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

Anden Acitelli, Collin Barack, Phillip Barker, Jennifer Humanchuk, Blaine Miller, Parth Parekh, Patrick Tosh

***Abstract*—Since 2016, The Underwater Robotics Team (UWRT) at The Ohio State University has designed and built new Autonomous Underwater Vehicles (AUVs) each year to compete at the AUVSI RoboSub competition. For the 2019 – 2020 school year, instead of building a new vehicle, the team focused on improving the existing systems of Puddles, the team's 2019 competition vehicle. The goal was to increase performance at competition by making each individual component more reliable. This included: rebuilding the codebase for controls, task code, and vision processing; iterating task mechanisms; and redesigning the robot's custom printed circuit boards (PCBs). Additionally, rigorous use of new testing applications, such as a simulator, showed an increase in reliability due to the changes made throughout the year.**

I. Competition Strategy

DURING the 2019 RoboSub competition, the Underwater Robotics Team at Ohio State University reached 11th place in semi-final runs, missing the threshold for the finals by 50 points. The vehicle was able to perform more tasks than in previous years, but untested task code and failures in mechanisms resulted in a lower score than the team had expected. Given these shortcomings, the team decided to not make any significant changes to the vehicle's chassis to give more time for the sub-teams to implement needed improvements. The goal with these changes was to make a more robust platform which would consistently perform at its peak capability. In addition, by continuing to focus on the same robot, senior members were able to improve their designs, fix previous mistakes, and transfer knowledge to new members.

A. Software

Due to the pandemic and a lack of fully realized tasks for the 2020 competition, the team designed new task code based on the obstacles and tasks used in the 2019 competition. A top goal for the team was to have the codebase ready for competition by the time the in-person event would have taken place. This would be made possible due to the restructuring of the software team into a SCRUM style workflow early in the school year, allowing for an improved project completion rate. In addition to the changes in organization, the team decided to rework the way code was written for the robot. One of the largest changes to the code base was the robot's task code. For the 2019 competition, the task code for Puddles was written during the summer leading up to the event, which led to a basic code design that could easily fail to reach parameters necessary for completing a task. These failures were addressed by making the code a series of action blocks, and combining the use of failure states and improved mapping. This would allow the robot to either retry or move on to another task if the current attempt was taking too long or had failed in some way.

In addition to restructured task code, newly improved detection and scanning protocols were created to allow the robot to observe the environment and mark the locations of key points. After this map was established, the vehicle would need to be able to accurately navigate it, which required revision of the team's controls system. Although the controls system from the previous year succeeded in making the robot stable, it failed to achieve the desired maneuverability due to gimbal lock and poor trajectory planning. To resolve these issues, the team improved on the controller's existing state, moving towards a simpler cascaded P control method and adding a trajectory manager. The team extensively tested all of the new code by performing function testing in the university's diving well whenever available. Implementation of a simulator, which began production at the start of the school year, became paramount once COVID-19 forced the team to cease all in person activities. The simulator enabled the team to test updated code and receive immediate feedback without needing a pool.

B. Electrical

The electrical team focused on addressing reliability issues with the printed circuit boards (PCB's) used for the 2019 competition. The stress of near constant testing put on the boards during competition highlighted design flaws, such as improper inter-integrated circuit (I2C) connections and bleeding current, which led to loss of control of the vehicle during runs. The electrical team redesigned four of the five boards, electing to reuse the backplane. This allowed the team to focus on reworking the actuator, electronic speed controller (ESC), battery balancer, and coprocessor boards. Special attention was paid to the actuator board, because the previous version of the board was created late in the summer and lacked proper function testing. For 2020, the team worked to ensure the functionality of each board with thorough testing and the robustness of the system as a whole was improved with the creation of backups for each PCB.

C. Mechanical

With the decision to re-use the previous chassis, the mechanical team had more time to improve on the task mechanisms which had failed during the 2019 competition. The marker dropper and torpedo launcher failed due to a lack of proper waterproofing of the electronics, and the Blue

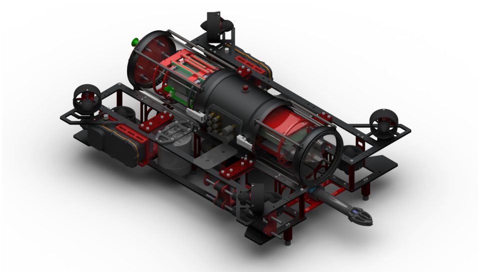


Fig. 1: CAD rendering of Puddles.

Robotics manipulator arm's failure point was an overspent motor due to a flaw inherent in the design. Re-engineering and thoroughly testing new mechanisms were the primary goals for a subset of the mechanical team. Not only were the new designs tested and thoroughly waterproofed, but backups were also designed for each system. Each component that needed electrical communications with the main housing of the vehicle was designed to use the same connection points to enable hot swapping and compact modularity. Additionally, time that would have been dedicated to remaking the chassis was spent instead on the first year of a two-year design process to create a new competition vehicle from the ground up for use in the 2021 competition.

II. Vehicle Design

The majority of the software, electrical, and mechanical improvements this year were done with a focus on reliability.

A. Software

Through the year, the software team made many changes to the robot in the name of simplicity and reliability: the team was able to add an Extended Kalman Filter (EKF) to a newly reworked controls system; the vision system was improved with new data handling and generation methods; and the team created a new, simpler way to tackle writing task code. Additionally, to accommodate the growing number of club members, the team created a robot-human interface called Robothoughts which can be accessed with any smartphone or computer.

1) Vision System: The team needed to rethink the robot's vision system after the 2019 competition year. The robot would not consistently reach the object recognition threshold required to progress to the next step of code due to low-resolution cameras and poor object recognition. The task failures and loss of points necessitated a rework of the visual detection systems, starting with the purchase of new and more reliable cameras. The new cameras, being a linked pair of stereoscopic detectors, are able to natively run the majority of the depth-to-target calculations formerly run on Puddles's graphics processor. This frees up much needed operating power to dedicate towards object recognition. In addition, the higher resolution of the new cameras allows for improved accuracy when finding tasks.

In addition to upgrading the cameras, the team sought to maintain its position at the forefront of data collection for the competition by implementing a new method of data generation. In years past, the team gathered visual data by taking footage of physical props in as many different pools as possible, and then would hand label the objects in each frame of the videos. To save time and produce a much wider array of images, the software team was able to work with a Gazebo simulation to generate thousands of pre-labelled images. The simulation loads a 3d model of the task props and takes snapshots from every angle with varied water and lighting conditions. This amalgam of vision data enabled the team to undergo rapid testing of various object detection techniques, leading to greater confidence in detection.

2) Control System: The previous controls system for puddles produced a stable robot, but the old system of seven layered controllers was pieced together over many years, resulting in a messy code-base, and the robot could not plan or follow optimal trajectories. To reorganize the controls in an easier to understand way and add trajectory planning, a new control system was created. This 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 worry about buoyant torque trying to force the vehicle upright. Once a trajectory is created, the system continues to feed the desired current state into the controller while the robot advances through the trajectory in real time, all the while ensuring it is in the right position, has the correct velocity, and is achieving the correct acceleration. 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. The previous system could theoretically enter a position aligned with a set of global axes that locked the orientation, and careful programming was needed to avoid this. The new system handles this 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 aspects of the robot. When trying to move the vehicle, separate commands needed to be sent each of the controllers, 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 responses.

3) Extended Kalman Filter: Due to the unreliability and noise inherent in each of the vehicle's sensors, the team implemented a method to combine all sensor outputs into one unified measure of robot positioning, known as an Extended Kalman Filter. This method allows for greater accuracy than relying on a single sensor's measurement. The filter, when fed data from the robot's DVL, Depth Sensor, and IMU, allows the team to better estimate the state of the robot, even if an individual sensor fails to transmit data. The EKF also minimizes the impact of magnetometric noise from the robot's thrusters by controlling sensor inputs against one another, ensuring consistent readings even during periods of high thruster activity. With the filter active, the robot's positioning in the mapping software is more accurate, which lets the team utilize it as another tool for autonomy.

4) Task Code: To better complete the tasks presented under the RoboSub checklist, the team wanted to move away from hard coded and difficult to understand task code. To accomplish this, the software team created a modular task code system using Flexbe in an effort to increase understanding and minimize development time. Flexbe, a graphical interface used to create robot movement behaviors, was decided upon due to its easy to understand interface that still allows for complex behaviors to be created. With the new system, the team can assemble individual actions that they would like the robot to complete, and connect them together to create larger behaviors. As an example, for the gate task, the larger behavior is made up of a search action, an alignment action, a positioning action, and a movement action to proceed through the gate. The simple nature of the programming results in an incredibly modular system that allows team members unfamiliar with programming to understand and create basic task code. While complex actions are still left to the seasoned programmers to implement, this visualization system greatly improved communication within the team.

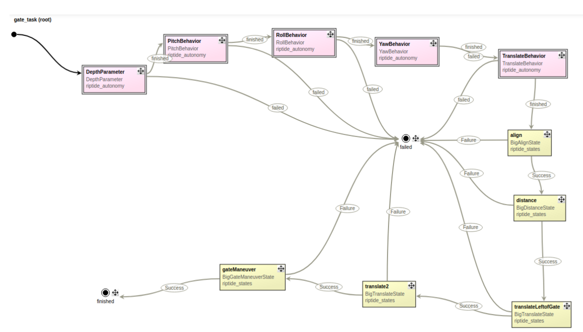


Fig. 2: FLEXBE tree for the gate task

5) RoboThoughts: A major goal of the team was to improve the experience of working with the robot for non-programmers on the team. To accomplish this goal, the RoboThoughts project was initialized. The project involved the creation of a web-based portal that enables anyone nearby to view the robot's critical information, including data such as current depth, orientation, the robot's current camera view, and a select few operating errors. This custom-built system was designed to be compatible with future changes, including expansion to more low-level sensors and real-time visual integration of the robot's mapping system. The project is built using a Python/Flask backend and a React frontend with a major focus on user friendliness, intuitiveness, and usability.

B. Electrical

Puddles's 2020 electrical system is similar in design to its 2019 structure. The system is composed of five custom PCB's: A battery balancer board to control and monitor the discharge of the batteries, an actuator board to control the task mechanisms, an electronic speed controller carrier board to hold all of the robot's ESC's, a coprocessor board to connect the electronics to the computer and provide a low level hardware interface, and a backplane board to connect all the other boards together. In addition to modifying four of the five PCB's, the team made use of a Peltier panel to control heat buildup in the vehicle. This year, the team focused on improving individual boards to increase reliability and ensure proper functionality.

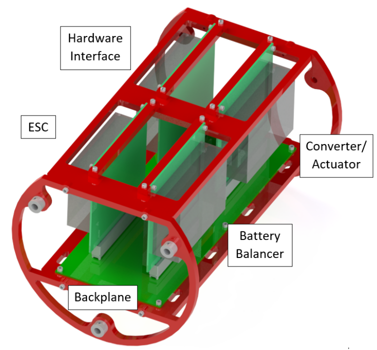


Fig. 3: Labelled CAD rendering of Puddles's PCB housing

1) Coprocessor Board: The coprocessor board acts as the central connection between the main computer in the robot and the custom boards. In addition, the board is responsible for developing the pulse width modulation signals for thruster control. The coprocessor board uses a microcontroller; a 32-bit ARM processor, the STM32F405RGT6; and two I2C communication lines. One of these I2C lines is responsible for communicating with the main computer while the other line communicates with the other custom PCB's. In addition to the I2C lines, the new coprocessor board also features: an ethernet port for interfacing through a network switch, eleven indicator LEDs for quick debugging, and two programming interfaces. The largest change to the coprocessor was a change in the software to publish state information as a ROS node. Previously, the robot's coprocessor hosted a webserver to display state information, which was prone to crashing. Because the new software uses a pre-established communication system instead of a custom one, the risk of crashing is greatly reduced.

2) Backplane: The backplane serves as the main mounting hub for the rest of the robot's PCB's. The design was chosen to allow boards to be removed and added individually for inspection, repair, and replacement. Two sets of parallel boards are supported by vertical connectors which route control signal and power lines, such as on/off signals, and 3.3V/5V power lines. Traces between the connectors are grouped by function and run the length of the board to split data to respective board input pins. In order to reduce individual connections to each board, sensor SDA and SCL ports are centrally routed from the backplane. Additionally, the board design allows for direct information transfer of the robot's mechanical kill-switch, incoming PWM signals, and debug indicator signals. Displaying the boards in an outward facing orientation with an empty corridor down the middle provides ample room for cooling air flow across the PCB's, and debugging LED's are easily visible from outside the robot's main hull.

3) Battery Balancer Board: The battery balancer circuit board serves the primary function of balancing the port and starboard battery inputs and converting them to varying voltages for use with the other PCB's. The battery balancing board also houses the safety switches and indicators for each powerline, as well as the robot's kill-switch. These additional components were aggregated from other board designs in previous years in order to have a centralized debugging panel. Due to the added electrical components, spacing and organization were the two main focuses of board re-design. As such, motor power and balanced out connectors were repositioned, and components that use the balanced power trace were vertically stacked to reduce the length of large traces. To improve troubleshooting, each power indicating LED is centralized to the subsection of the board pertaining to its function. For example, 12V, 5V, and 3.3V LED's are positioned alongside their respective converters for an intuitive way to locate each section's status. As an extra measure to reduce space use, segments of the board which include connectors were intentionally positioned on areas of the board closest to their cable destination to reduce wire length.

4) ESC Board: The electronic speed controller board provides a centralized connection point for thruster output and PWM signals. The design of the ESC carrier board has not changed much from 2019 to 2020, but it represents a leap forward from the team's previous robots, reducing the number of cables from 48 to two. It makes use of standard screw terminals to hold the Blue Robotics ESC modules, of which Puddles uses eight. The board holds all eight ESCs and fuses, and allows easy access and removal in the event of faults. The ESC board interacts with the coprocessor board for current and voltage monitoring through the use of an I2C bus. In the past, the team had implemented current sensors into the ESC board, but the design did not work as intended. In the 2020 revision the team ensured the proper functionality of the current sensors by selecting better fuses and increasing the size of the traces to better accommodate the current draw and improve efficiency.

5) Actuator Board: The Actuator board controls the marker dropper, claw, and torpedo coils. The board enables the team to send high currents with specific timings into the coils of the torpedo, toggle the marker droppers, and set the position of the claw using pulse-width modulation. In the past the team's actuator board also featured a power conversion module to convert the battery power to different voltage outputs, which crowded the board. For the 2020 competition the team sought to increase robustness of the design and improve the use of space, accomplished by moving the conversion circuits to the battery balancer board. The new design utilizes the freed space by moving the microcontroller to the front of the board, and many components were spread out to increase accessibility. New voltage indicator LEDs connected to voltage lines and the microcontroller were added. In addition, the programming connector was replaced with one that does not need a special adapter.

6) Peltier Panel: One novel aspect of the electrical design is the use of a Peltier cooler. Puddles electrical system produces a significant amount of heat and the implementation of a Peltier cooler helps manage the temperature. The team uses a blended software and electrical approach to control the Peltier module based on the temperature inside the robot. A downside to the module that the team found was that the module would cause condensation inside the main housing. To combat this, the team added desiccant at key locations, which effectively eliminated the condensation problem.

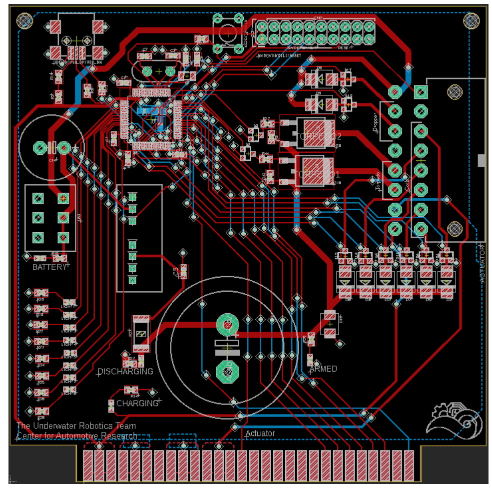


Fig. 4: EAGLE diagram of the actuator board

C. Mechanical

1) Chassis: The chassis used by UWRT for the 2020 competition is the same as the one used in 2019. It features a main tube housing with two removeable end caps, made primarily from 6061 aluminum, with acrylic panels in key locations. The acrylic sections create viewports both for the cameras to see out of, and for monitoring and debugging the electronics. The design also features mirrored body sections on either side of the main housing that provide a safety cage and mounting points for external sensors, thrusters, and task mechanisms. Wherever possible pieces of the chassis that had originally been made of aluminum were replaced with Delrin to reduce weight while maintaining structural integrity. With minimal changes needed for a chassis that had proven reliable, and a manipulator arm that had been repaired by the manufacturer, the team was able to focus on rebuilding and improving the torpedo launcher and marker droppers.

2) Torpedo Launcher: Last year, UWRT brought a novel concept yet to be seen in the RoboSub competition: a torpedo launcher that functioned purely with the use of electromagnetism, more commonly known as a Gauss gun. Similar in concept to a rail gun, the launcher propelled a solid steel slug forward by quickly passing a large current through a series of tightly wound coils of magnet wire. The current passing through the coils briefly generated a strong magnetic field that drew the projectile towards it, and using multiple coils with strict timing, the slug could be launched at high speed. UWRT's launcher used in the 2019 competition served mostly as a proof of concept, and was constructed using off-the-shelf coils and easy-to-find components. Although the design was simple to put together and use, the team wanted to move beyond the limitations given by retail parts in order to make a more powerful and reliable system. This led to the team recreating the mechanism entirely, starting with custom made coils, a 3d printed launch tube, and a novel electrical system. Each of the coils are wound around the segmented launch tube sections and then brought together and vacuum-potted to make them waterproof. The custom nature of the new system allows for finer control and a more predictable firing strength, able to be calculated with a simple aggregation of formulas. Unfortunately, with the pandemic and shelter-in-place rules, the manufacture of the torpedo launcher system could not be completed.

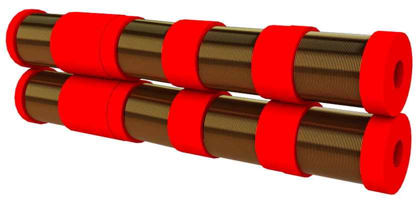


Fig. 5: CAD rendering of torpedo launch tubes.

3) Marker Droppers: For the 2019 competition, the team made use of electromagnets to create the robot's marker droppers. The mechanism worked by having a custom magnetic marker attached to each of the unpowered electromagnets. When the vehicle detected the correct conditions, a signal would power the electromagnet, reversing the bond between the magnet and the marker, and detaching it to fall into the bins. This system worked consistently up until its failure at competition due to such extensive testing that the waterproofing failed. Learning from this mistake, the new marker dropper system was designed on the same concept, with alterations made to make the waterproofing more robust. To improve upon the marker design, the team designed and machined a custom aluminum mold via CNC, into which a steel rod, large ball bearing, and several magnets are placed. Epoxy is then injected into the mold to give the markers their distinctive hydrodynamic teardrop shape and to bind the internal components together.

III. Experimental Results

Due to limited pool time and COVID, the team needed to effectively utilize out-of-water and virtual testing methods. The team attempted to have a pool test each month before the university was shut down, but mounting expenses were prohibitive. Because of this, the team developed new tools to test various parts of the robot without the need for a pool.

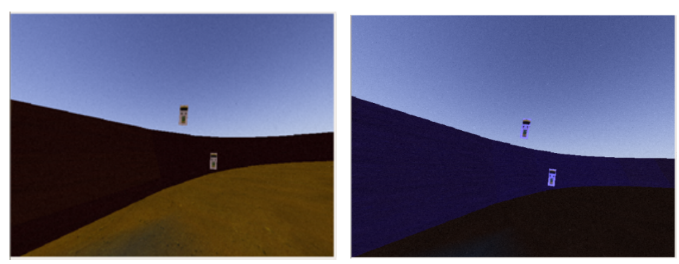
A. Mechanical

1) Finite Element Analysis: Because the team elected to reuse the chassis from last year, justification of the entire vehicle was not necessary to prove that the robot was soundly built; however, Finite Element Analysis was performed on various components to verify integrity with modifications made to reduce the vehicle's weight.

2) Torpedo Launcher Testing: In order to better tune the torpedo launcher's coil timings and strength, the team created a device to measure the strength of projectiles launched by the torpedo launcher. The design uses an impact plate attached with an L frame to a small load sensor on the opposing end to measure the raw input data. The data is then captured and interpreted with custom code on an Arduino, and stored on an SD card for further analysis. The system allows team members to see the exact launch strength of the projectile, which provides more empirical data to tune the timing of the launch coils. Previously this was done by observing the launch sequence and estimating strength based on how far the projectile was sent. Because production of the torpedo launcher was halted due to COVID, the completed test device could not be put to proper use

B. Software

1) Simulation: The team's simulator, created using uuv_sim on top of Gazebo, allowed the team to rapidly test new code and make changes to the controls and sensing systems without needing a pool test to gather data. The simulator has been expanded to allow an interface for rapid mass testing of the robot in randomly generated, parameterized environments. The team can easily generate an unlimited number of unique world files by randomly generating a list of key environment parameters or by use of a graphical user interface that allows for easy editing. The editable parameters include color tint of the world, "brackishness" of the water, props, prop locations, sun location, sun direction, and sun brightness. With this system, the team can expose the robot to a virtual TRANSDEC many times over with the goal of exposing the robot to all likely scenarios.



(a) Daytime environment (b) Nighttime environment

Fig. 6: Screengrabs of the simulation with different parameters

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 all of its sponsors who generously donated resources and materials: The Ohio State University College of Engineering, Honda OSU Partnership, FedEx, JLCPCB, Ohio Space Grant Consortium, Lord Sensing, Nortek, and Ford.

V. References

[1] Robot Operating Systems, www.ros.org

[2] YOLO, J. Redmon, "Darknet: Open Source Neural Networks in C," Darknet, 2013. [Online]. Available: https://pjreddie.com/darknet/.

[3] Google's Ceres Solver, http://ceres-solver.org/

[4] FlexBE, http://wiki.ros.org/flexbe

[5] Gazebo,http://gazebosim.org

[6] UUV Simulator, https://uuvsimulator.github.io

VI. Appendix A: Expectations

Table 1 lists the score expectations for RoboSub 2019 on page seven.

VII. Appendix B: Component Specifications

Table 2 lists components used in Puddles' system design on page eight.

VIII. Appendix C: Outreach Activities

UWRT's STEM initative 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. For MakerX, the team brought Puddles and the rest of UWRT's large robot family to show the wide range of forms and functions robots can take. At the Ohio State Fair, the team's STEMbot was placed in a small pool and community members were able to directly engage with the club by driving the vehicle around with a controller.

Fig. 7: UWRT's robot family.

STEMbot is a small ROV controllable via a PS3 controller, and this past year the team built a second one to improve STEM outreach capabilities. STEMbot is designed to demonstrate how easy it can be to get into STEM, being made with simple components such as PVC pipe and an Arduino.

Fig. 8: STEMbot, UWRT's outreach vehicle.

Further outreach events were cancelled due to COVID. Such events include include helping in a regional MATE competition with set up and judging, and putting on a workshop at the Center of Science and Industry (COSI), a local Columbus science center. The workshop would have involved teaching kids concepts such as buoyancy, creating watertight housings, and basic electronics. On campus, UWRT was invited to participate in Ohio State's Sesquicentennial celebrations by acting as a representative club to demonstrate the technical achievements of the college. Additionally, the team planned to host an event as part of Ohio State's Engineering Council Architecture and Engineering Week, where passing students and professors could take a test drive of STEMbot while learning more about the team.

The team has made efforts to adapt to the new climate by creating an online workshop targeted at low-income and underprivileged middle schools. The program's goal is to teach students the basics of coding and robot design through the creation of a virtual STEMbot in TinkerCAD. The free program will be implemented in early 2021.

Fig. 9: Kids surround a pool while one controls STEMbot.

TABLE I: RoboSub Expected Scores Based on 2019 Tasks

Subjective Measures			
	Maximum Points	Expected Points	Points Scored
Utility of Team Website	50	45	
Technical Merit (from journal paper)	150	128	
Written Style (from journal paper)	50	45	
Capability of Autonomous Behavior (static judging)	100	100	
Creativity in System Design (static judging)	100	80	
Team Uniform (static judging)	10	10	
Team Video	50	50	
Pre-Qualifying Video	100	100	
Discretionary Points (static judging)	40	20	
Total	650	578	
Performance Measures			
Weight	See Table 1/Vehicle		
Marker/Torpedo over weight or size by <10%	Minus 500/Marker		
Gate: Pass Through	100	100	
Gate: Maintain Fixed Heading	150	150	
Gate: Coin Flip	300	300	
Gate: Pass Through 60% Section	200		
Gate: Pass Through 40% Section	400	400	
Gate: Style	+100 (8x Max)	800	
Collect Pickup: Crucifix, Garlic	400/Object	400	
Follow the "Path" (2 total)	100/Segment	200	
Slay Vampires: Any, Called	300, 600	600	
Drop Garlic: Open, Closed	700, 1000 / Marker (2 + Pickup)	1400	
Drop Garlic: Move Arm	400		
Stake Through Heart: Open Oval, Cover Oval, Sm Heart	800, 1000, 1200/Torpedo (Max 2)	2200	
Stake Through Heart: Move Lever	400	400	
Stake Through Heart: Bonus - Cover Oval, Sm Heart	500	500	
Expose to Sunlight: Surface in Area	1000	1000	
Expose to Sunlight: Surface with Object	400/Object	400	
Expose to Sunlight: Open Coffin	400		
Expose to Sunlight: Drop Pickup	200/Object (Crucifix only)		
Random Pinger First Task	500	500	
Random Pinger Second Task	1500	1500	
Inter-Vehicle Communication	1000		
Finish the Mission with T Minutes (whole+factional)	Tx100		

TABLE II: Puddles' Component Specifications

Component	Vendor	Model/Type	Specs/QTY	Cost (if new)
Buoyancy Control	Not Present			
Frame	Custom	Custom	3' long x 2' wide 1' tall	$600
Waterproof Housing	Custom	Custom	3' long x 10" dia.	Re-used
Waterproof Connectors	MacArtney	Micro Circular	N/A	Re-used
Thrusters	Blue Robotics	T200	8x, 3-20V, 25A	Re-used
Motor Control	Blue Robotics	Basic ESC	8x, 7-26V	$25 Each
Propellors	Used T200 propellors			
Actuators	Custom	Electromagnets and coils	N/A	$100
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	Diamond Systems	Venus	i7-6600U Dual Core	Re-used
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)	Point Grey	BFLY-U3-132S2C-CS	3x	Re-used
Hydrophones	Aquarian Audio	H1C	4x, 0.98" dia.	Re-used
Manipulator	Blue Robotics	Newton Gripper	9-18V, 2.75" opening	Re-used
Algorithms: Vision	OpenCV			
Algorithms: Acoustics	Triangulation			
Algorithms: Localization and Mapping	"Conceptual" SLAM			
Algorithms: Autonomy	YOLO			
Open source software	ROS and OpenCV			
Team size	32			
HW/SW expertise ratio	8/3			
Testing time: simulation	200 hours			
Testing time: in-water	35 hours			