The Ohio State University Underwater Robotics *Tempest* AUV Design and Implementation

Collin Barack, Phillip Barker, Nathan Becker, Arko Chatterjee, Amber Dellacqua, Matthew Fisher, Brach Knutson, Isabella Richardson, Mitchell Sayre, Dylan Trainor, Cole Tucker

***Abstract*—Since 2016, The Underwater Robotics Team (UWRT) at The Ohio State University have designed and built five Autonomous Underwater Vehicles (AUVs). For the 2022 competition season, UWRT focused on the manufacturing of the newest competition robot—Tempest. The primary objective of this new design was to increase the score at AUVSI's RoboSub competition and establish a foundation built on solid engineering principles for future development. Reliability is at the forefront of the design goals which prompted exhaustive testing of the team's subsystems and of the whole Tempest vehicle. The team was able to construct, test, and iterate on designs to ensure the best performance in the competition.**

I. COMPETITION STRATEGY

A. General Strategy

THE Underwater Robotics Team's (UWRT) strategy was to increase points earned by attempting all tasks and decreasing points lost by designing a new system to avoid the pitfalls of previous vehicles. In 2019, the team lost points due to six major issues:

- Puddles exceeded the maximum weight parameter
- Puddles had a poor thrust to weight ratio
- Puddles had asymmetric drag due to large surface area in certain axes
- Puddles was unable to achieve desired pitch up and pitch down poses
- Puddles failed to complete tasks requiring actuators due to reliability issues
- Puddles housing was difficult to access for repairs

In 2019, the team did not attempt many of the competition tasks which resulted in a score 5,300 points below the winning team. 56 points were lost due to the vehicle exceeding the maximum weight limit. When designing UWRT's new competition robot, Tempest show in Fig. 1, the team tackled its strategy by focusing on a lighter, more maneuverable, and more serviceable design. These changes were in addition to the goal of increasing the reliability of Tempest so that the team could attempt more competition tasks. For the team, reliability is defined as the ability to complete competition tasks with repeatable results and without failure. Electrical, mechanical, and software robustness was fundamental to achieving the desired reliability.

B. Low Level Actions

Fig. 1: Image of Tempest compared to its CAD rendering counterpart.

To approach the competition tasks, the team put together a system to assemble high-level behaviors out of low level building blocks. These building blocks represent fundamental actions that Tempest can perform. Actions allow the robot to move, perceive, and interact with the competition space. At the lowest level, Tempest must reliably move from one location to another while maintaining the orientation commanded. In parallel, the team built a perception solution to guide Tempest through the competition space using a *Simultaneous Localization And Mapping* (SLAM) system based off visual perception from a *ZED2i* stereo camera connected to a *YOLO V5* [3] algorithm. Interactions with the competition space are handled by three subsystems: the claw, the torpedo, and the marker dropper, each with a mechanical, electrical, and software component. The team can rely on these actions; therefore, more competition tasks can be completed, and more points achieved overall.

C. High Level Behaviors

Building on the low-level actions, the team develops high-level behaviors that govern the autonomy of the vehicle. To manage these actions and behaviors, the team uses a software package called *BehaviorTree CPP V3* [1]. Fig. 2 shows UWRT's custom behavior tree. For the *Choose Your Side* task, the high-level behavior utilizes SLAM and the move action to locate the task objects and navigate to their location. Tempest has been instructed to proceed with the G-man side utilizing its perception solution. Choosing G-man allowed the development of a more sensitive vision model leveraging the high contrast of the G-man figure. Tempest finishes the *Choose Your Side* task by moving under the gate with the move action.

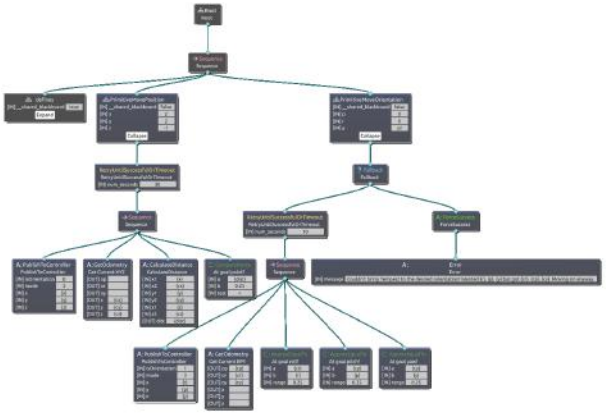

Fig. 2: Behavior Tree

For the *Make The Grade* task, Tempest takes a similar approach to *Choose Your Side*. The vehicle first navigates to the location of the task. Once arrived, the vehicle makes additional observations to determine a more accurate location for G-man. This position is then used as the location needed for the vehicle to touch the task object. The task is then completed by touching the badge with the move action.

After completing the previous behaviors, Tempest progresses to *Collecting*. The behavior utilizes the same actions as the previous behaviors to navigate to and align with the task. Once the task is reached, a fine alignment action is used to position Tempest in a location ready to grab the handle with a custom designed claw. Once set, the interact action is used to move the lid and the move action used to move Tempest to a position where it can drop the markers into the bin. As a final step, the lid is released to fall beside the bin.

Survive The Shootout is completed in a similar manner to the previous behaviors; however, a fine alignment is made with the holes in the buoys and the interact action fires the torpedo through the holes.

Given time constraints faced by the team, the decision was made to neglect the bottles at the *Cash or Smash* task and focus efforts on surfacing. Surfacing is built on the same actions as previous behaviors where Tempest navigates to and aligns with the octagon. Tempest is then disabled and utilizes the vehicle's positive buoyancy to float to the surface within the octagon, ending the run.

The team believes the goals of increasing points earned and decreasing points lost was met. By introducing a reliable building-block-based task execution system and building a robot that is 16 pounds (35.27 kg) lighter than its previous vehicle, UWRT estimates that it can increase its competition point gains to 6,980 in the 2022 competition.

II. DESIGN CREATIVITY

Ensuring that the system level design of Tempest is reliable and robust has been a requirement from the start. With every decision, the team evaluated the alternatives and sought the solutions that would offer the greatest advantages in longevity and reliability. One example of this is the migration of the team's codebase to *ROS2*, which affords the team easier and more reliable robot operation. This change ensures the team is not forced to re-build Tempest's core software with a new framework. Another example is the development of a power management system that can handle more than the team's current needs to ensure enhanced capabilities. A final example is the development of modular mechanical designs to ease assembly and adaptability for future advancements.

To create a better robot, the software team changed system framework from *ROS1* to *ROS2*. This change was driven by two major issues: the announcement of the end of development for *ROS1* by the *Open Source Robotics Foundation* (OSRF) and poor data handling on lossy networks. Poor data handling was caused by a reliance on *Transmission Control Protocol* (TCP) for communicating all data between systems which caused unreliable robot operation when connectivity was poor. To solve this problem, *ROS2* uses a *Data Delivery Service* (DDS) [5] that allows for software level prioritization of critical information across the ROS network while still providing improved reliability of communication for certain data streams using the *Universal Datagram Protocol* (UDP). Tempest's node graph under *ROS2* can be seen in Fig. 11 of Appendix C.

The change to *ROS2* afforded the team the use of *Micro-ROS*, a miniaturized version of the *ROS2* communication framework. *Micro-ROS* allows microcontrollers to join the ROS network directly and publish or subscribe to data streams. This addition meant that the Navionics team no longer needed to maintain a custom communication link for Tempest's electronic control system. The current atmosphere of global part inventory prompted a hardware change in the co-processor circuit. The co-processor circuit transitioned from an STM32 microcontroller to the RP2040 microcontroller. The usage of this microcontroller in a custom board is novel to RoboSub and provided the team with a longer-lasting design due to its ease of implementation, and ability to integrate more easily with the new software system. To verify the new design, the team built a prototype board centered around the implementation of the RP2040 shown in in Fig. 3. This board proved to be reliable and has become a drop-in microcontroller for the co-processor as well as the task mechanism circuits. This permits the team to save time by re-using the fully validated microcontroller hardware, and focus efforts on additional circuits.

The Navionics team worked to integrate a new power management system focused on reliable balancing, distribution, and voltage regulation, shown in Fig. 4. The team uses two 5 cell 8,000mAh LiPo batteries to fully power Tempest for a duration of three hours. Tempest uses a diode ORing controller that switches a set of P-Channel MOSFETs (PMOS) to balance the load across the batteries. Puddles uses an N-Channel MOSFET (NMOS) ORing controller for balancing which had

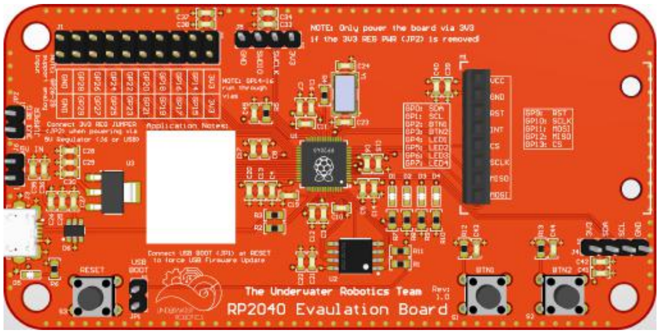

Fig. 3: Co-processor Evaluation Board

a small leakage current through the body diode, a design which failed to balance the batteries as accurately as desired. Tempest's PMOS based system has a smaller footprint due to a decrease in the Rds(on) characteristic of the MOSFET allowing more real-estate to be dedicated to other features. Additionally, the PMOS body diode does not leak current, and the team can more accurately balance the batteries. For distribution, the team uses high-side switch circuits to control all aspects of the robot from the command of the co-processor. For voltage regulation, the team uses switching regulator modules from *TDK lambda* to develop the 12V and 5V needed throughout the robot. Additional low dropout regulators are used to step down the 5V to 3.3V for logic and control circuits. The switching regulators can support 250W for current and future power needs.

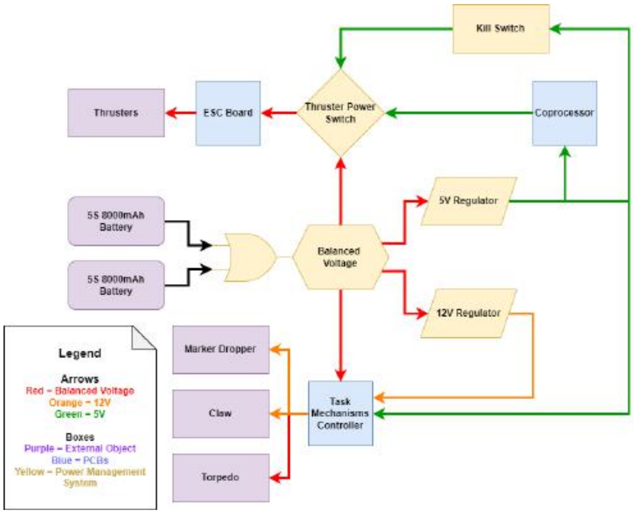

Fig. 4: UWRT Power management System

To better facilitate accessibility of the new electrical system, the mechanical team decided that Tempest's main hull would be a dual cylindrical housing shown in Fig. 5. A dual housing allows enough room on both lids for all subsea connectors (*SubConns*) required to run Tempest. In Puddles, *SubConns* were in the center of the robot making internal wiring troublesome, as connections were difficult to reach and required partial deconstruction of the electronics housing. In Tempest, this process is made simpler and tool-less requiring only taking off a lid and unplugging easy-to-reach connections. Tempest's electronics are mounted by two internal housings, or cages. The boards cage houses all custom electronics while the camera cage hosts the computer, camera, Inertial Measurement Unit (IMU), and a network switch. Separation of the cages into these distinct parts makes Tempest better organized and easier to service.

Fig. 5: Tempest Dual Cylindrical Housing

The dual housing was also chosen to satisfy the design constraint of being symmetrical. This requirement helps to ensure a minimal offset between the Center of Mass (CoM) and Center of Buoyancy (CoB), which aids in maneuverability. A small offset between the CoM and CoB requires less thrust to counteract the resultant torque, allowing the thrust to instead be utilized for maneuvers. In addition, the space between the housings allowed the *Doppler Velocity Logger* (DVL) to be centralized and aided the constraint of the CoM and CoB. Additionally, the wide footprint of the dual housing enabled the thrusters to be placed on the furthest edge of the robot, which aids in maneuverability by giving the thrusters a large moment arm. The centralized CoM and CoB as well as the position of the thrusters enable Tempest to reach poses and speeds unobtainable by Puddles.

III. EXPERIMENTAL RESULTS

In 2021 the team focused heavily on the design of Tempest, which allowed this year to be focused on its manufacturing and testing. To decide the testing priorities, the team made a timeline, shown in Fig. 6, detailing the outstanding tasks and what needed to be done to produce a competition-ready robot. Fig. 6 gave the team approximate testing time needed in the pool to fully validate the subsystems.

The team sought to overestimate the time needed in the pool and booked 40 hours in the water this year. However, due to significant system level changes the team underestimated the time needed. In addition to the 40 hours, the team put in significant time and effort to test and validate critical functions of the robot. The systems tested were

- Power Board
- Torpedo and Coilgun
- Marker Dropper
- Claw
- Software Codebase

The first iteration of the power distribution board suffered from brown-outs whenever the thrusters were engaged. To

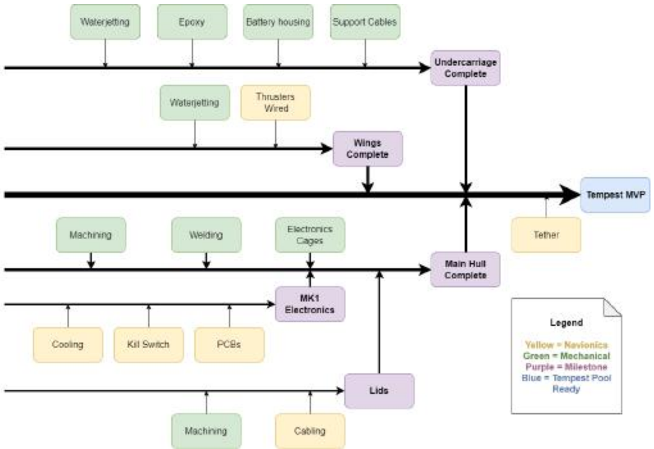

Fig. 6: Critical Path Timeline

seek a solution the team conducted load tests to determine the conditions that result in a brown-out. The team found that the engagement of the thrusters created common mode noise on the ground plane that disabled the voltage regulators. Insufficient bulk capacitance was thought to be the problem and further tests were done to determine the effect of different capacitance on the power rails. The addition of capacitors failed to solve the problem. Next, the team looked at separating the ground planes of the power regulators and the thrusters. This solution solved the brown-out, though required splitting the design into two boards temporarily. The team has devised a more permanent solution through the addition of common-mode chokes and separation of ground planes.

Next, the maximum current the power management system could deliver to the thrusters was investigated. The team used the modular wings from Tempest and submerged them into a dunk tank to test the load. The original power management system failed to support the maximum intended current of 64A, instead only delivering 16A. A second revision of the board was produced featuring larger MOSFETs, but was only able to support 32A. Further investigation showed an error in the schematic that placed the source and the drain incorrectly. The result of this was that the source was on the low side of the MOSFET placing it in the linear operating region instead of the desired saturation region. This fault, along with the backwards body diode limited the power distribution board's ability to deliver current to the thrusters. Correcting this error allowed it to deliver the desired 64A.

The team tested the torpedo interaction mechanism through the use of CAD software, and through bench testing. The torpedo's underwent *Computational Fluid Dynamics* (CFD) to determine the ideal geometry. The goal was to determine the design that would best obtain accuracy and travel long distances when fired by the launcher. Setting up the flow simulations required the consideration of the buoyant force along with the friction drag generated by the torpedo. A range of nose angles from 15 to 30 degrees and various geometric profiles were tested to find the design minimized the impact of these forces. The data gathered displayed that the double tapered torpedo with rifling along its body decreased the friction drag force and the turbulent flow as well as the buoyant force generated by the water. Specifically, the flow simulation displayed a constant flow of water distributed along the body of the torpedo with a stagnant velocity and evenly distributed pressure. Due to the results from the CFD simulations the double tapered torpedo was selected.

Profiles of the electrical characteristics of the coilgun that propel torpedoes were developed. The team profiled the coils by subjecting each one with a fixed AC source and measured the current. Each coil has a unique impedance that the team was able to calculate from the experiments. The average impedance was 250mH and the team was able to simulate the sequential impulse load in *LTspice*. The team used the simulation to determine the specifications needed for the high side switch circuits used to control the coils. The team followed the simulation with bench testing to validate that the MOSFETs chosen were adequately specified. The coilgun successfully fired the torpedoes on the bench and a more permanent design of the actuator control board was produced.

Tests of the marker dropper system to determine the voltage at which the markers are reliably dropped were investigated. Historically the marker droppers failed in their ability to consistently drop the marker on command. Incremental tests at increasing voltages identified that at least 5.5V are required to reliably drop the markers—0.5V higher than the previous design. Voltage applied to the markers droppers was increased from 5V to 12V to ensure reliable dropping. Additionally, changes in the team's task mechanism placement put the coilgun and marker droppers in close proximity. Because both devices operate using magnetic fields, the team chose to test the effectiveness of each device on the other to ensure no interaction between the two mechanisms occur. The tests were successful and concluded that proximity is no concern.

Tests were conducted on the claw to ensure proper function. Initially, the claw housing was submerged for 10 minutes to confirm it was waterproof. After confirming there were no leaks, the motor and magnetic torque coupler were installed, and tests found that the magnets were powerful enough to break the 3D printed layers of the torque coupler. To fix this, a stronger torque coupler was machined from aluminum. Next, the team connected the motor to a 12V power supply and incrementally adjusted the distance between the two magnetic disks until they could slip past each other to prevent the motor from stalling. After this adjustment, the claw was submerged, and was used to grab the PVC handle.

The software team used a simulator to test proper interaction of Tempest's software. The controller and its calibrations were reviewed to ensure that they gave reasonable outputs. This affirmation was valuable after the software team's switch to *ROS2*, as much of the codebase was re-written and had not yet been validated. When Tempest's software systems were completed in the simulator, the team was able to use it to test Tempest's behavior logic for basic tasks, specialized actions for more advanced tasks, and how each task performed in a pool environment.

Once the team was confident with the system level tests and software integration had been performed, Tempest was put in the water. During the spring testing campaign, the software team focused on tuning and optimization of the core components of the control system: state estimation, controller, kinematic solver, and system level validation. The first four pool tests were dedicated to setting up state estimation and initial controller calibration for Tempest until the vehicle could submerge. These tests were spent fixing bugs in components of the software pipeline that could not be tested in simulation, such as hardware configuration and firmware. The next five pool tests were used to continue validation of high-level control software including mapping and vision systems while performing significantly more rigorous tuning on Tempest's controller with the aid of the divers. The team developed a specific approach to allow the divers to act as a disturbance in the control loop, and the parameters were tuned around rejecting these transients. The last pool test of the spring campaign was dedicated to testing the low-level behaviors that Tempest can perform as part of the larger behavior tree system.

The testing discussed scratches the surface of the testing the team has done over the last year. The team plans to continue testing during the summer using more pool tests and simulations.

IV. ACKNOWLEDGEMENTS

The Ohio State University's Underwater Robotics Team is located within *Dreese Laboratories* under Ohio State's Department of Electrical Engineering. UWRT would like to sincerely thank the *Dreese Laboratory* and *Department of Electrical Engineering* staff for their contributions and help provided. Thank you to Dr. Serrani, Bill Thalgott, and Justin Ellis for helping to get the team set up and situated with the recent move to *Dreese Laboratories*. The team would also like to gratefully thank the efforts of Dr. Saeedeh Ziaeefard, UWRT's advisor and source of guidance for the improvements, innovations, and achievements year after year.

The creation of this year's robot, Tempest, and the longevity of the team's outreach and educational opportunities would not be possible without the dedicated support of UWRT's sponsors. UWRT would like to thank both its long-term and newly partnered sponsors, such as *Aptiv*, and *Advanced Power Drives*. Lastly, the team would like to thank *RoboNation* for their dedication in maintaining and orchestrating *RoboSub*, allowing teams of engineers across the world to participate in one of a kind robotics experiences.

Thank you to the UWRT members who volunteered their time and effort during the semester. The team recognizes that substantial contributions to the vehicle are difficult to balance with schoolwork and extends a thank you to the members who ensured that Tempest is competition ready. Additionally, the team would like to thank Nathan Ayer, Phillip Barker, Nathan Becker, Matthew Fisher, Brach Knutson, Robert Pafford, Isabella Richardson, Mitchell Sayre, Alex Schuler, Dylan Trainor and Cole Tucker for dedicating time in the summer to push Tempest over the finish line.

V. REFERENCES

[1] Michele Colledanchise and Petter Ögren. Behavior trees in robotics and AI: an introduction. *CoRR*, abs/1709.00084, 2017.

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

[3] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, TaoXie, Jiacong Fang, imyhxy, Kalen Michael, Lorna, Abhiram V, Diego Montes, Jebastin Nadar, Laughing, tkianai, yxNONG, Piotr Skalski, Zhiqiang Wang, Adam Hogan, Cristi Fati, Lorenzo Mammana, AlexWang1900, Deep Patel, Ding Yiwei, Felix You, Jan Hajek, Laurentiu Diaconu, and Mai Thanh Minh. ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference. February 2022.

[4] Nathan Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2149–2154, Sendai, Japan, Sep 2004.

[5] Steven Macenski, Tully Foote, Brian Gerkey, Chris Lalancette, and William Woodall. Robot operating system 2: Design, architecture, and uses in the wild. *Science Robotics*, 7(66):eabm6074, 2022.

[6] Musa Morena Marcusso Manhães, Sebastian A. Scherer, Martin Voss, Luiz Ricardo Douat, and Thomas Rauschenbach. UUV simulator: A gazebo-based package for underwater intervention and multi-robot simulation. In *OCEANS 2016 MTS/IEEE Monterey*. IEEE, Sep 2016.

TABLE I: Tempest's Component Specifications

Component	Vendor	Model/Type	Specs/QTY	Cost (if new)	Status
Buoyancy Control	Not Present				
Chassis	Custom	Custom	30” long x 3’ wide x 14” tall	$200	Installed
Camera-Side Waterproof Housing	Custom	Custom	1’ long x 8” dia.	$2,500	Installed
Board-Side Waterproof Housing			1’ long x 8” dia.	$2,500	Installed
Subsea Connectors	MacArtney	Micro Circular	N/A	$3,000	Purchased
Thrusters	Blue Robotics	T200	8x, 3-20V, 25A	Re-used	Purchased
Motor Control	Blue Robotics	Basic ESC	8x, 7-26V	$25 Each	
Propellors	Used T200 propellors				
Camera Cage	Custom	Custom	8.5” long x 6.75” wide x 6.25” tall	$85	Installed
Board Cage	Custom	Custom	9.0” long x 6.0” wide x 6.25” tall	$85	Installed
Battery Housings	Custom	Custom	9.0” long x 5.0” wide x 4.50” tall	$110	Under Construction
Claw Manipulator	Custom	Magnetic Coupler	7.5” long x 4.75” wide x 2.75” tall	$100	Under Construction
Torpedo Launcher	Custom	Electromagnets and Coils	1’ long x 3” wide x .5” tall	$100	Under Construction
Marker Dropper	Custom	Electromagnets	3.5” long x 1” dia.	Re-used	Installed
Kill Switch	McMaster-Carr	Magnetic Switch	1.5” long x 0.25” wide x 0.37” tall	$6	Purchased
Cooling Fans	NMB Tech Corporation	08015SS-12N-AL-00	80mm long x 80mm wide x 15mm tall	$14	Purchased
Peltier Panel					
High Level Control	BehaviorTree	N/A	N/A	N/A	
Battery	MaxAmps	Lithium Polymer	2x, 5S, 18.5V, 150C	Re-used	
Converter	TDK-Lambda	I6A4W(250W)	2x, 5V, 12V DC/DC Converter	$35 Each	
CPU/GPU	NVidia	Jetson Xavier	8-core ARM v8.2 64-bit CPU	$999	Purchased
Internal Comm Network	I2C				
External Comm Interface	Ethernet				
Programming Language 1	Python				
Programming Language 2	C++				
Compass	In IMU				
Inertial Measurement Unit (IMU)	LORD MicroStrain	3DM-GX4-25	1x	Re-used	
Doppler Velocity Log (DVL)	Nortek	DVL1000	1x	Re-used	
Camera(s)	Zed	2i	1x	$239	Installed
Hydrophones	Aquarian Audio	AS-1	3x 0.47”	$395	Not Purchased
Algorithms: Vision	YOLO				
Algorithms: Acoustics	Phase Difference	Custom			
Algorithms: Localization and Mapping	”Conceptual” SLAM				
Algorithms: Autonomy	BehaviorTree				
Open source software	ROS and OpenCV				
Team size	61				
HW/SW expertise ratio	8/3.				
Testing time: simulation	200 Hours				
Testing time: in-water	32 Hours				

VI. Appendix A: Component Specifications

VII. APPENDIX B: OUTREACH ACTIVITIES

UWRT's STEM initiative and goal of teaching others about underwater robotics extends from Ohio State's campus to the surrounding Columbus area. The team engages the local community by attending annual events such as the Ohio State Fair and MakerX (The Columbus Maker Expo). At both events UWRT helps host exhibits to educate the local community about ocean engineering, the importance of underwater vehicles, and the positive impact of STEM education.

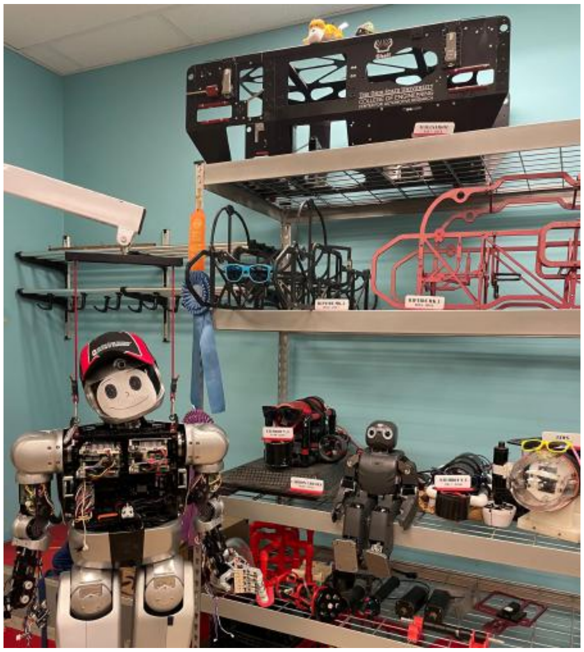

Fig. 7: UWRT's expanding robot family.

In the beginning of the year 2022, the team ran the pilot program of STEMBot, a 5-week after-school program at Rosemore Middle School. Each day UWRT members taught science and engineering principles to students. Students had the opportunity to assemble a STEMBot both in TinkerCAD, a kid-friendly CAD software, as well as it's physical form both electrically and mechanically. Students also learned the basics of programming and were able to compete their robots against each other in a mini-obstacle pool. All the students enjoyed the after-school program and constantly asked if UWRT would be returning the following year. The team was able to call the program a resounding success! Currently the team is taking feedback from the students and STEMbot volunteers to improve the program for the future.

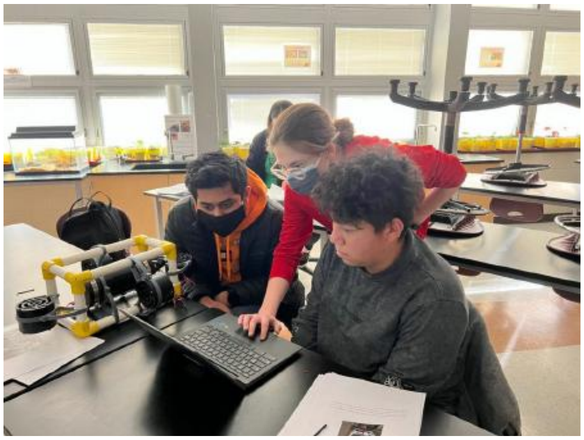

Fig. 8: Team member explaining TinkerCad to a middle student.

Fig. 9: Middle school students surround a pool while one drives a STEMbot through an underwater obstacle course.

VIII. Appendix C: Figures

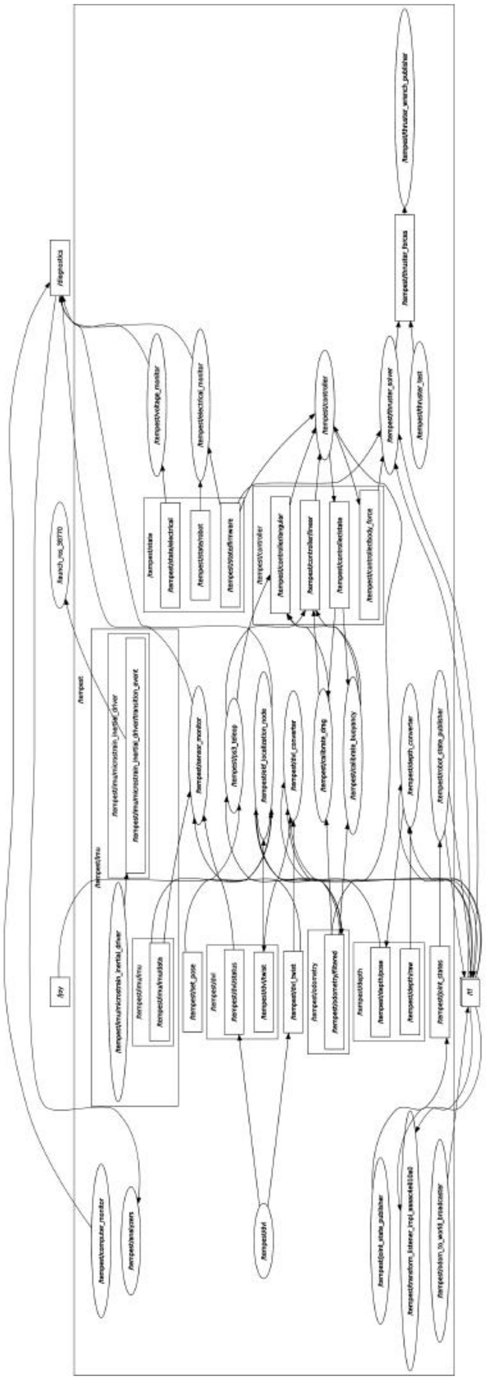

Fig. 10: A sanitized version of Tempest's node graph while in operation.

Fig. 11: Tempest's combined torpedo and maker dropper system featuring a Coil Gun design.