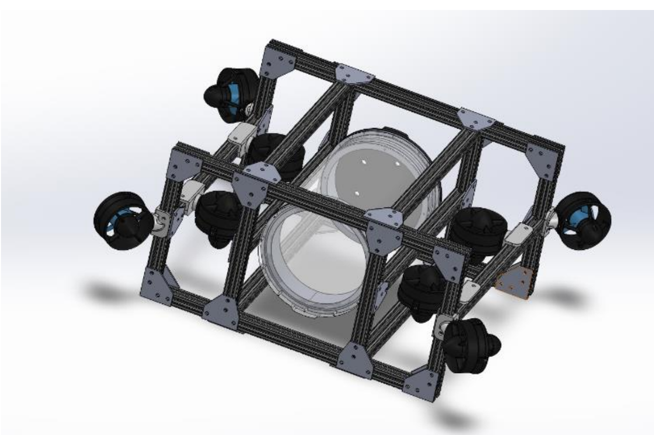
RoboSub 2023 Technical Design Report

Lehigh Underwater Robotics

Sinan Aksu, Yiting Chen, Jessica De Los Santos, Owen Funk, Layne Raczy, Nathaniel Snyder, Xinhao Tang, Kailun Wang

***Abstract*— Lehigh Underwater Robotics is entering RoboSub for the 2nd consecutive year, with a completely rebuilt drone named Albatross. Using the experience we gained last year, we aim to build upon what we learned and make Albatross capable of completing the coin flip, gate, and buoy tasks safely, efficiently, and reliably. By not increasing the scope of the tasks we intend to complete compared to Robosub 2022, we look to improve the reliability of our drone so it may be used in future training. Our other main objective is to design everything to be easy to disconnect which allows us to conveniently test items in isolation as well as assemble and disassemble our drone with ease.**



I. Competition Goals

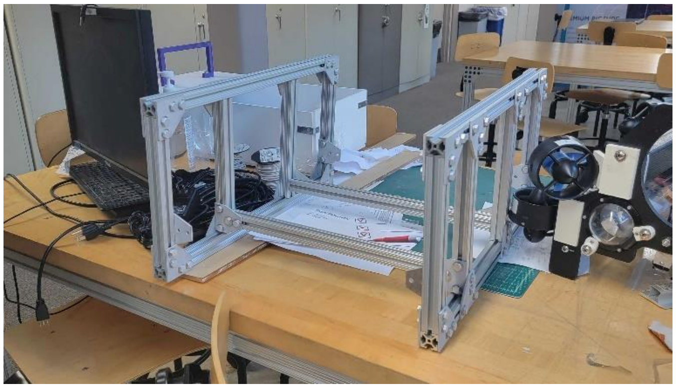
Our team this year, after careful consideration, is focusing on completing the coin flip, gate, and buoy tasks. The coin-flip and buoy tasks are synergistic as our drone will need to utilize computer vision for both of these tasks, which is advantageous as it ensures that our time spent training a sophisticated computer vision model can be leveraged to complete two different tasks. Also, there is a path pointed to the buoy, and since our team opted to omit any acoustic sensors, we can rely on computer vision to navigate to the task. The bin and torpedo tasks would both require adding additional peripherals to our drone, and considering our entire drone is being rebuilt, we decided it would be a safer strategy to try and ensure the core functionality is solid. By focusing on the coin flip, gate, and buoy tasks, we can allocate more time to refining and perfecting these essential capabilities rather than spreading ourselves thin by attempting more tasks. We have considered the trade-offs between system complexity and reliability and made a conscious decision to focus on these specific tasks to ensure a strong and robust performance in the competition.

II. Design Strategy (Mechanical)

The overarching goal of the mechanical team is to facilitate progress within the other two teams by creating easy to use and robust systems and testing environments. Because these elements need to be pool ready before training can begin, we look to provide a long-term drone that can be used for training in future years, so that training will no longer be bottlenecked by the mechanical team.

A. Frame

Due to the structural failures that occurred during the previous competition and our inability to adapt it to our evolving electronics, the mechanical design goal for our 2023 chassis was future-proofing. We wanted to ensure that we would not waste time and resources on basic construction in future years.



B. Buoyancy

To keep malfunctions from sending our drone to the bottom of whatever body of water it is working in, we required that our drone be positively buoyant. On top of this, we focused on controlling the center of buoyancy to be above and in line with the center of gravity. Having the center of buoyancy above the center of gravity creates a 'righting moment' which naturally keeps the drone upright. However, the gate task defines bonuses for acrobatic maneuvers such as barrel rolls and other rotations. In this case, the stability built into the drone harms our ability to successfully roll the drone. We opted to prioritize the stability of the drone and risk increasing the difficulty of the optional maneuvers.

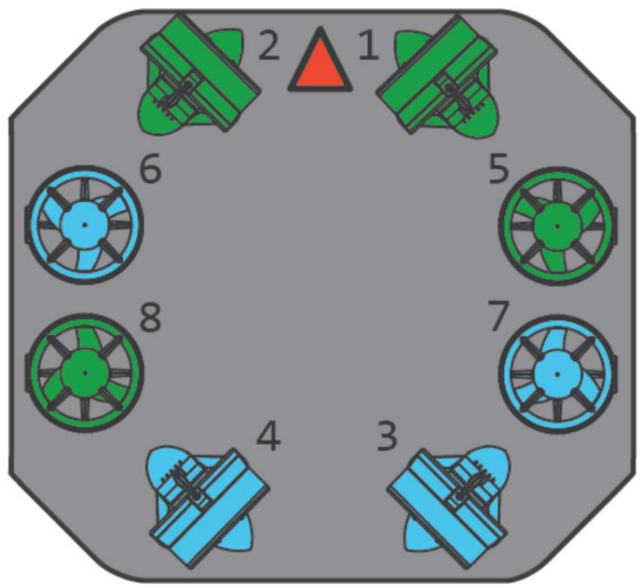
C. Corrosion

Because our overall design goal for this chassis was future-proofing, we worried about the effects of corrosion, specifically Galvanic corrosion, stopping our drone chassis from functioning in future years. Our constraints when solving this problem were the main material being aluminum 8020 struts and the structural integrity of our chassis. Thus, the main design decision was our choice in fasteners. The tradeoff in fasteners was between the strength of stainless steel and the chemical stability of matching aluminum fasteners with the aluminum frame. We reached a compromise between these options, relying on aluminum fasteners in less critical situations.

D. Thruster Configuration

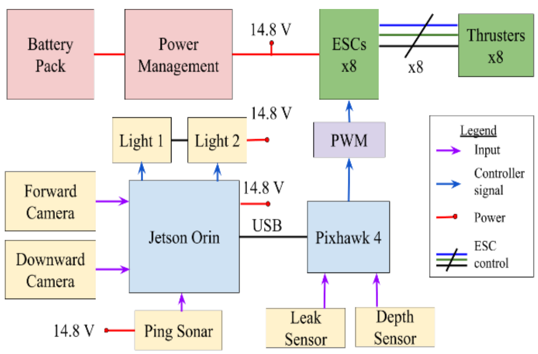
For this competition, we continued using the eight-thruster configuration we had used in the previous competition. This includes four vertical thrusters and four horizontal thrusters, with the horizontal thrusters angled inwards by 45 degrees. This setup allows all four horizontal thrusters to engage during all four cardinal directions of movement. This increases the stability of the drone, as the thrust is spread around the center of mass. The split between vertical and horizontal thrusters keeps vertical and horizontal movement independent of each other, which simplifies movement through the environment. However, this independent movement comes at a cost of resources and space; with the mass of our drone relatively small, 8 thrusters is more than necessary to propel our drone. Because of this, we needed a larger chassis and more wiring to accommodate an 8 thruster configuration. We valued ease of use and reliability over size and complexity due to our drone still being smaller than average.

E. Thruster Layout

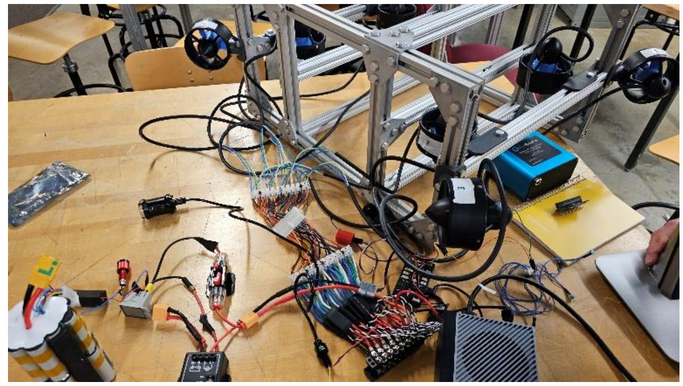


After choosing our thruster configuration, we needed to decide on physical placement of the thrusters on our chassis. Translational movement requires that the center of mass is aligned with the center of thrust, which can be adjusted with weights, but rotational movement has more interesting decisions with tradeoffs. Thrusters far away from the center of mass create large torques with low top rotational speeds, whereas thrusters close to the center of mass can offer higher rotational speeds, but at the cost of responsivity due to the low torques involved. When designing with autonomy in mind, responsivity and slower actions are favorable as they help to stabilize the drone. As such, we positioned the thrusters towards the edges of the drone.

III. Electrical Design



The goal of the electrical team was to provide safe, secure, reliable, and easy to detach connections between all of the major electrical components to create an environment that the software team can use to maneuver our drone. As a second year team we focused on refinement of last year's design: keeping what worked and improving upon what didn't. The majority of the components from last year's drone carried over: we use a Pixhawk 4 as our main flight controller along with 8 T200 thrusters and ESCs. We also equip our drone with lights, cameras, a leak sensor and a depth sensor. Using much of the same overall design allowed us to spend more time improving the cable management within the drone. To do this, we replaced much of the wiring to ensure the proper gauge wire was used for each component, every connection was secured and properly insulated, and that the wires were just the right length to avoid too much slack within the drone. In addition, where possible, connectors were used instead of direct wiring to allow for improved ease of use in the assembly / disassembly of the electrical system.



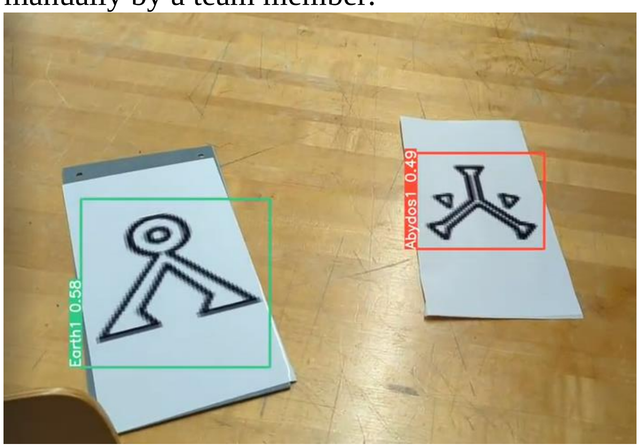
One notable change is the computer powering the system - which was replaced with the more powerful Jetson AGX Orin in order to improve performance The other major component added to the electrical system is a ping sonar, which was added in order to improve our drones maneuverability by allowing our software team to create a 3d map to improve our drones autonomous efficacy. Another electrical component we added was a tether, which was used solely for testing because it gave us a way to connect to a topside computer and control the drone without the drone being fully autonomous. We implemented the tether in a way so that it was easy to add or remove to meet our needs of making our drone autonomous or manually controlled. In our final design, no tether was used as the drone is required to be fully autonomous.

A. Electrical Design Challenges

Originally, the team aimed to revamp the electrical design of the drone by replacing our flight controller with a separate micro-controller. The idea behind this was to have tighter integration of all of our sensors, while also saving space inside of the drone. However, we ultimately decided to focus more on reliability rather than system complexity, and in doing so decided to not use a separate micro-controller and instead use the flight controller that was used last year. Using a separate micro-controller would make the software team's job much more difficult and be a big jump in system complexity, which goes against our overall design strategy of sticking with what is reliable and improving our system complexity little by little.

IV. Software Design

This year, our team is working to implement a more sophisticated computer vision model to improve our drone's performance in several tasks. We investigated several different pieces of software to determine what would provide the best balance of performance and ease of use. Some of the programs and algorithms we looked into include: OpenCV, which would allow for low power consumption object detection, and allow us to capture the depth from our depth cameras [1], YOLO v8 [2] which would provide more robust object detection for our drone at the cost of more required compute power, and utilizing a Simultaneous Localization and Mapping (SLAM) algorithm [3][4], which would allow our drone to determine its location relative to objects in the environment, while also being easy to implement into ROS. In addition to evaluating these different software solutions, we also created a machine learning dataset using a service known as Roboflow, built using footage recorded from a separate underwater camera apparatus operated manually by a team member.



A. Libraries Being Used

Our team is using several pre-existing libraries in order to accomplish the tasks. The primary software being used to control our drone is Robotic Operating System (ROS) 2 [5], which allows us to efficiently link all of our different software and hardware components together. This is in conjunction with MavROS and pyMavLink, which are ROS 2 packages that interface with our flight controller to allow us to easily control our drones thrusters. OpenCV is an open source program we are utilizing both for its object detection capabilities, as well as its built-in function to record depth data from a camera feed. This depth data can then be used to create a map of the environment using the SLAM algorithm, which will improve our drones navigational abilities.

V. Testing Strategy

Our overall testing strategy was to ensure functionality at each step towards the drone’s completion.

1. Without access to a pool, our software team experimented with using the SITL simulator [6], which allows us to emulate the sensors present on our drone, so that the software is ready to be tested on our physical drone when pool time becomes available.
2. When constructing the electrical system, our team made sure to test each component individually outside of the drone in order to ensure they functioned as expected. Part of this also included wiring up all the components outside of the drones enclosure before installation. In addition to this, we tested to ensure thermals were not an issue by running a stress test on our primary computer inside our drone enclosure, while monitoring the temperature using a probe.
3. While the electrical system for our drone was still being tested, we also constructed a separate apparatus, containing a lower end Nvidia Jetson, a battery, and our cameras, contained in a six-inch acrylic tube with 3d printed handles. This apparatus was manipulated by a team member under the water to gather video that can be used to train our computer vision and object detection models, without requiring our drone to be assembled and submerged.

4. For water tests, our drone has an ethernet tether that allows our team to connect to the drone while it is submerged. This allows for faster debugging and development of the drones software, while also allowing us to record sensor data from the drone in real time.
5. Finally, when testing the drone's autonomy, we removed the tether and set up a mock course to see how it would do in a competition-like environment. In order to ensure that our drone performed well at the competition, we targeted 80 percent efficacy for each of the tasks.

ACKNOWLEDGMENTS

Our team's continued success is thanks to the people who supported this project over the last year. Specifically, we would like to thank Lehigh University's ECE department; Professor Rosa Zheng's guidance and with the project; and Bill Maroun for assisting with our technical design.

Additionally, we would like to thank Lehigh's Office of Creative Inquiry's Mountaintop program.

REFERENCES

[1] *OpenCV.* [Online]. Available:https://opencv.org/

[2] *Yolo V8,* Ultralytics. [Online]. Available: https://github.com/ultralytics/ultralytics

[3] Rahman S, Quattrini Li A, Rekleitis I. SVIn2: A multi-sensor fusion-based underwater SLAM system. The International Journal of Robotics Research. 2022;41(11-12):1022-1042. doi:10.1177/02783649221110259

[4] *slam_toolbox,* Steve Macenski. [Online]. Available:https://github.com/SteveMacenski/slam_toolbox

[5] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, W. Woodall, "Robot Operating System 2: Design, architecture, and uses in the wild," Science Robotics vol. 7, May 2022.

[6] *SITL,* Ardupilot project. [Online]. Available:https://www.ardusub.com/resources/downloads.html

Components

Components	Vendor	Model/Type	Specs	Custom/Purchased	Cost	Year of Purchase
ASV Hull Form/Platform	Blue Robotics	8" Watertight Enclosure	8" diameter	Purchased	$461	2022
Waterproof Connectors (9)	Blue Robotics	Wetlink Penetrators	Anodized Aluminum	Purchased	≈ $12	2021
Propulsion (8)	Blue Robotics	T200	5.25 / 4.1 kg f	Purchased	$200	2022
Power System	Blue Robotics	Power Sense Module	Voltage and Current Sensing for Pixhawk	Purchased	$80	2021
Battery	Blue Robotics	Lithium-ion Battery	14.8V, 15.6Ah	Purchased	$330	2021
CPU	Nvidia	Jetson AGX Orin	2048 Cuda Cores, 64 Tensor Cores	Purchased	$1,999	2022
Motor Controls (8)	Blue Robotics	Basic ESC	1900 µs - 1100µs at 400 HZ maximum	Purchased	$36	2022
Flight Controller	Blue Robotics	Pixhawk 4	Accel/Gyro: ICM-20689 Accel/Gyro: BMI055 Magnetometer: IST8310 Barometer: MS5611			
Inertial Measurement Unit (IMU)	Blue Robotics	Pixhawk 4 (see above)				
Camera	Sony	IMX322	1080p 30FPS	Purchased	$63.99	2022
Camera	Sony	IMX 219	3280 × 2464 resolution (per camera)	Purchased	$60	2022
Algorithms		SLAM				
Vision		OpenCV YOLO v8				
Localization and Mapping		SLAM				
Autonomy		ROS 2 MavROS pyMavLink ArduPilot Python C++				
Open-Source Software		ROS2 OpenCV YOLO v8 Python C++				