preprocesses them with specialized CUDA kernels, and sends them into a shared memory staging buffer that our YOLO model then runs inference from automatically. YOLO was also separated from the vision pipeline, allowing it to be toggled depending on whether we wanted to use classical computer vision or YOLO. By making these architectural changes as well as optimizing our vision pipeline for a new version of the camera SDK, we doubled our effective frame rate.

Finally, we improved developer operations around pool testing and deployment. We introduced clearer GitHub branching and merging rules, including creating issues for new pool test branches and merging them through a weekly precheck process. We also built our own Docker registry at `docker.cuauv.org` to support more controlled and reproducible container builds by allowing image tagging and rollbacks. Together, these changes reduce configuration drift, make experimental work easier to track, and increase confidence when deploying software during pool tests.

Appendix H
Fishbowl Simulation and Visualizer Improvements

This year's simulation work pursued two parallel objectives: increasing the physical accuracy of the Fishbowl dynamics engine, including realistic sensor simulation to support full state estimation without hardware, and raising the visual fidelity of the Visualizer to the point where a simulated camera running YOLO is sufficient for validating mission logic.

Our custom rigid-body physics engine, Fishbowl, received a significant upgrade through improvements to both infrastructure and supported features. We originally considered replacing it with an off-the-shelf physics library, evaluating options including Bullet, Chrono, Gazebo Sim, and Stonefish. Gazebo and Stonefish were ruled out early: both are designed around ROS, which is incompatible with our SHM-based communication architecture, and neither offers straightforward support for the custom pool competition environments and task props we need to simulate. Bullet and Chrono were more viable as drop-in physics backends, but integrating either would have required significant rework of the simulation layer with limited benefit over extending Fishbowl directly, which already has deep hooks into our tooling and SHM data pipeline.

Therefore, we chose instead to build out Fishbowl's missing capabilities in-house. We added support for rigid body collisions using AABB bounding shapes, allowing us to collide with elements such as slaloms and buoys as well as adding a simulated manipulator arm for testing object manipulation tasks. To better simulate drag, we created water currents by applying a world-frame constant velocity offset to the vehicle's relative flow velocity before computing drag, giving the vehicle an ambient current for the control loop to fight against. Finally, we expanded our simulated sensor suite, going from just cameras to including accelerometer, DVL, depth sensor, and hydrophones. Each sensor is simulated by mimicking the internal workings of each sensor and following the OEM's precision guidelines, keeping the system fully under our control and tightly integrated with the rest of the stack.

For example, the simulated DVL models a Janus four-beam projection with each beam oriented 30 degrees from vertical in the body frame, matching the geometry of the Teledyne DVL used on Scylla. The altitude reading is computed per beam by raycasting to the pool floor and converting slant distance to vertical range, with Gaussian noise applied at the configured standard deviation. Velocity is computed at the DVL's mounting offset position in the world frame, incorporating both translational and rotational vehicle motion, then rotated into the body frame and noise-added before being written to SHM.

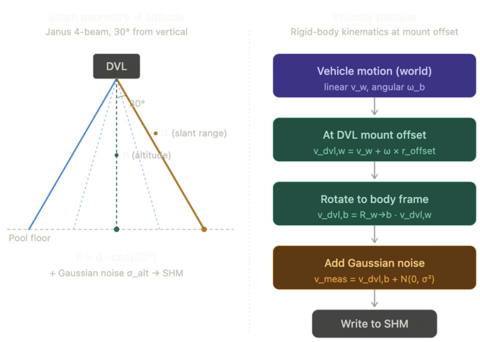


Fig. 23: Visualization of how the DVL is simulated in Fishbowl.

These realistic sensor outputs are fed directly into our live Kalman-filter pipeline, consuming simulated data exactly like on hardware. In parallel, Fishbowl was upgraded to model thruster forces directly from PWM commands issued by the control layer. These forces are mapped with the system identification pipeline and integrated with the previously identified Fossen dynamics model. This results in significantly improved simulation fidelity, allowing the simulator to more accurately reproduce real-vehicle dynamics and closed-loop control behavior.

The CUAUV Visualizer is a real-time 3D rendering system written in C++17 that provides a live view of the simulation environment. It connects to a running Fishbowl instance over TCP, rendering the vehicle and competition props with positions and orientations driven directly by the physics engine. This year, the visualizer received significant rendering upgrades aimed at replicating the challenging visual conditions encountered during pool testing and competition.

Underwater fog was introduced to simulate the limited visibility of pools, attenuating scene colors with distance to replicate the same degraded sight our cameras face in practice. Depth-dependent darkening further reduces brightness as objects descend below the water surface, capturing the rapid light absorption that makes deep targets difficult to detect. Procedural caustic projection, animated light patterns cast by the water surface, was added to more accurately reproduce the visual noise that complicates detection, which we experienced during competition last year. Together, these changes make the simulated camera feed a much closer stand-in for live footage, increasing our confidence that vision models and mission logic tested in simulation will transfer to the real pool environment.

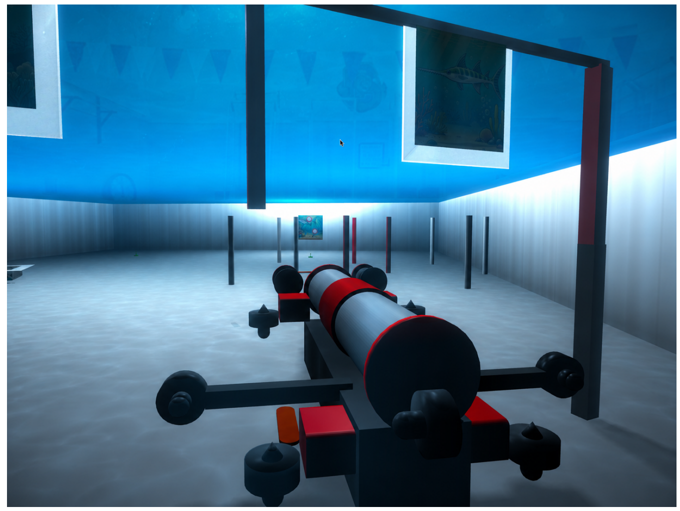


Fig. 24: Simulator running `robosub2025.cfg`.

The result of this year's effort is a fully custom simulator that faithfully reproduces real-vehicle behavior across the full software stack, from realistic vehicle movement in the water to reliable simulated runs of competition tasks. Combined with our new telemetry tools, this has made prototyping missions feasible in simulation, enabling dramatically more productivity for software without relying on in-person pool tests.

Appendix I
Shmlogger and Real-Time Telemetry Tools

This year, we wanted a more robust way to learn from past mission runs and amend mistakes. Previously, we relied only on camera footage taken during mission runs to debug potential issues. However, we wanted to record and learn from all available data stored in shared memory. This led us to develop our own in-house telemetry infrastructure via log files that work seamlessly with our shared memory state: `Shmlogger`.

`Shmlogger` is a utility service that creates readers into our shared memory buffer and then writes the data into our custom `.shmlog` file type. The `.shmlog` is a custom binary format designed for efficient logging of CUAUV's shared memory state. It utilizes Zstd compression to balance high-frequency data recording, defaulting to 200 Hz, with storage efficiency. We wanted the file format to be versatile as well, so we implemented database best practices by adding random-access seeking via periodic snapshots and a snapshot table for flexible querying. We also added GUI support for Shmlogger, allowing us to create

CSV and graphical exports for further data analysis, which has been useful for calculating key metrics such as root mean square error for our dynamical model's accuracy to real-world data.

Using Shmlogger, we have been able to debug various sensor errors that have plagued us for years prior, from solving the source of our accelerometer drifting due to not having the manufacturer's extended Kalman filter applied onto its inputs to being able to run system identification on our vehicle, unlocking the potential for open-loop controls. We plan on furthering the usage of Shmlogger by analytically tuning our PID constants for our controller using the Ziegler–Nichols closed-loop method, rather than through empirical trial and error.

However, Shmlogger only allows us to review our data retrospectively. Therefore, this year we also created a utility that allows us to plot and visualize specific data from multiple sources in real time. This allowed us to see discrepancies within various related sources of data in real time, such as deviations between a sensor's reading and the Kalman filter's estimation, which gave us clearer direction toward debugging issues on the vehicle.

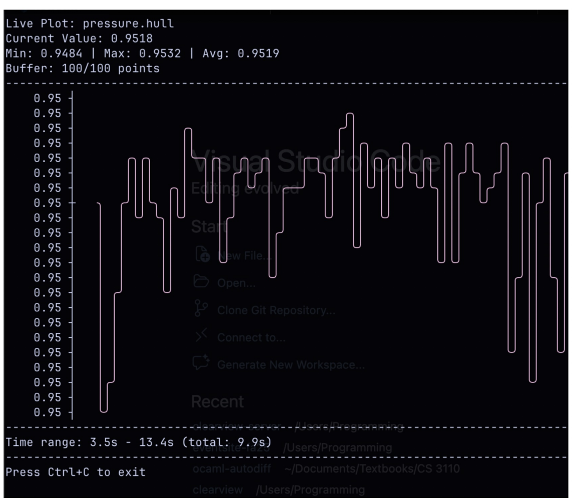


Fig. 25: Example of using shm-plotter to visualize the change in pressure for pressure checks.

Overall, expanding our telemetry and communication tools has massively supported pool operations. By making sensor outputs, estimator behavior, and mission-relevant state easier to monitor, we can debug failures faster and deploy changes safer.

Appendix J
Backplane Layout Comparison Between Orion and Scylla

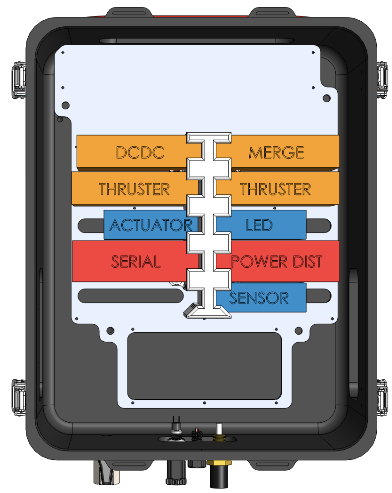


Fig. 26: Orion backplane layout housing nine electrical boards.

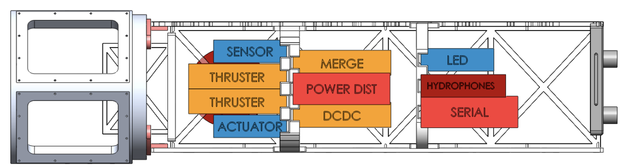


Fig. 27: Scylla backplane layout housing ten electrical boards.

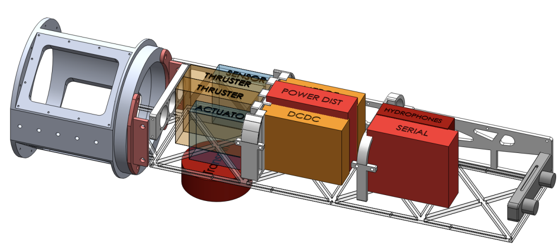


Fig. 28: Isometric view showing all eleven boards housed in Scylla's main pressure vessel.

Appendix K
Material Property Comparison

Table IV compares selected material properties of 6061 aluminum and high-density polyethylene (HDPE). These properties were considered when selecting HDPE for Scylla's main structural sheet, particularly because of its lower density and manufacturability.

TABLE IV: Comparison of 6061 Aluminum and HDPE Material Properties

Property	**6061 Aluminum**	**HDPE**
Density	2.7 g/cm^3	1.0–1.3 g/cm^3
Elongation at Break	3.4–20 %	2.5–100 %
Maximum Mechanical Temperature	170 °C	90 °C
Specific Heat Capacity	900 $J/(kg \cdot K)$	1330–2400 $J/(kg \cdot K)$
Axial Strength-to-Weight	13–42 points	6.9–17 points
Bending Strength-to-Weight	21–45 points	19–32 points
Ultimate Tensile Strength	130–410 MPa	24–80 MPa

Appendix L
Frame Mechanical Analysis

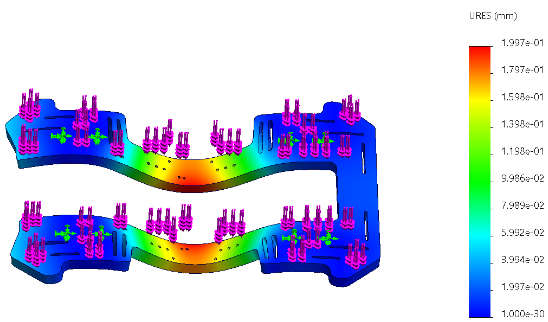


Fig. 29: FEA displacement plot of the top plate under applied lifting load, showing maximum deflection at the center span of approximately 0.20 mm.

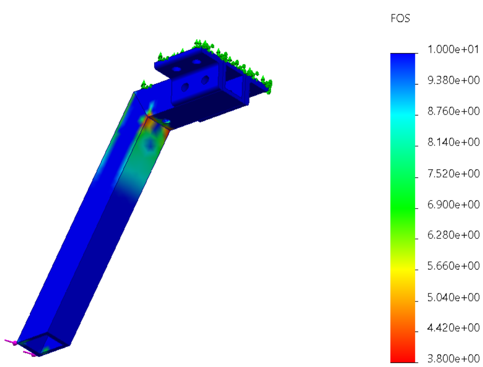


Fig. 30: FEA factor of safety plot for the front leg wall-impact loading case with a 5 mm stopping distance, showing a minimum FOS of 3.8 near the welded joint.

Appendix M
Foldable Thrusters Mechanical Analysis

To evaluate the required thrust for forward motion, the drag force acting on the vehicle was estimated using the standard quadratic drag equation:

$$F_D = \frac{1}{2} C_D \rho A v^2 \tag{1}$$

where F_D is the drag force, C_D is the drag coefficient, ρ is the density of water, A is the projected frontal area of the submarine, and v is the target forward velocity. The values used for this calculation are shown in Table V.

TABLE V: Parameters Used for Forward Drag Estimation

Parameter	Description	Value
C_D	Drag coefficient	1.05
ρ	Density of water	1000 kg/m^3
A	Projected frontal area	0.16 m^2
v	Forward velocity	1.3 m/s

Substituting the values from Table V into Eq. 1, the estimated forward drag force is

$$\begin{aligned} F_D &= \frac{1}{2}(1.05)(1000)(0.16)(1.3)^2 \\ &= 141.96 \text{ N}. \end{aligned} \tag{2}$$

The vehicle uses four forward thrusters, each mounted at 45° relative to the direction of forward motion. Therefore, the total forward thrust contribution is

$$F_D = 4F_{\text{thruster}} \cos(45^\circ). \tag{3}$$

Solving for the required thrust per thruster gives

$$\begin{aligned} F_{\text{thruster}} &= \frac{F_D}{4\cos(45^\circ)} \\ &= \frac{141.96}{4\left(\frac{\sqrt{2}}{2}\right)} \\ &= 50.19 \text{ N}. \end{aligned} \tag{4}$$

Thus, the estimated required thrust per forward thruster is

$$\boxed{F_{\text{thruster}} \approx 50.19 \text{ N}}. \tag{5}$$

The calculated thrust force was applied in an FEA simulation, shown in Fig. 31, to verify that the foldable thruster mount can withstand the increased moment arm without significant deflection.

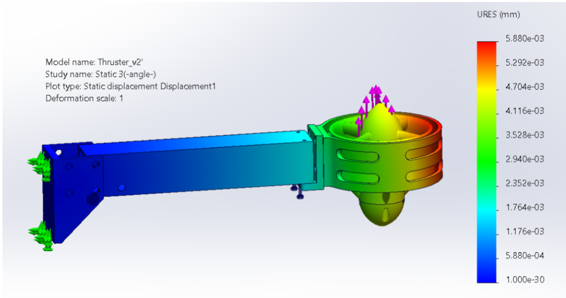


Fig. 31: FEA displacement plot of the foldable thruster under applied lifting load, showing a maximum deflection of approximately 5.88×10^{-3} mm at the end of the thruster.

A. Foldable Thruster Moment Arm Comparison

The foldable thruster design increases the lateral distance between Orion's depth thrusters from 25 in to 41 in. Since roll torque is given by

$$\tau = Fr, \tag{6}$$

where F is the thrust force and r is the moment arm from the vehicle centerline, increasing the thruster spacing directly increases the available roll torque. Using the calculated thrust force of 50.19 N, the torque comparison is shown in Table VI.

TABLE VI: Roll Torque Comparison for Orion Foldable Thrusters

Configuration	Thruster Spacing	Moment Arm	Torque
Original Orion	25 in	12.5 in	15.93 N m
Foldable Thrusters	41 in	20.5 in	26.13 N m

The percent increase in roll torque is

$$\begin{aligned} \% \text{ increase} &= \frac{\tau_{\text{foldable}} - \tau_{\text{original}}}{\tau_{\text{original}}} \times 100 \\ &= \frac{26.13 - 15.93}{15.93} \times 100 \\ &= 64.0\%. \end{aligned} \tag{7}$$

Therefore, the foldable thruster design increases the available roll torque by approximately 64%.

Appendix N
Scylla's Buoyancy Estimation

The displaced volume of Scylla was estimated by separating the vehicle into the main pressure vessel, battery pods, and remaining vehicle volume. The main pressure vessel was modeled as a cylinder:

$$\begin{aligned} V_{\text{MPV}} &= \frac{\pi d^2}{4} h \\ &= \frac{\pi (0.1937 \text{ m})^2}{4} (0.7255 \text{ m}) \\ &= 0.0214 \text{ m}^3. \end{aligned} \tag{8}$$

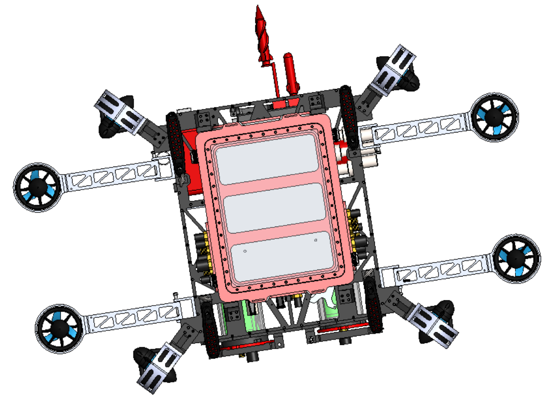


Fig. 32: Foldable thrusters in the downward open position on the full Orion assembly.

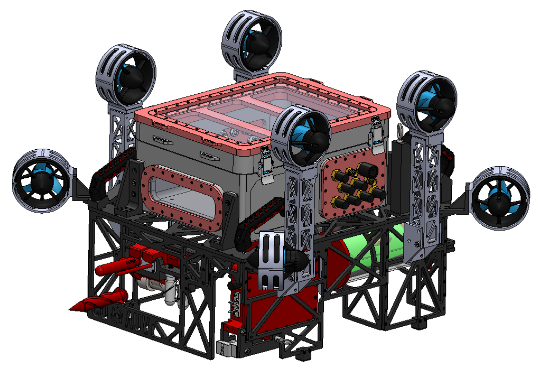


Fig. 33: Foldable thrusters in the folded position on the full Orion assembly.

The two battery pods were also modeled as cylindrical volumes:

$$\begin{aligned} V_{\text{battery pods}} &= 2\left(\frac{\pi d^2}{4} h\right) \\ &= 2\left(\frac{\pi (0.1168 \text{ m})^2}{4} (0.2248 \text{ m})\right) \\ &= 0.0048173 \text{ m}^3. \end{aligned} \tag{9}$$

The remaining vehicle volume was estimated as

$$V_{\text{rest}} = 0.01115 \text{ m}^3. \tag{10}$$

Thus, the total estimated displaced volume is

$$\begin{aligned} V_{\text{total}} &= V_{\text{MPV}} + V_{\text{battery pods}} + V_{\text{rest}} \\ &= 0.0214 + 0.0048173 + 0.01115 \\ &= 0.0373673 \text{ m}^3. \end{aligned} \tag{11}$$

Using Archimedes' principle, the buoyant force is

$$F_b = \rho V g, \tag{12}$$

where $\rho = 1000\ \text{kg/m}^3$ is the density of water and $g = 9.81\ \text{m/s}^2$. Substituting the estimated volume gives

$$\begin{aligned} F_b &= (1000\ \text{kg/m}^3)(0.0373673\ \text{m}^3)(9.81\ \text{m/s}^2) \\ &= 366.573\ \text{N}. \end{aligned} \tag{13}$$

The gravitational force was calculated using the measured vehicle mass:

$$\begin{aligned} F_g &= mg \\ &= (35.2\ \text{kg})(9.81\ \text{m/s}^2) \\ &= 345.312\ \text{N}. \end{aligned} \tag{14}$$

Therefore, the estimated net buoyant force is

$$\begin{aligned} F_{\text{net}} &= F_b - F_g \\ &= 366.573\ \text{N} - 345.312\ \text{N} \\ &= 21.261\ \text{N}. \end{aligned} \tag{15}$$

Thus, Scylla is estimated to be approximately

$$\boxed{21.3\ \text{N}} \tag{16}$$

positively buoyant.

Appendix O
Main Pressure Vessel Mechanical Analysis

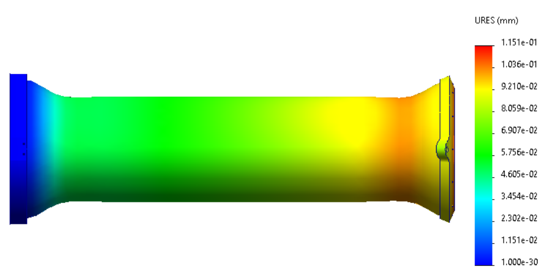


Fig. 34: FEA displacement plot (x489 scale) of the MPV outer hull assembly under 15 psi of pressure with max displacement of 0.11 mm

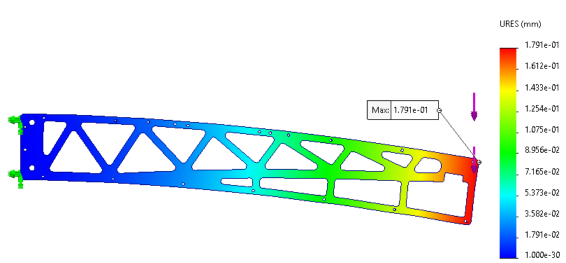


Fig. 35: FEA displacement plot (x285 scale) of one out of two cantilevered trusses supporting the racks with max displacement of 0.18 mm under 22.5 N

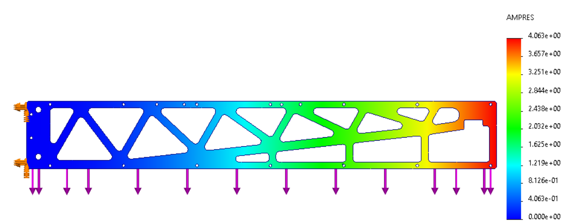


Fig. 36: FEA frequency plot of one out of two cantilevered trusses supporting the racks with the first frequency being 13 Hz

Appendix P
IVC-Coordinated Random-Pinger Mission Schedule

Each vehicle is equipped with a hydrophone array that estimates the bearing, or direction of arrival, of the active acoustic pinger. A single vehicle can therefore infer the direction of the pinger more reliably than its absolute global position. By exchanging bearing and confidence estimates over IVC, Orion and Scylla can compare acoustic observations from two separated locations. These estimates are combined with prior mission-state knowledge of the approximate Torpedoes and Octagon regions to form a consensus on which task corresponds to the active random pinger.

After this classification step, the mission scheduler branches according to the inferred pinger task. If Torpedoes is selected first, Orion proceeds to Torpedoes while Scylla later continues to Octagon; if Octagon is selected first, Scylla proceeds to Octagon while Orion completes Flyby Bins and Torpedoes. This makes IVC behavior-changing: the acoustic estimate exchanged between vehicles changes the order of task execution rather than only reporting status.

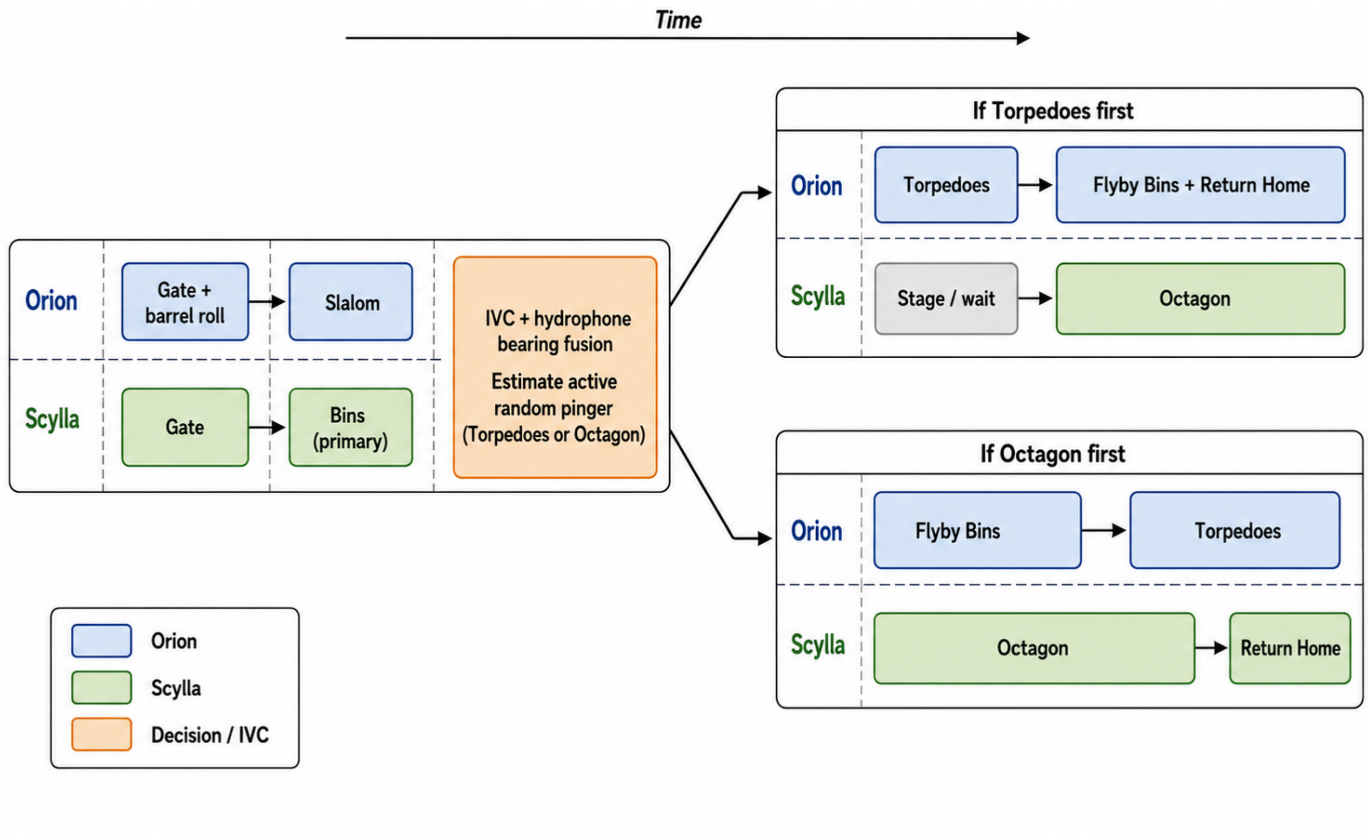


Fig. 37: IVC-coordinated mission schedule for Orion and Scylla. After the shared Gate–Slalom–Bins block, hydrophone bearing fusion and IVC determine whether Torpedoes or Octagon is executed first.

Appendix Q
CFD Simulation Results

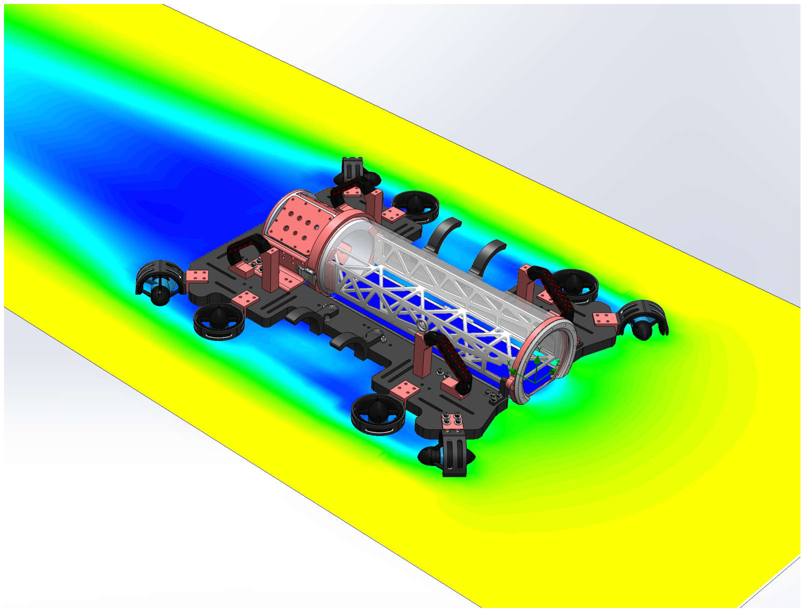


Fig. 38: SolidWorks Flow Simulation result for Scylla, used to estimate lumped drag and cross-axis coupling coefficients.

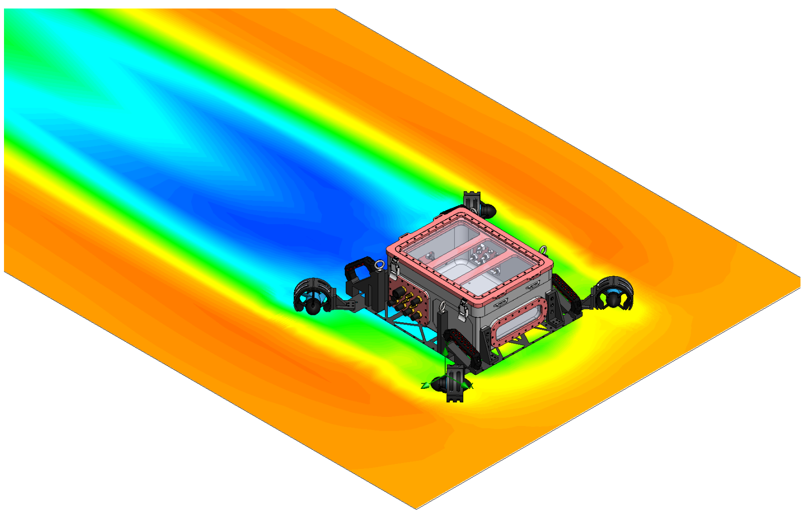


Fig. 39: SolidWorks Flow Simulation result for Orion, used as the baseline comparison for Scylla's drag and coupling estimates.