Qubo VII: RoboSub 2026 Technical Report

See Appendix A for the Team Roster

***Abstract*—The Robotics at Maryland (R@M) team at the University of Maryland dedicated the 2025–2026 academic year to converting Qubo VII's demonstrated potential into consistent competition performance. With a mechanically mature platform as a foundation, the team focused on reliability-driven hardware improvements, efficiency improvements for our computer vision and autonomy pipelines, and significantly expanded testing infrastructure, all in service of a competition strategy centered on consistent end-effector task completion. This year also marks continued development of our autonomous torpedo, a fully self-contained vehicle within torpedo size constraints, serving as an independent scoring platform alongside Qubo. For the first time, comprehensive full-system testing in a competition-like environment has validated these developments throughout the year.**

Fig. 1: Qubo by the testing pool.

I. Competition Strategy

A. General Strategy

Over the past several competitions, our goal has been to complete the initial navigation tasks followed by at least one of the end effector tasks, as we have observed that reaching this threshold is generally what separates teams that advance to the finals from those that do not. Despite electrical failures that prevented Qubo from demonstrating its full capabilities at previous competitions, we have come closer each year to this goal, even achieving it in parts during practice runs both before and during competition. This trend has continued in the most recent competition, motivating us to maintain this goal and preserve our core task strategies. With our task plans stabilized, we have focused on consistent full-system testing throughout the year in a mock competition environment. This effort served both to consolidate navigation tasks we have previously demonstrated and to improve the consistency of the bin and torpedo end effector tasks. Meanwhile, this testing exposed and resolved hardware and software issues long before competition.

B. Course Strategy

Qubo has the potential to complete nearly all tasks at competition; this year we have prioritized converting this potential to successful task runs through consistent testing. Our overall strategy remains the same: leverage our proven navigation systems to complete the first few tasks, and then use our end effectors to complete as many of the remaining tasks as possible.

1) Heading Out & Begin Assessment (Coin Flip & Gate): We will start our run with a coin flip to determine our starting orientation, then rely on our object detection system to find and align laterally with the gate as Qubo rotates to search. Once locked on the gate, Qubo will align with the side of the gate corresponding with our desired role as given by the second coin flip or visual indicator. Then, utilizing our pose estimation, position tracking and movement control systems, Qubo will calculate a point beneath the image and navigate to it, completing the task with a 720° yaw for style points.

2) Avoid Debris (Slalom): Maintaining a preset depth within the slalom bounds, Qubo will first align with the slalom to ensure passage through the same side of the slalom as the gate. It will search for the closest red pipe found using its object detection system. Then, Qubo will use its pose estimation systems to calculate a point between the closest white and red pipe, and navigate there. This will be repeated twice more to complete the slalom.

3) Recon (Bins): After exiting the slalom, Qubo will rotate to search for the bins. Given the bins' distinct new structure this year, we are confident that our object detection system will reliably identify them. Then, maintaining a depth near the surface, Qubo will navigate towards the bins until all are visible to the downward-facing camera. Based on our role, Qubo will select the two bins with relevant images and align with each in sequence, centering each image in its vision and descending to a depth where its droppers will reliably land in the bin while maintaining detection of the image for alignment.

4) Deploy (Torpedoes): After dropping a marker in the second bin, Qubo will search for and navigate toward the torpedo board. Using the depth map from Qubo's stereo camera setup, it will close to a specified distance from the board. Then, Qubo will use feedback from its pose estimation system to align its orientation with the board, keeping it centered in our view while rotating to end parallel with the targets. Based on our assigned role, Qubo will select the two relevant targets. Finally, Qubo will align to a position relative to each target that has been determined to produce reliable torpedo hits, and fire in sequence. The speed and straight trajectory of our power torpedoes allows Qubo to fire from distances far enough from the board to earn the maximum distance-based points.

The powered torpedoes serve primarily as a backup to the autonomous torpedo project, which aims to complete the torpedo task and additional tasks as an independent vehicle.

5) Resupply (Octagon): In future years, we hope to use our newly improved claw to complete the object interaction and sorting portion of this task, once we can reliably perform the other end effector tasks at competition. However, given the significantly greater complexity and lower completion rate of this task compared to the other end effector tasks, we have decided this year to focus on the bin and torpedo tasks to maximize expected points. After firing our torpedoes, Qubo will search for and head to the Resupply Table, centering it in the downward camera's view. Then, Qubo will surface in the Octagon for additional points and eligibility for the time bonus.

6) Return Home: Based on the water clarity observed at previous competitions, we are confident that Qubo will be able to detect the back of the gate from across the pool. As such, after descending from the Octagon, Qubo will navigate back through the gate to conclude our run.

II. Design Strategy

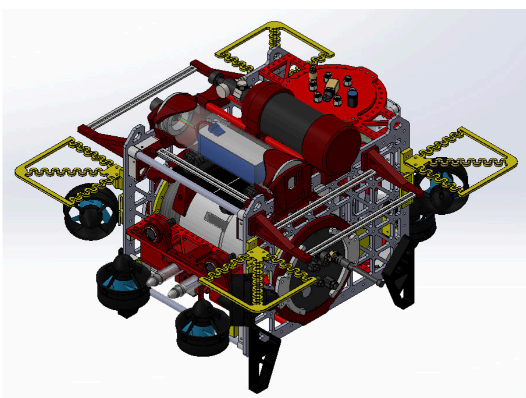

Fig. 2: Rendered CAD model of Qubo.

With Qubo's core mechanical structure in a mature state and its electrical system reaching a stable condition, hardware development has been directed largely toward Qubo's end effectors and toward reliability and usability improvements to its core systems. This hardware stability, combined with a sustained focus on mock competition testing, has enabled our software team to develop and refine Qubo's autonomy systems throughout the year.

A. Pneumatics

The pneumatics system actuates the torpedo launcher, marker dropper, and claw end effectors. Compressed air is regulated from a paintball tank into an enclosure containing four 12V solenoid valves. These valves direct air to four double action pneumatic pistons that each control an end effector. During the 2025 competition, a leak in the flow regulator allowed water to enter the solenoid valves and drain to the bottom of the pneumatics hull, damaging the PCB inside. To prevent recurrence, the PCB was relocated from the bottom to the top of the pneumatics hull and mounted upside down, so that any water ingress drains away from it. A shallow lip is also designed into the enclosure to prevent water from dripping directly on the PCB. Additionally, the flow regulator, solenoid valves, and one-way valve were replaced with robust corrosion-resistant components.

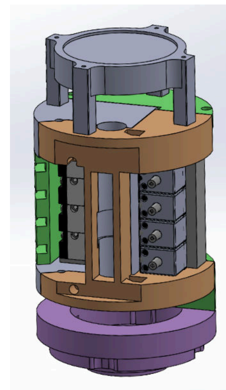

Fig. 3: Rendered CAD model of the pneumatics system.

B. End Effectors

1) Marker Dropper: The dropper system was designed around the linear actuation of the piston to enable precise marker deployment into designated bins. This approach simplifies the mechanical design while enhancing control and speed during competition runs. Using Qubo's pneumatic piston actuators, the dropper releases one marker per actuation and reloads for subsequent use. It is capable of housing two markers simultaneously to support deployment into two separate bins. To operate, the piston extends and pushes a triangular wedge downward. The wedge interfaces with an angled slot on the active cylinder; as the wedge moves down, it pushes the active cylinder forward and out of the main body, releasing the first marker by gravity. When the piston retracts, the second marker falls into the active cylinder, ready for the next actuation.

The dropper underwent several design iterations during its development. The legacy dropper mounted the piston horizontally, directly attaching it to the active cylinder. This design choice was revised to reduce the form factor, as the legacy design was T-shaped and occupied significant space on Qubo's side plate. Additionally, the piston protruded and partially obstructed one of Qubo's thrusters. The current design eliminates both these issues with a sleek and vertically oriented form factor that minimizes its footprint on Qubo's frame.

2) Powered Torpedo V2: This design is an improved version of the 2024 powered torpedo, focused on maintainability and usability, that has now been tested and deployed throughout the past year. Its maintainability is a core feature: rather than being sealed shut, it uses a double O-ring piston seal

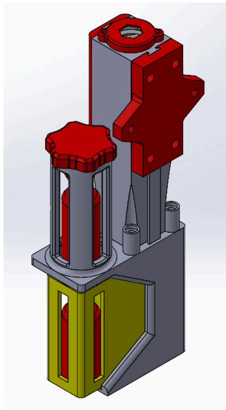

Fig. 4: Rendered CAD model of the dropper system.

to keep its electronics dry and accessible in the event of a failure. Additionally, the internal charging port further reduces electronics exposure to water.

The torpedo activates upon detecting a negative magnetic gradient, triggered by the removal of a north-pole magnetic field. It must first be engaged by the presence of this magnetic field, facilitating charging by removing the requirement for a magnet. Additionally, a custom charger displays the capacitor's voltage and charging current for convenient monitoring.

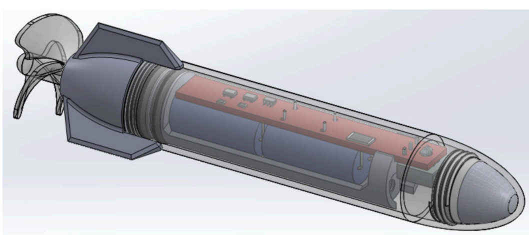

Fig. 5: Rendered CAD model of the updated powered torpedo.

3) Claw System: During the competition, Qubo must pick up irregularly shaped objects and place them in bins to score points. To accomplish this, the mechanical team developed a claw using a rack and pinion mechanism actuated by a pneumatic piston. Extending the piston pushes the rack forward, rotating the gears and opening the arms; retracting the piston closes them. The grippers are 3D-printed in TPU and designed to conform to the shape of the object being grasped. Their triangular structure allows them to easily compress and curl around objects. Each gripper has two fins that allow them to slide into the claw arms. To hold the claw together, two brackets securely encase the piston and fasten together with screws that thread into heat-set inserts.

This year, the gear teeth were changed from spur to helical profiles. This decision was made to reduce backlash in the gear meshing. The spur gears had a small gap that enabled the claw arms to rotate ten degrees even with the gears held in place. Given the length of the claw arms, this backlash produced an inch of play at the tips of the arms. Such imprecision made the legacy claw's grip unreliable and prevented it from grasping slender objects. Switching to helical gears nearly eliminated this backlash and the resultant play. Additionally, the gripping surfaces were textured with spikes and a coating of Flex Seal to improve friction compared to the legacy claw's smooth TPU surface.

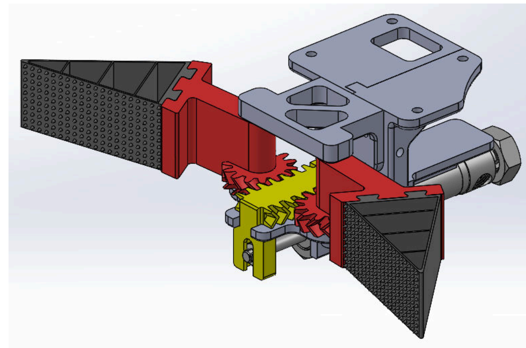

Fig. 6: Rendered CAD model of the claw system.

C. Electronics Updates

Qubo's main electrical hull (Fig 7) features a backplane system with four daughter card PCBs: a power card, two thruster cards, and a miscellaneous card. Additionally, it uses a separate pneumatics card to actuate solenoid valves controlling Qubo's end effectors. This year, the power card, backplane, and pneumatics cards were redesigned.

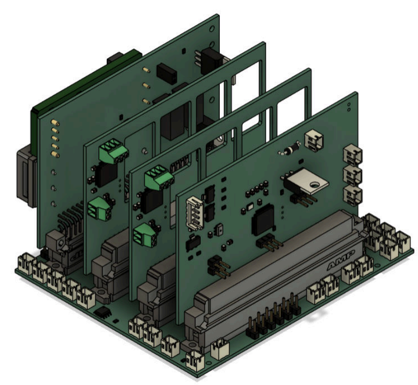

Fig. 7: Rendered CAD model of Qubo's Backplane electrical system.

1) Power Card Improvements: In previous years, Qubo's autonomous performance was severely limited by a lack of data collection. This was caused by the competition requirement that divers must engage the kill switch prior to returning the robot to begin the next run. When this kill switch was engaged, volatile memory on Qubo's Jetson was immediately lost. This volatile memory constituted all data from the autonomous run, effectively limiting software development to tethered testing time. To combat this, a second kill switch was added to Qubo's power daughter card. This new kill switch opens power supply circuitry for all of Qubo's components except the Jetson. This will allow Qubo's thrusters and end effectors to be safely disengaged while retaining volatile data to be later downloaded using the tether.

Separately, Qubo has also faced an issue where heat generated from the DC-DC converters on the power card caused Qubo's adjacent T200 ESCs to fail. The legacy power card addressed this issue by repositioning the DC-DC converters to face away from the thruster daughter cards. This proved effective in practice, though no quantitative verification was performed. More broadly, there remained no method of quantitatively monitoring the temperature within Qubo's electrical hull to prevent hardware failures. As such, a new Microchip Technology MCP9804 digital temperature sensor was incorporated on the new power card design and connected to Qubo's I2C bus to enable this data collection.

2) Backplane Improvements: One of the primary focuses of this year was to maximize Qubo's uptime for software testing. This meant that leaving hardware disassembled for extended durations was no longer acceptable, and Qubo needed to be brought back to operational status at the completion of any hardware installation or repair. The increased frequency of hardware changes drove up mating cycles on backplane connectors significantly, resulting in several mechanical failures. Compounding this issue, the legacy backplane did not consider ergonomic assembly; it only optimized electrical layout convenience. As such, many receptacles on the new backplane were repositioned to minimize wire run length and associated wire strain. At the same time, the new backplane included additional traces to implement the aforementioned temperature sensor and new kill switch from the power card.

3) Pneumatics Card Redesign: In conjunction with the mechanical redesign of the pneumatics hull, the pneumatics solenoid controller PCB was also redesigned to address several known limitations. The legacy design faced challenges with current inrush, communication protocol restrictions, and lack of testability. This year's design makes key improvements to address these issues. First, an Adafruit TCA4307 Hot-Swap I2C Buffer was installed on this card for slave side protection. This was included with the intention of mitigating the risk of I2C bus faults from hot-swapping, a failure mode that ended the team's competition run in 2025. Additionally, a custom ramp-rate controller was introduced on the 12V bus to limit inrush transient current upon startup or opening of a solenoid valve. Second, an STMicroelectronics STM32G0B1RE microcontroller replaced the legacy Adafruit PCA9685 PWM controller for solenoid valve actuation. While including this STM32 increased design complexity, it was ultimately incorporated to allow for communication protocol hardware flexibility while retaining backwards compatibility during our embedded systems' transition from I2C to CAN bus. Third, the new pneumatics card featured 16 1mm pitch male pinheader test points that may be easily probed with the lab oscilloscope. These test points are located at key nodes in the circuitry of the card to best support debugging efforts, namely the I2C bus, CAN bus, power rails, and solenoid valve excitation rails. Together, these changes position the pneumatics card to support reliable operation in future competition runs.

D. Embedded System

Updating Qubo's firmware is an ongoing project from the past several years. Motivated by I2C signal attenuation over long wires and scalability concerns, a CAN Bus replacement has been in development. Qubo's electrical system features four cards with an STMicroelectronics STM32G0B1RE microcontroller: two thruster cards, one pneumatics card, and one miscellaneous card. As such, the embedded system is distributed between four controllers. Since each controller only handles a small number of fast tasks, we have opted not to use an RTOS in favor of simpler, single-purpose embedded programs. Each controller monitors the CAN bus, determining which messages are addressed to itself using the ID of the CAN frame. It processes the frames' data according to its function: each thruster controller drives four ESCs, the pneumatics controller drives four solenoid actuators, and the miscellaneous controller interfaces with each peripheral in its respective protocol. Debugging and repairs for this project were facilitated by the backplane's aforementioned modular design.

E. Localization

Qubo has historically tracked its motion using two primary sensors: a VectorNav VN-100 [5] IMU that measures its orientation, angular velocities, and linear accelerations, and a Bar02 Pressure Sensor [6] for depth calculation. The measured orientation and depth were treated as ground truth without sensor fusion, and full position information was unavailable. Over the past several years, we have focused on estimating Qubo's position by combining the information from these sensors with a new WaterLinked DVL-A50 [7] and integrated IMU that senses linear velocities and orientation. This was done with an implementation of an Extended Kalman Filter from robot_localization [8], configured to fuse these data sources to refine estimates of direct sensor outputs and estimate Qubo's lateral position from correlated measurements.

F. Controls & Autonomy

Qubo's controls system consists of PID feedback controllers on all six axes of motion, capable of navigating Qubo to any given target position and orientation. Each axis controller can be independently configured between controlling the position or velocity of that axis, or set to inactive. Furthermore, the system can toggle between controlling Qubo's lateral position in either its world or body frame. Using these configuration options combined with detailed documentation for each behavior we write, we can easily configure Qubo to perform a wide variety of tasks. This year we also transitioned from using the BehaviorTree.CPP library [11] to using py_trees [12]. Using py_trees enables us to make changes immediately during limited pool testing times by eliminating compilation times, and offers a simpler framework while retaining the core functionality we require.

G. Computer Vision

In previous years, we used a fine-tuned YOLOv11 Nano model [4] for object detection. However, this fully supervised approach required thousands of manually labeled images, demanding many man-hours of nightly labeling and delaying deployment testing until the second day of competition. This year, we developed a human-in-the-loop automatic labeling pipeline that enabled training and evaluation on the first competition day. We replicated the YOLOv11 Nano architecture and incorporated self-supervised contrastive learning, augmentation consistency, semi-supervised pseudo-labeling, and hard negative mining for the "red_pipe" and "torpedo_circle" classes. This produced a strong detector from only 100 labeled images. We then ran inference on the remaining unlabeled images using the LOST [21] labeling platform, allowing labelers to quickly approve or refine predicted boxes. Finally, continued fully supervised pre-training generated the model deployed on Qubo. The entire pipeline requires only a few hours.

To estimate task position and orientation, we use morphology, clustering and Perspective-n-Point [10] methods. After obtaining bounding boxes from the detector, K-means clustering groups pixels by color while reducing background noise by sampling roughly 7% of pixels. This isolates stable feature points that are mapped from image coordinates to object world coordinates, producing translational and rotational vectors. For the torpedo board task, we isolate the red circular targets using Mini-Batch KMeans clustering on the cropped image, selecting the cluster with the strongest red dominance score. The resulting mask is refined with median blur and thresholding, and contour detection identifies the largest connected component. An ellipse fitted with OpenCV's fitEllipse provides the circle center and axes, and the ellipse's four cardinal points are used as the 2D correspondences for pose estimation with Perspective-n-Point. The autonomy stack uses these results to align with the torpedo board in the optimal torpedo firing position.

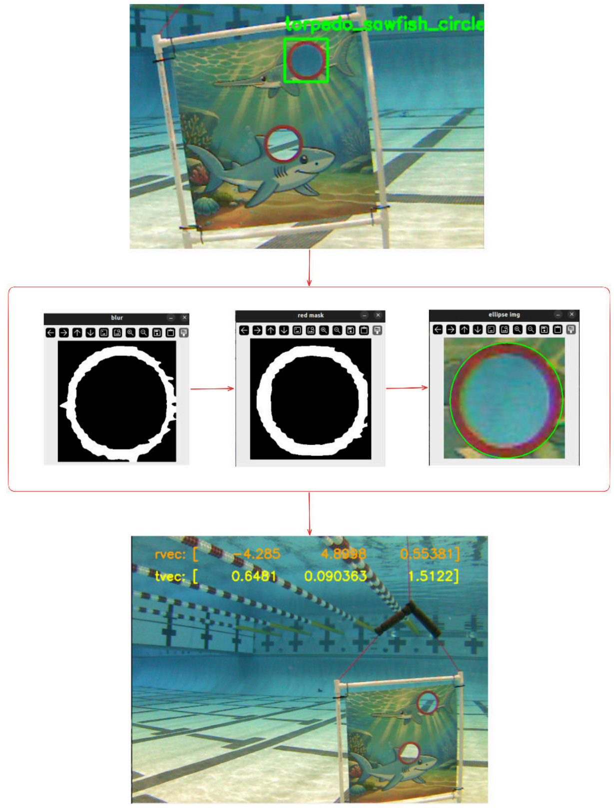

Fig. 8: Pose estimation of a torpedo target.

H. Autonomous Torpedo

The autonomous torpedo shown in Fig. 9 has been in development over the past year after it grew from the powered torpedo project with the goal of creating a fully autonomous vehicle that fits within the size constraints of a torpedo. For its strategic success, maintainability, maneuverability, and strict adherence to the 2x2x6 inch bounding box are the primary design requirements. For that reason, the design and component selection were inspired by small FPV drones.

The mechanical design of the autonomous torpedo is divided into three parts: the body, the head, and the inner skeleton. To achieve maintainability, the torpedo has a double O-ring piston seal between its body and head, with all of the inner electronics on the skeleton connected to the head. This seal facilitates removal of internal components for maintenance. Likewise, maneuverability required minimizing net buoyancy. This motivated the use of a non-circular cross-section for the body to reduce wasted volume, although this imposed even stricter dimensional constraints.

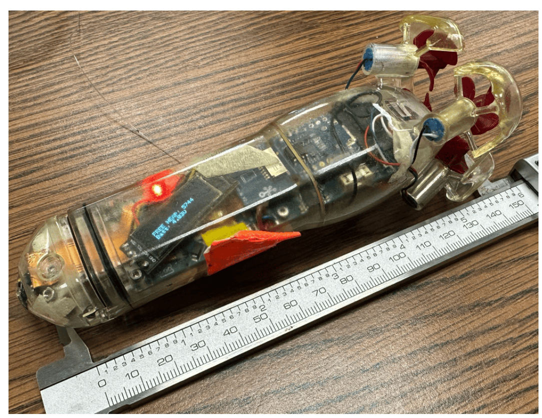

Fig. 9: Photo of the autonomous torpedo with a ruler for scale.

Given the autonomous torpedo's strict size constraints, less documented and precise microcontrollers and sensors were selected in favor of their compact form factor. This necessitated the development of the custom main PCB and sensor flex PCB, shown in the electrical block diagram in Fig. 10. The main controller used is the STM32G473RCT6 due to its small size and numerous GPIO pins. Using the main I2C bus, this module controls the Grove AI vision module V2 [13] which is used for torpedo target identification. The most important sensors used are the ICM-42688-P accelerometer and gyrometer [14],

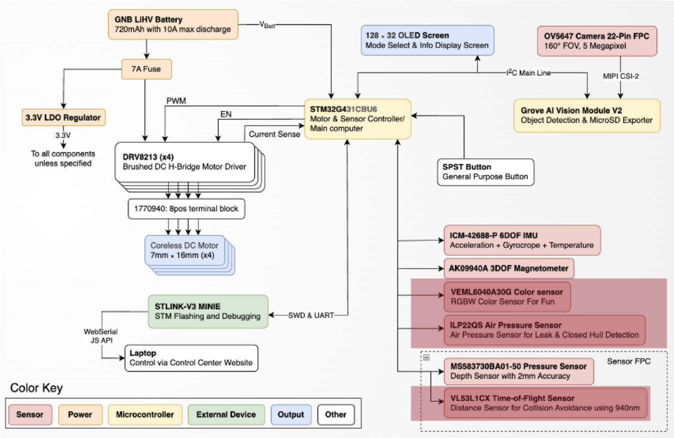

Fig. 10: The electrical block diagram of the autonomous torpedo.

the MS583702BA01-50 depth sensor [15], and AK09940A magnetometer [16], all of which are used for localization. The system is powered by the Gaoneng 720mAh single-cell LiHV battery [17] as is commonly used in FPV drone racing. The electrical system is protected by a 7A fuse on the main power supply while the STM32 monitors battery voltage with the ability to cut power to the entire board via a MOSFET latching circuit.

III. Testing Strategy

A. Autonomous Torpedo Systems Testing

Because the autonomous torpedo uses an STM32 chip as the main computing platform with limited memory (32KB) and library support, a large amount of the testing and debugging framework for the autonomous torpedo had to be custom-built. The Autonomous Torpedo Control Center as shown in Fig.11, provided live logging, plotting, and viewing of the autonomous torpedo's sensor and localization outputs. The website also provided a convenient mechanism for users of the torpedo to control the device, as Linux and ROS2 are far too resource intensive for the device.

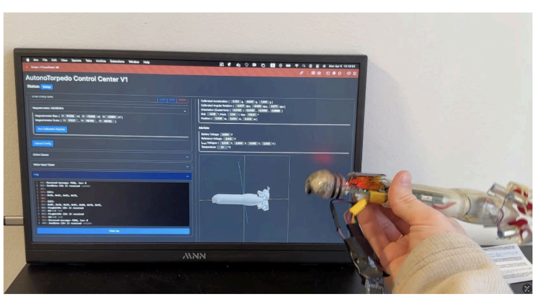

Fig. 11: Photo of the autonomous torpedo control center website opened while connected to the autonomous torpedo.

B. Qubo Systems Testing

1) Simulation: In previous years, our simulation, built in Gazebo [18], modeled our home pool in the Neutral Buoyancy Research Facility (NBRF). This simulation only included the prequalification gate and pipe; it did not feature any competition tasks. This year, we implemented a simulation in Gazebo that closely replicates the competition pool at the Woollett Aquatics Center, including all tasks from this year and textures that match the markings of the pool. Blender [19] was used to model each task and the pool, then apply all textures. The simulation enables us to test our behavior validity and stability before physical pool testing, and supports multiple configurations of task placement and orientation.

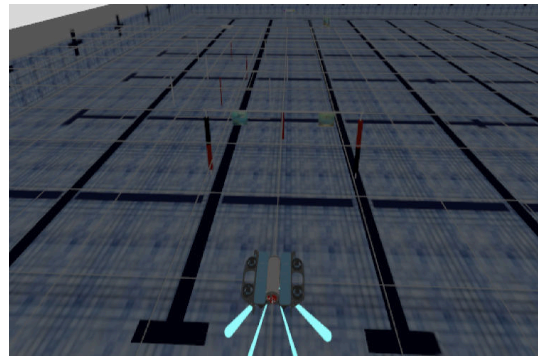

Fig. 12: Gazebo simulation of the competition pool and tasks.

2) Competition Pool: As with our simulation, in previous years our physical testing was was restricted to the NBRF pool, whose relatively small lateral size and large depth of which limited the tasks we could test. This year, we have secured access to our university's Olympic-size swimming pool for testing. Combined with custom-built replicas of the competition tasks, this has enabled us to create a testing environment that closely matches the competition environment in the Woollett Aquatic Center. See Appendix E for details on our individual component and full system testing plans and results.

Acknowledgments

Robotics @ Maryland would like to thank the Space Systems Laboratory at the NBRF for lending their lab, time, and tools to the team. The team would also like to thank Dr. Akin specifically for acting as the team advisor over the years, as well as our additional advisors, Dr. Mitchell and Dr. Zhang. We also welcome Russell Shomberg as a new advisor this year, who contributed the design concept for the claw grippers [20]. We are grateful to Ivan Penskiy for allowing the team to use space and tools in the Robotics and Autonomy lab on campus, as well as the Eppley Recreation Center and staff for use of their swimming pool for testing. R@M also thanks the Aerospace, Electrical and Computer Engineering, Mechanical Engineering, and Computer Science departments for lending their space, time, and money to the team. Finally, the team thanks TRX Systems, Robotic Research LLC, the Department of Mechanical Engineering and the Maryland Robotics Center for their generous donations.

References

[1] Open Robotics, "Robot Operating System: Humble Hawksbill." https://docs.ros.org/en/humble/index.html
[2] H. Kalbhor et. al., "Qubo VI: RoboSub 2025 Technical Report", 2025.
[3] STMicroelectronics, "Mainstream Arm Cortex-M0+ 32-bit MCU." https://www.st.com/en/microcontrollers-microprocessors/stm32g0b1re.html
[4] Ultralytics, "YOLOv11." https://docs.ultralytics.com/models/yolo11/
[5] VectorNav, "VN-100 IMU/AHRS." https://www.vectornav.com/products/detail/vn-100
[6] Blue Robotics, "Bar02 Depth/Pressure Sensor." https://bluerobotics.com/store/sensors-cameras/sensors/bar02-sensor-r1-rp/
[7] Water Linked, "DVL-A50." https://docs.waterlinked.com/dvl/dvl-a50/
[8] T. Moore et al., "A Generalized Extended Kalman Filter Implementation for the Robot Operating System," Charles River Analytics, 2023.
[9] DeepWater Exploration, "exploreHD 3.0 Underwater ROV/AUV USB Camera." https://dwe.ai/products/explorehd
[10] OpenCV, "Perspective-n-Point (PnP) pose computation" https://docs.opencv.org/4.x/d5/d1f/calib3d_solvePnP.html
[11] Auryn Robotics, "BehaviorTree.CPP Library." https://www.behaviortree.dev/
[12] Daniel Stonier, "py-trees package." https://pypi.org/project/py-trees/
[13] Seeed Studio, "Grove AI Vision Module V2." https://wiki.seeedstudio.com/grove_vision_ai_v2/
[14] TDK InvenSense, "ICM-42688-P: High Precision 6-Axis MEMS MotionTracking Device" https://invensense.tdk.com/wp-content/uploads/2020/04/ds-000347_icm-42688-p-datasheet.pdf
[15] TE Connectivity, "MS5837-30BA: Ultra-small, gel-filled, pressure sensor with stainless steel cap" https://www.te.com/commerce/DocumentDelivery/DDEController?Action=showdoc&DocId=Data+Sheet%7FMS5837-30BA%7FB1%7Fpdf%7FEnglish%7FENG_DS_MS5837-30BA_B1.pdf%7FCAT-BLPS0017
[16] AsahiKASEI, "AK09940A: Ultrahigh precision 3-axis Electronic Magnetometer" https://www.mouser.cn/datasheet/2/1431/ak09940a_en_datasheet_myakm-3244294.pdf
[17] Race Day Quads, "Gaoneng GNB 3.8V 1S 550mAh 100C LiHV Whoop/Micro Battery w/ Plastic Head - A30" https://www.racedayquads.com/products/gaoneng-gnb-3-8v-1s-720mah-100c-lihv-whoop-micro-battery-w-plastic-head-a30
[18] Open Source Robotics Foundation, "Gazebo Simulator." https://gazebosim.org/docs/
[19] The Blender Foundation, "Blender'.' https://docs.blender.org/
[20] Black Ram Industries, "Fin Gripper (Robotic/Prosthetic Hybrid) - Mark VI" https://www.printables.com/model/26078-fin-gripper-roboticprosthetic-hybrid-mark-vi
[21] J. Jäger et. al., "Label Objects and Save Time.“ https://lost.training

APPENDIX A: MEMBERS

Admin

Harshita Kalbhor
Kurt Kovacs
Amy Naylor
Reese Snook

Mechanical

William Trasatti
Tommy Wolcott
Ava Pavese
Michael LeVesque
Collin Rogers
Maanav Subramanian
Ken Schumer
Kaan Gin
Praneel Mangalapurapu
Ethan Mulcahy
Andrew Li

Electrical

Eli Leopold
Anthony D'Costa
Liam Brosnan
Nitya Sharma
Kirankishore Jayaprakash
Marcus Allen
Fizza Jafry
Sophie Grigg
Arjun Sundaram
Sebastien Pop
Kaan Gin

Software

Ishaan Ghosh
Jai Rastogi
Vaishnavi Samprati
Hengyu Tao
Anusha Sharma
Michael Kittredge
William Conner
Leo Jiang
Max Jin
Eric Xu
Nicolas Preda
Alvin Shen

Appendix B: Outreach

Robotics @ Maryland (R@M) began as a small team focused solely on building autonomous underwater vehicles (AUV) for the RoboSub competition. Over time, that mission grew. Today, R@M builds three autonomous systems: Qubo, an AUV; Terry, a Mars rover prototype; and Testudog, a quadruped robot. While our technical reach has expanded, our foundation remains people-driven: using robotics as a platform for education, mentorship, and community.

This year, R@M's outreach efforts were demonstrated on **Maryland Day**, held on April 25th. With an expansive course setup designed, our goal was to make robotics tangible for all. Visitors could swim Qubo through a Robosub course recreation, drive Terry across terrain, and watch Testudog walk in real time. Props and themed displays mimicked competition exactly, drawing in crowds who may never have considered engineering as something accessible to them. The energy attracted parents, kids, alumni, and even the the President of the University of Maryland. Members staffed stations for hours, breaking down PID loops, sensor fusion, and autonomous navigation into conversations that met each visitor where they were. Our goal was the same as ever: make engineering feel less like a mystery and more like a possibility.

Our **Design Review** events extend that mission into the academic community. Held each semester and open to faculty, sponsors, peers, and anyone curious about the engineering process, these nights offer a transparent look under the hood of everything we build. We share what works, what is confusing us, and how we plan to improve. We talk honestly about system tradeoffs, failed approaches, and open design questions. This openness invites real feedback. Sponsors and faculty push back on our assumptions, point out gaps in our reasoning, and offer perspectives we would not have reached on our own. The result is better engineering and stronger communication skills across the team.

R@M also builds community from the inside out. Each spring, we carry on our **Pi Day Tradition**: members stand outside McKeldin Library and get pied in the face by anyone willing to make a donation. What started as a lighthearted fundraiser has become one of our most visible moments on campus, a reminder that engineering clubs are made of real people who do not take themselves too seriously. Passersby stop, laugh, and ask what we do, which opens more conversations about robotics than any flyer ever has.

In the fall, R@M co-hosted a **Cookout** with Terrapin Rockets, another engineering organization on campus. Over burgers and volleyball, members from both clubs connected around a shared identity: students who build things, solve hard problems, and want to be around others who do the same. Events like these strengthen the broader engineering community at Maryland and remind us that collaboration starts with simply showing up for each other.

Fig. 13: Teaching kids how to drive Qubo on Maryland Day.

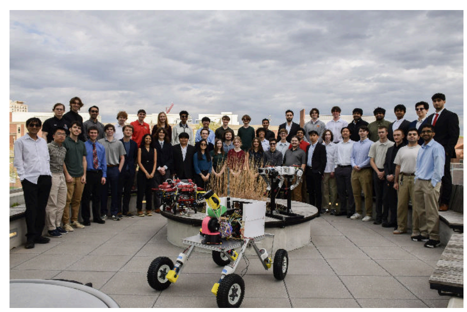

Fig. 14: R@M members at the Spring 2026 Design Review.

Fig. 15: Photo Collage from Pi Day

Fig. 16: R@M co-hosting a cookout with Terrapin Rockets.

Appendix C: Component List

Component	Vendor	Model/Type	Specs	Custom/ Purchased	Cost	Year of Purchase
Buoyancy Control	Neutral Buoyancy Research Facility Stock	Foam	Pink	Custom	$0	N/A
Frame	N/A	N/A	Aluminum, Water Jetted, 0.4 x 0.3 x 0.3 meters	Custom	$200	2017
Waterproof Housing	Blue Robotics	6" Watertight Enclosure	Acrylic and Aluminum Endcaps	Purchased	$400	2022
Waterproof Connectors	MacArtney	BH3F SubConn connectors	3 Pin Connector, Female	Purchased	$77.64	2024
Thrusters	Blue Robotics	T200	11.2 lbs forward thrust, 350 Watts	Purchased	$200	2017
Motor Control	Blue Robotics	Basic ESC	7–26V, 30 Amps max	Purchased	$288 ($36 each $\times$ 8 ESCs)	2017
High Level Control	N/A	PID Control	N/A	Custom	$0	N/A
Actuators	Bimba	6498K632	Double-Acting, Universal Mount, 3/4" Bore Size, 1" Stroke Length	Purchased	$183.56 ($45.89 each $\times$ 4)	2026
Propellers	Blue Robotics	BR-100042-001	T200 CCW Propeller	Purchased	Included in Thruster Cost	2017
Battery	Zeee	6S6000	6S LiPo Battery 6000mAh 22.2V 100C with EC5 Connector	Purchased	$120.99	2026
Converter	Flex Power Modules	2166-PKB2213DPIHS-ND	DC DC Converter 12V 240W	Purchased	$62.10	2025
Regulator	N/A	N/A	N/A	N/A	N/A	N/A
CPU	NVIDIA	Jetson Orin Nano	6-core Arm Cortex-A78AE CPU with 8GB LPDDR5 memory and 1024-core NVIDIA Ampere GPU (32 Tensor Cores)	Purchased	$650	2024
Internal Comm Network	N/A	N/A	Ethernet, USB, I2C, CAN Bus, UART	Custom	$0	N/A
External Comm Interface	N/A	N/A	Ethernet	Custom	$0	N/A
Compass	N/A	N/A	N/A	N/A	N/A	N/A

Continued on next page...

Component	Vendor	Model/Type	Specs	Custom/ Purchased	Cost	Year of Purchase
Inertial Measurement Unit (IMU)	Vectornav	VN-100	2 degree accuracy heading and tilt	Purchased	$1100	2023
Doppler Velocity Log (DVL)	Waterlinked	DVL-A50	$\pm$3.75 m/s range, $\pm$0.4% accuracy	Purchased	$6950	2024
Manipulator	N/A	N/A	Air piston operated rack and pinion mechanism with flexible grippers	Custom	$0	N/A
Algorithms	N/A	Custom	N/A	Custom	$0	N/A
Vision	DeepWater Exploration Inc.	DWE-EHDUSB-WL	exploreHD 3.0 (400m) Underwater ROV/AUV USB General Vision Camera	Purchased	$974.97 ($324.99 each $\times$ 3)	2025
Acoustics	N/A	N/A	N/A	N/A	N/A	N/A
Localization & Mapping	Charles River Analytics, Inc.	robot_localization	Extended Kalman Filter	Custom	$0	N/A
Autonomy	Daniel Stonier	py_trees	N/A	Custom	$0	N/A
Inter-Vehicle Communication	N/A	N/A	N/A	N/A	N/A	N/A

Software Algorithms & Languages

Component	Type	Usage
Algorithms: Vision	OpenCV	Various basic vision processing algorithms
Algorithms: Other	Kalman Filter	State estimation and sensor fusion
Programming Language 1	C++14	N/A
Programming Language 2	Python 3	N/A
Programming Language 3	C11/Armv6-M/Armv7E-M	N/A
Open-Source Software	ROS2 Humble Hawksbill	N/A
Open-Source Software	Docker	N/A

Autonomous Torpedo Component List

Component	Vendor	Model/Type	Specs	Custom / Purchased	Cost	Year of Purchase
Frame	JLCPCB	SLA Print	8001 resin with oil spraying finish	Custom	$9.85	2025
Thrusters	Hobbypower	DC coreless brushed motors	7mm x 16mm, 55000rpm	Purchased	$9.58	2025
Motor Control	Texas Instruments	DC motor driver	1.65-11V, 4A, with current sensing	Purchased	$0.95 x 4	2025
PWM control	ST Microelectronics	STM32G431	4 PWM channels used	Custom	$2.26 x 2	2025
Propellers	N/A	FDM printed PLA propellers	0.95mm shaft hole	Custom	$0	2025
Battery	Gaoneng	GnB-GNB5501S100PHV	3.8V, 550mAh, 100C, 1S	Purchased	$7.49	2025
Main Computer	Seeed Technology	Xiao ESP32-S3	ESP32-S3R8	Purchased	$7.99	2025
Internal Comm Network	I^2C, UART					
External Comm Network	UART					
IMU	TDK Invensense	ICM-42688-P		Purchased	$2.51	2025
Magnetometer	AsahiKASEI	AK09940A	10nT / LSB	Purchased	$7.09	2025
Vision	Waveshare	OV5647	5MP camera with 160deg FOV	Purchased	$4.67	2025
Vision Module	Seeed Studio	Grove AI Vision Module V2	30FPS	Purchased	$17.99	2025

Appendix D: Autonomous Torpedo Testing Procedure and Results

A. Torpedo waterproofing panel

	Powered Torpedo o-rings	Autonomous Torpedo o-rings 1	Autonomous Torpedo o-rings 2	Camera Seal 1	Camera Seal 2	Pressure sensor seal	Wire penetrator seal
Before water tests							
Weight measurement 1 (g)	23.7	12.1	12.5	15	14.7	12.9	6
Weight measurement 2 (g)	23.71	12.14	12.6	14.99	14.86	12.91	5.94
After quick dip in the water							
Weight measurement 1 (g)	23.8	12.1	12.6	14.9	14.8	13.3	5.9
Weight measurement 2 (g)	23.79	12.17	12.61	14.99	14.9	13.37	5.94
Notes	Feels dry inside	No visible water	No visible water	No visible water	No visible water	No visible water	No visible water, but some sloshing
After 2min at 8ft							
Notes	Feels dry inside	No visible water	No visible water	No visible water (hard to see)	No visible water (hard to see)	No visible water (hard to see)	Absolutely has some water
After 12min at 8ft							
Weight measurement 1 (g)	24.1	12.3	12.9	15	14.8	13.8	6.8
Weight measurement 2 (g)	23.99	12.35	12.94	15.06	14.89	13.76	6.8
Notes	Feels dry inside	Feels dry inside and popped open	No pop, but feels dry	No visible water	No visible water	No visible water, sensor still works	Half full of water
Results							
Success/Failiure	Success	Success	Success	Success	Success	Success	Failiure

Fig. 17: Waterproofing panel test results.

For this test, all of the objects were tied to the same weight (Fig. 18) and then measured individually on two different scales. First all objects were lowered in by hand, swished around and then measured and observed. Then the tests were lowered to 8 feet for 2 minutes and brought back up to be observed. Finally, everything was lowered back to 8 feet for 10 more minutes and then brought back up and measured and observed. Whenever a test was being weighed, all water on the surface was wiped off.

Overall, only the wire penetrator test failed, but since the pressure sensor also had a wire pentrator and didn't fail, everything was proven to be waterproof-able so the final 3D-prints were ordered. All of the tests gained a small amount of water weight, but this is because all materials absorb water to some extent. The pressure sensor test gained a decent amount more than the others, but that was due to the masking-tape labels on the output wires which caused this.

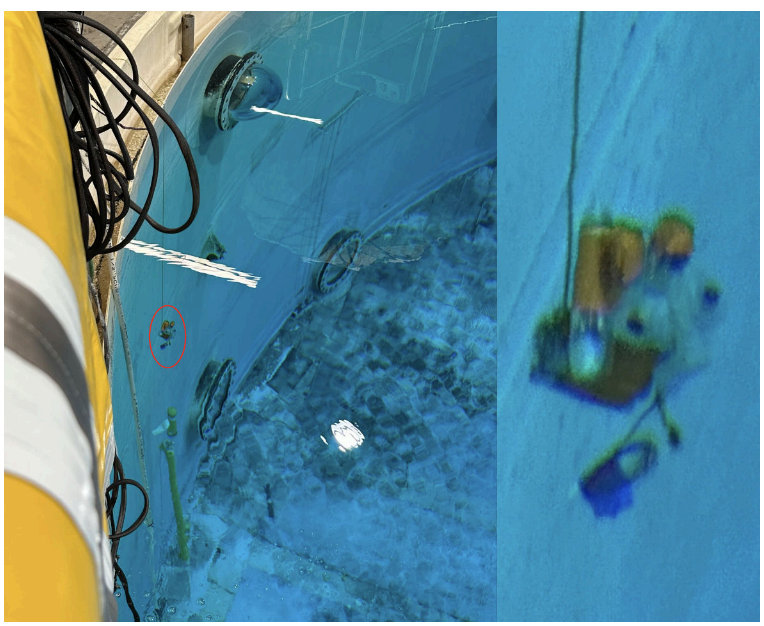

Fig. 18: Waterproofing panel test setup, with the panel zoomed into on the right.

Appendix E: Qubo Testing Procedure and Results

A. Individual Component Testing

Before any integrated pool testing, all critical subsystems were validated individually using repeatable diagnostic procedures. These tests ensured that failures found beforehand could be isolated to specific hardware or software components and fixed before full system testing. Component-level testing also significantly reduced debugging complexity when issues arose during full-system pool sessions. By validating sensors, actuators, communications, and software independently before deployment, the team could rapidly isolate failures encountered during autonomous operation as well; overall, this layered validation strategy improved reliability during competition testing and reduced time lost to hardware debugging at the pool.

1) Actuation and Electrical Testing:

- **I2C Device Validation**: All thruster cards, pneumatics cards, and depth-sensor interfaces were tested using low-level I2C diagnostics prior to operation. Address scans and register reads verified communication with each embedded device and helped rapidly isolate electrical or wiring failures during debugging sessions.
- **Thruster Testing**: Thruster validation was performed in two stages:
  * direct PWM-based individual thruster testing, followed by
  * full-system thrust testing using ROS wrench commands

  Full-system tests verified coordinated vehicle motion along commanded axes, while individual tests isolated failures to specific thrusters, ESCs, or wiring channels. This methodology proved critical during spring testing, where asymmetric motion was traced to intermittent thruster-card connectivity issues.
- **Pneumatics Testing**: The pneumatic system was tested by repeatedly actuating the claw, dropper, and torpedo launchers under both dry and underwater conditions, examining solenoid operation, air-pressure retention, and launcher responsiveness.

2) Sensor Validation:

- **Doppler Velocity Log (DVL)**: The WaterLinked DVL-A50 was tested prior to each deployment to verify network communication and velocity output consistency. Diagnostic procedures confirmed Ethernet connectivity, ROS node initialization, and valid velocity estimates before the sensor was submerged. Because the DVL can overheat outside water, validation was intentionally performed immediately before deployment.
- **Inertial Measurement Unit (IMU)**: The VectorNav VN-100 IMU was validated through both direct serial communication and ROS integration tests. Raw sensor streams were inspected to verify stable orientation and acceleration outputs, after which the ROS driver was launched to confirm correct initialization and message publication. These tests ensured reliable heading estimation for localization and control.
- **Pressure Sensor**: The Blue Robotics Bar30 Pressure Sensor was tested independently by verifying stable pressure-derived depth estimates on the ROS /depth topic while running a corresponding ROS node that reads pressure values from the sensor.
- **Cameras**: Each DeepWater Exploration exploreHD 3.0 Underwater ROV/AUV camera was individually validated prior to pool testing by verifying USB enumeration, image streaming, and ROS/OpenCV compatibility. Diagnostic scripts identified camera USB indices were valid and ensured that all forward- and downward-facing cameras produced stable image feeds before deployment.

3) Software and Communications Testing: Subsystem launch tests ensured that vision, localization, controls, and teleoperation nodes initialized correctly and could communicate across the onboard ROS2 network. Logging and diagnostic visualization tools such as Foxglove (see Figure 22) were also used to verify live telemetry during testing. Before each deployment, the team verified:

- SSH connection availability
- Ethernet communication consistency under load
- ROS topic functionality and communication to landside computer
- Position tracking, movement control and vision systems startup

B. Full System Testing Plan

Throughout the 2025–2026 development cycle, Qubo was tested in staged environments ranging from bench validation and the Neutral Buoyancy Research Facility (NBRF) to full-system pool testing in the Eppley Recreation Center Olympic pool, in which runs were conducted in competition-like conditions using full-scale RoboSub props, enabling evaluation of vision, localization, controls, and end-effector reliability under realistic competition conditions. Each testing session defined clear objectives beforehand and logged observations during and afterwards, including system failures, environmental effects, and task completion results. This process enabled rapid iteration on hardware and software while maintaining traceability between observed issues during pool testing and fixes implemented later on.

Fig. 19: Team members preparing Qubo for deployment into the mock competition pool.

1) Eppley Recreation Center Pool Testing Schedule: Due to the cost of reserving a lane in our university's pool and the significant time needed to transport Qubo and all competition materials back and forth from the Eppley Recreation Center, we decided to have four tests for 4-6 hours in our fall semester, and six tests in the spring semester as our development cycles quickened. These tests served as test runs for pool tests at competition, and all of our hardware and software project development plans revolved around these dates as checkpoints at which we could assess our progress in all areas. The weeks in between each test allowed us to fix problems that we found and iterate on possible improvements that we noticed during each test.

TABLE II: Pool Testing Schedule

Pool Testing Times	
Fall Semester	**Spring Semester**
Oct. 18, 2025 3–9 PM	Feb. 28, 2026 6–9 PM
Oct. 19, 2025 3–9 PM	Mar. 1, 2026 3–7 PM
Nov. 2, 2025 5–9 PM	Mar. 28, 2026 4–9 PM
Nov. 23, 2025 5–9 PM	Apr. 11, 2026 4–7 PM
	May 2, 2026 3–8 PM
	May 8, 2026 4–9 PM

2) Risk Assessment: Water testing with an AUV presents specific risks related to electrical safety, pool usage, and human interaction with equipment. The primary risks identified include water intrusion into electronic components, swimmer injury, equipment damage, communication failure, pool contamination, and interference from other pool users. To mitigate these, the team has established multiple layers of protection. All electronic housings underwent leak testing in the Neutral Buoyancy Research Facility (NBRF) before each deployment, and fuses were installed to prevent electrical hazards. The tether provided constant physical control of Qubo, while both a topside hard kill switch and a wireless emergency stop allowed for immediate shutdown in case of malfunction. The swimmer assigned to assist Qubo has been specifically trained in underwater handling of the robot, and all operations occurred within a reserved and clearly marked lane with one lane buffer to prevent contact with bystanders. Qubo is also equipped with 3D printed safety fins to prevent any damage from the thrusters to the swimmer or the pool wall in case the AUV moves off course.

Emergency procedures are in place for all foreseeable incidents. In the event of loss of control, the operator immediately activates the E-stop and the swimmer will retrieve Qubo only when it has fully stopped moving. If any sign of leakage or electrical failure is detected, the test stops immediately, the robot is removed from the pool, and electronics are dried and inspected before resuming. Should a swimmer exhibit distress, the test is halted and the on-duty lifeguard notified to take control of the response. To ensure accountability, the team maintains a written test log documenting all activities, results, and safety notes for each session. Any incidents, however minor, are reported to both the faculty advisor and RecWell coordinator. Through these protocols, Robotics@Maryland ensures that all testing is conducted responsibly, safely, and in full compliance with university and facility standards. Below is a table (Table II: Pool Testing Risk Assessment) that summarizes the above information:

TABLE III: Pool Testing Risk Assessment

Risk	Description	Likelihood	Impact	Mitigation
Electrical short or leak	Water intrusion into electronics	Low	High	Conduct full waterproof testing before operation; ensure sealed connectors and fuses.
Swimmer injury	Collision with Qubo or props	Low	Medium	Maintain buffer lane; swimmer trained in AUV handling.
Equipment damage	Impact with pool walls or floor	Medium	Medium	Operate at low speeds; maintain tether control.
Communication failure	Loss of signal or control	Low	High	Use tethered operation and hard kill switch.
Pool contamination	Tools or materials left behind	Very Low	Medium	Enforce ”nothing left in pool” rule.
Unauthorized pool interference	Other users entering test area	Medium	Medium	Reserve lane; clearly mark buffer lane; coordinate with RecWell staff.

C. Results Summary

These full-system tests constantly revealed issues and areas of improvement across our hardware and software systems, allowing us to iterate and refine our systems consistently throughout this year. Below is a summary of some of the results we found during our tests:

1) Localization and Controls Testing: Movement control and localization were validated incrementally through teleoperation, waypoint tracking, and autonomous behavior execution. Initial October testing confirmed that Qubo's localization estimates matched observed motion in the pool and that point-to-point control could move the vehicle to commanded poses. Throughout the year, repeated tests refined PID tuning and robustness under varying buoyancy conditions.

2) Vision and Autonomy Validation: Computer vision and autonomy systems were evaluated continuously throughout the year using iterative data collection and retraining. Early pool tests in October 2025 focused on validating stereo depth estimation, gate detection, and torpedo-board localization. Initial runs demonstrated reliable gate recognition when the gate was fully visible, but unstable detections when partially occluded or viewed from extreme angles. Subsequent testing sessions were used to improve robustness through targeted data collection and retraining. During November testing, the team evaluated new torpedo-board detection pipelines and pose-estimation systems. The introduction of ellipse-fitting pose estimation and improved target segmentation worked towards developing alignment behaviours across varying board orientations.

In Spring 2026, testing expanded to full autonomous task sequences; by May 2026, among other things, Qubo could successfully:

- complete the gate task with a given role and random starting orientation
- search for, align with, and score droppers in a bin autonomously
- align with torpedo targets using vision-guided control

These tests also revealed several issues that were fixed or accounted for by changes in the functioning of our autonomy behaviours over the course of the year. Repeated testing allowed us to tune vision-based controllers to be accurate across many different system and environment conditions, find alignments for the most reliable scoring from end effector systems, and streamline the process of data collection and object detection model training through tests of this pipeline.

3) Hardware/End-Effector Systems Validation: As they are the focus of our strategy this year, the torpedo and dropper systems were tested extensively under pool conditions. Early runs focused on verifying pneumatic actuation and torpedo propulsion, and later runs transitioned to tests as a part of full task runs. Over the course of these tests, several reliability issues and facts were found and fixed or worked around in the systems onboard Qubo and those used in preparing the end effectors to be used during runs, including:

- insufficient air pressure preventing many sequential actuations
- delayed or broken torpedo firing systems due to hardware misalignment
- air compressor failures in pneumatics preparation
- dropper system errors and incomplete piston actuation preventing proper sequential dropping of markers

Alongside revealing these issues, these tests mimicking pool testing conditions at competition have given us the opportunity to optimize the process of preparing our end effectors for competition runs, minimizing the time needed to charge and load our torpedoes and droppers and prepare our pneumatics systems for use.

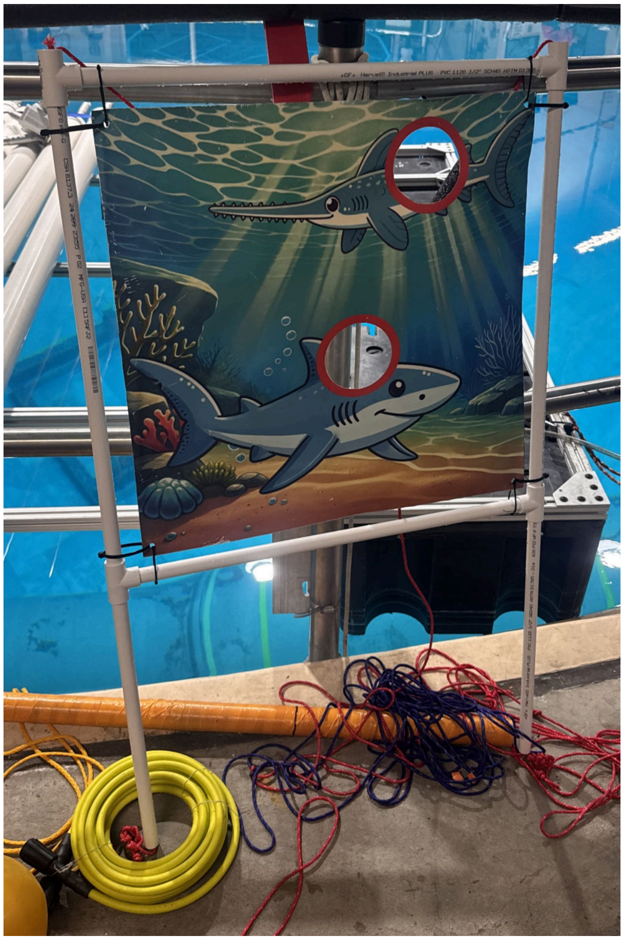

Fig. 20: Scale model of the torpedo board used for testing.

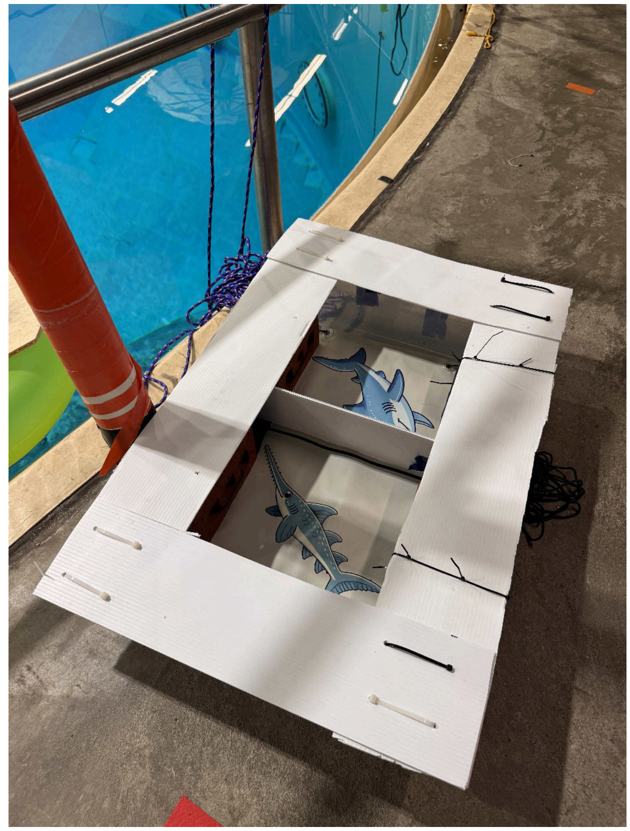

Fig. 21: Scale model of the bin used for testing.

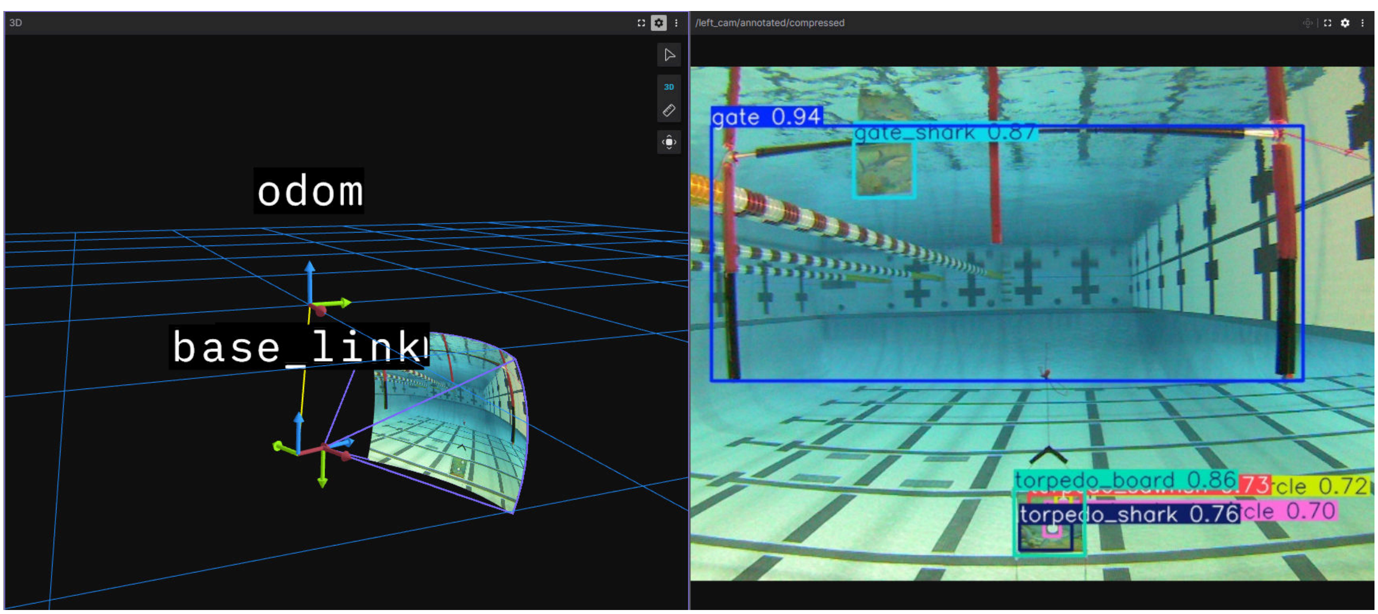

Fig. 22: An example real-time visualization of localization and vision systems during testing.