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

Anden Acitelli, Collin Barack, Phillip Barker, Arko Chatterjee, Amber Dellacqua, Matthew Fisher, Hayden Gray, Parth Parekh, Isabella Richardson, Mitchell Sayre, Dylan Trainor

***Abstract*—Since 2016, The Underwater Robotics Team (UWRT) at The Ohio State University has iterated on the foundations of a single Autonomous Underwater Vehicle (AUV) each year to compete at the RoboSub competition. Breaking from tradition, the team decided to take the 2019-2021 school years to design and build a new vehicle to compete in the 2021 competition. Featuring an entirely new hull design, refactored software, and an improved electrical system, UWRT has created its brand-new vehicle, Tempest.**

I. Competition Strategy

FOR the 2021 RoboSub competition, The Underwater Robotics Team at Ohio State University decided to approach the competition and its course with a new robot, Tempest. This robot created opportunities for the team to improve upon previously implemented designs, while exploring new concepts based on the team's past experiences. The team focused on creating a robot that was easy to repair, adaptable to various situations, and user-friendly for all team members. Tempest is designed to be able to complete each competition task and the team has built backups of each system for quick replacement should a part fail during competition.

A. Mechanical

One of the mechanical team's goals for the 2021 competition was to manufacture and design a robot that would weigh less than 60 lb (27 kg) to prevent a point reduction at competition. This material was chosen in place of aluminum because it is lighter in weight while remaining highly machinable. Second the frame was reduced in size and complexity. Rather than over-engineering the robot, as was done in the past, care was taken to properly engineer the chassis to meet design constraints and safety standards.

Another high-priority goal for Tempest was to make the electrical systems easy to access. Previously the electrical team faced difficulties reaching the electronics due to a laborious disassembly process and unideal cable management. Over time, wires would tangle around the hardware, making it difficult to locate components that needed to be repaired or replaced. To increase accessibility of the robot, the mechanical and electrical teams designed Tempest's dual housings to focus on keeping the electronics organized. Cages were designed to prioritize cable management while compartmentalizing the electrical boards and computers. Both cages are attached to drawer slides using sliding dovetails, allowing for toolless removal of the electrical system.

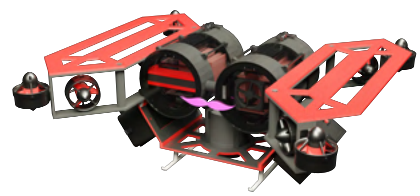

Fig. 1: CAD rendering of Tempest.

B. Electrical

The electrical team's goal in designing the system for Tempest was to increase the reliability of existing hardware. The electrical team divided this task into four primary circuit elements: computer interface, power management, thruster control, and task mechanism control. These subsystems are necessary for the computer to maintain control over the robot's motion while completing tasks. The most notable change in the circuitry is the addition of redundant features such as transient suppression and doubling key current carrying capabilities were implemented to prevent total system failure during operation.

C. Software

The software team sought to improve on competition success by remedying two specific issues: an overreliance on pool time for testing, and inconsistent task performance. The development process was formalized to address both of these issues. Before being flashed to the robot, new code was first required to run successfully in the simulator and function well remotely during pool tests. This rigorous testing made pool tests more efficient.

In order to improve the other main issue, unreliable task performance, the team refactored the code through a state machine structure for decomposition to made it easier to debug. These technical system alterations improved the perception, mapping, and task code of Tempest.

II. Vehicle Design

A. Mechanical

1) Housing: A leading requirement in the design of Tempest's hull was to reduce the number of steps to access any component to two. This requirement arose in response to the team's previous vehicle, where the internal components were difficult to reach and maintain. To address this issue, Tempest's watertight housing was designed to make use of two short cylinders, contrary to Puddles's longer single housing. This double-housing system allowed the team to better compartmentalize and organize the internal electronics into two smaller systems that can be easily accessed.

Each side of the housing is comprised of an aluminum structure on the fore and aft with a central polycarbonate section. The aluminum-polycarbonate interfaces are sealed with gaskets kept in compression by stainless steel tension rods. Puddles's housing was known to leak at a similar interface in cold water due to thermal contraction in its gaskets. To rectify this Tempest combined the tension rods with a spring screw system to allow the sealing to dynamically adjust to changing dimensions. The gasket is also made of a more thermally resistant material to further minimize chance of failure.

In between the two cylinders are a pair of hollow tubes that join each side of the housing and allow wires to be passed between the compartmentalized electronics. Both of the electronic subsystems can be viewed through polycarbonate panels on the front faces of the housing. These transparent faces are needed for the camera and other sensors to monitor the environment. The internal electronics are accessible via two lids on the aft of the vehicle. Because these subsystems are mounted on drawer slides, they can be easily withdrawn for maintenance. Outside the hull, on the aluminum structures at the fore and aft of the vehicle, are mounting points for Tempest's chassis.

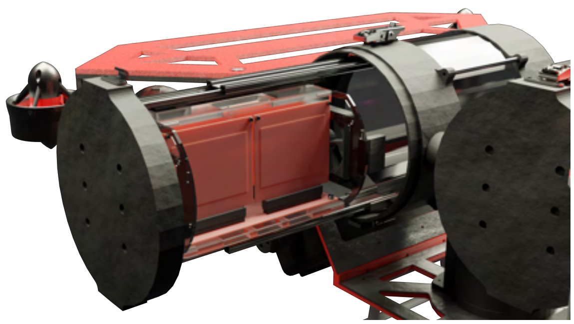

Fig. 2: Expanded housing with electronics cages

2) Electronics Cages: The double housing of Tempest allows the internal electronics to be organized into two compartmentalized systems: the Camera Cage and the Board Cage. This reduces the steps required to access the electronic components and aids in cable management. To further reduce the steps to access, the electronics cages are attached to drawer slides using sliding dovetails, enabling the cages to be removed without tools.

The Camera Cage situated on the starboard housing of Tempest contains the robot's computers and internal sensors. To ensure a stable fit, the camera is supported by two 3D printed mounts. A rubber dampening pad is placed in between the camera mounts and camera cage to reduce vibration caused by the fan, thrusters, and other electrical components. The remaining components, the Graphics Processing Unit (GPU), Inertial Measurement Unit (IMU), network switch, and breakout board are fastened onto an electronics mount that was designed to be easy to replace. The breakout board is a specially designed board to accommodate the double housing design. It limits the number of cords that need to be run in between the housings to enable the two sides of Tempest to communicate. The components in the camera cage produce substantial heat; which is reduced to optimal temperatures by the cooling system. It utilizes a Peltier panel, heat sink, and fan attached to the hull. Three subsea connectors run through the lid to the Doppler Velocity Logger (DVL), acoustics housing, and tether.

The less complicated sister of the Camera Cage is the Board Cage. Situated in the port housing, this cage holds the team's custom designed Printed Circuit Boards (PCBs). The board cage takes the signals from the computers in the camera cage and distributes them through to the rest of the robot. The four custom PCBs are attached to the backplane through edge connectors and fixed to the top bar of the cage with screws. Because the components in this cage do not produce much heat, only a fan is needed for cooling. On the front side of the cage sits the kill switch, which is made from a magnetic contact alarm switch so it can be triggered externally. The electrical portion of the switch sits inside the cylinder while the magnet is attached on the outside.

3) Wings: The chassis of Tempest is and open frame design comprised of two sets of stacked trapezoidal wings on either side forming a cross pattern. This allows for the four horizontal thrusters to be equally placed far from the center of mass. The thrusters are in a vectored configuration at 45° angles enabling each to be used for horizontal movement, thus giving greater control over Tempest's motion. Mounted on the ends of the wings are the four heave thrusters for vertical translations, pitch, and roll. The chassis is primarily constructed out of HDPE; this material was chosen to both reduce weight and cost while maintaining enough strength to support the rest of the robot.

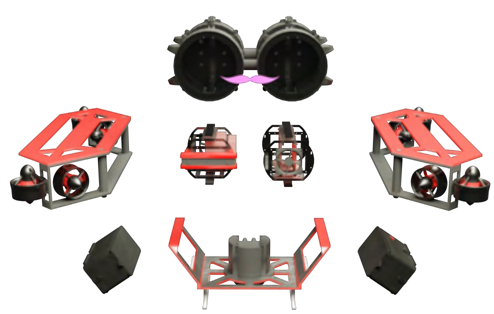

Fig. 3: Exploded view of Tempest

Another main focus of this design was to prioritize simplicity and modularity. This is achieved by designing the chassis to minimize the number of screws used and reduce the number of tools necessary to construct the robot. For example, both wings can be removed by taking out four screws and, when removed, come away as a single unit. This allows the robot to be easily broken down and flat packed to reduce shipping costs.

4) Undercarriage: The undercarriage of the robot serves as a mounting point for the battery housings and DVL. To ensure stability, the undercarriage was designed symmetrically, with the DVL in the center and the batteries aligned with the center of the vehicle along the x-axis. The side walls of the undercarriage are slanted so that the batteries are easier to access and to give the robot a smaller footprint. To prevent bowing from the weight of the vehicle, 2 mm (0.08 in) aluminum plates were attached to the inner frame. The lamination prevents warping while allowing the total weight to remain less than that of a solid aluminum structure. Beneath the Undercarriage there are a set of sled runners that act both as protection for the DVL and as a support for the rest of the vehicle when not in use.

5) Battery Housings: The battery housings on Tempest are based on the design of Puddles, but with slight alterations. The previous system was difficult to use because of how involved the process was to access the internals. The eight screws and flat gasket of Puddles were replaced with a captured gasket and four latches, removing the need for tools to open the housings. The lid of the housing has an outer ridge to make it easier to overcome the pressure difference inside the housing when opening it to remove the batteries. To further increase ease of access, the housings slide into position on rails, and are fixed by a single bolt.

6) Attachment Panels: One of the broader goals of Tempest was to allow for the robot to be used in a variety of situations outside of the competition. To accomplish this Tempest uses standardized attachment panels placed upon the wings on each side of the robot. This allows task mechanisms to be easily attached to any location on the wings, making Tempest well-suited for a diverse array of environments and tasks. These panels are designed to be easily removed from the wings and exchanged for ones with different tool packages to allow for greater modularity of the robot.

7) Torpedoes: In previous years, UWRT has worked to develop a torpedo launcher with an electromagnetic firing mechanism. Through a series of four tightly wound coils of magnet wire, the launcher propels a solid carbon-steel bullet forward by creating a strong magnetic field that drives the torpedo through a barrel incrementally. Using last year's prototype launcher, the team wanted to innovate and produce a launcher with more accurate control and predictable firing strength. Based on the dimensions of the polycarbonate barrel the team created a formula in MATLAB to determine optimal coil length. To enhance the hydrodynamic profile of the torpedoes, multiple versions were drawn and simulated in SolidWorks. This ensured that the most dynamic silhouette was selected for use at competition. To enhance the effectiveness of the coils, vents were added along the length of the barrel to allow water to be pushed out of the way of the projectile.

8) Claw: For the 2021 competition, the team made the decision to incorporate a custom-designed manipulator for Tempest in place of a BlueRobotics Newton Subsea Gripper from last year. The claw was designed with a magnetic torque coupler, a disk embedded radially with six magnets, that was attached to a DC motor contained within the housing of an arm. The arm is paired with a similar disk mounted on the external face of the housing. Due to the magnetic attraction between the two disks, rotational motion can be imparted across the sealed surface. This external disk is attached to a worm gear which is used to drive the rotation of the two jaws of the claw and gives Tempest the ability to manipulate task devices. This was done to eliminate the need for active sealing reducing potential failure points in the system.

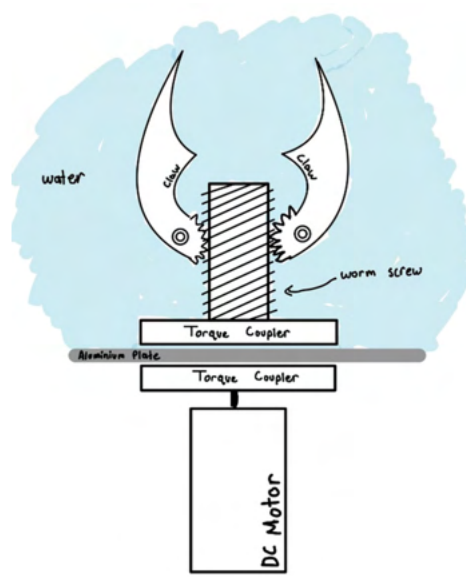

Fig. 4: Concept design for the claw

9) Thruster Shrouds: Optimizing the health and safety of team members and other personnel interacting with Tempest was a goal for the team this year. To reduce the possible interference of external factors such as wildlife and human extremities, shrouds were designed to protect the propeller of the T200 thruster. The open slots on the shrouds allow water to pass freely while preventing foreign objects from hitting the propellers. 3D printed in ABS plastic, this two-component design rests on each face of the thruster and connects to the motor's supports via four clips and its nose cone.

Fig. 5: Thruster safety shrouds

10) Lighting: The team designed an external lighting system which utilized a custom-built LED ring light. The light was attached into the front port side acrylic panel of the main housing to prevent glare on the camera in the starboard housing. The additional lighting allows for Tempest to operate in dark or murky waters and improves functioning at competition.

B. Electrical

Tempest's electronic system contains five custom designed PCB's: the actuator board, the power distribution board, the coprocessor board, the electronic speed control (ESC) board, and backplane board, each handling a different aspect of the robot. The Actuator board serves as the controller for each of Tempest's competition task mechanisms. The Power Distribution board translates, monitors, and distributes power from the batteries to the rest of the robot. The Coprocessor board (CoPro) is responsible for connecting the electronics hardware with the software that controls the robot. The ESC board is a central connection point for all thruster control. The Backplane board is used for mounting Tempest's four primary PCB's and passes signals between them.

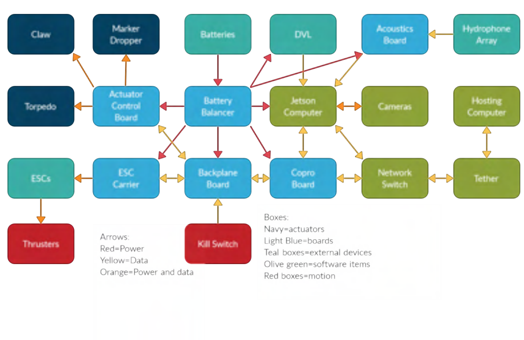

Fig. 6: Electrical System Map

1) Actuator Board: For the 2021 competition season the team updated the actuator board to better handle the load requirements of the new torpedo system by adding additional current carrying transistors. In addition, the team used transient suppression techniques for inductive loads by adding diodes and a varistor. These parts control the back emf from the torpedo coils and ensures the firing circuit remains reliable. The team also added control circuitry for the new claw mechanism while maintaining old circuitry for the previously used manipulator. This allows for the use both the primary and backup claws.

2) Power Distribution Board: The Power Distribution board's primary function is to balance the port and starboard battery inputs and convert them to 3.3V, 5V, and 12V for use with the other PCB's. Additionally, the power distribution board acts as the centralized connection point for all power within the robot. The board records power usage information and transmits it to the CoPro for system monitoring. To build a more robust system the team determined the exact power specifications for each component rather than over-engineering. This reduced space usage, allowing for better organization of traces and the addition of fault lights for troubleshooting.

3) Coprocessor Board: The coprocessor is the middleman between the hardware and the software within the robot. It provides critical diagnostic information and interprets commands from the computer to PWM signals for the thrusters. The team overhauled the firmware for the CoPro board by adding diagnostics reporting and implementing better fault detection and error handling, in turn increasing the reliability of the robot. The coprocessor now reports critical information back to an operator for live monitoring during pool tests and dry runs. The new reporting features of the CoPro include, internal temperature, battery consumption rates, and more.

4) ESC Board: The ESC board takes in power from the power distribution board and PWM data from the CoPro which allows the ESC modules to output the signal to the thrusters. In addition, the ESC board features eight current sensors, and a new fault indicator circuit. To improve on mounting the ESC's, the team used snap lock connectors in place of the prior screw terminal mounting system. This allows the ESC modules to be mounted externally and snapped into the board when ready. Whereas, in the past, the ESC's were difficult to place correctly due to little clearance between the screw terminals and the short wires from the modules.

5) Acoustics: The acoustics system allows for the robot to triangulate its position in the test pool at competition. This is done by using 3 hydrophones on the bottom of the housing that are spaced 1.6 cm (0.63 in) apart. This year the team opted to increase the reliability and improve the design of the acoustic system by changing the central algorithm that previous designs have been based on. To achieve this, a complete redesign was necessary. In the past, the team found the distance from the pinging device using time difference. This year the team chose to use phase difference and develop a more robust system. The phase difference approach provides the team with some constraints, namely size limitations and a need for faster and more accurate signal processing. However, it offers benefits in the reliability and accuracy when determining location. Since the team was working toward implementing an entirely new system, the choice was made to rely on a DAQ to pass data from the hydrophones to the computer. This decision allowed the team to focus on building the detection circuitry and the algorithm to use the data. To incorporate the changes a new housing was built, it is comprised of three separate parts: the housing case, the lid, and a scaffold that allows the electronics to be easily removed and serviced.

6) Team Engagement: This year the team decided to spread some joy during the pandemic by promoting art in engineering. The team had members submit artwork to be printed onto Tempest's PCB's. For a look at the art that made it into the robot visit section VIII figure 15.

C. Software

1) Perception: In order to navigate through the course while moving and interacting with objects, the robot must successfully perceive its surroundings. This year, the team implemented a major system alteration which enhanced how the robot perceives objects in the environment. The previous version of the perception system utilized a YOLO Darknet,

Fig. 7: Exploded View of the Acoustics Housings

which detects objects and outputs an estimated x and y coordinate of where the object is located in the camera view. While effective the team wanted a system with more usable data. The new perception system uses machine learning models trained for detection and estimation of the position and orientation of objects relative to Tempest. These models were trained in a version of the ROS package Deep Object Pose Estimation (DOPE), which was modified by the team to fit specific needs and computation limitations of Tempest. With this system, Tempest can estimate true pose (x, y and z as well as roll, pitch and yaw) of competition objects.

In the past to train the perception model to accurately estimate an object's position and orientation relative to the camera, the team manually complied a large dataset by labeling each frame of a training video taken by the camera. Because of this labor-intensive process the team developed a more efficient approach. No real images of objects were used, rather, each frame in the data sets was programmatically generated with 3D rendering. Utilizing NVISII, a python-based 3D rendering tool, the object was rendered with random background and lighting conditions. The object's current position and orientation was extracted from the rendering tool and saved with the associated frame. By letting this automated process run, the team generated datasets of 50,000 unique frames in under 2 hours. Data augmentations such as digital noise, varying object orientation, and blur were then applied to the frames in each training iteration. Thus, ensuring that the model could identify the object trained and estimate its pose for a variety of conditions.

2) Mapping: The 3D mapping system uses the object pose estimation data from the perception system. The addition of the mapping system allows the robot to track objects of interest and remember the location and orientation of tasks, allowing for improved competition accuracy. The mapping system is given an initial estimator of the position of tasks, and then utilizes data from Tempest's perception and localization systems to actively update and refine a positional map of objects in the surrounding environment. The mapping system feeds the perception system's data through several filters to check the validity of the data, which is then merged with the current data for a given task using Gaussian probability density functions.

While mapping is common in many robots, Tempest's system is novel in many aspects. Firstly, the mapping system only tracks objects of interest and does not attempt to completely map its environment. Since keeping track of the entire environment would require a greater amount of data, this approach decreases the amount of noise in the system and strain on Tempest's computers. Mapping is a new improvement as previously the vision system fed data directly to the task code. In the robot's old system, decisions were made based on a single frame of vision data, whereas the new system merges data from each frame the object was detected, making a more reliable estimate. This also enables Tempest to carry out tasks without keeping the object of interest in frame.

3) Task Code: For the software team's third technical systems improvement, the team switched from using a custom solution to FlexBe [8], which is a more reliable state machine decision structure for task code. The new system allows team members to easily make adjustments and locate potential errors while prototyping, allowing more efficiency in writing task code. This creates a more modular library of behaviors for the robot which easily incorporates new technologies like local pathing and navigation through MoveIt. [2]

From previous years' experience, the team knows that rapid software development is important to fielding a competitive robot. After analyzing the complexity of a custom solution versus the relative ease of FlexBe, the team decided to use its library. The task code system was switched to utilize state machines which gives the advantage of being able to build the task code as a series of blocks comprising different behaviors — such as movement, searching, and manipulation — that can be placed and grouped together. This change makes it easier design task code from a higher level and promotes reuse of code. When used in conjunction with a graphical interface built through FlexBe, the system is now more user friendly for all members of the team to create and modify tasks as needed. This allows rapid iteration of task code from not only the members of the software sub team, but also members of the electrical and software sub teams.

4) Navigation: Through the more advanced data generated by the perception and mapping systems, the team was able to expand the navigation system. The team decided to use MoveIt, a motion planning library that provides different types of motion planning algorithms for robots. MoveIt was evaluated and chosen by the team because it fit with the goal of quick and modular core. This tool allows the team's robots to use perception and mapping data to locally path around obstacles. Due to this tool, the robot is much more consistent and is now able to find specific parts of tasks to complete. These paths then feed to the control systems.

5) Controls: The control system receives all of its data from the navigation system. It then uses a cascaded P controller to stabilize and execute the instruction it is given. This year, the control system has improved to be much more stable despite dynamic forces. This system is a more reliable design with a modular structure that allows for easy iteration in the future.

The control system was restructured this year to facilitate development. It was made modular so that it can be swapped out with different algorithms at will, thus creating a fuzzy logic with controllers. The base system uses two cascaded P controllers (a P controller outputting to a P controller) to control the linear and angular motion of the robot; while physical effects such as drag and buoyancy are removed mathematically. With buoyancy removed, the P controllers can correct linear position without having to deal with buoyant torque trying to force the vehicle upright. Once a trajectory is sent from the navigation system, the controller continues to feed the desired current state into the controller while the robot advances through the trajectory in real time. The controller will output to a thruster solver which finds an optimal way to turn the thrusters, taking into account power consumption and which thrusters are in or out of the water. Another advantage of the new control system is the absence of gimbal locking. A common problem with controllers is that they could theoretically enter a position aligned with a set of global axes that locks the orientation. The new system handles this case through the use of quaternions.

The new code also has many structural improvements. The previous system made use of seven independent controllers to handle depth, alignment, orientation, and other state variables of the robot. When trying to move the vehicle, each controller needed to be individually addressed, which resulted in difficult to understand code. Now, all movement goes through one main controller and the vehicle can independently hold a linear position while setting an angular velocity. The P controllers keep movement simple while still outputting immediate corrective response. As mentioned before, this controller can be easily swapped out with a different style of algorithm at will –even using two different controllers during the same run- because of how it is modularly written. The team plans to expand to LQR and Linear Gaussian controllers in the future to be able to move around more efficiently compared to P controllers.

III. Experimental Results

Due to Covid-19 the team had restricted access to pools, workshops, and other university facilities. This required the team to utilize out-of-water and virtual testing methods to effectively test the robot.

A. Mechanical

1) Torpedo Testing: The team designed and manufactured a coil gun and torpedoes for the target-based part of the competition. Multiple torpedo designs were tested and simulated in SolidWorks to determine what design could best attain high accuracy and achieve long ranges when fired. When selecting each feature of the projectile, it was crucial to account for the water's buoyant force and skin friction drag. To minimize the impact these forces a range of nose angles, from 15 to 30 degrees, and combinations of profile characteristics were simulated. After analyzing flow simulation data, it was noted that a double tapered end measuring at 20 degrees was the most effective design. The simulation showed that the water flowed across the projectile body with constant velocity, minimizing turbulent flow. The pressure exerted by the water was distributed along with the torpedo evenly, unlike other designs such as a single tapered end at 20 degrees to which the pressure was concentrated on the non-tapered end only. Rifling was added to the body of the torpedo to give the projectile gyroscopic stability.

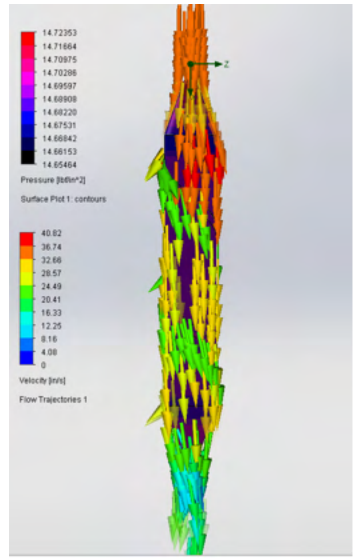

Fig. 8: Flow simulation of torpedoes with rifling.

2) Claw Testing: The team's initial design made use of a permanent magnet and an electromagnet spaced slightly apart. By reversing the current running through the electromagnet, the claw would theoretically be able to switch between attracting and repelling the permanent magnet, allowing the claw to open and close while maintaining a sealed chamber for the electronics. Unfortunately, upon testing the magnets acquired, it was found that the repellent force of the electromagnet would never overcome the attractive force of the permanent magnet to the magnetic frame of the electromagnet and as such, the design had to be scrapped. The team then switched to the current manipulator design previously mentioned.

3) Structural Adhesion Testing: Because the team was making use of non-metal materials, new fastening methods had to be used. With materials that were too soft to reliably hold a thread under stress, the team used a series of epoxies to mate plastic components, particularly on the undercarriage. To ensure the epoxies would hold up under use, the team tested several bonding agents with scrap pieces of HDPE and left them to soak in water for several days. After being removed from the water, the bonds were stressed to their failure point. Critical joints were tested further after assembly by placing known weights equaling more than 1.25x the vehicles total weight on top of the joints.

4) Center of Buoyancy and Mass: To calibrate the control system for Tempest, both the magnitude and location of the center of buoyancy and mass were required. This was accomplished using Archimedes' principle, which states that a body submerged in a fluid at rest is acted upon by a buoyant force with a magnitude equal to the weight of the fluid displaced by the body. To accomplish this, a complete SolidWorks assembly of the robot was replaced with solid bodies made of water. Then, using SolidWork's mass properties tool, the magnitude and location of the displaced force was obtained.

5) Finite Element Analysis: To reduce the weight of Tempest, the team used HDPE in place of aluminum in the chassis. The use of this alternate material decreased the total weight by 25 pounds (11.34 kg). Finite element analysis was used to ensure that the replacement material would hold up to the stresses of normal use. Because of the team's findings, slight modifications were made to the undercarriage, including extra support struts for the DVL and the addition of the 2mm (0.08 in) aluminum lamination.

B. Electrical

The Electrical Team designed, built, and tested each circuit in Multisim. The tests consisted of simulated function tests which allowed the electrical team to view how the boards would fair under different types of loads. After passing these tests, the team was able to move on to production of the electrical system. Due to limited access to the lab and other COVID constraints, the production of these boards has not fully completed yet. Once this is fully completed, the electrical team will move on to simulated load testing. This process is rigorous as each board requires a full system test and checkout. For instance, the power distribution board is pushed to its operating limits to ensure reliability at the most extreme conditions. A full system test and checkout will be performed one board at a time. for instance, the Power distribution board will be pushed to the full extent of operating loads with a pseudo load to ensure reliability across conditions. The ESC board will undergo an endurance test with 8 thrusters to ensure no mishaps. The CoPro will be put through the ringer on error handling and packet loss as well as communication reliability tests. The actuator board will be thrown into a rig to actuate the task mechanisms and ensure consistent and reliable operation

C. Software

1) Perception: Throughout developed of the perception system, footage from the simulator was used extensively, serving as a benchmark for whether changes made in the training process and detection pipeline improved, diminished, or resulted in no change in model performance. When successful detection and pose estimation had been achieved with simulator footage, the team applied extensive data augmentations to each frame in the training process. These data augmentations served to generalize the models, meaning they succeed in any setting with variable visual conditions, including those observed underwater. Once the model succeeded in detection and pose estimation with all of the team's available real-world footage, it was integrated it into the robot's system fully. Observing successful detection, our model had achieved a general fit, meaning it succeeded in many environments with drastically different conditions.

2) Mapping: The mapping system was tested for both reliability and robustness. Most testing was done by changing the system's initial positional estimate for objects. Specifically, the position and certainty of the initial estimate was varied. This enabled the team the chance to see how the robot would react to being very sure or unsure of where an object is. Certain parameters pertaining to the system's estimate filter were also

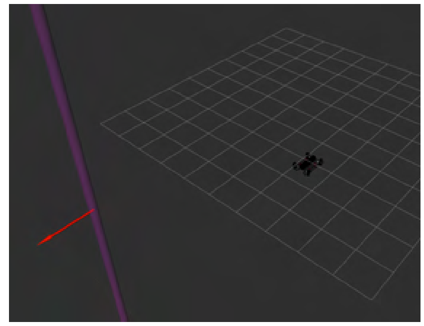

Fig. 9: Visualization of the covariance of a pole obstacle (position covariance is represented by the purple cylinder on the left) as seen by the robot in RVIZ (visualization tool).

tested. This filter is used to sort through all the information that is received from the perception system and rule out any false positives or unrealistic data. The team tested different cutoff values for this filter to see how it affected the system's performance.

The team's approach to testing relied heavily on simulation due to decreased water time caused by the COVID-19 global health crisis. Specifically, most simulation testing of the mapping system was done by detecting and tracking a pole in the water. The mapping system was given a rough initial estimate of where it should expect the pole to be, as well as a certainty value of the estimate. It would then update this estimate as it received frames of data from the perception system.

Testing the system with a low covariance (an x and y covariance of $1.0m^2$ resulted in an extremely accurate estimate of the object's position, resulting in a covariance of less than $0.0025m^2$ after 2 minutes see figure 10. Since estimates are merged by combining probability densities, the mapping system can close in on an object's position even after the system has accrued its own uncertainty of the robot's position.

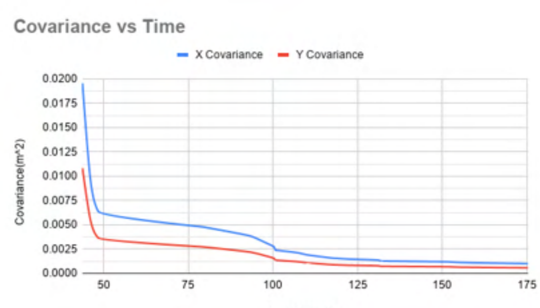

Fig. 10: Graph of covariance over time with an initial covariance of $1.0m^2$ for both X and Y. Shows covariance over time after the first detection was received from the perception system.

3) Task Code: Testing the task code consisted of three stages, compiling the states, simulating the behaviors, then running the tasks during pool tests. Compiling and running the states when not connected to the simulator allowed the team to find which states were causing errors without a heavy load of trying to simulate the bugged code and gave an opportunity to fix them easily. Next, the simulator approximates how the task code would work in the real world which allows the team to see if there were any logical flaws in the design of the behaviors without the need to meet in person or schedule a time for a pool test. Finally, the tested behaviors are used at a pool test to test the accuracy of the simulation and to practice implementing them on the robot in real life. All this testing was built into the development process and timeline when first creating the states and was a high priority in minimizing problems in development.

IV. Acknowledgements

UWRT would like to thank everyone who helped the team over the course of the past year, notably: Dr. Saeedeh Ziaeefard, the team's advisor, who kept the team leads on track and pushed the whole team to improve; and Hollie Hinton, the Director of Corporate Relations at Ohio State University, who helped the team secure funding and sponsors.

Additionally, the team would like to thank The Ohio State University College of Engineering and The Center for Automotive Research.

V. References

[1] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. http://ceres-solver.org.

[2] David Coleman, Ioan A. Șucan, Sachin Chitta, and Nikolaus Correll. Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt! Case Study. In *Journal of Software Engineering for Robotics*, volume 5, page 3–16, May 2014.

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

[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] 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.

[6] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew Ng. Ros: an open-source robot operating system. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics*, Kobe, Japan, May 2009.

[7] Joseph Redmon. Darknet: Open source neural networks in c. http://pjreddie.com/darknet/, 2013–2016.

[8] Philipp Schillinger, Stefan Kohlbrecher, and Oskar von Stryk. Human-Robot Collaborative High-Level Control with an Application to Rescue Robotics. In *IEEE International Conference on Robotics and Automation*, Stockholm, Sweden, May 2016.

VI. Appendix A: Component Specifications

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	Under Construction
Board-Side Waterproof Housing			1' long x 8" dia.	$2,500	Under Construction
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	9.5" long x 5.0" wide x 4.50" tall	$110	Under Construction
Claw Manipulator	Custom	Torque Coupler	2.5" long x 5.5" wide x 7.75" 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	FlexBE	N/A	N/A	N/A	
Battery	MaxAmps	Lithium Polymer	2x, 5S, 18.5V, 150C	Re-used	
Converter	TDK-Lambda	I6A4W(250W)	3x, 3.3V, 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	C++				
Programming Language 2	Python				
Compass	Not Present				
Inertial Measurement Unit (IMU)	LORD MicroStrain	3DM-GX4-25	1x	Re-used	
Doppler Velocity Log (DVL)	Nortek	DVL1000	1x	Re-used	
Camera(s)	Mynt	EYE S	1x	$239	Installed
Hydrophones	Aquarian Audio	AS-1	3x 0.47"	$395	No Purchased
Algorithms: Vision	OpenCV				
Algorithms: Acoustics	Phase Difference	Custom			
Algorithms: Localization and Mapping	"Conceptual" SLAM				
Algorithms: Autonomy	YOLO				
Open source software	ROS and OpenCV				
Team size	40				
HW/SW expertise ratio	8/3.				
Testing time: simulation	200 hours				
Testing time: in-water	0 hours				

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 marine engineering. This year, due to the pandemic, these events were cancelled and the team did not participate. The team looks forward to the future where knowledge and excitement about STEM can be spread.

Fig. 11: UWRT's robot family.

During the pandemic the team wanted to expand outreach opportunities and further promote STEM education within the community. To accomplish this goal the team developed the STEMBot Workshop, an outreach event tailored to a 10 day after school program for late elementary and early middle school classrooms. The program has gained traction and the team is set to take it to a local middle school in the fall. The program focuses on the building and programming of a small ROV. Students will be lead through activities in building the ROV and in the end the students will compete in a small obstacle course.

Fig. 12: STEMbot, UWRT's outreach vehicle.

Fig. 13: Kids surround a pool while one controls STEMbot.

VIII. Appendix C: Figures

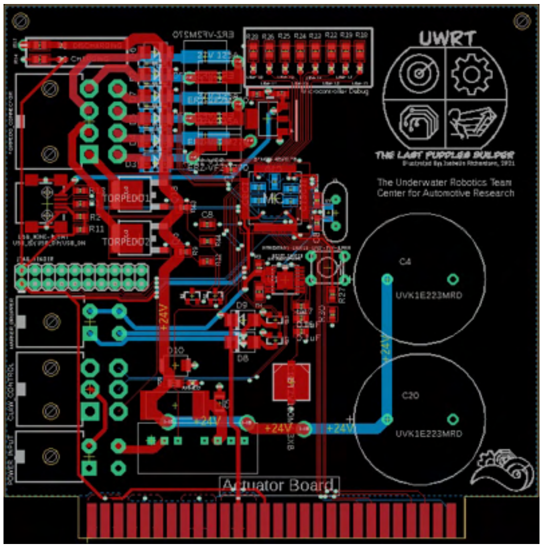

Fig. 14: Actuator Board Art

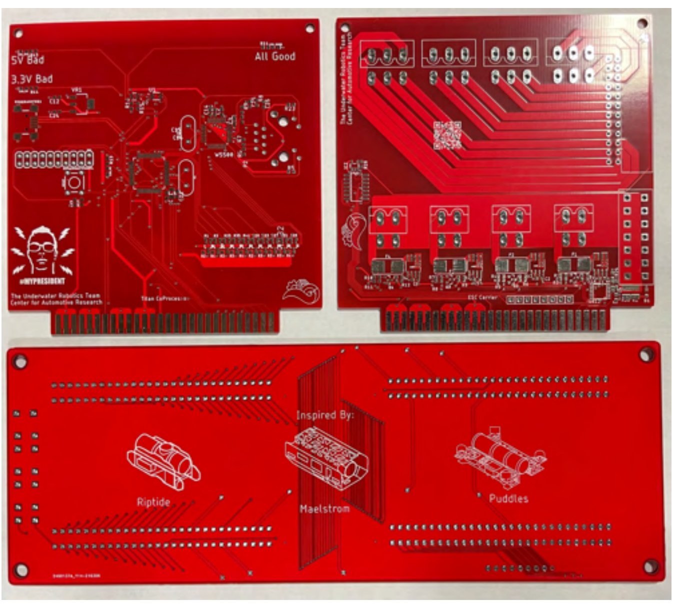

Fig. 15: Team designed board art