Refining Talos: Ohio State Underwater Robotics' Strategic Design and Performance Enhancements

Amber Dellacqua, Brach Knutson, Mitch Oinonen, Ethan Rosati, Alex Schuler, Sam Tomlinson, Cameron Tucker, John Ulm

***Abstract*—This paper reviews the design and enhancements of Talos, an Autonomous Underwater Vehicle (AUV) developed by The Ohio State University's Underwater Robotics Team (UWRT). Over the past year, the team's focus has been on validating existing systems and integrating new capabilities to enhance Talos' performance for the upcoming Robosub 2024 competition. The competition strategy emphasizes robustness and reliability, aiming to maximize task completion and improve overall competition performance, with a priority focus on the *Enter the Pacific*, *Hydrothermal Vent*, *Mapping*, and *Ocean Temperatures* tasks. Key improvements include robust task mechanisms, a refined electrical architecture, and advanced control and perception algorithms. These enhancements were driven by the competition strategy and influenced the team's system architecture, design, and engineering decisions. The paper outlines the design strategies, implementation details, and testing approaches adopted by UWRT, highlighting the impact of these developments on Talos' functionality and competitive edge.**

I. Introduction

Last year, The Ohio State University's Underwater Robotics Team debuted Talos, its new autonomous underwater vehicle. This year, the team continued to develop Talos, validating existing systems and increasing its capabilities with new task mechanisms, operator improvements, a condensed electrical architecture, and improved controller and perception algorithms. With these additions, UWRT aims to complete more tasks at Robosub 2024 than attempted in previous years. This paper details the advancements made to Talos, focusing on the design and implementation of the new features, and evaluates their impact on the vehicle's performance in competitive scenarios.

II. Competition Strategy

A. General Strategy: Completeness & Robustness

UWRT's competition strategy for RoboSub 2024 emphasizes the completeness and robustness of Talos in both functionality and performance. Based on UWRT's performance at the 2023 RoboSub competition, the team has refined its strategy to enhance Talos with advanced hardware and software capabilities such as:

- Robust torpedo/marker launchers for reliable completion of *Mapping* and *Ocean Temperatures*
- Condensed electronics system, allowing the team to create additional functionality like an acoustics system, to achieve the random pingers points
- Robust control and an improved perception system, giving Talos the precision to complete tasks such as *Mapping*
- Reliable and configurable claw, allowing for completion of *Collect Samples*

B. Competition Strategy: Course Approach

The three criteria set in UWRT's general strategy were paired with several parameters to prioritize tasks for Talos' autonomy runs:

1) Point values attributed to tasks
2) Talos' ability in terms of mechanical components, electronic hardware, and software architecture
3) Duration allocated to task validation testing

Based on these criteria and the team's bandwidth for the current academic year, UWRT has decided to prioritize completing the Mapping and Ocean Temperatures tasks, along with surfacing in the octagon at the end of each run. UWRT aims to consistently maximize its point acquisition during each run by leveraging this strategic approach and building upon a reliable AUV system.

C. Competition Strategy: Task Execution

1) Gate - Enter the Pacific: Talos will opt to perform the coin flip, leveraging its AprilTag origin frame to ensure its navigation system remains robust despite changes in starting position and orientation [5]. Talos will begin by submerging and searching for the gate around its initially estimated position, using the search algorithm detailed in Appendix C. Upon locating the gate, Talos will perform a "miniature search", positioning itself at an optimal distance to locate both the "hot" and "cold" sides. Talos will then maneuver under the "hot" side of the gate, as its red color is easier to see underwater. Finally, Talos will employ a novel stunt controller to perform two somersaults, as these maneuvers are more effective than a roll based on the design of the robot, maximizing style points.

2) Buoys - Hydrothermal Vent: Talos will use the same foundational actions used in *Enter the Pacific* to complete *Hydrothermal Vent*. First, Talos will search for the buoy around its initially estimated position. Once located, Talos will circumnavigate the buoy in the clockwise direction, matching the "hot" side designation.

3) Random Pingers: UWRT opted to locate the random pingers due to their high point value, low risk, and ease of integration with the rest of the system. To achieve this, the team designed a new acoustics system that can determine the pinger signal's angle of arrival. With this system, Talos will determine which task to perform by continually sampling signals while executing *Enter the Pacific* and *Hydrothermal Vent*. It will triangulate the pinger's position and compare it with the estimated positions for *Mapping* and *Ocean Temperatures*. Talos will complete the task closer to the pinger before moving on to the other one.

4) Torpedoes - Mapping: Talos will complete *Mapping* in sequence with *Ocean Temperatures*, as determined by the random pingers logic. To achieve the precision required for hitting the smallest holes and earning maximum points, UWRT overhauled its perception system. Talos will use this new perception system to first locate the overall prop and then perform its "miniature search" to identify the two smallest targets. Talos will then align each torpedo individually and fire them into the smallest target first, followed by the second smallest target.

5) Bins - Ocean Temperatures: UWRT opted to complete *Ocean Temperatures* due to its high point value and potential for even more points from the random pingers. To decrease the risk associated with the task, UWRT redesigned its torpedo launching and marker dropping mechanism to eliminate the need to pitch Talos and risk losing Doppler Velocity Log (DVL) lock. The team initially tried tilting the AUV to perform the task, but this approach caused issues with maintaining DVL tracking. To fix this, they built a downward-facing camera (DFC) and modified its dropper mechanism to point downwards. This enabled the task to be performed without tilting the AUV, avoiding the risk of losing DVL tracking. Talos will use a new downward search pattern to locate the bin (Appendix C). Upon locating it, Talos will position the marker dropper above the "hot" section of the bin and drop both markers into the bin.

6) Octagon & Table - Collect Samples: After completing the prioritized tasks, Talos will proceed to *Collect Samples*. To best allocate its resources and maximize scoring efficiency, Talos will use the downward search to locate the table and the samples on it. Then, Talos will employ its new claw to pick up the samples and move them to different bins. Based on knowledge from other organizations attempting a similar challenge, UWRT implemented a protocol into the AUV's thruster solver to reduce thruster down-wash, minimizing the impact of the robot's presence. This allowed for a mechanically simpler claw design, reducing project cost and testing requirements. Afterward, Talos will surface in the octagon to end its run.

III. Design Strategy & Implementation

Design changes for this year focus on enhancing the reliability of the AUV's peripheral systems and validating the existing systems. These changes primarily address issues encountered during Robosub 2023 and introduce improvements needed to advance AUV autonomy in 2024. This section describes these improvements and how they enhance the team's competition strategy.

A. Mechanical Subsystems

With the overall chassis and underlying system design completed in the previous year, the Mechanical Subteam this year focused on improving and implementing the remaining task mechanisms and the AUV's everyday operator experience.

1) Servo Housings: One of the most important mechanical changes required for Talos to complete the *Mapping*, *Ocean Temperatures*, and *Collect Samples* tasks was updating the waterproof servo housings. During RoboSub 2023, Talos' 3D-printed servo housings were prone to leaking, causing damage to the servo motors and functionality issues in the torpedo and marker-dropping systems. The new housings required both a redesign in the seals as well as a material change to aluminum to prevent water leaking through the walls, bypassing the seals. The new design features a CNC-milled aluminum body and two radial seals over the existing singular face seal, which makes it more tolerant to positional errors in mounting. The O-ring and groove dimensions were determined according to Parker Hannifin's Dynamic O-ring design tables [1]. For a standard bore size of 0.5 in, the selected size was a number 012 X-ring, chosen for the X shape's better dynamic sealing performance. For this seal, the groove diameter was chosen as 0.381 in, which is slightly deeper than the recommended 0.389 inch diameter to account for using an X-ring. This maintains the stretch at 4.67% and squeeze at 12.18%, which is within the recommended ranges for a dynamic seal. The groove width was kept at the recommended 0.093 in.

The new servo housing design is universal between all task mechanisms, only requiring a different output shaft specific to the motion required. Due to the Dynamixel servo's chainable functionality, housings can also be linked together using SubConns for simplicity.

Fig. 1: Redesigned Torpedo Launcher/Marker Dropper Mechanism and Servo Housing

2) Torpedo Launcher & Marker Dropper: Since the team is prioritizing completing both the *Mapping* and *Ocean Temperatures* tasks, a slight redesign in the torpedo and marker dropper system was also important to allow for more easily controllable and stable marker drops. To avoid pitching downward and risking the loss of tracking on the DVL, the assembly was split in half so that the torpedoes remained pointed forward while the markers pointed downwards. To maintain the simplicity of the single servo design, a pair of bevel gears was used to transmit torque between the upper and lower sections. The final assembly can be seen in **Fig. 1**, with torpedoes facing forwards and markers facing downward. With the addition of the DFC, Talos can now accomplish both the *Mapping* and *Ocean Temperatures* tasks without unnecessary movements.

3) Claw: While the *Collect Samples* task is not prioritized for the team at Robosub 2024, development on a new claw design was completed to begin building a foundation to complete similar tasks in future years. The new claw was designed to easily detach from Talos, ensuring straightforward use and control. For simplicity, it was built to operate with only one servo. The final design uses a rack and pinion driven by an identical servo setup as the one used in the torpedo/marker dropper system, with a static mount angled at 45 degrees downwards. The 45-degree angle was determined to be optimal for picking up the objects in the *Collect Samples* task as the tube worm is difficult to grasp when mounted downwards, and the nautilus is challenging when mounted forward. The claw also utilizes interchangeable and customizable fingers, allowing the team to best suit the objects it must pick up. With the rated maximum torque of the Dynamixel servo of 0.84 N*m and a pinion pitch radius of 0.01m, the claw design has a maximum calculated grip strength of 42 N, about 9.4 lbs, per side.

4) Operator Experience & Buoyancy : Although not necessary for task completion, several minor mechanical improvements were added to Talos to make both assembly and common mechanical maintenance easier. Turnbuckles were added to the tensioning cables so that the tension can be adjusted or removed completely if the AUV needs to be disassembled. The square tubing on the front and back of Talos used for task mechanism mounting was replaced with L channels to avoid dropping screws inside. With the addition of new task mechanisms, more buoyancy was required for the AUV, so metal water bottles were mounted to the top of the main housing. The buoyancy additions this year are shown in **Fig. 2**, highlighted in blue. The threaded rods used to hold the battery housings on Talos were thickened to ¼-20 from 8-32 to prevent bending during transport and repeated use. Lastly, to improve the operator experience, a new tether spool was designed (Appendix M), color codes were added to all SubConns to prevent misconnections, and rubber bumpers were added to all thrusters and battery housings to avoid injuries and damage to pools.

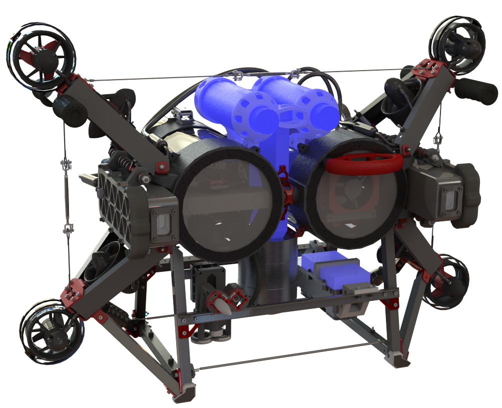

Fig. 2: Rendering of Talos with Buoyancy Additions Highlighted in Blue

B. Electrical Subsystems

Following the introduction of the Mark 2 electronics system last year, the Navionics Subteam prioritized condensing and improving it. As part of this effort, the team developed new boards (Appendix G) and firmware (Appendix O) to take on additional tasks, such as detecting random pingers and adding quality-of-life enhancements, such as a charger for the smart battery system (Appendix F).

1) PoActuator Board: The new PoActuator board was developed to replace both the Power and Actuator Boards used in last year's competition. Simplifications in the actuator setup allowed these components to be combined with the larger power system onto a single board. This board is responsible for properly maintaining the electrical system's battery power, 12V, and 5V rails while also controlling the claw, torpedo, and marker dropper actuators. In addition to saving space for the implementation of new boards, the PoActuator board includes significant improvements such as:

- A regulatory diode to prevent back-flow current
- An update to kill switch circuitry to better handle high voltage MOSFETS
- A new aux switch connection
- Improvements to communication with actuator mechanisms

2) RS422 PCIe Card: As more sensors were incorporated to improve Talos' competition performance, it became evident that the Jetson AGX Orin, serving as the central processor, had insufficient connectivity. To amend this, a PCB was designed to convert the M.2 slot on the Orin's carrier board into an RS422 serial interface. This board was created in electrical and mechanical compliance with the M.2 2230 specification [2] and ultimately proved useful in connecting a Fiber Optic Gyro (FOG) to the Orin.

3) Acoustics Board: UWRT wanted to expand Talos' task execution capabilities this year by designing an acoustics system to triangulate the random pingers. This system uses two hydrophones, each connected to a superheterodyne receiver, described in Appendix E. The signal is then amplified and sent through an envelope detector for smoothing, and a comparator to convert it to a digital pulse. The time difference between the pulses is used to determine an angle of arrival This angle can be used to refine a position as the AUV moves through the course.

4) LED Board: To improve the visibility of Talos' status LEDs in all water conditions, the team created a dedicated LED board powering 6 amps of LEDs, which is six times the current of the previous LED strip. Due to the heat generated by the LEDs, the team decided to use two different boards, one for control and another for the LEDs themselves. The control board uses automotive-grade LED drivers and the RP2040 microcontroller to connect with the rest of the electrical boards. To manage the heat, the LEDs have been mounted to an aluminum PCB with a large heat sink on the bottom.

5) Downward-Facing Camera: The DFC, located in an external housing due to spacing constraints in the main hull, minimizes the cost of camera equipment and cabling between the external housing and the central hull by processing images internally and sending results over the CAN bus. It also allows the team to offload compute from the Orin, allowing more compute for other processes. The DFC consists of a Raspberry Pi 5 with two Raspberry Pi Global Shutter cameras, a Google

Coral Edge Tensor Processing Unit (TPU) accelerator and a custom Raspberry Pi Hardware Attached on Top (HAT) shown in **Fig. 3**. The HAT sockets on top of the 40-pin connector and provides regulated power and a CAN bus interface from the external connector. It provides a PCIe link between the TPU and the Raspberry Pi, and auxiliary trigger connectors to ensure both cameras capture frames simultaneously.

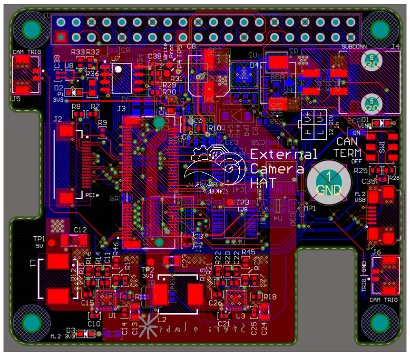

Fig. 3: Downwards Facing Camera Pi HAT

C. Software Subsystems

The Software Subteam focused on increasing the reliability of their codebase through the hardening of foundational systems, enabling more pool time to be dedicated to completing competition tasks. The team implemented a new controller, perception system, and various smaller fixes to the codebase.

1) Controller: To increase steady-state stability, travel speed, and tunability, UWRT overhauled Talos' controller (Appendix K). This replaced UWRT's previous cascaded P control scheme, streamlined the thrust allocation algorithm, and introduced closed-loop control on each thruster.

A Sliding Mode Controller (SMC) was hybridized with a PID controller and implemented using MATLAB Simulink to control AUV position. The control schemes were designed modularly such that they can be assigned to control individual degrees of freedom. These combinations can also be changed in real-time. The modular nature of the control scheme allows for complex AUV motion such as summersaults and barrel rolls. To run the model on the AUV's primary computer, as well as minimize processing delay in the control loop, the team used Matlab's Simulink Coder toolbox to generate C++ code (Appendix I).

The use of a PID/SMC hybrid controller increased AUV speed without sacrificing stability. The SMC provides the benefits of a velocity controller, such as consistent target speed over a wide range, without drawbacks like drift at the setpoint. The PID controller allowed for robust control of the static axes (pitch and roll) without the constant thruster direction reversal of the SMC, greatly reducing thruster wear while increasing system stability.

Making a feedforward term individually tunable minimized the time spent tuning the controller. Little additional tuning is required, as the PID/SMC hybrid controller is robust enough to handle mechanical changes.

Thruster control in previous years, which was implemented as a command controlling the percent output of the thruster, has lacked robustness due to factors like inconsistent battery levels. To increase overall stability and minimize thruster response time, firmware-level closed-loop control was implemented on each individual thruster. This control loop utilizes RPM feedback from the ESCs and a PI technique to drive a thruster at a particular speed. This controller can maintain thruster velocity to within 40 RPM, or approximately 0.4N, and run at 1200Hz.

2) Vision System: Vision integration for the AUV was rebuilt for RoboSub 2024. The new vision system uses YOLOv8 [3] for object detection and segmentation, OpenCV's Good Features to Track (GFTT) algorithm to identify 2D feature points within the bounding box, and segmentation masks to ensure the points lie on the object [6]. These points are then mapped to the depth map from the ZED 2i camera to create 3D points. Singular Value Decomposition (SVD) is applied to these 3D points to fit a plane and determine the object's orientation. The center of the bounding box provides the x and y coordinates for the centroid, and the z coordinate is derived from the SVD. For the DFC, the Iterative Closest Point algorithm is used to detect non-planar objects, as SVD is insufficient.

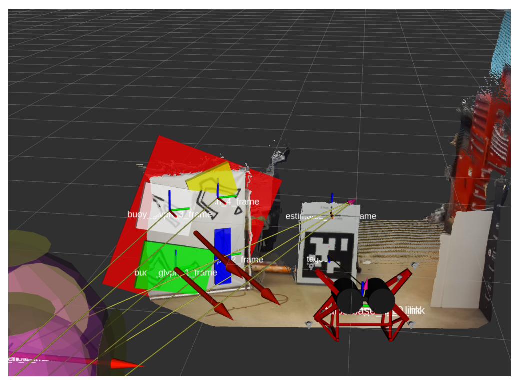

Fig. 4: Talos' vision detections, Zed2i point cloud, and April-Tag in RVIZ

A data augmentation pipeline was implemented to enhance the robustness of the YOLOv8 model. Starting with labeled videos, the pipeline applies augmentations such as mirroring, rotation, zooming, cropping, and pasting detections onto different images, as seen in Appendix L. These augmentations increase dataset variability, improving the model's generality and performance under novel conditions.

3) Mapping System: UWRT rebuilt its mapping system to enhance the precision tracking of task objects and improve recovery of Talos' location within the map when odometry drifts. The new system assumes that initial estimates of the task object positions are accurate relative to each other as they are mapped out before each run. Instead of adjusting individual object frames upon receiving detections from the vision system, the mapping system adjusts a map "offset" frame that shifts all objects uniformly, allowing the AUV to recover its position within the map if odometry has drifted. The mapping algorithm estimates object position and confidence using sample averaging and combined standard deviation, respectively. The system also supports looking up frame transformations at the precise time at which object (vision) detections originate rather than looking up the most recent data. This eliminates error from latency due to image processing in the vision system. Finally, to avoid sudden changes in the map at close range to objects when performing tasks, the new mapping system includes a “lock” feature. This lock feature allows the autonomy system to notify the mapping system when it plans to be close to an object. This eliminates the potential for inaccurate detection positions, which could be produced by the vision system in close range.

4) GUI: This year, UWRT continued to enhance its RViz dashboard, making AUV operations even more user-friendly. The team added four new panels:

- Parameter Panel: Allows users to easily change ROS parameters in runtime.
- Feed-Forward Panel: Enables users to easily apply a wrench to the AUV and quickly figure out an appropriate wrench to counteract natural forces.
- Mapping Panel: Facilitates inspection and configuration of the mapping system, calibration of the map frame using the previous year’s custom AprilTag, and control of Zed SVO recordings and recordings on the team’s new DFC.
- Electrical Panel: Enables users to perform basic integration with firmware to perform actions like power cycling the IMU.

The team also added useful improvements to the older panels, mainly to increase detailed detection and reporting of codebase and vehicle operation errors. The improved reporting includes internal controller metrics such as requested thrust or the rate at which thrusters change direction. Additionally, the panels can alert users to ROS issues, such as duplicate nodes in the network.

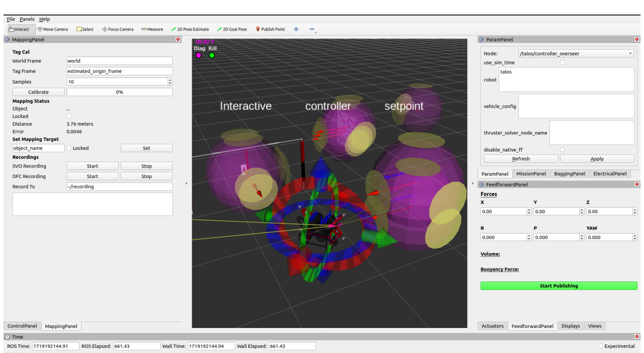

Fig. 5: New RViz window, featuring the map, interactive setpoint, simulation ground truth, and new panels

To improve operation and debugging, several items were added to the 3d space displayed by RViz:

- Ground truth position of objects in the simulator
- Vectors representing detections from the vision system
- Point cloud from the team’s stereo camera
- An interactive triad that allows users to easily modify the controller setpoint

5) AUV Reliability Fixes: One of the main issues UWRT faced during the 2023 competition was the significant time required to launch the code on the AUV. Starting with an idea from their third chance run in 2023, UWRT created a headless interface for the AUV to deploy the vehicle at a run with minimal interaction and setup (Appendix J). The headless interface, in conjunction with the autonomy system, uses Talos’ onboard LEDs to indicate its state and guides users through the entire deployment process.

The team also removed the launch panel used in the 2023 competition and replaced it with a small web app served from the Orin. This change keeps all internal launch states on the Orin rather than requiring them to be duplicated on two machines. Storing all states on the Orin makes the launch service capable of handling starting and checking the AUV systems before being lowered into the water at a run.

Throughout the year, the 32 nodes and network traffic heavily strained the ROS2 network (see Appendix H). To alleviate this, UWRT set up the FastDDS discovery server, centralizing the storage of node information. This greatly reduced system-level issues such as broken publishers and subscriptions and increased network resiliency to large data rates.

IV. Testing Strategy

UWRT emphasized rigorous testing of its AUV this year, aiming to ensure that all task behaviors are thoroughly tested before the competition. To achieve this, UWRT committed to at least eight hours of in-water testing each month, occasionally traveling outside of Columbus for suitable testing environments. During the spring semester, UWRT began conducting tests at Ohio’s Indian Lake, which proved to be a practical and cost-effective option for tests not involving the perception system. For a comprehensive description of the test procedures used by UWRT to prepare for water testing, see Appendix B. By the end of the spring semester, Talos saw time in the water at least once a week. In scenarios where pool time could not be obtained, or systems needed testing before in-water time, UWRT utilized a simulator that can be run with hardware in the loop.

A. Testing Approach

To ensure high confidence in task execution during the competition, the team prioritized testing and validation of the controller during the fall semester. This approach ensured that Talos’ controller was designed effectively and tested thoroughly. UWRT focused its pool time on validating task behaviors during the spring semester. The team prioritized the more challenging tasks, such as *Mapping* and *Ocean Temperature*. These tasks exercise the same systems as the gate and buoy tasks but offer higher point values and require greater precision, providing more rigorous tests.

B. Simulator

A custom simulator was developed to address UWRT’s need to test software outside of dedicated pool time. This system fulfilled the team's requirements for seamless integration with the ROS2 network, maintainability and accuracy. At the heart of this simulator is an advanced algorithm based on a fourth-order Runge-Kutta solver. This algorithm uses the robot's current state, physical properties, and thruster output to accurately predict its future states in real-time. These predicted states are then translated into corresponding sensor data and fed back into the AUV's software stack. Sensor noise and thruster output delays are artificially introduced, ensuring the simulated environment closely mirrors real-world conditions.

Visual simulation was introduced, a UWRT first. Using the OpenGL framework, a digital twin of the Woollett Aquatics Center pool was created, complete with caustics, reflections, and water surface waves. This visual simulation enabled UWRT to validate the AUV's pose estimation and object detection system as it navigated the virtual competition environment.

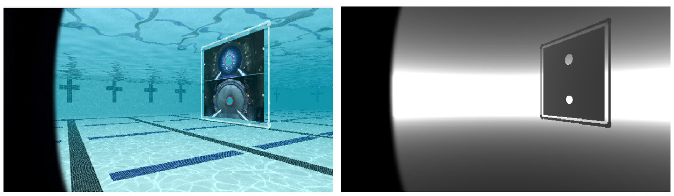

Fig. 6: Camera and depth images produced by the UWRT simulator

Throughout the year, UWRT utilized simulation to validate its control system, FastDDS discovery server, time-accurate transform lookups, behavior trees, and perception system. The FastDDS discovery server was used to fix issues of broken publishers and subscriptions, while time-accurate transform lookups in the perception system were implemented to counter latency in the vision system. Integration of the vision and mapping system was achieved, alongside improvements in vision models and behavior trees. This comprehensive testing saved UWRT countless hours of pool time that would have been spent fixing unit-level bugs.

C. Thruster Characterization

A critical component of UWRT's controller redevelopment was a complete characterization of the propulsion system on Talos. To perform this characterization, UWRT constructed a thruster force gauge – consisting of a thruster and load cell connected via a lever arm – allowing for real-time thrust measurement inside a test tank. Using this gauge, UWRT mapped thrust output to thruster RPM and DShot (thruster power) input for multiple thrusters and battery voltage levels.

Using this collected data, UWRT determined that significant differences between individual thrusters were due to variations in RPM output for the same DShot input and battery level variance. The data informed UWRT's decision to design the firmware-level thruster controller. Completing the thruster characterization project leads to a more consistent thrust between thrusters and battery levels and a system that works in standardized physical units (Appendix B).

D. IMU/Prequalification

While testing the controller during the fall semester, UWRT discovered inaccuracies in the yaw rate reported by Talos' IMU. After ruling out the driver, the next culprit was interference from ferric materials in the pool (Appendix N). The team tested the IMU in various configurations, including with the magnetometer disabled. UWRT conducted these tests by having Talos perform prequalification with the IMU in each configuration, due to the long and straight course requiring precise navigation.

The tests revealed that the IMU performs well with the magnetometer disabled but struggles to handle severe acceleration disturbances, such as hitting an object or a wall; such disturbances cause the IMU to enter a constant drift, disrupting the navigation system. These findings informed the purchase of a FOG, which would be more resistant to magnetic and acceleration disturbances.

V. Conclusion

By enhancing its reliable AUV platform, UWRT has allowed itself to focus on more advanced competition tasks by implementing reliable task mechanisms, an improved electronics system, a robust controller, a well-tested perception system, and reliable autonomy. The improvements to the AUV align with the team's competition strategy, and the rigorous testing approach has enabled the identification and resolution of critical issues in higher-level parts of the system before the competition. With greater confidence in Talos' existing capabilities and additional features added this year, UWRT plans to complete more competition tasks at Robosub 2024 than in previous years.

VI. Acknowledgements

UWRT owes its accomplishments this year to the generous support and contributions from many individuals and organizations. First, UWRT would like to thank its sponsors: Honda, Ford, Nortek, VectorNav, Advanced Power Drives, Eaton, Caterpillar, and The Ohio State University College of Engineering. Special thanks go to the Electrical and Computer Engineering Department and Dr. Saeedeh Ziaeefard, its faculty advisor, for providing lab space, equipment, and invaluable guidance.

UWRT would also like to thank Bill Schalz and Sean Danekind, who each provided UWRT with pool time for testing, which was instrumental in Talos' development for this year's competition. Additionally, UWRT would like to thank the staff at The Ohio State Recreation and Physical Activity Center, the Columbus Aquatics Center, and Aquatic Adventures Hilliard for their help with in-water testing of the AUV.

Talos' continued progress would not be possible without the dedication and hard work of UWRT's team members, who consistently go above and beyond to achieve competition goals.

Finally, UWRT would like to thank RoboNation for its commitment to the RoboSub competition and for inspiring future engineers in robotics.

VII. References

[1] O-Ring Division. *Parker O-ring handbook*. Parker Hannifin, Parker Seal Group, 2007.

[2] Peripheral Component Interconnect Special Interest Group. ”pci express m.2® specification revision 5.0, version 1.0”, 2023. [Online] Available: https://members.pcisig.com/wg/PCI-SIG/document/19525?uploaded=1 [Accessed: Jan. 12, 2024].

[3] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8, 2023.

[4] Ha Le Nhu Ngoc Thanh Tuan-Tu Huynh Mien Van Quoc-Dong Hoang Mai The Vu, Tat-Hien Le and Ton Duc Do. Robust position control of an over-actuated underwater vehicle under model uncertainties and ocean current effects using dynamic sliding mode surface and optimal allocation control, 2021. [Online] Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7865870/ [Accessed: Sept. 20, 2023].

[5] Edwin Olson. Apriltag: A robust and flexible visual fiducial system. In *2011 IEEE International Conference on Robotics and Automation*, pages 3400–3407, 2011.

[6] Jianbo Shi and Tomasi. Good features to track. In *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

Appendix A: Component Specifications

TABLE I: Talos' Component Specifications

Component	Vendor	Model/Type	Custom/Purchased	Cost	Purchase Year
Acoustics Housings		Aluminum 6061, Polycarbonate	Custom	$98.00	2024
Algorithms: Autonomy		BehaviorTree.CPP v3	Custom		
Algorithms: Localization/Mapping		Extended Kalman Filter/Custom mapping system	Custom		
Algorithms: Vision		YOLOv8	Custom		
AUV Chassis		Talos, Aluminum 6061	Custom	$900.00	2023
Battery	MaxAmps	Li-Po 8000 5S2P 18.5v	Purchased	$599.98	2023
Buoyancy Control	Blue Robotics	Subsea Buoyancy Foam; R-3312	Purchased	$70.00	2023
Camera	Stereolabs	Zed 2i	Purchased	$499.00	2022
Communication Network		CAN Bus	Custom		
Converter	TDK-Lambda	I6A4W020A033V	Purchased	$68.34	2022
CPU	Nvidia	Jetson AGX Orin	Purchased	$2,374.00	2022
CPU Carrier	Connect Tech	AGX202	Purchased	$971.00	2024
Downwards-Facing Camera (DFC) CPU	Raspberry Pi	5 Model B+	Purchased	$79.99	2024
Downwards-Facing Camera (DFC) Housings	Xometry	Aluminum 6061, Polycarbonate	Custom	$327.66	2024
Downwards-Facing Camera (DFC) TPU	Google	Coral	Purchased	$39.99	2024
Doppler Velocity Log (DVL)	Nortek	DVL1000	Purchased	$15,000.00	2018
Hydrophones	Aquarian Audio	HC1	Purchased	$400.00	2018
Inertial Measurement Unit (IMU)/Compass	Vectornav	VN-100-T	Sponsored	$1,300.00	2022
Fiber Optic Gyro (FOG)	Fitzoptica	VG103S-2LND	Purchased	$2,870.00	2024
Microcontrollers	Raspberry Pi	RP2040	Purchased	$0.07	2021
Motor Control	APD	80F3	Sponsored	$245.00	2022
Open Source Software		ROS2/OpenCV/Pico-SDK/BTCPP			
Programming Languages		C/C++/Python			
Smart Battery Housings	Xometry	Aluminum 6061, Polycarbonate	Custom	$634.00	2023
Task Mechanism Servo	Robotis	DYNAMIXEL XL430-W250-T	Purchased	$49.90	2023
Task Mechanism Servo Housings		Aluminum 6061	Custom	$92.12	2023
Thrusters	Blue Robotics	T200	Purchased	$1,432.00	2021
Waterproof Connectors	MacArtney	MC/HP/D Series	Purchased	$4,000.00	2015-2023
Waterproof Main Housing	Xometry	Aluminum 6061, Polycarbonate	Custom	$2,233.00	2021

Appendix B: Test Plan and Results

A. Talos pre-test checks

1) Required equipment:

- Talos AUV
- Ethernet Tether
- UWRT Safety Stack
- UWRT Operator laptop (or an Ubuntu 22.04 laptop with ROS2 humble installed)

2) Steps:

1) Unit test and validate individual components of AUV being targeted at the test on the bench. The procedure for this varies with respect to the things being tested.
2) Ensure the kill switch is removed from Talos
3) Power Talos AUV with benchtop power supply
4) Launch simulator launch code on Talos AUV
5) Launch RViz on operat laptop
6) Use hardware-in-loop simulator to validate behavior trees to be tested with vision.
7) Shut down simulator launch code on Talos AUV
8) Launch bench launch code on Talos AUV
9) Ensure that Talos is staying in the middle of the map in RViz
10) Pitch Talos up and down and ensure that the Talos displayed in RViz follows.
11) Repeat step 8 with the roll and yaw axes.
12) Place props to be used during the test in front of Talos.
13) In RViz, ensure that prop positions in RViz converge to their true values. This can be roughly evaluated using a tape measure.
14) Shut down benchtop launch on Talos AUV.
15) Insert kill switch and ensure thrusters do not move.
16) Use SSH to access Talos' computer.
17) Run the command: "ros2 run riptide_controllers2 thruster_test_canbus"
18) Ensure that thrusters spin in the correct order and direction. Thruster IDs and directions are printed on the physical thrusters.
19) After all thrusters have been tested, remove kill switch and ensure that thrusters stop
20) Stop the command that was run in step 17

3) Pass/Fail criterion:

- The test is considered passing if the steps can all be completed without error.
- The test is considered failing if the steps cannot be completed due to reasons like code exceptions, hardware errors, network issues, and more.

4) Results: This test was executed before most pool tests that UWRT ran this year and was found to significantly reduce the amount of pool time spent debugging basic AUV configuration issues. It also reduces the number of errors encountered at pool tests which can be replicated and fixed on the bench.

B. *Watertight Testing of Servo Housings*

1) Required Equipment:

- Servo housing (bottom and top half)
- Servo shaft
- $\geq$ 20ft rope
- $\geq$ 15ft deep pool

2) Steps:

1) Install both the shaft and main housing O-rings into the respective grooves on the shaft and lower housing half
2) Install the shaft onto a known dead servo using 4x 6mm M2x0.4mm screws and ensure free rotation of the shaft
3) Put on nitrile gloves and coat all O-rings with a pea-sized amount of O-ring grease
4) Place servo and shaft into the lower housing half, and install 2x BlueRobotics penetrator blanks onto the lower housing half
5) Fit the upper housing half onto the lower housing half, passing the shaft through the upper housing hole and pressing firmly together to seat O-rings
6) Install 10x 0.5" 4-40 screws in a star pattern to bolt together both housing halves, tightening the screws to 3 in-lbs
7) Tie a 20ft length of rope to the servo housings and release into a pool at a depth of 15ft. Leave sitting in water for at least 1 hour
8) Retrieve the servo housing, disassemble, and check for water present in the housings or on the servo
9) Reassemble the housing according to steps 4-6, and give to a swimmer to hold underwater at least 2ft deep
10) Have the swimmer rotate the servo shaft 180° at least 50 times
11) Repeat step 8

3) Pass/Fail Criterion: The test is deemed successful if there is no water or moisture present in the housings after both disassembly steps, and is deemed a failure otherwise.

4) Results: On Feb 4, 2024, this test procedure was completed at The Ohio State Recreational and Physical Activity Center dive well. The servo housing was kept underwater for a majority of the 4-hour long pool test, and checked three times throughout the test. In every instance, the test was a success as there was no water found in the housing.

C. Talos Controller Tuning

1) Preamble: To ensure optimal vehicle performance, tuning the PID/SMC hybrid controller is essential. Before beginning the tuning process, some setup is required. It is recommended that a stationary operator, as shown in **Fig. 7**, monitor the motion of the vehicle and steady it between trials. To obtain real time feedback data, use a plotting tool such as Plot Juggler or the MATLAB Simulink interface. Additionally, the controller can be tuned without redeploying the configuration file to the AUV by using the RVIZ parameter panel tool. Finally, ensuring that the vehicle is in the correct physical configuration and batteries are correctly positioned on the adjustable rails will increase tune longevity.

2) Required equipment:

- Talos AUV
- Ethernet Tether
- UWRT Safety Stack
- UWRT Operator laptop (or an Ubuntu 22.04 laptop with ROS2 humble installed)
- Trained Swimmer

3) Steps:

1) Begin by roughly tuning the heave feed forward term – the robot should not be moving vertically at a significant rate, but some movement is allowable at this point. Feedback control should be disabled at this point.
2) Next roughly tune the pitch and roll axis feed-forward such that motion in these axes are at a minimum when the AUV is in the desired nominal position.
3) Continually, adjust the heave, then roll and pitch axis to minimize movement in all three axes. Some movement in these axes will always occur, however, with fine tuning, it should be barely noticeable.
4) Optionally, adjust the feed-forward terms for the yaw, surge, and sway axis. However, the feedback control system can typically overcome slight motions and adjusting these terms can make tuning heave, pitch, and roll axis much harder.
5) Enable the feedback control system. Set all axes to be disabled. Adding axes one at a time simplifies the tuning process. The control mode for each axis is suggested but either SMC or PID works for all axes. Generally, SMC results in a more stable linear position, however, for the angular axes, particularly the roll and pitch, the unstable nature of the desired state results in rapid thruster reveritsal.
6) Enable the heave axis and assign SMC as the desired control mode . Tune the 0th order eta term such that the vehicle oscillates around the setpoint with minimal motion. Next, tune the lambda term again to minimize oscillation. At the end of this process, the vehicle should have a peak-to-peak amplitude of less than 1 cm.
7) Enable the roll axis and assign the PID as the desired control mode. Begin by tuning the proportional gain so that the vehicle is within 2.5 degrees of the setpoint. Steady state error is preferred over oscillation at this point.
8) Enable the pitch axis and assign the PID as the desired control mode. Begin by tuning the proportional gain so that the vehicle is within 2.5 degrees of the setpoint. Steady state error is preferred over oscillation at this point. Ensure the roll axis is still within 2.5 degrees and adjust as necessary.
9) Adjust the damping gain on the roll then pitch axis to minimize oscillation.
10) Slowly increase the I gain so that the vehicle converges to the setpoint relatively quickly (¡2 seconds from 2.5 degrees from the setpoint). By republishing the gains to the controller, the accumulators will be reset to allow for response analysis. This should allow the vehicle to maintain the nominal position in these axes within one degree at all times.
11) Enable the surge, then sway axis and repeat the process used to tune the heave axis.
12) Enable the yaw axes and assign the PID as the desired control mode. Tune the P-gain such that the vehicle converges to the setpoint at a reasonable rate. Then tune the damping gain so that oscillations are less than one degree peak-to-peak – reduce P gain if necessary.
13) Adjust the SMC velocity profiles to represent the desired vehicle velocities relative to the distance from the setpoint. Higher velocities may result in significant deviations from the setpoint in the angular axes due to the difference in the center of mass and the center of drag.

4) Pass/Fail criterion:

- The test is considered passing if at the end of all steps, the Talos AUV can maintain its position with 1 cm in the heave, surge and yaw DOF, 1 degree peak-to-peak in the yaw DOF and 2 DOF peak-to-peak in the roll axis. The robot should be commanded into various states and move between states in a reasonable fashion. Additionally, the AUV must be able to recover its position when subjected to a substantial shove.
- The test is considered a failure if the AUV cannot meet any of the requirements listed above or the tuning process cannot be completed.

5) Results: This test was executed multiple times throughout the year mainly after significant physical changes to the AUV were made - i.e. the addition of buoyancy bottles. After the latest control system released, the tuning process has always succeeded. However, the time required to complete the process varies - typically around 20 minutes - based on the extent of adjustment needed.

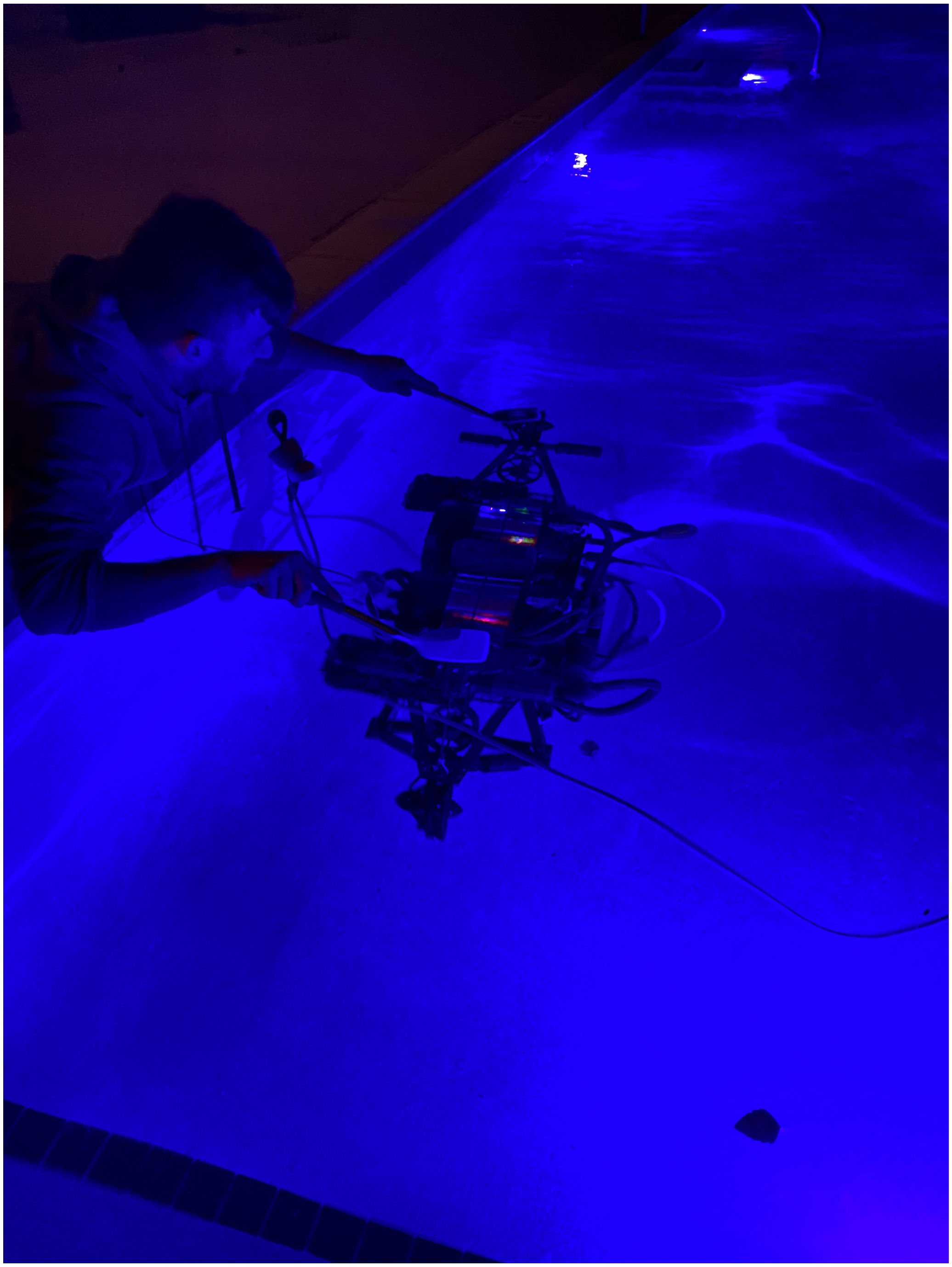

Fig. 7: An experienced operator tuning the AUV's feed-forward

D. Thruster characterization

To improve control accuracy, per-thruster measurements were taken to relate DSHOT commands with thruster forces and RPMs. This was done using a load cell and an Arduino-based control script, as shown below.

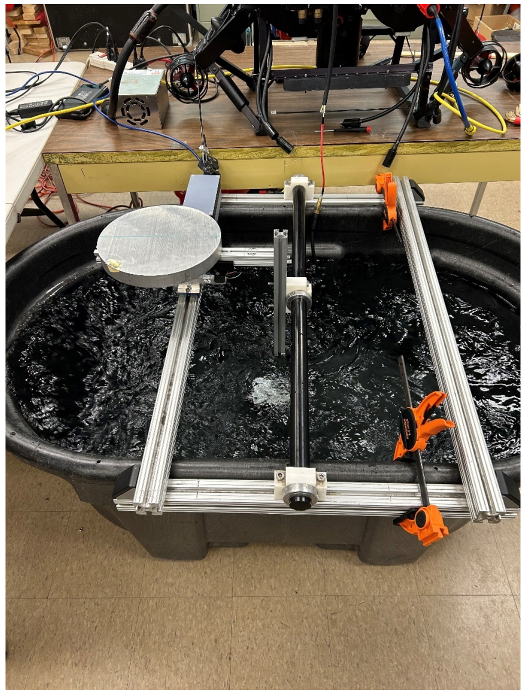

Fig. 8: The apparatus used to characterize the thrusters

1) Required equipment:

- Thruster force gauge (**Fig. 8**)
- Talos AUV
- Operator laptop with ROS2

2) Steps:

- Attach Arduino on force gauge to operator laptop
- Launch calibration script
- Make call to the action spawned by the thruster calibration script
- When prompted, publish a std_msgs/msg/Empty to the trigger topic received by the calibration script.
- When prompted, reverse the thruster direction on the force gauge and publish the Empty to the trigger topic
- The script will generate a .cvs file when the calibration is done

3) Pass/Fail critereon: The test is considered passing if the calibration script does not report any errors, and failing otherwise.

4) Results: Using this characterization, the team was able to determine that using raw DShot commands to control force created inconsistencies between thrusters or batteries, as shown in Fig. 9. However, as shown in Fig. 10, commanding an RPM output yielded more predictable force.

This characterization motivated UWRT to create the firmware-level PI controller described in the controller section. The force output of the thrusters using that firmware-level controller is shown below:

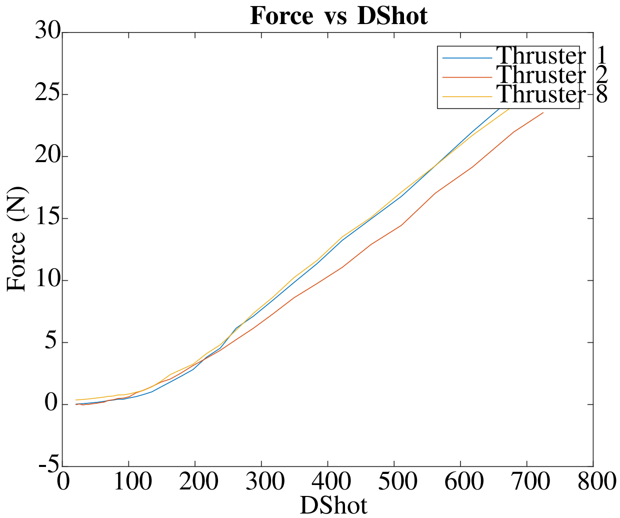

Fig. 9: Force vs Dshot command for three Blue Robotics T200 thrusters.

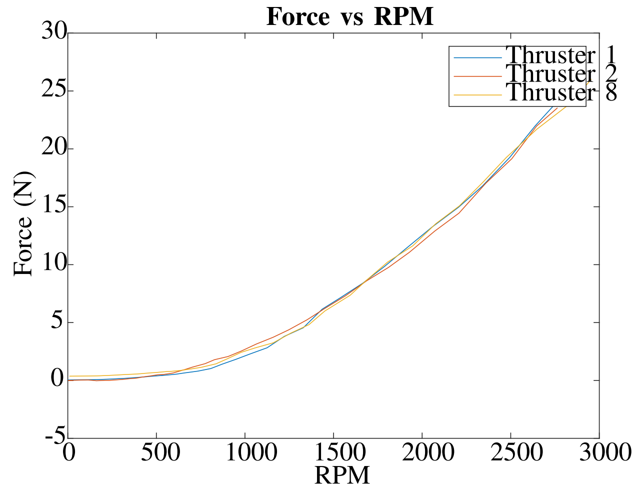

Fig. 10: Force vs RPM for the same Blue Robotics T200 thrusters as in fig. 9

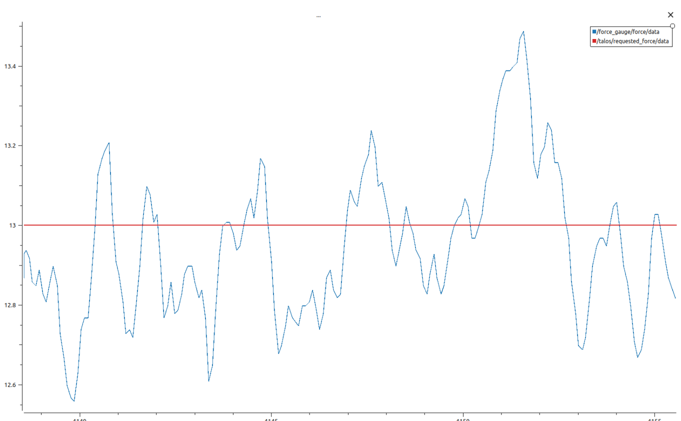

Fig. 11: Firmware-level controller force output on a different thruster and ESC

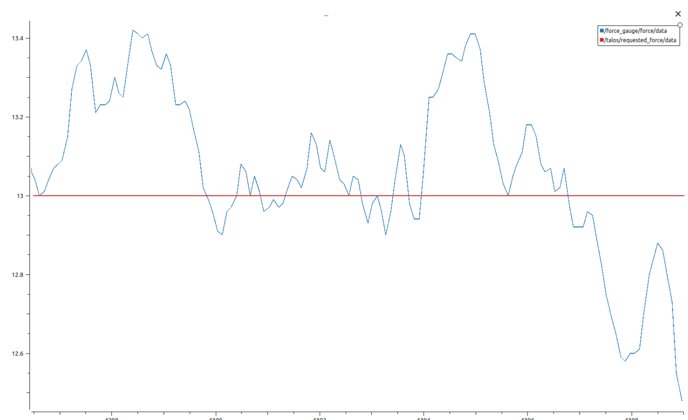

Fig. 12: Firmware-level controller force output on a low battery

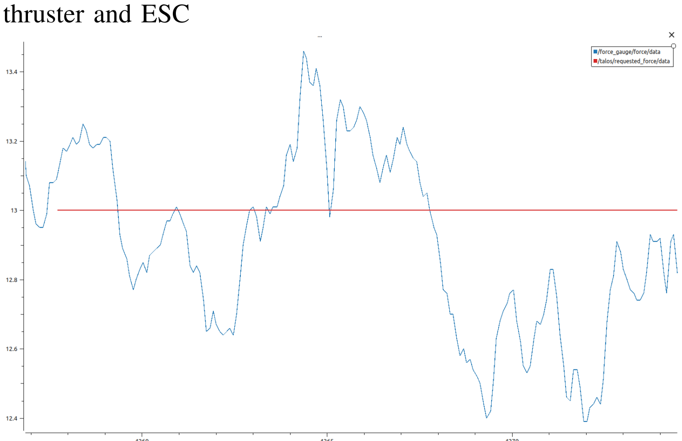

Fig. 13: Firmware-level controller force output on a full battery

APPENDIX C: SEARCH ALGORITHMS

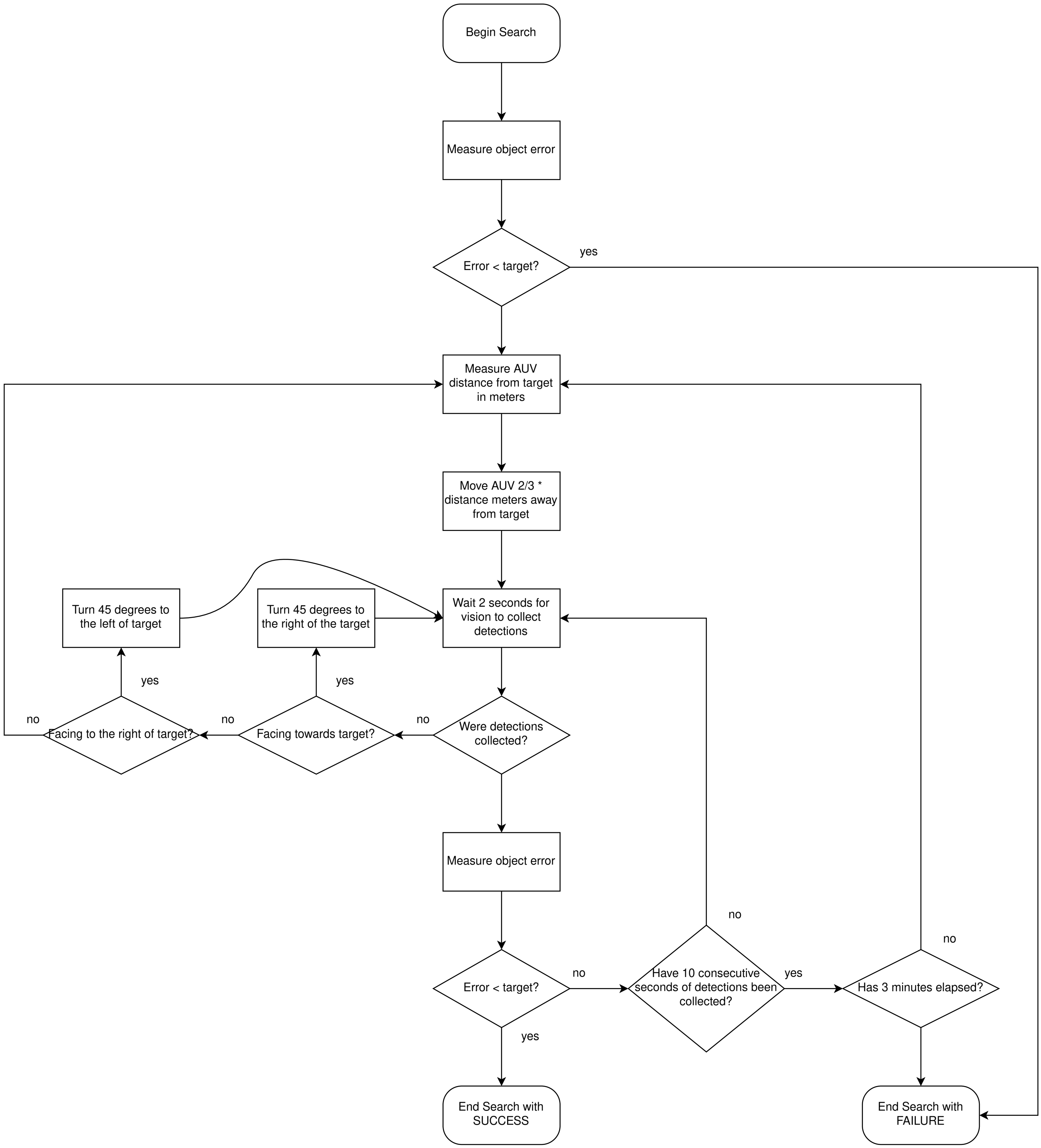

Fig. 14: Autonomy Search algorithm flow chart

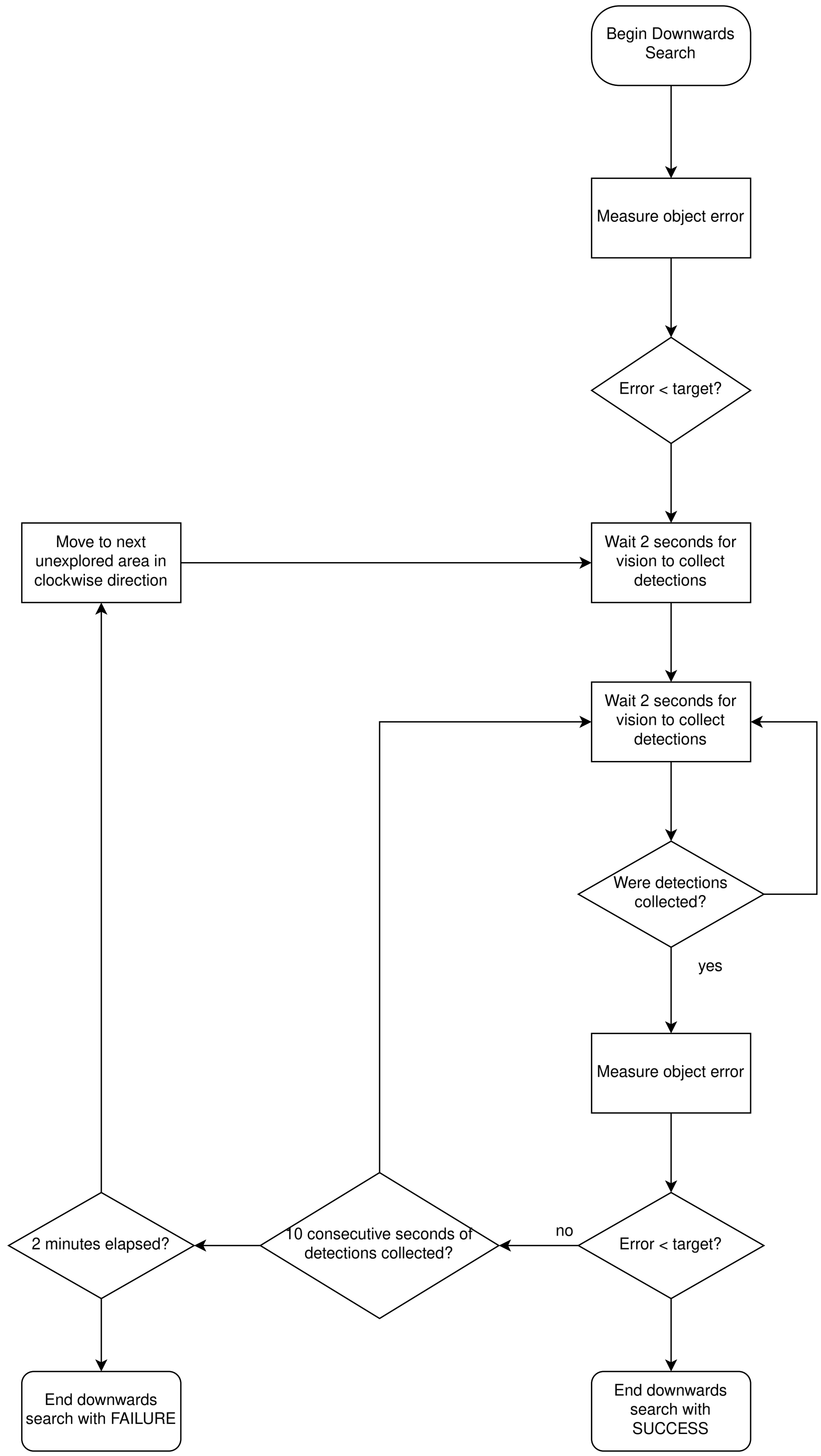

Fig. 15: Autonomy Downwards search algorithm flow chart

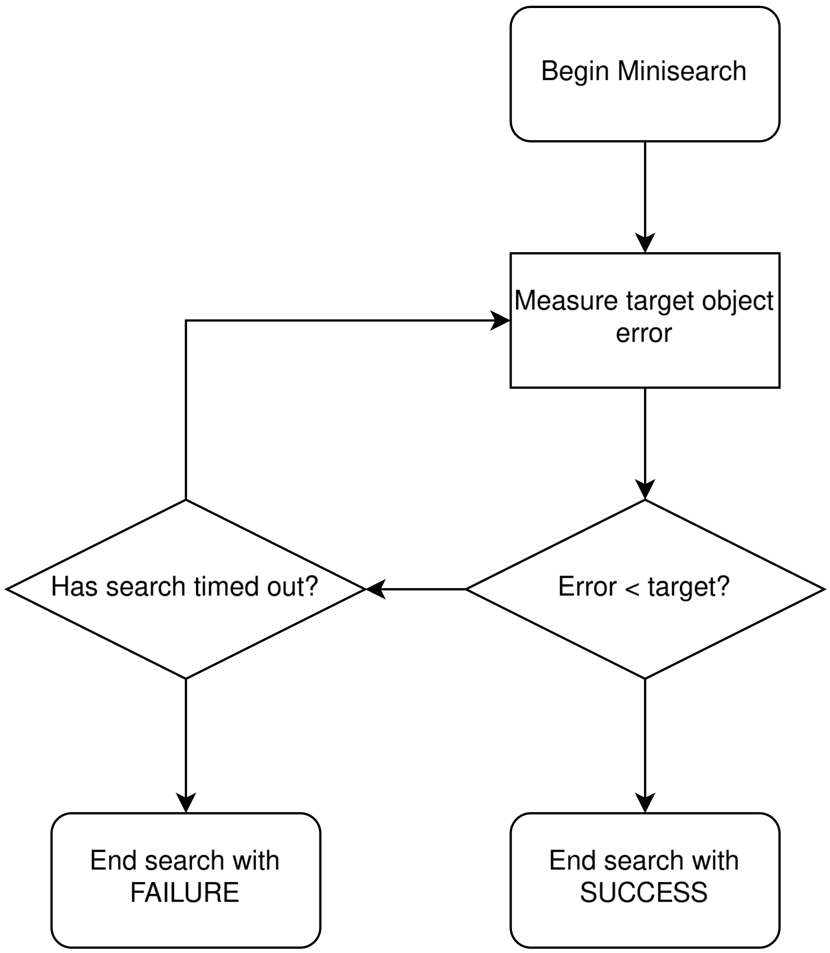

Fig. 16: Autonomy miniature search algorithm flow chart

Appendix D: Electrical Cage Explosions

The board cage is based on 3D printed parts leveraging the PCBs for internal rigidity. More compact TPU mounts were designed to allow for a fifth board in the cage. Notable changes to the board cage include the removal of the power and actuator board - now the POAC board, the addition of the acoustics board, the addition of the LED boards, and a AUX switch - a hall effect sensor.

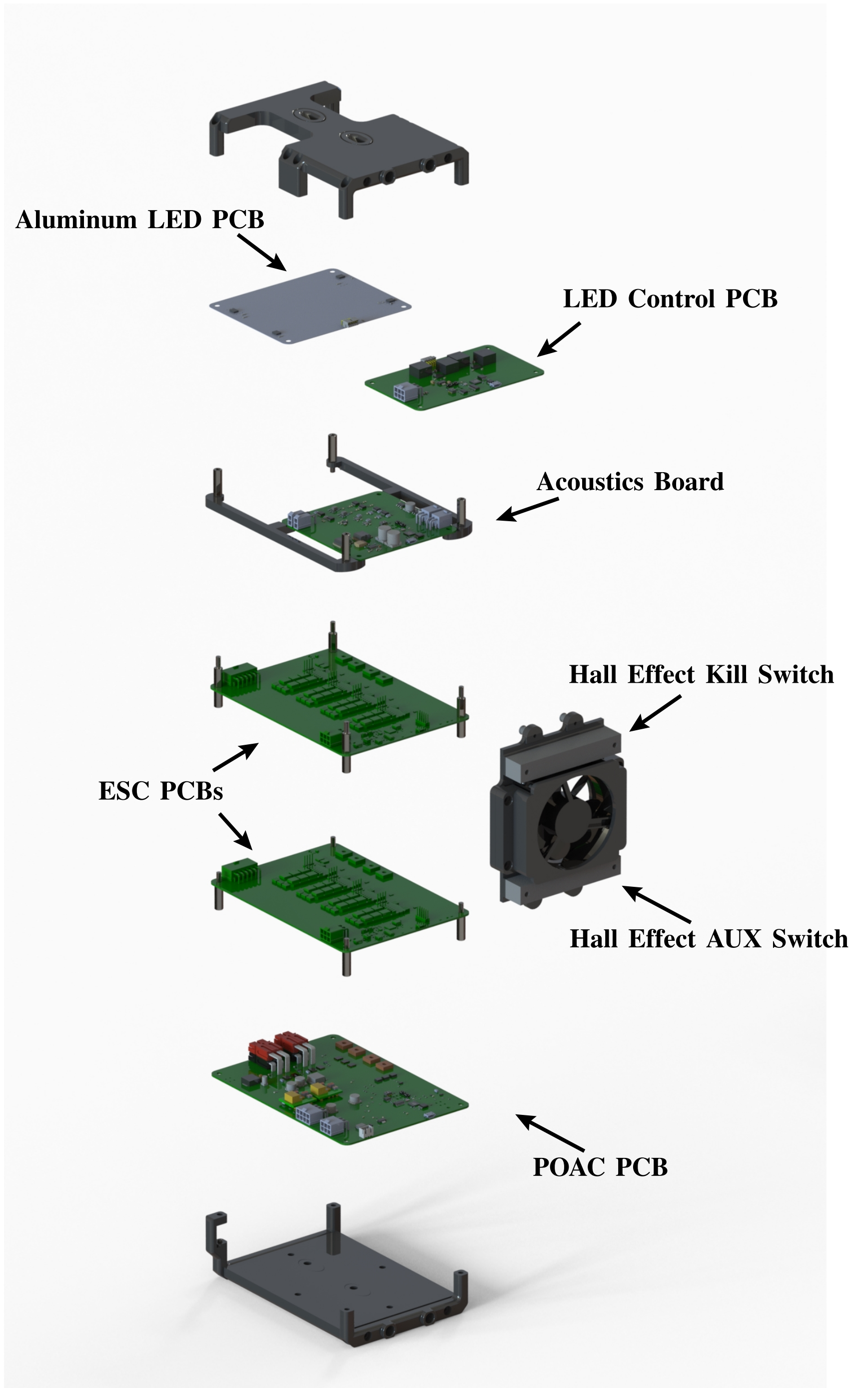

Fig. 17: Board Cage Exploded View

The camera cage was re-designed with a focus on spacial efficiency and wire management. To fit UWRT's new fiber optic gyroscope, a shelf style design was employed to house components such as the network switch, IMU, FOG and breakout board while maintaining accessibility. To keep wiring organized, UWRT designed a reusable, 3D-printable "clip tie" to hold wires in place.

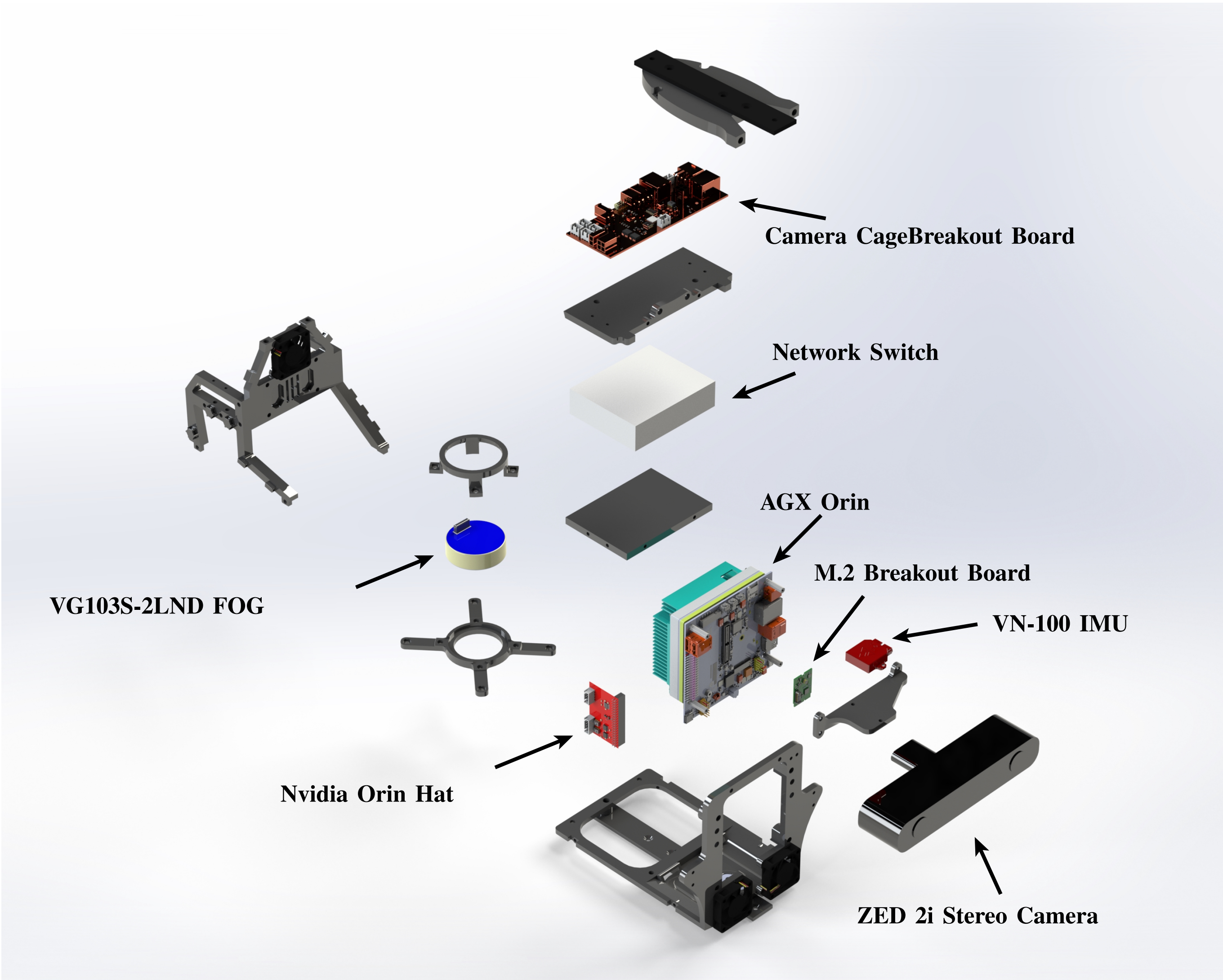

Fig. 18: Camera Cage Exploded View

APPENDIX E: ACOUSTICS DIAGRAM

The superheterodyne receiver in the acoustics system mixes the output of each hydrophone with a variable frequency created by a Local Oscillator (LO). The resulting signal is put through a high Q factor bandpass filter tuned to 107.5KHz. The system can tune to any pinger frequency by setting the LO's frequency to 107.5KHz minus the frequency of the pinger being tracked.

Fig. 19 shows a time delay sample calculation made in MATLAB. *Hydrophone Signal*, shows two sample hydrophone signals sampled at 300 ksps. Both signals consist of sinusoidal waves at 25, 30, 35, and 40 KHz with the 30 KHz wave being the greatest in amplitude. Hydrophone 2's 30 KHz wave is delayed by 200 samples. *Mixed Signal* shows both signals mixed with a $107.5\ \text{KHz} - 30\ \text{KHz} = 77.5\ \text{KHz}$ wave. *Filtered Signal* shows both signals after a steep bandpass filter centered at 107.5 KHz. *Signal Envelope* shows the envelope of the filtered signals. Finally, *Pulse,* shows the envelope converted to a pulse based on signal strength. The time difference between both hydrophones is found to be 0.667×10^{-3} seconds which which matches the expected calculation $\frac{200\ \text{samples}}{30\times10^3 \frac{\text{samples}}{\text{second}}} = \frac{1}{1500} = 0.667 \times 10^{-3}$ seconds.

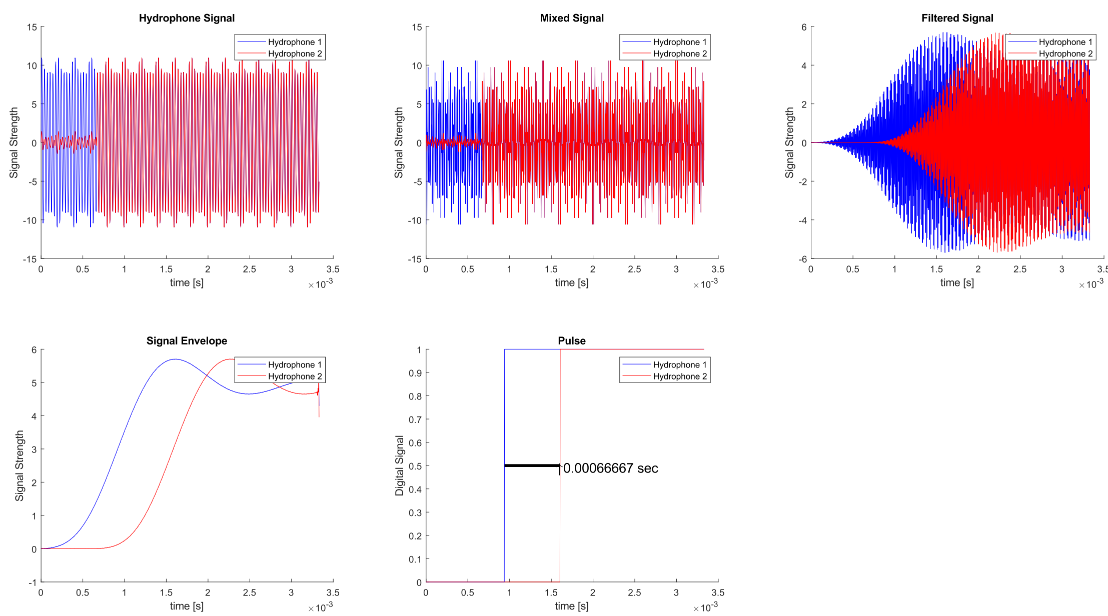

Fig. 19: Acoustics Sample Waveforms

Appendix F: Smart Battery Charger

Last year, the team revolutionized Talos' batteries with the smart battery system. Despite this advancement, the charging process remained convoluted, shown in the charging procedure below. To address this, a custom charger was designed to streamline the process and implement various safety features. Notably, fuses were placed along the main power rails, battery pack manager circuits were implemented to prevent overcharging, and the RP2040 controller chip was powered from multiple sources to improve reliability. The new charger also protects the fuses in the battery housing which, if blown, require the housing to be opened. Additional features include a capacitive touch screen and custom interface firmware, higher charge by increased cell balancing, and smart charging using either USB-C or wall adapter barrel jack connection.

Charging Procedure:

1) Connect power cable to battery 1 and charger
2) Connect data cable to battery 1
3) Remove power cable from charger and quickly replace with data cable
4) Connect power cable to battery 2 and charger
5) Start charging batteries

Fig. 20: Smart Battery Charger 1

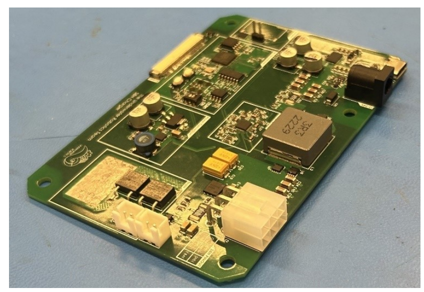

Fig. 21: Smart Battery Charger 2

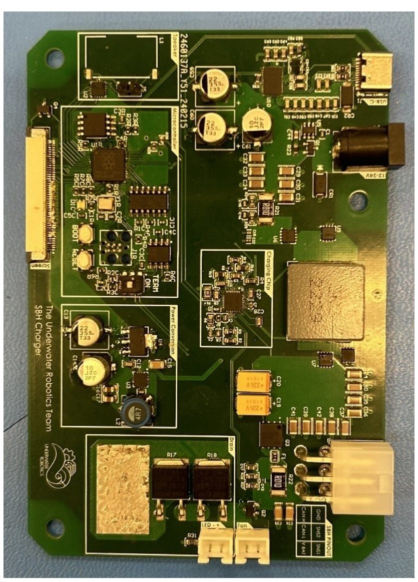

Fig. 22: Smart Battery Charger 3

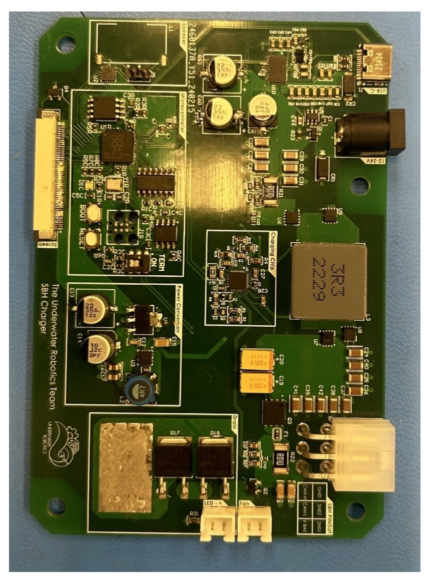

Fig. 23: Smart Battery Charger 4

APPENDIX G: ELECTRICAL SYSTEM BLOCK DIAGRAMS

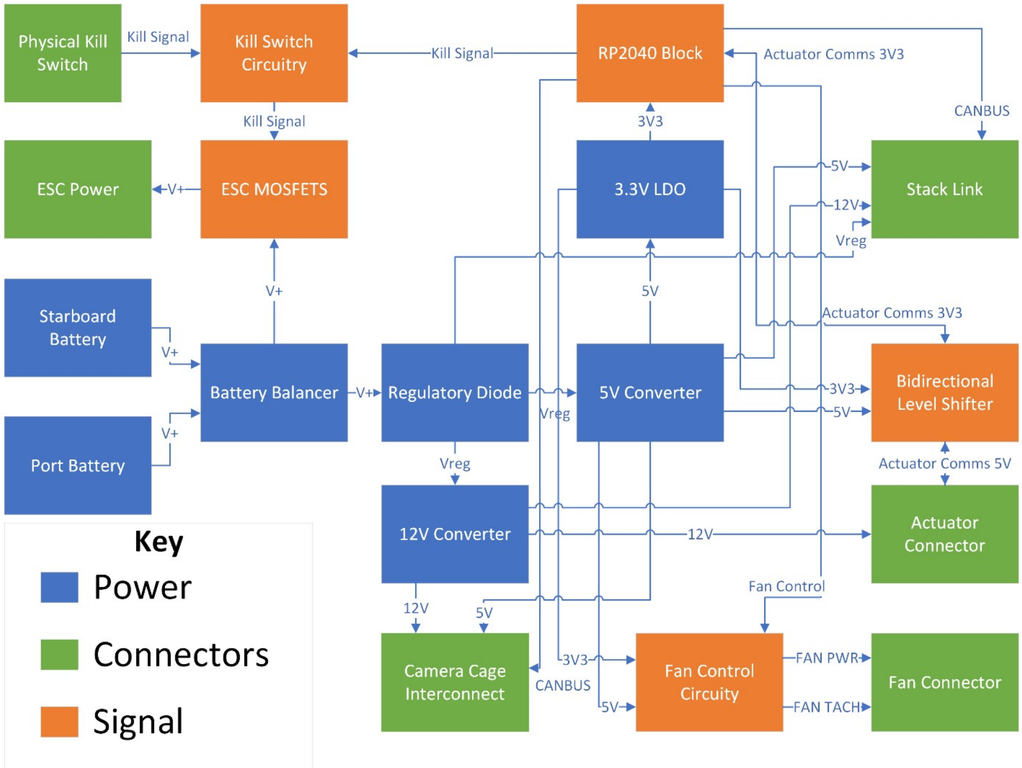

Fig. 24: Poactuator Electrical Block Diagram

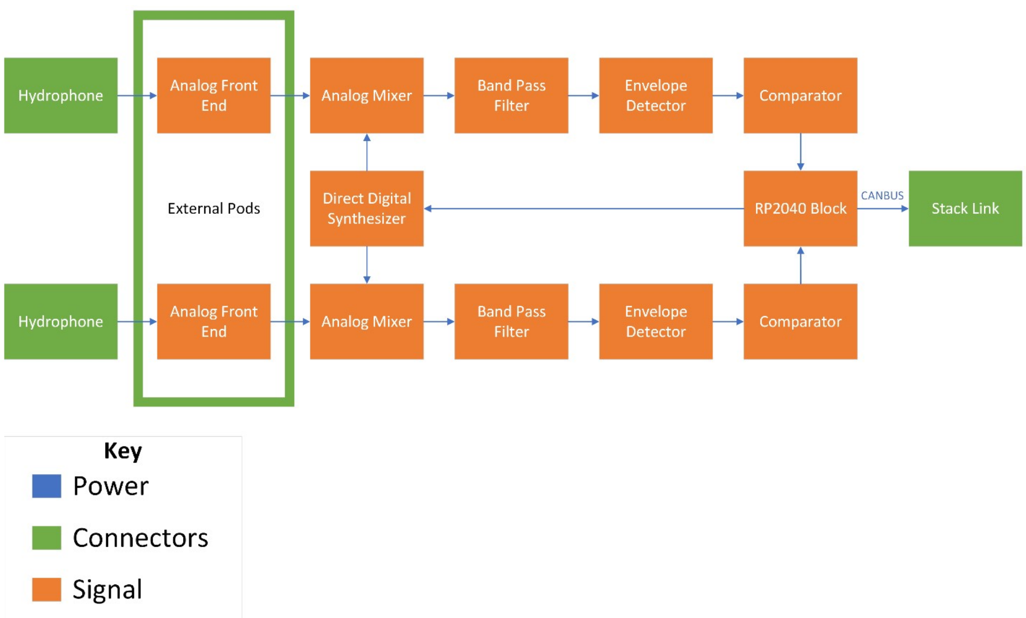

Fig. 25: Acoustics Electrical Block Diagram

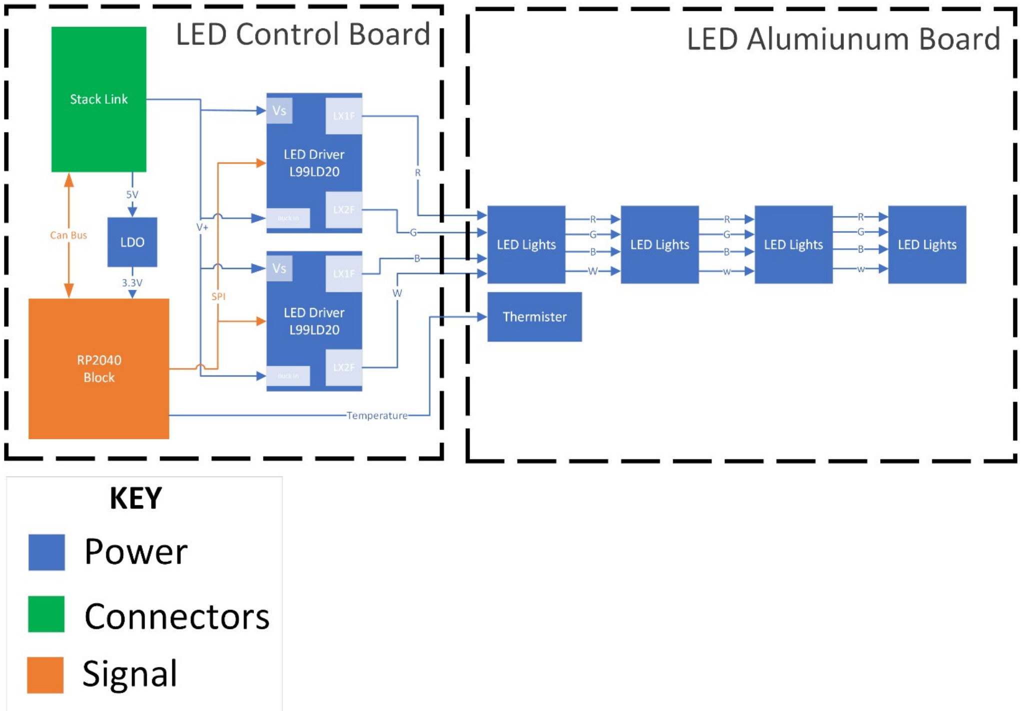

Fig. 26: LED Electrical Block Diagram

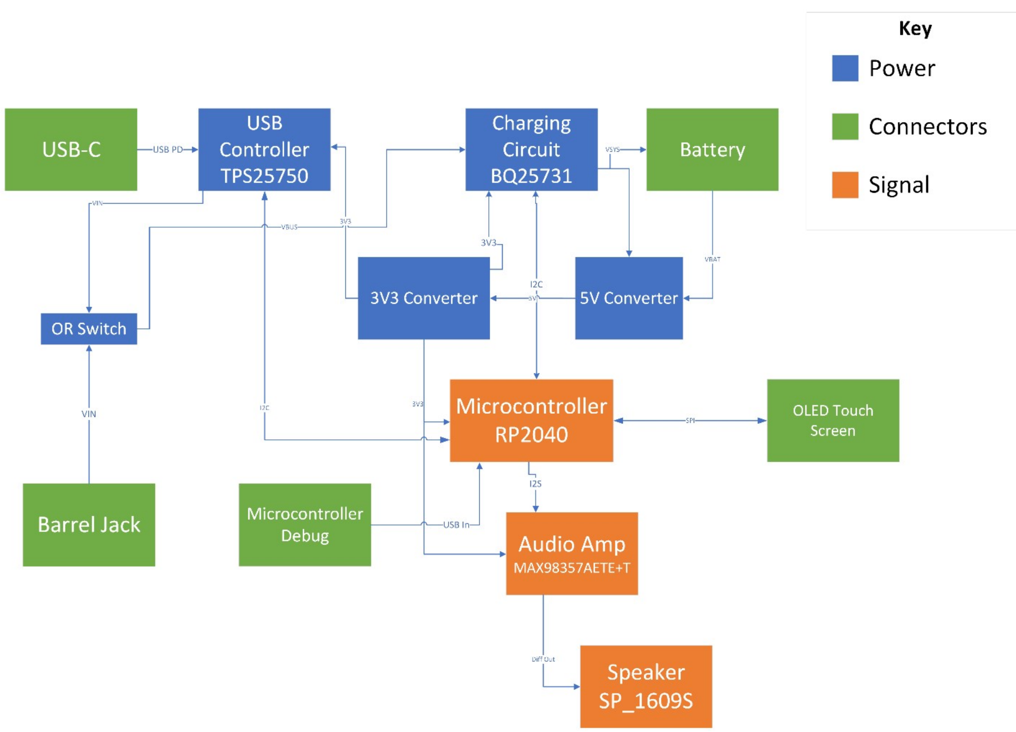

Fig. 27: Smart Battery Charger Electrical Block Diagram

APPENDIX H: RIPTIDE SOFTWARE ARCHITECTURE

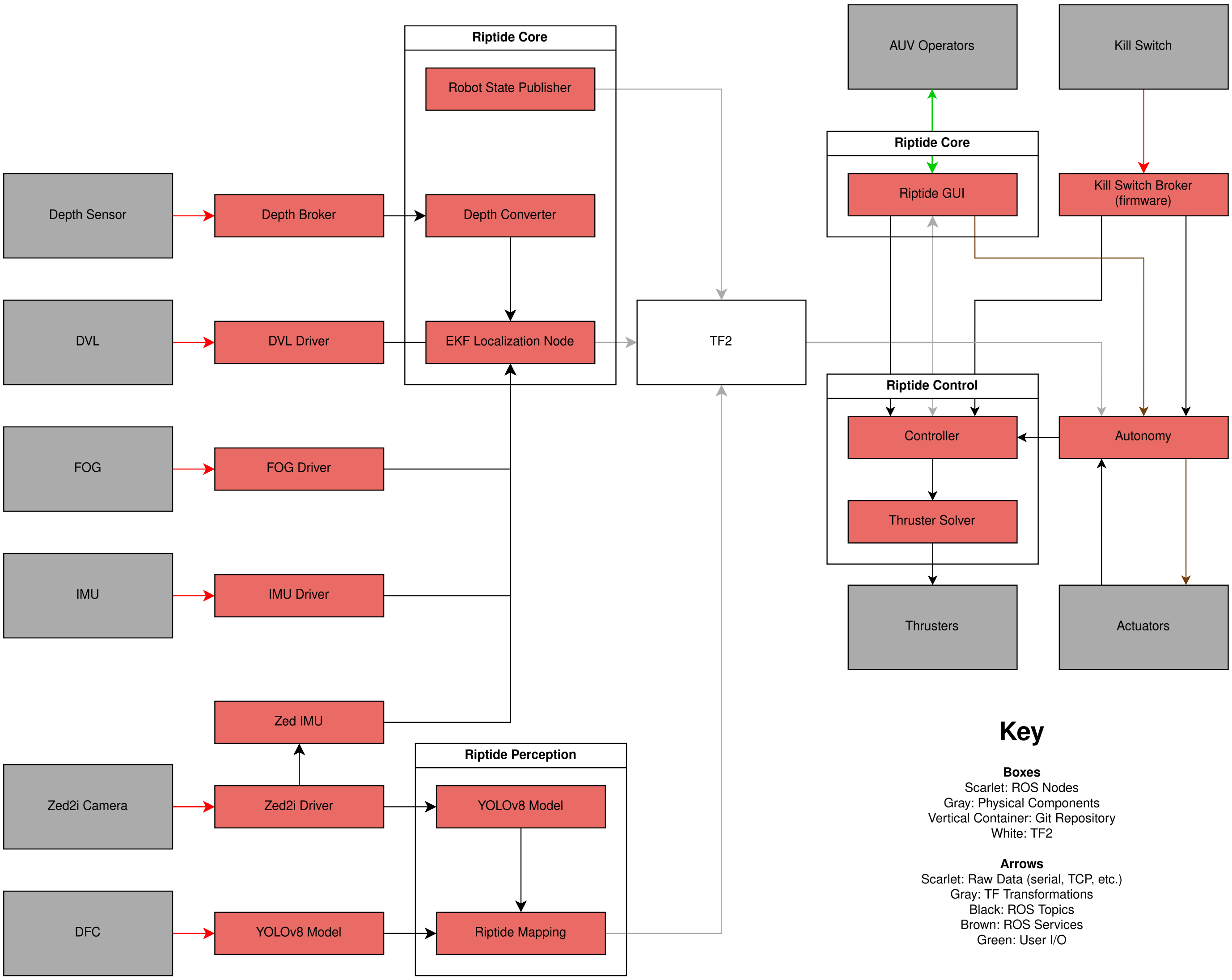

Fig. 28: Riptide Software Architecture Diagram

Appendix I: MATLAB Integration in ROS2

The team took several steps relating to easing the integration of MATLAB and the team codebase. Simulink Project was set up for the control repository, allowing for basic Git features to be used to handle simple model changes and merges. Additionally, a MATLAB installer script was created to automate an involved MATLAB setup which is required for MATLAB to work with the codebase. This setup, if implemented properly, allows for users to run Simulink models in the editor during vehicle testing:

- MATLAB R2023A must be installed to ensure that all members can work on the Simulink models themselves.
- Support for UWRT's custom ROS message types must be built into MATLAB so the controller models can ingest commands - this is a several step process
- A specific set of toolboxes must be included in the install:
  - Aerospace Blockset
  - Aerospace Toolbox
  - Control System Toolbox
  - Embedded Coder
  - MATLAB Coder
  - ROS Toolbox
  - Simulink
  - Simulink Coder
  - Simulink Test
- The libstdc++.so.6 file shipped with MATLAB must be renamed or removed, or code generated from the models will not build on the local system due to undefined references.
- Python 3.9 must be installed to use certain ROS toolbox features. The UWRT MATLAB installer builds it from source such that it does not interfere with the version of Python already on the system.

To assist with the generation and management of the C++ code from Simulink, UWRT created a ”model manager” script. This script, which can be run with the ROS2 command line interface (CLI), can manage batches of Simulink models with a simple set of command arguments. Its primary function is to generate Colcon packages for all models detected in the workspace for both ARM and x86 machines, reducing the number of commands needed to maintain each model by up to five. Additionally, it can organize the resultant archives for easy hosting, as well as download and set up archives from any GitHub release, and automatically build and deploy the generated or downloaded archives to the AUV. The manager accomplishes this while being conscious of the machine architecture.

Appendix J: AUV Headless Interface State Diagram

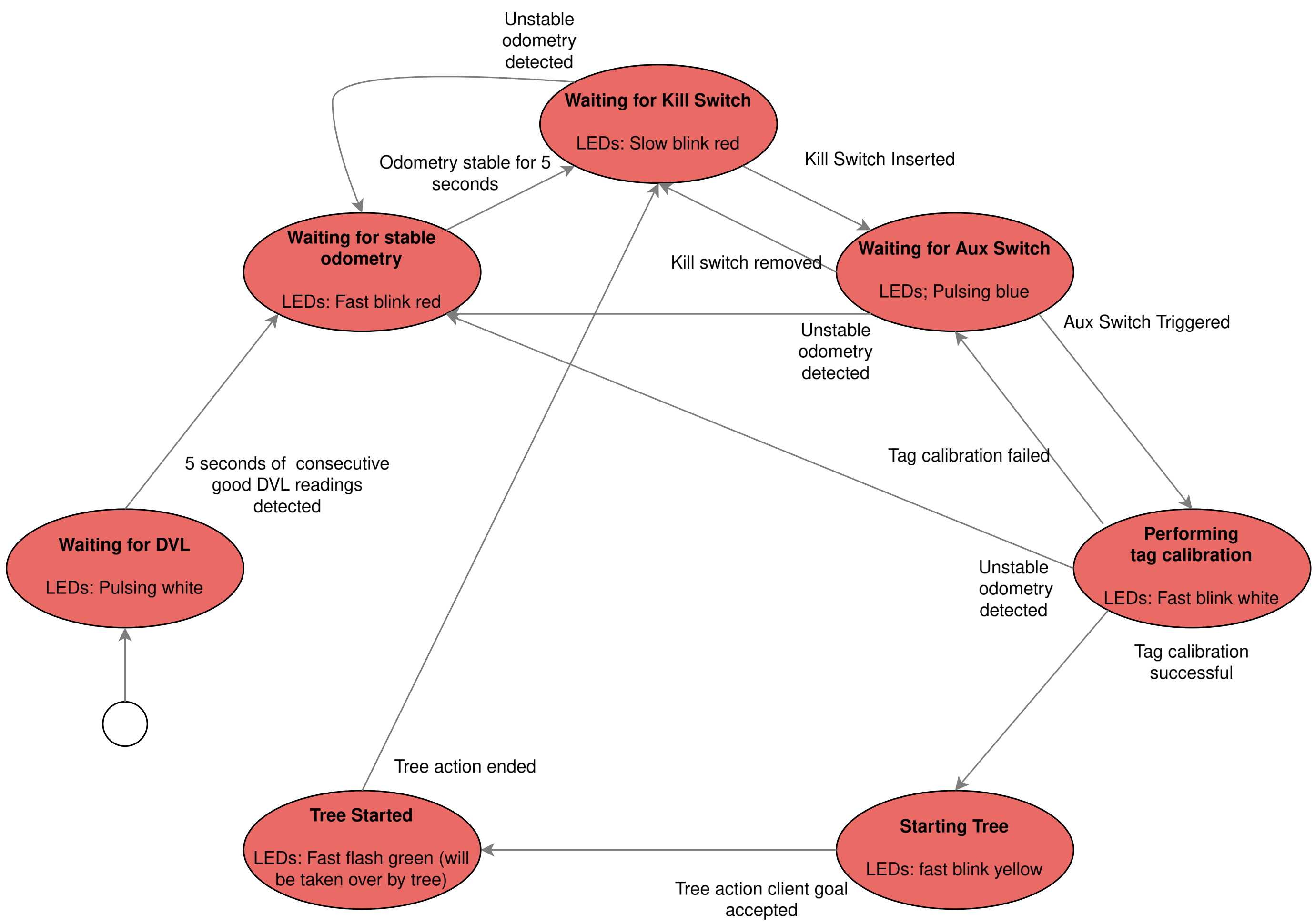

Fig. 29: The state diagram of the AUV headless interface

APPENDIX K: CONTROLLER DIAGRAM

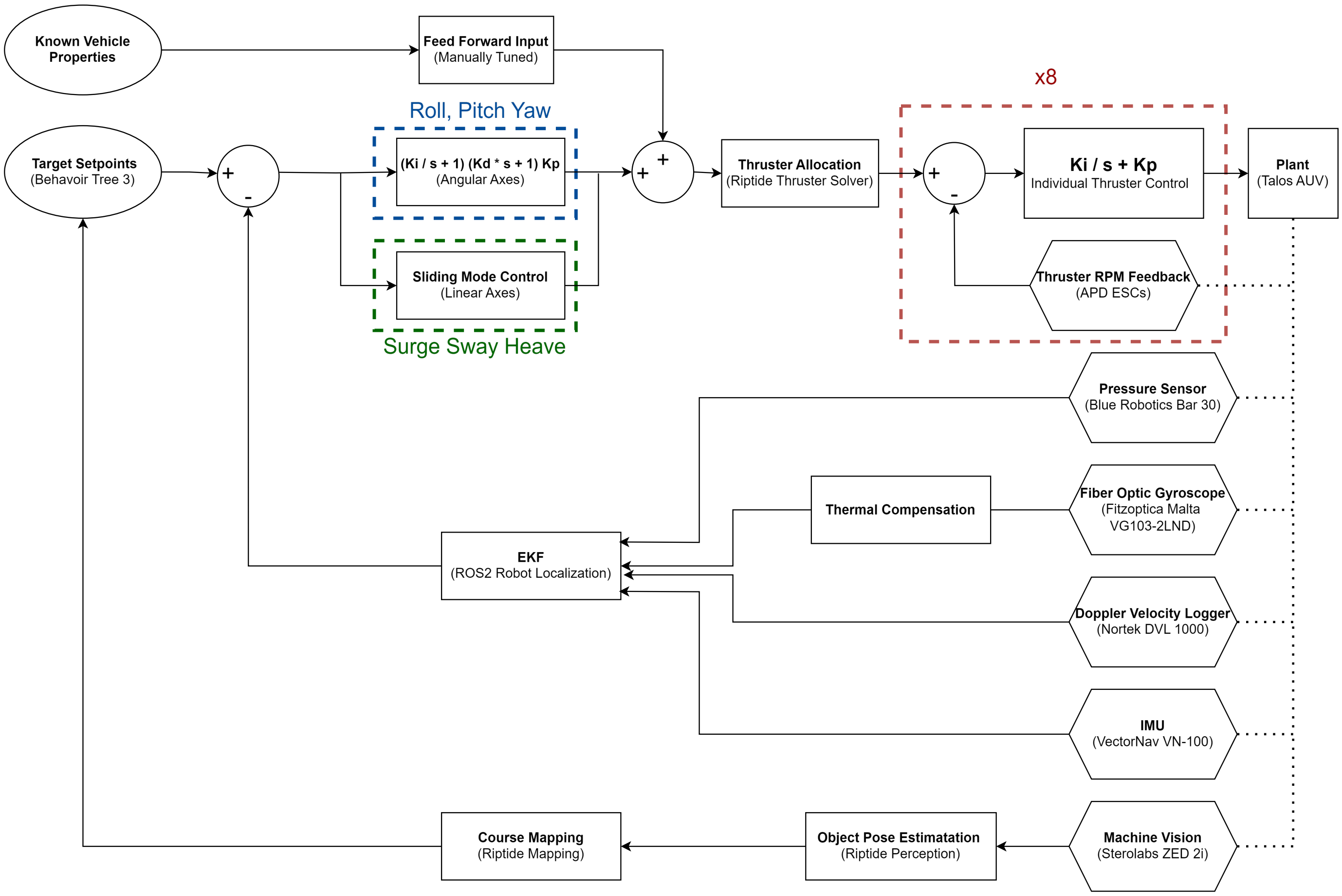

Fig. 30: The positional control feedback diagram

APPENDIX L: AUGMENTED DATA

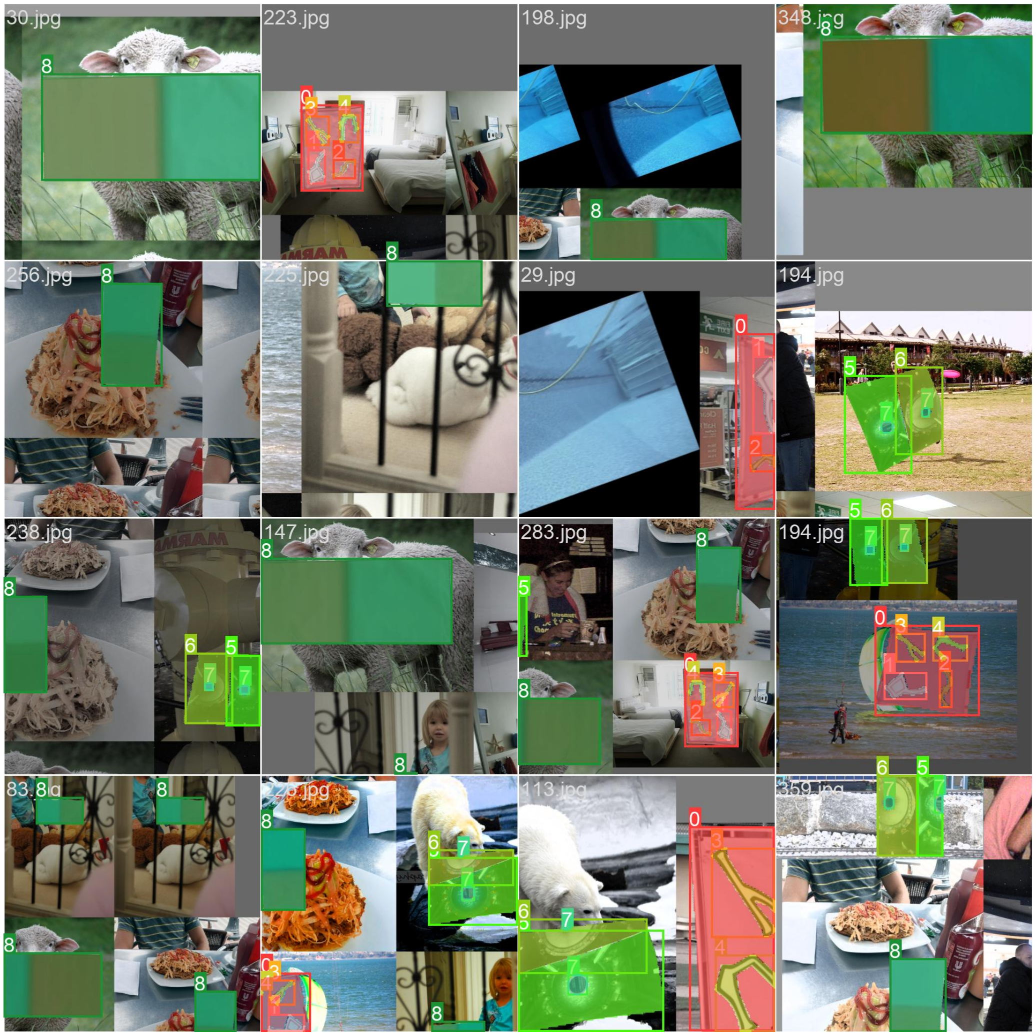

Fig. 31: Augmented Data for the YOLOv8 Model

Appendix M: Tether Spool

Whilst it might not be the most glamorous of projects, the UWRT family grew this year with the introduction of the beloved tether spool. This project eliminated a significant hurdle in AUV testing: tether management. The tether spool utilizes an Ethernet slip-ring, allowing the tether to be wound and unwound without torsional strain. This innovation has significantly increased overall system reliability and eliminated the need to disconnect the AUV to extend the tether length.

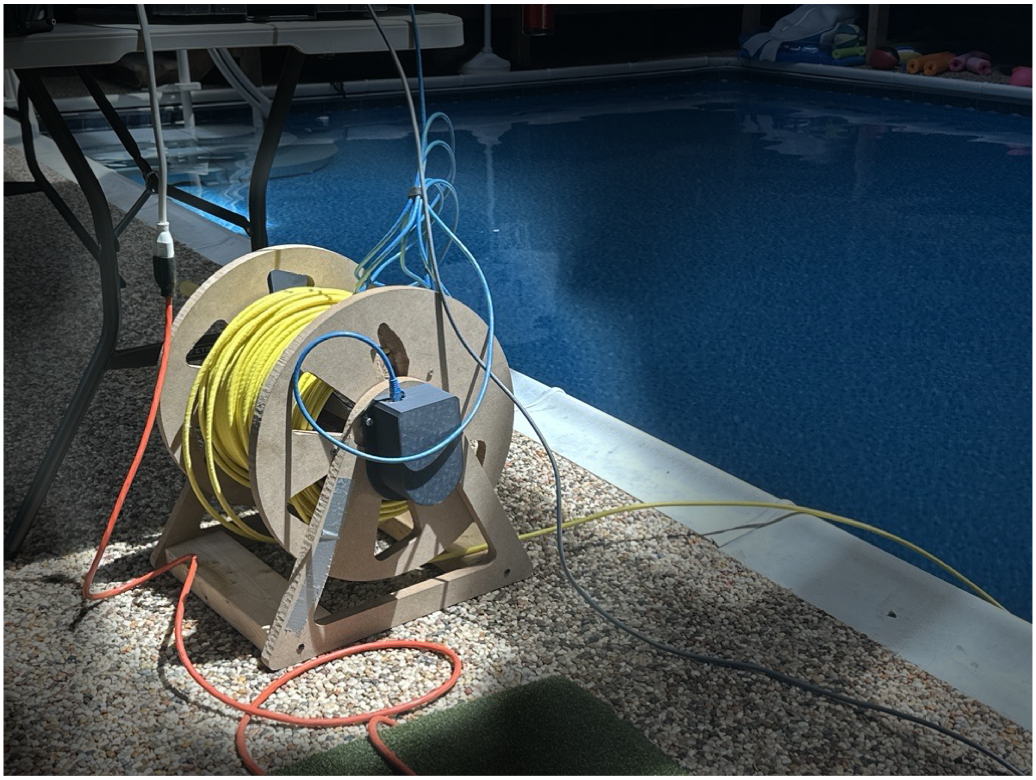

Fig. 32: The UWRT Prototype Tether Spool

Appendix N: IMU Driver

During RoboSub 2023, the team encountered issues with Talos' IMU driver, causing incorrect data to be fed to the navigation system, making the controller ineffective. A new driver was developed to connect the VectorNav VN100 IMU to the ROS network to improve the navigation system's robustness. Previous iterations of this software provided multiple locations to configure the sensor, which occasionally led to misconfiguration. Thus, the new driver was created with the goal of simplicity and only permitted writing settings directly to the sensor's registers through VectorNav's Control Center.

Appendix O: Firmware Introspection

For this competition, a large push for the embedded system firmware was improving debugging ease and insight into the current state of all embedded devices in the AUV. A custom command line interface was developed to view debugging information over the remote secure shell (SSH) session available via the AUV tether. This provides in-depth information about any crashes and faults, their locations, and other useful debugging data. Additionally, custom commands can be registered by each firmware project to perform low-level debug and configuration actions on hardware, such as provisioning task mechanism IDs or viewing CAN bus error rate statistics.

Appendix P: Community Outreach

UWRT's STEM outreach initiative and goal of educating others about the field of underwater robotics expands from Ohio State's campus to the greater Columbus area. The team engages youth in the community through a five-week after-school program called STEMBot. UWRT returned to Rosemore Middle School with an improved STEMBot program that allowed middle school students to gain more hands-on experience with developing underwater robots. Students had the opportunity to assemble a STEMBot both in TinkerCAD, a kid-friendly CAD software, as well as its physical form both electrically and mechanically. Students also learned the basics of programming and were able to compete with their robots against each other in an underwater obstacle course. At the end of February, The Ohio State Underwater Robotics Team had the wonderful

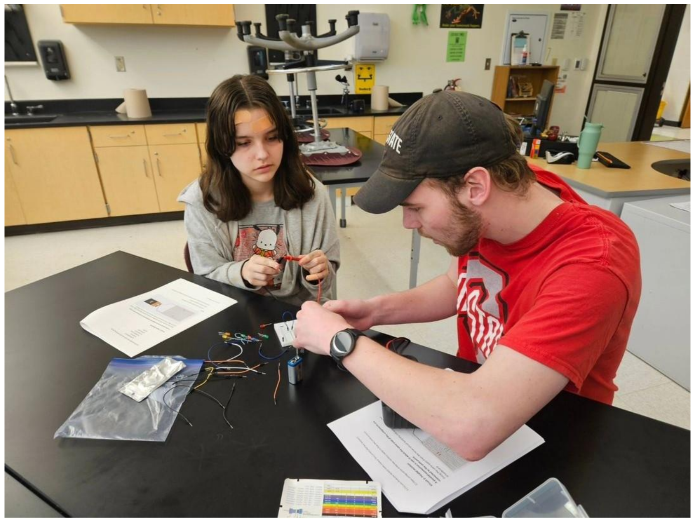

Fig. 33: A UWRT member teaches a young student about circuits during the Stembot Workshop

opportunity to expand their outreach efforts to the Columbus School for Girls. The event focused on encouraging young girls to explore opportunities in STEM fields. One highlight was the team's president, Amber, sharing her personal journey in engineering, from high school to post academia, offering valuable insights into her experiences and how they ignited her interest in underwater robotics. Amber emphasized the importance of active involvement and making the most out of STEM classes and events, highlighting how such experiences can foster curiosity, creativity, and problem-solving skills. During this event, the girls had the unique opportunity to interact with Talos, the team's autonomous underwater vehicle. Seeing Talos up close and learning about its components such as PCBs, actuators, and retired hull components, provided a tangible and exciting glimpse into the world of robotics and engineering. The students enjoyed asking the team questions about the design and functionality of Talos and were curious about how they could implement engineering practices into their own classes and interests outside school. Some lucky attendees even got to take the reins and maneuver Talos through obstacles underwater, experiencing firsthand the thrill the team feels at RoboSub. UWRT expanded its outreach efforts by partnering with Troop

Fig. 34: UWRT's former President answers questions from a young student about the team's AUV

6755, a local Girl Scout Troop, with the goal of introducing young girls to the significance of underwater robotics. As part of the event, a hands-on activity centered around buoyancy and center of gravity was conducted, allowing the girls to balance boats equipped with sails, effectively illustrating key principles in an understandable manner. Furthermore, the troop enjoyed an exclusive tour of UWRT's laboratory facilities, providing them with the opportunity to interact with current team members and Talos, thereby enriching their experience.

Fig. 35: Troop 6755 poses for a picture with Talos after a tour of UWRT's Lab

Appendix Q: Glossary

- ESC: Electronic Speed Controller; A device which drives brushless motors.
- SMC: Sliding Mode Controller; A robust control technique which uses a discontinuous control signal (e.g, bang-bang) to control a customized ”sliding surface” equation to 0, guaranteeing convergence of a system in finite time
- PID: Proportional Integral Derivative: A control technique which uses weighted error, sum of error, and derivative of error, to determine an output control signal which drives a system to a setpoint.
- Cascaded-P: A control scheme which uses two proportional controllers, one for position, and another for velocity. The position controller output feeds the velocity controller input (e.g. ”cascaded”)
- UWRT: UnderWater Robotics Team
- DVL: Doppler Velocity Log; A device which uses the Doppler Effect to determine its velocity with high precision.
- AUV: Autonomous Underwater Vehicle
- DFC: Downward-Facing Camera
- ROS2: Robot Operating System 2
- CNC: Computer Numerical Control
- CLI: Command-Line Interface
- SSH: Secure Shell; A secure remote terminal tool.
- FOG: Fiber-optic gyro; a device which uses a long, wound, fiber optic coil and the Sagnac effect to determine its rate of rotation with high precision.
- IMU: Inertial Measurement Unit; a device which measures its specific acceleration and rate of rotation in all axes
- ICP: Iterative Closest Point; An algorithm which uses a point cloud as a reference to determine the orientation of an object in an image
- SVD: Singular Value Decomposition; An algorithm which uses matrix math to determine the orientation of a planar object in an image
- GUI: Graphical User Interface
- AprilTag: A visual fiducial marker which can be located in 2D and 3D space using classical CV techniques
- SubConns: Underwater connects which UWRT uses to connect peripherals like the thrusters, batteries, and DFC
- Segmentation: In machine vision; a model supported by YOLOv8 which outputs object detections as tightly bounded polygons rather than boxes
- PCIe: Peripheral Component Interconnect Express; a standard for computer expansion