Qubo VI: RoboSub 2025 Technical Report

See Appendix A for the Team Roster

***Abstract*—The Robotics at Maryland (R@M) Team at the University of Maryland dedicated the 2024-2025 academic year to optimizing existing systems on their competitive Autonomous Underwater Vehicle (AUV), Qubo, for RoboSub 2025. This was achieved by improving the placement and mechanical design of the end-effectors and improved aesthetics for the mechanical framing system. The backplane electrical system was redesigned and replaced for increased functionality. Moreover, the team streamlined their internal software tools, running a ROS-based software system that allows for flexible execution of competition tasks [1]. These advancements in optimizing existing systems enabled Qubo to successfully complete prequalification for RoboSub 2025 and gain a competitive advantage for this year's autonomy challenge.**

Fig. 1: Qubo by the testing pool.

I. Competition Strategy

A. General Strategy

The team's strategy for the 2025 competition remains the same as the strategy for the 2024 competition: focus on completing at least one end-effector task along with the navigation tasks. We decided to repeat this strategy as the trends found by the team remained valid for the 2024 competition, and because we were unable to execute this strategy last year due to an unfortunate failure of the electrical system that ceased motor functionality after the pre-qualification runs ended. This, in combination with Qubo being in a mature mechanical state, led to a laissez-faire approach from the mechanical team to increase the amount of electrical and software optimization and development during the year. Such an approach has ensured that the electrical system is more robust and that the software team had more time for data collection this year. With this strategy in mind, the team has ensured reliability so as to not waste testing time during competition.

B. Course Strategy

With an array of improved end effectors combined with revamped controls and vision systems, Qubo has the potential to complete almost every task in the competition. Our plan this year is to rely on our navigation for the first tasks, then take advantage of our dropper, torpedo, and claw systems to complete as many of the end effector tasks as possible.

1) Heading Out and Collecting Data (Coin Flip and Gate): We will start our run with a coin flip, and rely on our fine-tuned YOLO [4] object detection system to identify the Gate and its attached images as Qubo searches for it. Once the Gate is detected and fully in view, we will use a combination of our sterero camera, OpenCV segmentation, and gate's dimensions to estimate a target point between our chosen side of the Gate. This choice will be made and stuck with depending on which of the Reef Shark or Sawfish is more visible to our system. Qubo will then move to the calculated target point and track its position using our dead reckoning algorithms. The target point will be continually refined while the Gate is in vision. We will conclude with a yaw 720° for style points.

2) Navigate the Channel (Slalom): After completing style, we will achieve and maintain a preset depth within the plane of the pipes. Stereo depth information from our vision systems will show which pipes are closest to us, and similarly to the gate also allow us to calculate a target point between the closest two pipes (which will also determine which side of the slalom Qubo enters). Qubo will head toward the target point, and upon reaching it repeat this process twice more to complete the Slalom.

3) Path and Drop a BRUVS (Bin): Last competition, we found that the Bin task is the most difficult for our object detection systems to find on our forward-facing cameras. So, we will utilize the Path marker pointing towards the Bin from the Slalom to orient Qubo towards the Bin. A combination of object detection and OpenCV methods applied to our downward-facing camera will allow us to determine the angle of the Path relative to our current orientation, which Qubo will align with. Then, Qubo will head forward until it detects the Bin and its images. Despite the lack of vertical stereo depth, we can use Qubo's tracked depth and knowledge of the depth of the pool to calculate how far the Bin is. Depending on our choice at the gate, we will align our dropper with the chosen image, descend to a predetermined small distance from the Bin at which our dropper performs well and we can still detect and maintain our alignment with the image, and drop our markers.

4) Tagging (Torpedoes): Unlike the Bin, we found that the clarity of the water and the larger size of the Torpedo task allowed our object detection systems to reliably detect the torpedo board from many points in the pool, so we will use this to find and head towards this task. Once Qubo is within a distance at which our stereo depth functions well on the board, we will use our vision systems to calculate a target point in front of our chosen image that aligns our torpedo with the target, head there and fire, back up and repeat with the other image. Our newly updated powered torpedoes' maintainability and ability to travel fast and straight allows us to fire from target points far enough from the board to gain extra points.

This year, the powered torpedoes are primarily a backup to the autonomous torpedo project, which aims to complete the torpedo task along with other tasks as its own vehicle.

5) Ocean Cleanup (Octagon): Without a passive sonar system, we rely on our object detection systems to detect most tasks, and we found last year that detecting the Octagon was less reliable. For this reason, this task is a reach goal. If we are able to navigate to the Octagon, our main focus will just be on surfacing within the ring. Upon securing those points, we will then attempt to sort the trash, using our downward-facing camera for claw alignment.

6) Return Home: Due to the clarity of the water observed at last year's competition, we are confident that we will be able to detect the back of the gate from across the pool and head back through the gate at the end of our run.

II. Design Strategy

This year, the team has chosen to optimize and improve the systems from last year [2]. Qubo has been competitive since 2023, and is almost fully developed barring some optimizations that were not made last year to time constraints. The plan this year is to finish those optimizations.

A. Pneumatics

The pneumatics system for dropper, claw, and torpedo actuation remained largely unchanged from the 2024 competition. This system could not be as optimized as other systems and subsystems because of the limited use during and after last year's competition. The electrical failure mentioned earlier not only limited use during competition but also after, as there was damage to the electronics used to operate the solenoid valves. These setbacks simply limited the testing we could perform. The team already knew this system was mechanically functional from the 2024 team's testing and other more pressing issues took priority over pneumatic system testing.

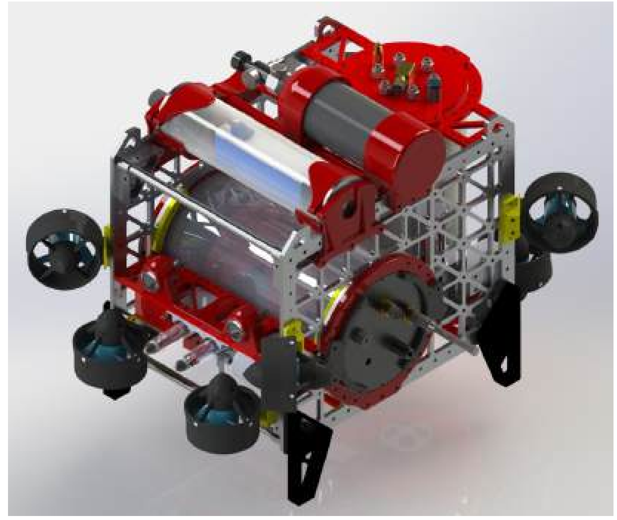

Fig. 2: Rendered CAD model of Qubo.

B. End Effectors

All major end-effectors needed to be updated. Questionable design choices were implemented to fit the dropper and the claw on Qubo, as they were not visible through the vision system. The powered torpedoes were overhauled as the design and function was not optimized for competition use.

1) Marker Dropper: The dropper end effector is designed to drop a marker into a bin of the correct color. This task is crucial to R@M's competition strategy because it focuses on completing a task historically overlooked by other teams. The dropper system was designed around the linear actuation of the piston to ensure precise marker deployment into the designated bins. This approach simplifies the mechanical design, enhances control, and optimizes speed, making it well suited for competitive environments where accuracy and performance are critical. Utilizing Qubo's pneumatic piston actuators, the dropper can release one marker per actuation and reload for subsequent uses. It is capable of housing two markers simultaneously, allowing for two distinct drops for redundancy. To operate, the piston extends forward and pushes the active cylinder out of the main body. The marker falls due to gravity. When the piston retracts, the reserve marker falls into the active cylinder. The piston can then actuate again and drop the second marker.

Throughout its development, the dropper underwent several iterations with design improvements made along the way. The core mechanism remained consistent with enhancements that focused on improving functionality and ease of use. The previous iteration of the dropper had a main body that took over 12 hours to 3D print. Additionally, the main body could only be mounted to a specific location on Qubo. The new iteration of the main body is broken into three separate parts screwed together with embedded hex nuts. This makes the dropper significantly easier and faster to 3D print if a piece breaks or a design changes. Additionally, the modularity of the new design allows the dropper to be screwed to almost any location on Qubo's framing.

2) Powered Torpedo V2: This is the new and improved version of the previous year's design focused on improving maintainability and usability. In contrast to the previous year, the powered torpedo is no longer sealed shut, instead using a double o-ring piston seal to keep its electronics dry and accessible in the case of a failure. Additionally, with the charging port now on the inside, there are two fewer hull penetrators to potentially leak water in.

Electrically, the new version only runs once the north-pole face of a magnet has moved away. This has improved usability as it can now be charged without a magnet present. Additionally, the charging circuitry is now in a custom charger which displays the capacitor's voltage and charging current.

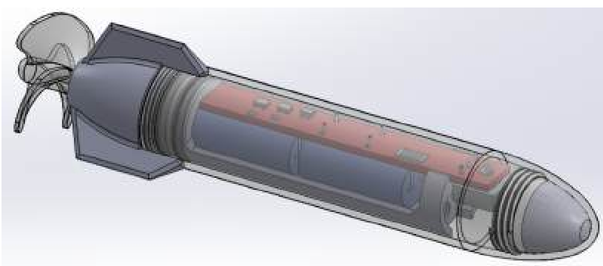

Fig. 3: Rendered CAD model of the updated powered torpedo.

3) Claw System: During the competition, Qubo will need to pick up irregularly shaped objects and place them in bins to score points. To do this, the mechanical team developed a claw that uses a rack and pinion mechanism actuated by a pneumatic piston to open and close. To open the claw, the piston extends and pushes the rack forward. The rack turns the gears and rotates the arms open. The reverse happens to close the claw. The grippers are made of 3D printed TPU and are designed to conform to the shape of whatever they are grabbing. Their triangular structure allows them to easily compress and curl around an object. Each gripper has two fins that allow it to slide into the arms of the claw. 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 mounting was changed because the previous mounting system was cumbersome. The two brackets that encased the piston hooked around horizontal bars on the underside of Qubo. This meant that, to take the claw on and off, either one of the bars needed to be removed or the claw had to be disassembled. For the new mounting, the plate on the right side of the claw screws into the inner side plate of the frame. With this new design, removing the claw is as simple as taking off a few screws without any disassembly required.

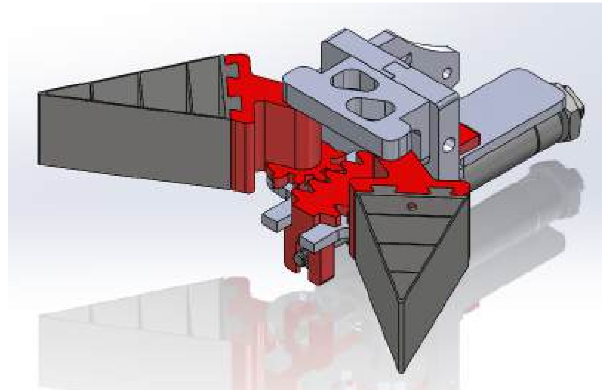

Fig. 4: Rendered CAD model of the claw system.

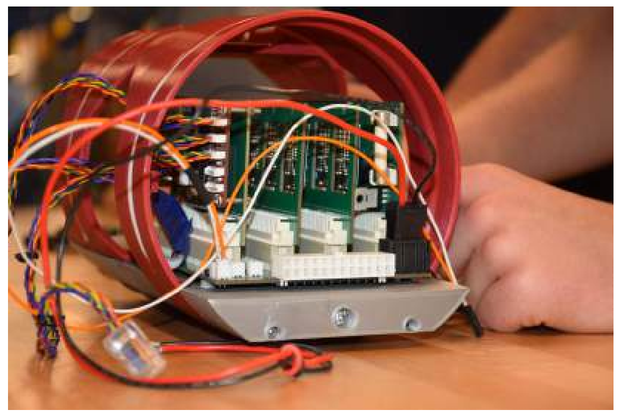

Fig. 5: Qubo's electrical hull, featuring a Backplane electrical system.

C. Electronics Updates

Last year's competition saw several new designs implemented on the electrical hull of Qubo, from a custom Backplane system to a power distribution board. However, it also saw new challenges, including attenuation of the I^2C signal and overheating of the power card. In spite of this, the systems implemented on the electrical hull of Qubo were largely successful. New designs for this year were made focusing on improving the flexibility and maintainability of last year's Backplane and daughter cards.

1) Backplane Improvements: Last year, the Backplane was an effective solution to reduce the bulk of wire in the electrical hull and increase modularity. However, it did not include any signal infrastructure, which relied on high-impedance WAGOs. In addition, Qubo faced difficulties with the attenuation of the I^2C signal in its many components. This year, after 14 redesigns, both I^2C and CAN bus signals are integrated in the new Backplane, and are selected with short circuit jumper caps. In addition, the new Backplane includes USART infrastructure for debug individual cards.

2) Power Card: Another card that was introduced last year was the power card. This card accommodated power supply for 12V, 5V and 3.3V buses. However, its 12V regulator caused overheating issues on adjacent cards. This year, the power card has been redesigned so that the hot 12V regulator is facing away from other cards on the Backplane. This will prevent adjacent cards from overheating, especially in the challenging outdoor thermal environment that Qubo will operate in at competition. In addition, the new power card includes the infrastructure to implement a parallel electronic fuse system as a more dynamic alternative to the existing chemical fuse.

3) Thruster Card Redesign: This year's thruster card features major upgrades. These thruster cards are essential for Qubo's success, as they are the most direct interface for all of Qubo's motions. Each thruster card now incorporates a new, on-board ST microcontroller, enabling enhanced thruster control. In addition, the motor signals were improved, with new traces and hardware facilitating CAN and I^2C options. Their signal integrity was also improved through use of more unbroken ground planes for EMI shielding. Overall, these changes will ameliorate the I^2C signal attenuation issues the team faced last year.

4) New Miscellaneous Card: There are many miscellaneous systems throughout Qubo's electrical hull that are essential for its success. This year, these systems have been collected onto a single card. This card eliminates the use of an external Arduino for controlling the LED strips in Qubo. It also incorporates a real-time clock for the Jetson.

D. Embedded System

Qubo's firmware was a continued development from last year. Motivated by problems with I^2C signals being attenuated over long wires, the team decided to begin development exploring a CAN Bus replacement. The chosen hardware controller was an STM32G0B1RE. This solution also has the benefit of being more scalable in the coming years, enabling the team to add more embedded devices to the robot with comparatively little additional effort, such as power sensors and an OLED diagnostics screen.

E. Localization

In the past, Qubo had two main sensors tracking its movement: a VectorNav VN-100 [5] IMU that meausures its orientation, angular velocities, and linear accelerations, and a Bar02 Pressure Sensor [6] for calculating depth. The measured orientation and depth were treated as ground truth without sensor fusion, and full position information was unknown. Over the last couple 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]. Our configuration is to fuse these data sources to refine estimates of direct sensor output, and estimate Qubo's lateral position from related measurements.

F. Controls

Tracking Qubo's full state gave us the opportunity to build a controls system consisting of PID feedback controllers on all six axes of movement that can take Qubo to any given target position and orientation. For each axis, its controller can switch between controlling the position or velocity of that axis. The system can also switch between controlling Qubo's lateral position in either its world or body frame. These configuration options allow us to do simple point-to-point movements or execute more complex maneuvers depending on the task. For example, to rotate smoothly around a target like the prequalification pipe, we can choose body frame and put our X (forward), Y (horizontal) and yaw axes in velocity mode so that the vision systems command yaw velocities to track the pipe maintaining a constant horizontal velocity while the other axes hold their positions.

G. Computer Vision

In past years, the team had a stereo ZED 2i [9] camera facing forward, enclosed in an acrylic waterproof casing. The casing added another two layers of refraction beyond the camera lenses: air to acrylic and acrylic to water. This made classical computer vision techniques very inaccurate. This year, we replaced the ZED with 2 exploreHD 3.0 [10] fisheye-lens cameras, set up in stereo. This eliminated the acrylic casing, and we integrated this into our existing systems by applying camera calibration, stereo calibration, and stereo rectification to form images that replicate the pinhole model. While this significantly reduces the setup's field-of-view, it is roughly equivalent to that of the ZED while also now allowing for geometry-based vision techniques. We combine this new capability with a fine-tuned YOLOv11 [4] object detection model, OpenCV segmentation methods, and knowledge of the tasks' dimensions to calculate the relative pose of tasks in our field of view and position targets for our controls systems. An example of our pipeline is shown in Fig. 6. The first image shows the YOLO model isolating the region where the gate lies. The middle images show a series of OpenCV methods applied to that section of the image, converting it to

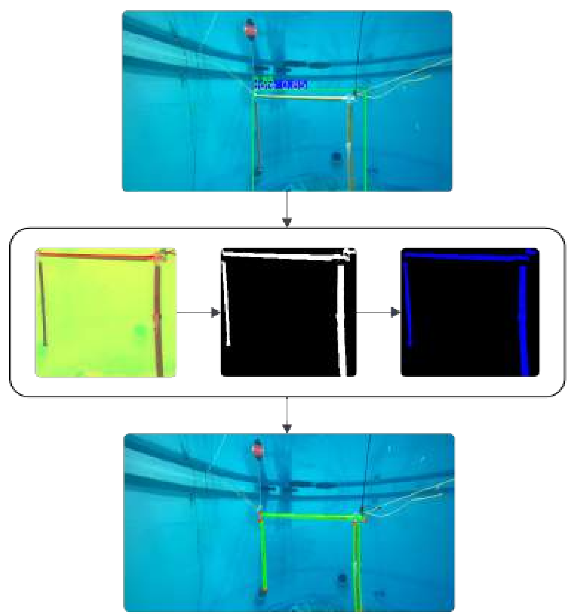

Fig. 6: Segmentation of the prequalification gate.

HSV space, applying masks to segment the gate, and using morphological operations to refine these masks by closing small gaps and removing noise. The last image shows the final result of OpenCV contour and corner detection applied to the filtered mask. These corner points are then used in Perspective-n-Point [11] to estimate the gate's pose relative to Qubo.

H. Autonomy

High-level autonomy focuses on taking our plans for each task (described earlier) and translating them into programs that command our controls systems and end effectors based on information from our localization and vision systems. Over the last couple years, we have replaced simple state machines we used in the past with behavior trees, using the BehaviorTree.CPP [12] library which integrates with ROS. This framework allows for easier visualization and increased modularity and efficiency with reusable behaviors.

I. Autonomous Torpedo

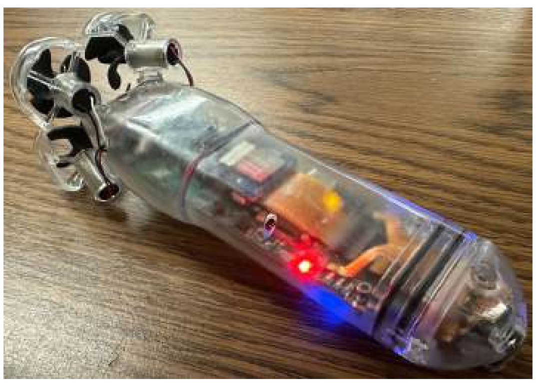

Fig. 7: Photo of the autonomous torpedo.

The autonomous torpedo shown in figure 7 grew out of the previous year's powered torpedo project with the goal of creating another fully autonomous vehicle that fits entirely within the size constraints of a torpedo. For its strategic success, maintainability and maneuverability are key along with the strict size constraints of the 2x2x6 inch bounding box. For that reason, the design and component selection were inspired by small FPV drones.

The mechanical design of the autonomous torpedo is split into 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 allows everything inside the torpedo to completely slide out for easy maintenance. Likewise, maneuverability required minimal net buoyancy which motivated the non-circular cross-section of the body to reduce wasted volume, although that came with even stricter size constraints.

For all of the microcontrollers and sensors on the autonomous torpedo, size was key. More finicky and less documented integrated components had to be chosen for this sake which required the development of the main PCB and sensor flex PCB, shown in the electrical block diagram in figure 8. The main controller used is the Xiao ESP32-S3 due to its small size and simplicity as a self-contained module. Over the main I^2C line, this module controls the Grove AI vision module V2 [13] which is used for torpedo target identification and the STM32 which acts as a motor driver and sensor controller. The most important sensors used are the ICM-42688-P accelerometer + gyrometer [14], the MS583702BA01-50 depth sensor [15], and AK09940A magnetometer [16], all of which are used to estimate the position and orientation of the torpedo in the water. Everything is powered by the Gaoneng 550mAh single-cell LiHV battery [17] as is commonly used in FPV drone racing. The power is sent through a 7A fuse for safety and the STM32 monitors battery voltage and can shutoff power to the entire board through a mosfet latching circuit.

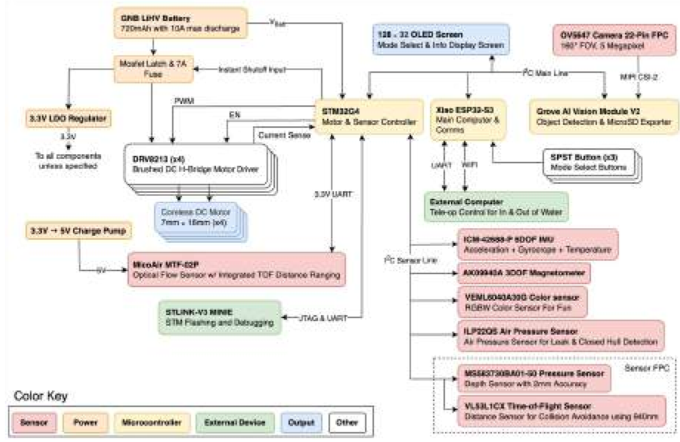

Fig. 8: The electrical block diagram of the autonomous torpedo.

III. Testing Strategy

In an effort to increase the reliability of Qubo during competition, this year R@M has put testing at the forefront of its development process. Time for testing was incorporated into every project timeline, along with leeway to anticipate iterations. This was an effective change from last year as creating new, unique designs from scratch creates variability and leaves room for error. The team's failure to account for these steps last year led to design changes and end effector integration the week before competition. All new systems, from the pneumatic system to each end effector and the backplane, were systematically tested on a strict timeline and in parallel. This included waterproof testing for the torpedoes and multiple enclosures, rapid prototyping throughout the design process, and productive pool tests that iterated methodically on progress, limiting blowback if something went wrong. Overall, Qubo's increase in reliability is credited to a focus on systematic testing from start to finish.

A. Torpedo Systems Testing

The torpedo testing focused on waterproofing, balancing, and the thrust characteristics of various coreless DC motors.

Fig. 9: Torpedo waterproofing panel to test each type of hull-penetrator/opening. From left to right these are testing: the depth sensor and wire sealing, camera sealing, the autonomous torpedo o-rings, wire sealing, and the powered torpedo o-rings.

The powered torpedo kept its 8520 coreless DC motor from the previous version, and its testing was based largely around the waterproofing and balance.

On waterproofing, before the final 3D prints for the autonomous torpedo and powered torpedo were ordered, test prints for each of the hull-penetrations/openings were made to verify their waterproofing. Figure 14 in Appendix D, subsection A, shows the results of this test.

The balancing tests were more qualitative, the powered torpedo would float on the water and the positions of the inner weights would be moved until it showed no preference to any specific orientation. The same approach was followed for the autonomous torpedo, and it is important to note that this process is enabled by the design of both torpedoes allowing the hull to be opened up easily so the inner weights could be moved around.

Lastly, the thrust tests were conducted to evaluate different coreless DC motors. While these motors are primarily advertised by their dimensions alone, the power characteristics can vary wildly for the same dimension of motor depending on the motor's winding. Motors (of the same dimensions) can be easily separated by measuring the current draw at no load, and then further tested qualitatively by feeling its propulsion force underwater with a propeller. These steps lead to the choice of 716 DC coreless motors for the autonomous torpedo which have a decent balance of weight vs power draw vs thrust.

B. Low-Level System Testing

The goal of low-level testing was to ensure the system had motor control functionality and was able to receive data from the sensors. This was done in parallel with implementing new hardware.

The software team developed a suite of testing utilities to verify functionality of various peripherals on the robot. Thrusters are run at low speeds outside of water to check for basic operation and correct orientation. Since the same PCA9685 PWM generator IC is used for controlling PWM for the pneumatics solenoids, the same testing software was used.

The DVL on Qubo was integrated and tested by taking advantage of the web server hosted on the DVL computer. Simply visiting the website when connected to the robot's network confirms that the DVL is on and responding to motion. When working on it outside of water it is submerged in a dish of water to prevent it from overheating.

Testing for the new electrical backplane was also straightforward. Since no new features were added to the electrical system, testing simply consists of comparing Qubo's behavior to last year, which is done by running the same tests.

C. High-Level System Testing

The first round of testing was to verify that Qubo performs competition tasks in the same way as last year with new hardware and end effectors. This was done by running the same pre-qualification code as last year, to obtain a basis for updating controls software. To test our new controls and autonomy systems before placing Qubo in the water, we use a Gazebo [18] simulation recreating the pre-qualification environment. This was used to validate implementation of the theory described earlier. Once controls and autonomy sequences are verified in simulation, we move to Qubo to test further in water.

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. Russell Shomberg is a new advisor this year who suggested the design for the claw grippers from this website [19]. The team is grateful to Ivan Penskiy for allowing the team to use space and tools in the Robotics and Autonomy lab on campus. 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. Additionally, the team thanks Robotic Research LLC, the Mechanical Engineering Department 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] D. Capalongo et. al., "Qubo V: RoboSub 2024 Technical Report", 2024.
[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] Stereolabs, "ZED 2i Stereo Camera." https://www.stereolabs.com/store/products/zed-2i
[10] DeepWater Exploration, "exploreHD 3.0 Underwater ROV/AUV USB Camera." https://dwe.ai/products/explorehd
[11] OpenCV, "Perspective-n-Point (PnP) pose computation" https://docs.opencv.org/4.x/d5/d1f/calib3d_solvePnP.html
[12] Auryn Robotics, "BehaviorTree.CPP Library." https://www.behaviortree.dev/
[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-550mah-100c-lihv-whoop-micro-battery-w-plastic-head-a30
[18] Open Source Robotics Foundation, "Gazebo Simulator." https://gazebosim.org/docs/
[19] Black Ram Industries, "Fin Gripper (Robotic/Prosthetic Hybrid) - Mark VI" https://www.printables.com/model/26078-fin-gripper-roboticprosthetic-hybrid-mark-vi

Appendix A: Members

Business

Harshita Kalbhor
Amy Naylor
Sriman Selvakumaran

Mechanical

Reese Snook
William Trasatti
Tommy Wolcott
Sam Bentz
Michael LeVesque
Daniel Gregory
Ryan Buchler
Collin Rogers
Maanav Subramanian
Tyler Huneke
Ken Schumer
Ava Pavese
Nathaniel Chen

Electrical

Kurt Kovacs
Anthony D'Costa
Brian Zagalsky
Eli Leopold
Hannah Devine
Jack Danser-Anger
Liam Brosnan
Nitya Sharma
Sam Ondrusek
Vassay Iqbal

Software

Alexander Yelovich
Ishaan Ghosh
Thomas Li
Jai Rastogi
Leo Jiang
Max Jin
Eric Xu
Brant Jiang
Grace Cai
Neha Rajasekaran
Alvin Shen
Jai Patel
Aditya Shelke
Vaishnavi Samprati
Michael Kittredge

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 centered on building lasting, meaningful connections, starting with **Maryland Day** on April 26th. Thousands of people visit campus that day to celebrate Maryland, many without any background in robotics. Our challenge was to make the field accessible without oversimplifying it. We brought out interactive demos. Our remote setup let visitors control actuators and observe real-time sensor feedback. For many, it was the first time they had seen engineering beyond a screen. Members rotated through stations for over eight hours, explaining PID loops, stereoscopic vision, and CAN bus communication in simple terms to children, parents, alumni, and prospective students. We opened our workspaces for guided tours. These were not polished displays, but real spaces in use. The authenticity resonated.

Our **Design Review** events took that a step further. Held each semester, these reviews are open to all of campus, including faculty, sponsors, peers, and anyone interested. While presentations dive into software architectures, mechanical tradeoffs, and testing protocols, what makes these nights impactful is the cross-disciplinary conversation. Students from unrelated majors often attend and ask thoughtful questions that push us to rethink assumptions. This year, we had to rework how we visualized our proposed chassis design after an aerospace engineering professor pointed out a flaw in our modeling. These moments strengthen not just our systems but our ability to explain and defend our designs under pressure.

Mentorship also plays a major role in how we give back. In the spring, R@M hosted a group of **FLL students from Urbana and Clarksburg Middle Schools** who had qualified for the state championship. They toured our lab, saw our AUV, and asked detailed questions, ranging from how we waterproof components to how we handle sensor calibration underwater. Rather than a scripted session, our members shared personal experiences, including how they first got into robotics, what failed during their first builds, and why they stuck with it. The visit lasted longer than scheduled because the students wanted to keep talking.

Internally, R@M also invests in community. Our Pi Day tradition (an open invite to throw whip cream pies at R@M members for fundraising), team dinners, hikes, and spontaneous snack runs are not just for fun. They are how we stay motivated through long build nights and tight deadlines. We are teammates, but also friends who support each other's growth. At every level, R@M is committed to using our resources, time, and passion to uplift others and foster curiosity. Through open labs, honest conversations, and technical transparency, we aim to make robotics less of a mystery and more of a possibility.

Fig. 10: R@M showing off Terry at Maryland Day.

Fig. 11: R@M members at the Spring 2025 Design Review.

Fig. 12: R@M leading a tour of the lab and Qubo.

Fig. 13: R@M hosting a group hike.

Appendix C: Component List

Component	Vendor	Model/Type	Specs	Custom / Purchased	Cost	Year of Purchase
Buoyancy Control	NBRF Stock	Foam	Pink	Custom	$0	N/A
Frame	Custom	N/A	Aluminum, Water Jetted	Custom	$200	2017
Waterproof Housing	Blue Robotics	6in	Acrylic and Aluminum Endcaps	Purchased	$400	2022
Waterproof Connectors	Blue Robotics	Penetrators	M10 Potted Connectors	Purchased	$5	2022
Thrusters	Blue Robotics	T200	11.2 lbf forward thrust, 350 watt	Purchased	$200	2017
Motor Control	Blue Robotics	Basic ESC	7-26 V, 30 amps max	Purchased	$36 x 8	2017
PWM control	NXP Semiconductors	PCA9685	16 PWM channels	Purchased	$14.95 x 2	2023
Air Tank	Maddog	N/A	Aluminum, 20 oz.	Purchased	$28.95	2023
Solenoids	WIC Valve	3V210	0.25 in. 3 Way, 2 Position	Purchased	$26.52 x 4	2023
Pistons	Bimba	6498K632	Double-Acting 3/4" Bore 1" Stroke, 0.25 in. Rod	Purchased	$45.89 x 4	2025
Check Valves	BONOMI	100012LF-1/4"	Brass 0.25 in. 400 psi	Purchased	$24.24 x 2	2023
Battery	Gens Ace	GA-B-45C-5000-4S1P-Deans	14.8v, 5000mah	Purchased	$36	2017
Converter	Custom PCB		12V, 5V, Fuse, Current and Voltage Monitoring	Custom	$50	2017
Main Computer	Nvidia	Jetson Orin Nano	16 GB RAM	Purchased	$699	2025
Internal Comm Network	Ethernet, USB, I^2C, CAN Bus					
External Comm Network	Ethernet, Sonar					
AHRS	Vectornav	VN-100	2° accuracy heading and tilt	Purchased	$1,100	2023
DVL	WaterLinked	DVL-A50	±3.75 m/s range, ±0.4% accuracy	Purchased	$6,950	2024
Vision	DeepWater Exploration	exploreHD 3.0	Underwater ROV/AUV USB General Vision Camera	Purchased	$300 x 3	2024

Algorithms: Vision	OpenCV	Various basic vision processing algorithms	N/A	N/A
Algorithms: Other	Kalman Filter	State estimation and sensor fusion	N/A	N/A
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: 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. 14: Waterproofing panel test results.

For this test, all of the objects were tied to the same weight (Figure 15) 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.

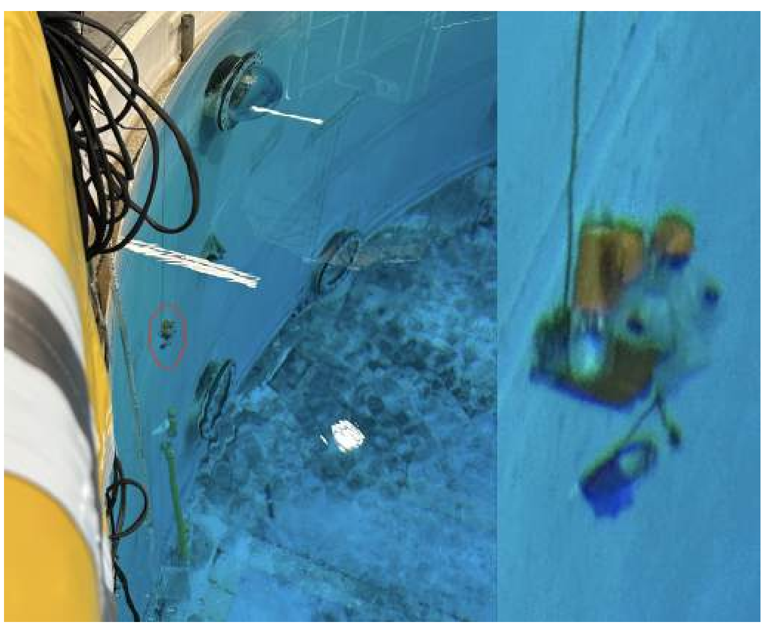

Fig. 15: Waterproofing panel test setup, with the panel zoomed into on the right.