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

Alex Borchers, Anthony Condo, Kyle Fathauer, Jennifer Humanchuk, Benji Justice, Erika Klek, Noah Limes, August Mason, Blaine Miller, Jun Nishikawa, Diego Quevedo, Ted Sender, Cam Spicer, Gus Workman, Hao Yu

***Abstract*—The Underwater Robotics Team is a student organization at The Ohio State University that fabricates Autonomous Underwater Vehicles (AUVs) to compete at the AUVSI RoboSub competition. The team is made up of three subteams - mechanical, electrical, and software. This year, the team designed a sophisticated vehicle, surpassing previous vehicles abilities across the board. With creativity in mind, our new AUV, *Maelstrom*, consists of versatile mechanical, electrical, and software platforms, with redesigned pneumatics and acoustics subsystems. *Maelstrom* also has basic forms of autonomy and is capable of securing a spot in finals.**

***Index Terms*—AUV, ROS, YOLO, Sheet Metal, STM32**

I. Competition Strategy

THE AUVSI RoboSub 2018 competition tasks were broken down into three levels of difficulty: beginner, intermediate, and advanced. Beginner tasks (*Casino Gate* with random start orientation, *Path Marker*, and *Shoot Craps*) require only object detection and basic controls, with no external mechanisms necessary. Intermediate tasks (*Buy Gold Chip* and *Play Slots*) are more difficult, requiring the addition of an operational pneumatics / mechanical subsystem and competent vehicle control. Lastly, the advanced tasks (*Play Roulette* and *Cash In*) add the additional challenge of requiring an acoustics subsystem with exceptional software integration.

Over the past year, a majority of the team's time and resources were spent creating dependable mechanical and electrical systems to establish the foundation for a stable and reliable control system. Rugged mechanical, electrical, and control platforms are also required for a successful machine learning (ML) focused autonomous software stack. Given the areas we prioritized, our strategy for approaching competition tasks was to focus on the beginner and intermediate tasks, while making stretch goals for the advanced tasks. *Maelstrom*, illustrated in Figure 1, is able to accomplish all of the beginner tasks and part of each Intermediate and Advanced task. With the intermediate tasks, *Maelstrom* is capable of: pushing the plate for *Buy Gold Chip*, pulling the *Play Slots* handle, and shooting the torpedoes through the slots. For the advanced tasks, *Maelstrom* can drop a marker onto the roulette wheel and surface inside the *Cash In* square, so long as *Maelstrom* locates these tasks via the acoustics system or ML.

A. Software

The challenges faced in RoboSub are no different than years past. The competition focuses on precise controls, accurate computer vision, and autonomous task-handling systems. To advance *Maelstrom's* performance, our software enhancements emphasized the development of a robust control system, computer vision software, and preliminary forms of autonomy. The entire control system was rewritten to fine-tune *Maelstrom's* motions while efficiently integrating all sensors into the larger software stack. To improve computer vision systems / autonomy, a ML algorithm, YOLO (You Only Look Once) [1], was implemented with custom training data.

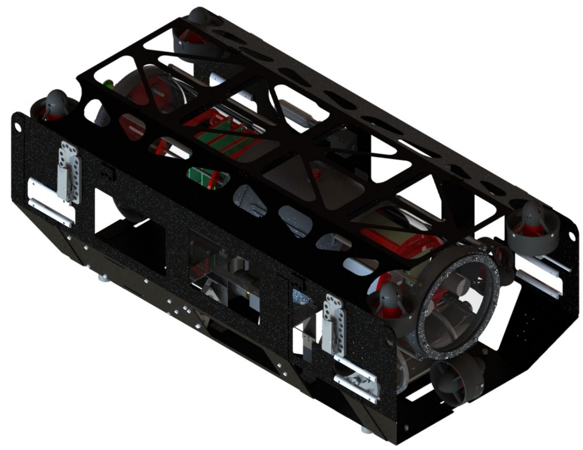

Fig. 1. CAD rendering of *Maelstrom*.

1) Control System: Controlling a 6 degrees-of-freedom vehicle in water was a challenging task due to complex equations of motion (EOM) and inherent uncertainties. Of the existing control methods, Proportional-Integral-Derivative (PID) controllers were favored for their ease of implementation and use over other options, such as Linear Quadratic Regulators (LQR). While LQR can provide greater control with improved stability, it requires advanced vehicle characterization, a deeper understanding of control theory, and still suffers from system nonlinearities.

Maelstrom's control system consists of seven individual PID controllers one for depth, three for angular motion (roll, pitch, and yaw), and three for linear motion (x, y, and z). To reduce complexity, each of the three angular and linear motion controllers were grouped into a single attitude and alignment controller, respectively, resulting in three primary PID controllers.

2) Sensors: The vehicles sensors constitute the hardware side of the control system. The sensors include: one LORD MicroStrain 3DM-GX4 inertial measurement unit (IMU), two

Point Grey USB 3.0 cameras, one Blue Robotics Bar30 depth sensor, and four Aquarian Audio H1c hydrophones. *Maelstrom* utilizes each sensor to its maximum potential to provide a sufficiently-controlled vehicle with minimal overshoot.

3) Autonomy: In previous RoboSub competitions, teams attempted to improve their autonomy platforms by using state-of-the-art object detection algorithms, such as YOLO. A thorough cost-benefit analysis of YOLOs performance metrics and implementation difficulty over traditional, hard-coded strategies, such as OpenCV [2], was performed. Since YOLO outperforms many alternatives, the team decided to incorporate this advanced algorithm. To prevent processing losses on the mission computer, a NVIDIA Jetson TX1 with an Auvidea J140 carrier board were purchased to handle the graphics-intensive operations. Using MATLABs Ground Truth Labeler app in tandem with a custom MATLAB script accelerated the data formatting process to easily train a neural network with Darknet, the library YOLO is built on.

4) System Complexity: The entire software base, illustrated in Figure 2, is built upon Robot Operating System (ROS) [3] due to its simple structure of nodes, publishers, and subscribers. Using ROS allowed us to focus on higher level software development without having to expend resources on the more tedious developmental aspects, such as memory-sharing and threading. ROS architecture of nodes communicating via messages allows for easy-to-write, modular programming because nodes can be written and deployed independently of one another. Combined with the simple structure of node handling, ROS streamlined the code development process by making, reading, writing, and debugging code straightforward and efficient.

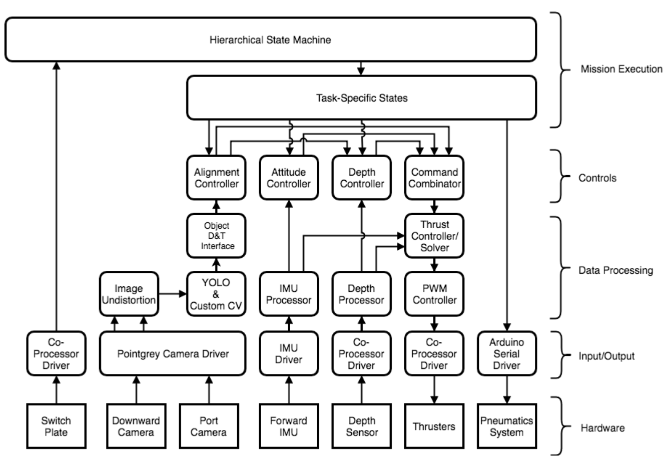

Fig. 2. Software Stack.

B. Electrical

To support *Maelstrom's* advanced software stack, the electrical system ensures that all electrical devices are properly powered without failure. In the past, printed circuit boards (PCBs) were overly complex and prone to failure; during the first three days of last year's competition, more than half of the electrical system was hastily reassembled with just an Arduino Mega. Since then, our aim was to build a well-organized electrical management system that was easy to troubleshoot.

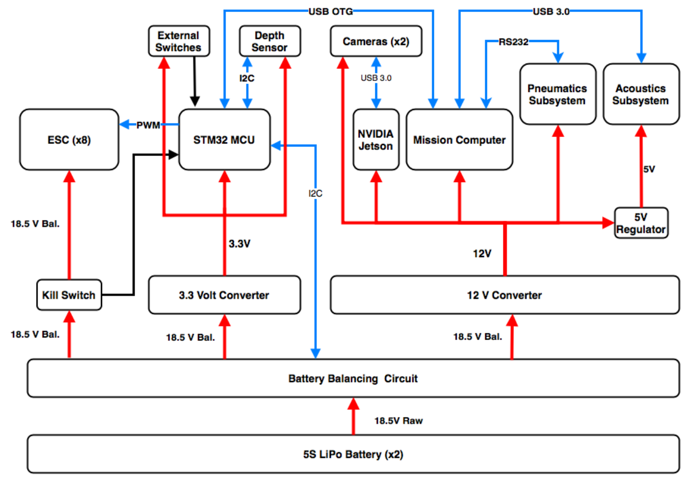

Fig. 3. Electrical Block Diagram.

1) Simplicity: The electrical design prioritized simplicity above all else to facilitate ease of use and debugging. The electrical system was broken down into three main components: battery management, power conversion, and hardware control. Initial board design iterations focused on an uncomplicated composition to verify each board could satisfy its criteria. Further iterations focused on additional features to improve debugging and system monitoring, such as status LED's and temperature, current, and voltage sensors. Figure 3 illustrates the electrical block diagram, which shows both power and signal distribution.

2) Modularity: A backplane constitutes the main electrical system. The backplane holds three vertical PCB's and one horizontal PCB, as displayed in Figure 4. All vertical PCBs are supported by aluminum braces to increase durability. The backplane's edge connectors allow systems to communicate with one another and supplies boards with logic level voltage. For improved reliability, functionality, and speed, we switched from a series of Arduino Mega processors to a single STM32ARM Cortex Microcontroller. The STM32 has increased available pinout and communication protocols over a more open programming platform to enhance system modularity. Regulating all electronic speed controllers (ESCs) with only the STM32 was pivotal to a consistent electrical system.

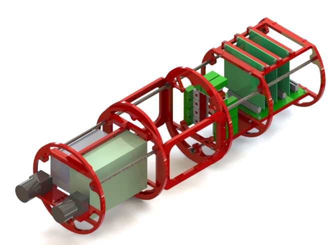

Fig. 4. CAD rendering of the Internal Electronics Mounting.

3) Control of External Components: Although the team has manufactured external subsystems for previous vehicles, *Maelstrom* is the our first AUV with two fully functional external subsystems: pneumatics and acoustics. While the software stack is not built up enough to perform advanced control maneuvers or localization and mapping algorithms, each subsystem is electrically operational. All pneumatic components can be controlled on command, and the acoustics system can relay information to the mission computer regarding the AUVs position relative to a pinger task.

C. Mechanical

Participating in RoboSub has provided us with sufficient knowledge to make both a watertight vehicle and a structurally sound chassis. Entering our third year with this established base, we aimed to create a versatile vehicle with mechanically operational subsystems to support the expanding electrical and software platforms.

1) Chassis Experimentation: Since our AUV's housing has been watertight with minimal concerns, we decided to direct effort towards improving the chassis. Our previous chassis would have required a considerable redesign to accommodate the newly fabricated external housings. Therefore, we chose to experiment with other methods of construction, with the objective of creating a scalable chassis for future years. Based on the success of other student project teams at Ohio States Center for Automotive Research (CAR) with sheet metal, we chose to follow suit for this year's chassis. After assembling an initial version of the chassis with thinner aluminum, necessary modifications were made on the final design to properly support all of the vehicles components. Our aim with a sheet metal chassis was to improve hydrodynamic motion, and to clean up the vehicles constituents.

2) Design Simplification: While past vehicle designs were mechanically rigid, they had unnecessary support beams located in key access areas, resulting in time-consuming maintenance. One of the goals was to design a frame capable of supporting the vehicles components while maximizing access. Our sheet-metal chassis creates an open structure that is rigid enough to handle loads from the thrusters and the frequent removal of our main housing end caps to access internal hardware.

3) External Housings: In previous years, external housings served as weight blocks to maintain vehicle buoyancy. To attain our objective of accomplishing some of the intermediate and advanced tasks, we overhauled older designs to pave way for improved pneumatics and acoustics subsystems. The pneumatics system was redesigned to deliver compressed CO_2 to two torpedo tubes and a marker dropper, with two extra ports for expandability. Figure 5 is a render of the pneumatics housing. Furthermore, the acoustics subsystem was redesigned to incorporate a fourth hydrophone to meet the specifications of our localization algorithm.

II. Design Creativity

Throughout the development of *Maelstrom*, numerous issues were addressed to guarantee success at competition. The nature of each issue, whether it was carried over from last year's vehicle or due to limited resources, encouraged the team to devise innovative and cost-effective solutions with the materials available.

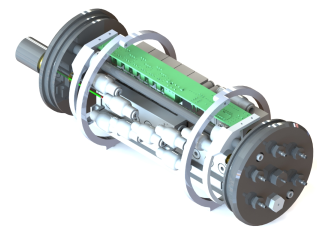

Fig. 5. CAD rendering of Pneumatics subsystem.

A. Software

1) Google's Ceres Solver: While using professional CAD suites like SolidWorks can streamline the design process, using them to obtain exact characteristics about an assembly can be inaccurate due to design uncertainties. The resulting moment from *Maelstrom's* center of buoyancy (CoB) about its center of mass (CoM) is consequential and affects every controller; tuning our PID controllers for precise control maneuvers required an inventive solution to determine the vehicle's unknown CoB relative to its CoM. We already deploy Googles nonlinear Ceres solver to calculate thruster outputs based on *Maelstrom's* six EOMs. For convenience, we configured Ceres to solve a separate set of equations to approximate the location of the CoB relative to the CoM. Utilizing Ceres to our advantage was critical in reducing the time required to adequately tune the control stack.

2) Graphics-Dedicated Computer: With the intent of using YOLO for ML, the team chose to dedicate the intensive graphics computations to a secondary, onboard computer, a NIVIDIA Jetson TX1. Offloading all computer vision and ML computations to a graphics processing unit (GPU) prevented the mission computer from encountering process overloads. Thus, the mission computer was able to focus on the important control and other autonomy algorithms. Since both computers would have to work in tandem towards the same goal, we leveraged ROS ability to connect multiple computers on the same ROS network. We merely indicated to the Jetson that the ROS master server is on the mission computer.

3) Remotely Operated Vehicle: To maximize efficiency at summer pool tests, we developed a ROS node that would temporarily transform our AUV into a remotely operated vehicle (ROV) via a Sony PS3 Controller. Deploying *Maelstrom* as a ROV provided the ability to easily maneuver the vehicle through the course to collect valuable camera footage and to see firsthand how difficult it is to program a robot to autonomously complete the various tasks. Having video footage also enabled us to segment out the objects of interest within each frame for ML training and aid in the development of adequate computer vision algorithms. Furthermore, with ROS features, we could playback video footage to perform

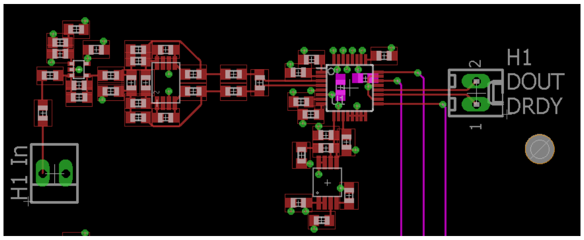

Fig. 6. Board layout of single Acoustics channel.

software-in-the-loop (SIL) simulations and verify the controller outputs.

B. Electrical

1) STM32 Arm Cortex Microcontroller: Switching the hardware controller to an STM32 microcontroller introduced a new set of challenges to the team's hardware/software integration. Previously, communication between the mission computer and ATMEGA utilized ROS for ease of use. However, due to the lack of supported ROS libraries for our specific STM32, we created a custom, robust serial protocol. The serial protocol operates similarly to a ROS node; the mission computer relays information and gathers appropriately packaged sensor data. The custom serial protocol was also implemented in the external subsystems; while using ROS on both processing chips was possible, it was not feasible due to maintenance requirements. Communication with the mission computer over serial also prevents potential issues, as the microcontrollers never have to be reprogrammed.

Because the STM32 manages most of the hardware, communication speed is important. Rather than using a slow, standard firmware loop, we implemented a real time operating system (RTOS) for its multi-threaded processing capability to control different parts of the robot. With RTOS, the microcontroller can simultaneously run the inter-board communications, thrusters, sensors, and serial communications to the mission computer with maximum efficiency.

2) Acoustics: Until this year, we had never attempted to create an acoustics system as the majority of our attention was fixated on creating the main electrical system. With frequent advice from Dave Guidry, an engineer at Texas Instruments, we created a custom digital signal processing (DSP) board to handle all aspects of analog to digital processing. Board design centered around a 512 kSPS ADC chosen for its high sampling rate. The DSP board individually amplifies each H1c hydrophone input and filters the analog signal according to the predefined frequency range for our competition. One of the four channels of this board is shown in Figure 6. Another challenge was making a ZEM5305 (FPGA) control the DSP such that each of the hydrophone signals were sampled simultaneously, the key to solving this task. With properly collected data, a triangulation algorithm was applied to output a relative position and heading relative to a pinger.

3) Debug Lights and Status Board: A priority with the new electrical system was the ability to identify the robot's electrical state and mission status. On each of the three main circuit boards, debug LEDs were added to quickly verify all input, output, and intermediate states. For non-binary statistics, we designed a status board consisting of three 4-digit displays, colored LEDs, and a warning buzzer to indicate the overall status of the robot. The displays relay *Maelstrom's* temperature and each battery's current or voltage. The LEDs indicate the software status and the buzzer activates in extreme cases, such as overheating or low battery voltage.

C. Mechanical

1) Chassis: Our past AUV chassis were based on water-jet quarter-inch aluminum chosen for its rigidity. As aforementioned, *Maelstrom's* chassis is constructed out of sheet metal to leverage the knowledge base of our fellow student project teams and mentors. The advantages of using a two-piece, sheet-metal frame include: increased freedom for component placement and diminished assembly time, as holes no longer had to be machined into oddly shaped aluminum pieces. Once the frame was bent into its final octagonal shape, the entire chassis was assembled over the course of three days. Despite being less rigid than previous designs, this chassis facilitated better placement of components, minimizing maintenance efforts.

2) DVL Plate: The vehicles Doppler velocity logger (DVL) extension was repurposed. The team's first AUV included this main housing extension should we acquire a DVL sensor. Last year, the extension was on the top of the main housing, acting as a switch plate. This year, we rotated the main housing 180 degrees which placed the DVL extension on the bottom. The new vehicle design utilizes this extension to address multiple issues from our past vehicles, including drag due to the large, bulky power cables that bent outward in a U-shape. Rerouting the battery cables through the bottom of the DVL plate allowed the cables to be streamlined relative to the flow field, reducing drag.

The second advantage to the redesigned DVL extension is the relocation of two important sensors: the downward-facing camera and the IMU. On last year's vehicle, the downward-facing camera's field of view pointed through the acrylic end-cap's curved surface, resulting in images too distorted to process. Additionally, the IMU was too close to the forward thrusters, subjecting its magnetometer to considerable electromagnetic interference. To address both problems, the repurposed DVL extension, Figure 7, includes a flat acrylic plate for the downward-facing camera to look through and places the IMU a minimum of ten inches away from each thruster.

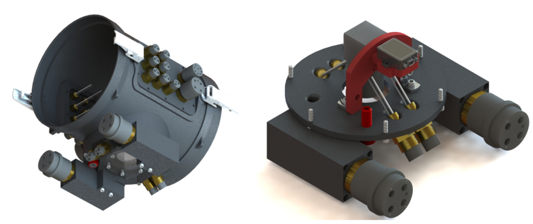

Fig. 7. CAD rendering of DVL Plate Modification. Integrated to main housing on the left, DVL Plate on the right.

3) Switch Panel: An improved switch panel was designed to provide user control of the vehicle at runtime. The switch panel, located on the vehicles aft, contains six switches - one thruster kill switch, and five others for user-defined actions, such as system check, mission start, and computer reset. Reed switches were used to reduce the number of holes machined in the panel to reduce the number of sealing surfaces.

III. EXPERIMENTAL RESULTS

Paradoxically, our team favored both failure and success. Provided our team had yet to design an AUV of *Maelstrom's* breadth, unsuccessful tests provided just as much knowledge as successful ones. At every test, we hoped to not only accomplish something new, but also find every possible manner in which the code base failed so it can be addressed. Like with most engineering designs, failure is not acceptable in the field.

During fall and spring, testing focused on tuning the depth and attitude PID controllers. Unfortunately, due to a misunderstanding of how the control system functioned, many tests were unsuccessful - controllers were incapable of being properly tuned and the vehicle could not attain a specified orientation. Over the summer, we were able to identify and correct all issues in the control stack.

After the school year ended, the team's plan was to test as often as possible, as testing time was deemed directly proportional to success. Because we were behind schedule in assembling the new chassis and needed a fair amount of time to address issues in the control stack, a majority of pool tests emphasized tuning vehicle parameters (mass, volume, and the CoB) and PID parameters. A less than optimum amount of time was focused on writing and testing vision code; the majority of vision testing was performed with collected video footage with limited in-water optimization.

Since making the pneumatics and acoustics housing required intensive efforts, minimal in-water tests were performed. Most experimentation for the two systems were performed separately from the entire vehicle for convenience. The hydrophones were frequently tested out of water along with MATLAB scripts to verify the triangulation algorithm. As for the pneumatics system, preliminary testing was conducted to determine ideal torpedo profiles that could sustain the best trajectory. Once the mechanical, electrical, and software components were completed, the pneumatics system was tested as a separate unit from the rest of the vehicle.

Due to many technical problems, *Maelstrom* was never in a completed state for any pool test. Although minimal testing is not ideal when developing an autonomous vehicle, we are confident in our ability to create a completed AUV ready for competition, once each component functions properly as an individual unit.

IV. ACKNOWLEDGEMENTS

Dave Guidry, from Texas Instruments, Inc. assisted us in designing the acoustics system.

Matt Little, from CAR, assisted us with the mechanical design and manufacturing.

Our sponsors, who generously donated resources and materials. We would like to thank The Ohio State University - College of Engineering, The Ohio State University - Center for Automotive Research, Honda OSU Partnership, Battelle, FedEx Freight, Delphi, Shell, LORD MicroStrain Sensing, Shell, Diamond Systems, B&G Tooling, NHT Global, and Danco Anodizing.

V. REFERENCES

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

[2] OpenCV, www.opencv.org

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

VI. Appendix A

Table 1 lists components used in *Maelstrom*.

VII. Appendix B

A. Outreach Activities

One of UWRT's goals is STEM outreach in the Columbus area. This year, the team had a display at the Ohio State Fair and MakerX. Students explained basic AUV concepts to interested parties and had a remote operated vehicle (ROV) that children were able to drive with a PS3 controller. The team also had a workshop at the Center of Science and Industry (COSI) to educate middle schoolers about underwater robots, as well as presentations to summer camps at the PAST Foundation, both located in Columbus. UWRT also helped students repair their robots at local MATE ROV competition.

TABLE I
AUV Components

Component	Vendor	Model/Type	Specs/QTY	Cost (if new)
Overall Robot	N/A	Made in House		
Waterproof Connectors	MacArtney	Micro Circular		
Thrusters	Blue Robotics	T200	8X, 3-20 V, 25A	
Motor Control	Blue Robotics	Basic ESC	8X, 7-26 V	$25
High Level Control	SMACH			
Battery	MaxAmps	Lithium Polymer	2X, 5S, 18.5V, 150C	
Converter	Cinocn	CQB150W	12V DC/DC Converter	
Regulator	Traco Power		3.3V DC/DC Converter	
CPU	Diamond Systems	Venus	i7-6600U Dual Core	
GPU	NVIDIA	Jeston TX1		
Internal Comm Network	I2C			
External Comm Interface	Ethernet			
Programming Language 1	C++			
Programming Language 2	Python			
Inertial Measurement Unit (IMU)	LORD MicroStrain	3DM-GX4-25		
Depth Sensor	Blue Robotics	Bar30		
Camera(s)	Point Grey	BFLY-U3-132S2C-CS	2x	
Hydrophones	Aquarian Audio	H1C	4X	
Algorithms: Vision	OpenCV			
Algorithms: Acoustics	Triangulation			
Algorithms: Localization and Mapping	"Conceptual" SLAM			
Algorithms: Autonomy	YOLO			
Open Source Software	ROS and OpenCV			
Team Size	17			
HW/SW Expertise Ratio	14/3			
Testing Time: Simulation	4 hours			
Testing Time: In Water	52 hours			