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

Collin Barack, Anthony Condo, Kyle Fathauer, Jennifer Humanchuk, Benji Justice, Noah Limes, Blaine Miller, Parth Parekh, Andrew Tian, Ted Sender, Hao Yu

***Abstract*—The Underwater Robotics Team from The Ohio State University designs and builds Autonomous Underwater Vehicles (AUVs) to compete at the AUVSI RoboSub competition. Building from the foundation of Maelstrom, the team's previous vehicle, the past year has been dedicated to upgrading all of the mechanical, electrical, and software systems. This year for RoboSub 2019, the team has created its newest AUV, Puddles, with the goal of securing a place in finals.**

I. Competition Strategy

THE Ohio State University's Underwater Robotics Team (UWRT) strives to keep learning and pushing the boundaries of what their autonomous underwater vehicles (AUVs) are capable of. In RoboSub 2018 UWRT competed with Maelstrom and placed 9th overall. UWRT's goal this year was to extend the capabilities of this platform to place as finalists in RoboSub 2019.

A. Team Organization

Due to the team's growth in size, UWRT devoted effort into strengthening its organizational structure to provide members with the necessary resources for success. The team leaders established consistent informational meetings and communication methods to ensure all members were up-to-date. Routine procedures were set in place such as taking meeting minutes (for both general meetings and design reviews) to a shared storage drive. In addition, a dedicated business team was formed to obtain more sponsorship opportunities. One of the business group's accomplishments was securing the funds from an OSU-Honda partnership to purchase the team's first Doppler Velocity Log (DVL) from Nortek at a discounted price. Furthermore, an iterative design process was followed to investigate both past failures and new designs before implementing them in the main vehicle.

B. Engineering Design Decisions

When RoboSub 2019 tasks were released the team assessed the robot's existing capabilities to determine which tasks were attainable and which required vehicle upgrades. From prior experience the team knew it could pass through the Gate with a fixed heading and identify a majority of the objects in the competition course. To improve on the previous year's performance, UWRT set the goal to pass through the Gate with ”Style” (such as a barrel roll while moving forward) and complete the Slay Vampires, Drop Garlic, Stake Through the Heart, and Expose to Sunlight Tasks.

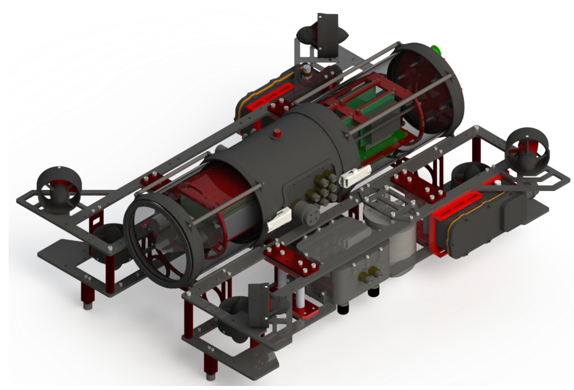

Fig. 1. CAD rendering of Puddles.

This goal required significant redesigning of Maelstrom's mechanical system, specifically the actuation mechanisms, chassis, and heat management methods. Maelstrom had an inconsistent pneumatic system because it used a pressure regulator not rated for underwater environments which output unpredictable pressures. Due to the team's background in electrical engineering, the decision was made to replace the pneumatic system with electromagnetic-based systems. The sheet metal chassis was structurally unstable, was subject to vibrations from thrusters, and made it difficult to access cables and external housings. To address these problems, the team decided to use an open-design, quarter-inch aluminum chassis for its structural integrity for mounting, ease of access to housings, and ability to allow for a vectored thrust configuration for improved vehicle maneuverability. Finally, overheating caused Maelstrom's main computer to lose communication to the electronics. This was addressed by adding fans, additional heat sinks to the computers, and reorienting the PCBs to be parallel to the vehicle's longitudinal axis to allow for more air flow.

Maelstrom's electrical system proved to work consistently, but there were some problems with electromagnetic noise and wire management. The electrical team addressed these by replacing over 40 wires with direct PCB-to-component interfaces and by placing filters around parts. Other changes involved using a new microprocessing architecture and designing a system which enables remote control of individual subsystems.

The software team began by using Maelstom's software stack as a foundation to build from. The area's chosen to improve upon were: enhancing the existing PID control system, integrating the DVL with sensor fusion algorithms, and improving the state machine for high-level decision making. Concurrent with these other projects were two other research projects: a custom, open-source Guidance, Navigation, and Control (GNC) library for AUVs and an alternative framework for developing the state machine.

The result of applying these changes over the course of the year was Maelstrom's successor, UWRT's newest AUV, Puddles.

C. Task Completion

The proposed order for task completion is as follows: Gate, Slay Vampires, Drop Garlic, Stake Through the Heart, and then Expose to Sunlight. To start the run, Puddles will pass through the gate, regardless of starting orientation, and pass through the smaller section while earning style points with four 90-degree rotations in the yaw and roll axes. Puddles will then drive to the Slay Vampires task, touch the single buoy and the far side of the three-sided buoy. Next, Puddles will maneuver to the Drop Garlic bins and drop custom markers in the open bin. Puddles will then move to the Stake Through the Heart task, shoot a torpedo through the heart, move the lever, and shoot another torpedo through the newly-opened oval. Finally, Puddles will travel to the Expose to Sunlight task where it will grab the vampire in the open coffin and surface in the octagon to end the run. Expected scores can be found in Appendix A.

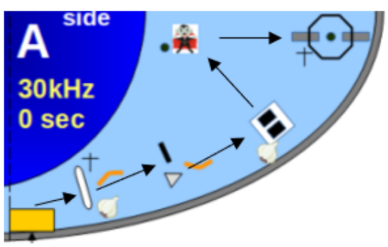

Fig. 2. Proposed plan for completing tasks.

II. Vehicle Design

Throughout the system, every aspect of Puddles' design was carefully considered and implemented, resulting in Puddles being UWRT's most advanced and agile AUV yet.

A. Mechanical

1) Chassis: The decision was made to have Puddles use the same cylindrical aluminum main housing as in previous years as it seals easily, is the optimal design for enduring pressure loads, and would save the team time by not having to create a new one. The new chassis is a quarter-inch aluminum frame which can be broken down into two mirrored ”wings” and connecting cross-beams which the main housing is attached. This chassis design provided stability for the external housings and mechanisms to be mounted to any location and be accessible from all angles. It also supports a vectored thruster configuration where each corner of the vehicle contains a single heave and vectored thruster.

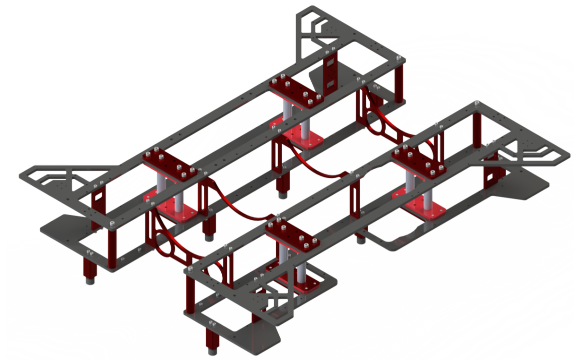

Fig. 3. CAD rendering of Puddles' chassis.

2) Actuators: The torpedo launcher consists of two custom coil guns. Each uses a series of six coils which activate in sequence to propel a 0.25” by 3.25” metal slug through the water. The system is powered with 18 - 21V through Puddles' main batteries and controlled with a timing module. The coils are Peter Paul Electronics 5C-7-K24 wire coils which were received from a donor. The slug is made from low carbon steel to maximize the driving force exerted on the projectile by the coils' magnetic fields.

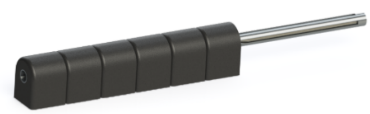

Fig. 4. CAD rendering of torpedo launcher system.

There are two marker droppers, each composed of a electromagnet and a magnetic marker. When the electromagnet is unpowered the marker will stay in place, only to be released when powered. The markers are made with a steel ball, two magnets, and a solenoid actuator rod molded together using epoxy. The entire mechanism is contained within a 3D printed shroud for stability and mounting to the vehicle.

Fig. 5. CAD rendering of marker dropper system.

For a manipulator the team purchased a BlueRoboics Newton Subsea Gripper to be attached to the front of the vehicle.

B. Electrical

1) System Architecture: Puddles' internal electronics is composed of five custom PCBS: the backplane, a battery balancer board, a voltage converter and actuator control board, an electronic speed controller (ESC) board, and a hardware interface board. The battery balancer draws in power from the two batteries (18-21V each) while ensuring one does not charge the other. The voltage converter inputs the battery voltage and outputs the required voltages for all on-board electronics. The ESC board is a dedicated PCB for the eight ESCs used for the thrusters. Previously, the ESCs were wrapped in a bundle and required 48 individual wires to be plugged in; by mounting each ESC between two screw terminal blocks on a single board the number of wires was reduced to a single power input and ESC output line. The pulse-width modulation (PWM) signals controlling the ESCs were then routed through the backplane. Finally, the hardware interface board consists of a STM32 microcontroller with the micropython framework, allowing it to be programmed in Python for simplified coding and eliminating problems using Real Time Operrating System (RTOS) libraries.

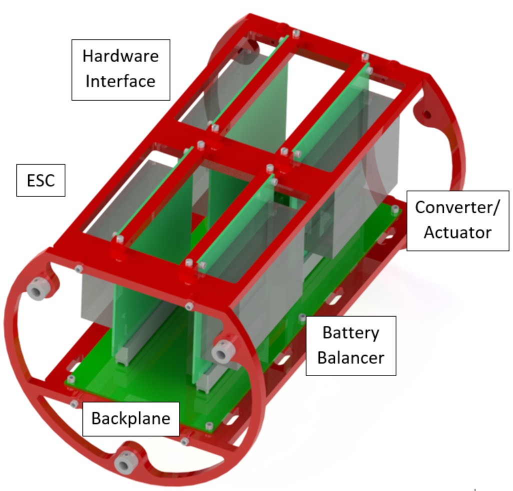

Fig. 6. CAD rendering of the internal electronics.

On each board debug LEDs, voltage sensors, and current sensors were added to gain a better understanding of the system's load. Further, when possible digital controls were enabled on electrical components and connected to the hardware interface board through inter-integrated circuit (I2C) communication lines. This design allowed for additional functionality because the thrusters, computers, cooling systems, and/or the entire vehicle could be monitored, recorded, and controlled through a single system.

2) Operator Console: To capitalize on the statistics being gathered throughout the electrical system, an operator console website was designed. When any computer is connected to the robot it can open up a webpage which reports battery statistics, current draw, temperature of the system, etc.. There are also additional buttons to turn on and off the voltage supplies for the 12V and 5V power lines in the robot along with shutting down the thrusters. This functionality allows the robot to be restarted without having to open up the main housing, saving time during testing.

3) Acoustics: Using Maelstrom's system as a starting point, Puddles used the same H1C hyrdrophones, external housing, and base design for the digital signal processing (DSP) board. The alterations to the board design were to minimize the size by replacing components and having it directly plug into the ZEM5305 FPGA. The custom algorithm used locates the ping by finding the point of steepest increase in amplitude of the incoming signal, which indicates the start of the ping. Once this point is found, OpenCV's match template [1] is used to isolate and match the starting point from the other incoming hydrophone signals to find the time difference between them. From this time difference calculations are performed to find the angle of arrival from the pinger.

C. Software

1) System Architecture: Puddles' software architecture is based on the Robot Operating System (ROS) [2] framework. ROS provides the ability to code different processes that are stand-alone, yet can easily pass data to and from each other over a publisher/subscriber interface. Within this framework the four primary software groups are: hardware-software interface, sensor and data processing, controllers, and state machine/mission execution.

2) Object Recognition: Like with Maelstrom, Puddles has a dedicated graphics computer, a Nvidia Jetson TX1, for vision processing and machine learning. Puddles uses the You Only Look Once (YOLO) [3] framework to utilize the team's prior experience with the platform and its ability to identify objects when trained. The team has already collected and labeled a diverse data set for all the tasks in the competition from which YOLO is trained on.

3) Stereo Vision: Stereo vision is able to identify distances to objects by comparing images between two adjacent cameras. By knowing the separation of the cameras on Puddles, calculations are performed via a ROS stereo vision package to generate a 3D point cloud. From experimentation this proved to work well, as the water filters objects (especially those far away), leaving behind only objects in close range. Stereo vision is used in conjunction with the YOLO detection output to provide spatial information of an object. Using these two sensors together also aids in obstacle avoidance and mapping of the competition area.

4) Improved PID Controllers: Maelstrom's primary control system wass composed of seven PID controllers with their outputs tied together in a single non-linear solver (Google's Ceres Solver [4]) to determine the appropriate thrusts. With Puddles, the PIDs were given a new configuration where motion profiles were used to provide smoother motion. Each position controller feeds into a velocity controller which calculates the appropriate velocity needed to prevent overshoot. After adding the DVL to Puddles' sensor suite, a Kalman filter was written to estimate the translational state of the vehicle to use with this new PID configuration.

5) State Machine Behaviors: The state machine executes the high-level decisions based on the information provided by sensors and other systems. ROS has a python-based state machine which integrates with the FlexBE [5] library. Each task is written in the ActionLib [6] framework and broken down into universal actions such as ”move” and ”find”, and then pieced together into ”behaviors” through a GUI, minimizing the amount of coding required. These behaviors can then become part of the state machine. The goal of this method of development is for any software team member to easily be able to write task code by using a series of basic states, pass in the needed information, and then build a behavior. This project is in working development and will continue to be improved over the course of the next year.

6) Guidance, Navigation, and Control Library: For six-degrees of freedom (DoF) vehicles, like an AUV, PID controllers are simple to use, but are limited with only three parameters to adjust. To keep improving control capabilities, the team began to develop an an open-source GNC library for AUVs, with a user interface designed for ROS. The Navigation package includes an asynchronous Kalman filter for estimating the translational states. The Guidance package allows the user to create a trajectory from arbitrary endpoints. The Control package uses the trajectory and Newton Euler equations to linearize about each reference state to solve for nominal thrusts. A Linear Quadratic Regulator (LQR) is then used to account for perturbations from the desired reference state. Due to time constraints on testing, only the Navigation package will be used during RoboSub 2019.

III. Experimental Results

Due to limits on pool availability and inclement weather conditions, in-water testing time with Puddles was minimal. As a result, the team sought to maximize out-of-water testing to more efficiently perform in-water testing.

A. Mechanical

1) Finite Element Analysis: Finite Element Analysis (FEA) was performed on various components to validate designs and ensure each component met specifications in terms of weight, strength, and volume. One example of performing FEA was in the redesign of the chassis. This analysis showed the design would be negatively buoyant, and gave the mechanical team time before production to alter characteristics of components to make them smaller or lighter weight without compromising the vehicle’s structural integrity.

B. Electrical

1) Worst Case Scenario Testing: Each circuit board was designed with requirements to meet the system needs and fail-safes were integrated wherever possible. However, the board designs were not tested enough to ensure each board met these requirements. Subsequently, failures occurred during in-water testing. For example, the current sensors on the ESC board had internal fuses which failed to burn open unless subjected to high enough currents for an extended time. As a result, traces on the board burned when tested above predetermined limits. Additionally, it was discovered the hardware interface board was unable to handle intermittent disconnections due to a bad connection from the tether. After these events, the team began performing worst case analysis on individual components before testing the whole system.

When unable to run the needed analysis, such as with the acoustics system due to the high cost of underwater pingers, the team resorted to other testing methods. To tune the acoustics system to recognize the a ping and not an echo, acoustic data was collected at varying locations and frequencies within a room with high reverb. From the collected data the algorithm was adjusted until it could correctly identify the direction of arrival of each recorded signal and was validated with real-time data.

C. Software

1) Dry Runs & Computer Simulation: Before every pool test, a set of goals and guidelines were recorded for what was to be accomplished and hardware-in-the-loop (HIL) testing was performed. Inputs were simulated through moving the robot to different orientations and falsified through software when needed. Code adjustments were made after each HIL test, which reduced the need for debugging while at the pool. Additionally, deadlines were set prior to each pool test for when electrical and mechanical changes were or were not permitted to allow enough time for software to be tested.

Alongside physical testing, the team began developing a software simulation package for Puddles using Gazebo [7], the simulator designed for ROS, along with the underwater plugins and environments provided by the open source UUV Simulator [8]. Gazebo simulates the physics of the vehicle while virtual sensors provide access to the simulated vehicle state. This state is then made available to Puddles’ software stack via a ROS interface, creating a software-in-the-loop (SIL) simulation. The simulation package will be used leading up to competion to test task code and will continue to be improved over the course of the next year.

IV. Acknowledgements

UWRT would like to thank everyone who helped the team over the course of the past year: Matt Little from Ohio State’s Center for Automotive Research, who assisted with mechanical design and manufacturing; Cory Baxter from Nortek, who assisted the team in acquiring a DVL and helped integrate it in Puddles’ design; Shawn Midlam-Miller, the capstone advisor for a group of the team’s seniors, who helped secure the extra funding from the university to purchase the DVL; and finally, Tom Ryan from Versalogic, who provided the team with a Baycat computer and helped integrate it into Puddles’ system.

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, B&G Tooling, FedEx, ExxonMobil, JLCPCB, Ohio Space Grant Consortium, Lord Sensing, Nortek, Ford, Danco, Diamond Systems, and Versalogic.

V. References

[1] OpenCV, www.opencv.org
[2] Robot Operating Systems, www.ros.org
[3] YOLO, J. Redmon, Darknet: Open Source Neural Networks in C, Darknet, 2013. [Online]. Available: https://pjreddie.com/darknet/.
[4] Google's Ceres Solver, http://ceres-solver.org/
[5] FlexBE, http://wiki.ros.org/flexbe
[6] ActionLib, http://wiki.ros.org/actionlib
[7] Gazebo,http://gazebosim.org
[8] 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 demonstrated its vision capabilities. At the Ohio State Fair, smaller remotely operated vehicles (ROVs) were showcased, where guests could actively control them in a small pool. One of which was the team's own design, STEMbot.

Fig. 7. STEMbot, UWRT's remotely operated vehicle.

STEMbot is a small ROV controllable via a PS3 controller and was redesigned over the past year with the purpose to participate in STEM outreach. STEMbot is designed to demonstrate what is possible with the simple components of PVC pipe and an Arduino. In April 2019, the team brought a small pool to Alum Creek Elementary School in Columbus, OH to teach fifth graders about the concepts involved with underwater robotics and also gave them an opportunity to drive STEMbot.

Fig. 8. Kids surround a pool while one controls STEMbot.

Other events included 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 involved teaching kids concepts such as buoyancy, creating watertight housings, and basic electronics. The team also visited the House of New Hope treatment foster care center in Columbus OH to host a booth at their annual Harvest festival. On campus, UWRT participated in the College of Engineering Dean's and Homecoming Tailgates to teach visiting alumni about the mission of the team. Additionally, the team hosted an event as part of Ohio State's Engineering Council Architecture and Engineering Week, where passing students professors could take a test drive of STEMbot while learning more about our team.

TABLE I
ROBOSUB 2019 EXPECTED SCORES

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	$15,000
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	$329
Algorithms: Vision	OpenCV			
Algorithms: Acoustics	Triangulation			
Algorithms: Localization and Mapping	"Conceptual" SLAM			
Algorithms: Autonomy	YOLO			
Open source software	ROS and OpenCV			
Team size	11			
HW/SW expertise ratio	8/3			
Testing time: simulation	20 hours			
Testing time: in-water	50 hours			