The Design of Team Inspiration's 2023 AUVs

Eesh Vij (team lead), Ashiria Goel, Colin Szeto, Jerry Hsieh, Jesus Mendoza Cortes, Khushee Goel, Kyle Jacob, Lindsay Wright, Mabel Szeto, Mia Chen, Noah Tang, Parth Saxena, Rishi Veerepalli, Sean Boerhout, William Gao, Isabelle Gunawan

Abstract—In its fourth year in RoboSub, Team Inspiration is embarking on developing multi-agent robotics cooperation capabilities, mastering new sensor use, and incorporating machine learning techniques, and using AI-on-the-edge with smart cameras aiming to complete as many missions as feasible. While preserving Græy's hardware (2021), we designed Onyx (2023) with a larger hull, and using an in-house custom power management Printed Circuit Board (PCB) adding hot-swap batteries capabilities and more power efficient regulatorsenableding a much cleaner electronics assembly for additional capabilities complementing Græy. Building on lessons learned from RobotX, we bolstered our perception capabilities with new hardware and machine learning (ML) algorithms while reorganizing our overall software architecture on which our navigation, mission planner, and new ML capabilities are integrated. Moreover, for 2023 we developed a Robosub introduction guide to ease the learning curve for newRobosub teams.

I. Competition Strategies

Building from our efforts in 2020 and 2021 to create a solid base for future competitions, we continued our strategy to compete with two Autonomous Underwater Vehicles (AUVs) as advantages provided by the rules far outweigh the cons of size constraints and weight penalties. Moreover, we believe that learning multi-agent robotics will open new opportunities in our academic and professional development. Two AUVs complement each other's abilities, enable us to complete more missions within the same amount of time, add fault tolerance at an AUV level, and eliminate the need for a one-sub-do-it-all that presents risks with integration and being a single point of failure.

Our approach is to keep the two AUVs simpler compared to one AUV with all capabilities. By keeping the core design the same as in previous AUVs, we enabled team members to focus efforts into developing new capabilities.

Additionally, communication capabilities between the two AUVs are an advanced capability warranting special points for the competition score. The AUVs will use the same mission planner loaded with two different mission paths. Inter-sub communication can optimize scoring capabilities and save water time in the competition while also providing a fail safe during mission runs: if one sub accidentally surfaces, our run continues per competition rules.

Team Inspiration endeavors to complete every mission. Both AUVs will pass through the gate, spin to earn style points, and follow the orange paths. Græy will hit the buoys, while Onyx will use hydrophones to locate the pinger marking the Octagon and Torpedo tasks; launch torpedoes, drop the team markers, surface in the Octagon, and complete the Chevron tasks. Through inter-sub communication, both AUVs will use the same Image set for missions on Earth or Abydos; Onyx will scan the image and transmit the information to Græy.

This year, we focused on improving the AUV's software system in perception, localization, planning, and navigation. We used ML techniques such as You Only Look Once (YOLO), GPU acceleration, and smart cameras that run embedded code on them with artificial AI-on-the-edge for Computer Vision (CV) to identify the missions and visual targets related to tasks corresponding to Earth and Abydos. Team members tested individual cameras asynchronously prior to integration. To improve localization and navigational accuracy, we added the Blue Robotics Ping360 scanning imaging sonar to Onyx.

We use Robot Operating System (ROS) to enable interprocess communication between the programs for a modular expandable software system. Utilizing ROS with our software baseline has allowed for a more sophisticated structure where programs run simultaneously without low level control for multithreading and inter-process protocol. The end result is that we believe that our codebase is cleaner and data transfers between inner process and across AUVs easier to implement and more robust

More details on our software architecture, including the decision of using ROS, can be found on our website.

We continued to build on the electrical knowledge gained in previous competitions by updating Onyx's electronics and Græy's enclosure harness. We replaced some key elements of the electrical system, such as our power distribution board (adding hot-swap batteries capabilities and more power efficient regulators), and a custom, high precision, hydrophone system. Onyx possesses a clean, organized wiring system, promoting more efficient electrical design.

II. Design Strategy

All subsystems balanced the KISS (Keep it Simple Silly) philosophy against our goal to push our manufacturing capabilities and software understanding. We also considered schedule, cost, performance, and risk factors when developing new capabilities. Onyx's design was largely based on Græy while using more custom parts and actuators beyond Græy's capabilities. Our software architecture and mission planner from previous years were stripped to a baseline, then completely rebuilt into a more versatile system. This allowed for easier integration of our new ML and CV systems.

A. Mechanical

We derived design requirements from the competition handbook and researched previous teams' successful configurations for each system and payload. Weighted decision matrices, evolution documents, and weekly design reviews all reinforced requirement-oriented design. Before manufacturing more complex components, we consulted mentors and industry professionals for best practices. Below are some specific details about Onyx that fulfilled our additional requirements and core philosophy.

Due to the benefits of quick assembly, cost efficiency, and modularity, we continued to use 80/20 extrusions for the base of the frame. We upgraded from an 11.75 in long by 8 in diameter enclosure to a larger 18.25 in long enclosure. With a greater than 55% increase in volume, we can accommodate additional equipment.

Several team members have matriculated into UC San Diego since Team Inspiration's last RoboSub participation in 2021, allowing for access to advanced manufacturing capabilities in the university labs. We manufactured our thruster mounts from CNC routed acetal due to its excellent corrosion and warping resistance.

This was the first year the team integrated the torpedo launcher, gripper, and marker dropper on the sub.

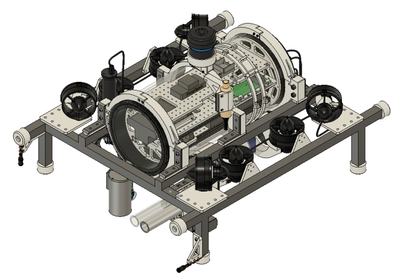

Fig. 1. Onyx Mechanical Design

1) Torpedo Launcher. To quickly learn the mechanics of torpedoes and launchers, we chose a simple spring and latch mechanism. We used Finite Element Analysis (FEA) to determine that the FDM manufactured torpedo housing was capable of handling repeated impulses from firing the torpedoes.

2) Gripper. The gripper uses last year's Newton Subsea ROV gripper with modified jaws, enabling us to manipulate both chevrons and the bin handles. Working with our existing 80/20 frame design, we designed a sliding mount and a magnetic quick release which would deploy the gripper below the sub before each run, saving space and making alignment and acquisition of game elements easier.

3) Marker Dropper. Our requirements for the marker dropper were to use only one servo while reducing the footprint as much as possible. After creating a trade study of previous teams' designs, we chose the Kyutech Kyubic design due to its weight, size, fabrication cost, and reliability. Our design implemented a 2000 Series Dual Mode Servo, control arm, and barrel that holds two spherical markers. We used 3D printing to complete rapid prototyping initiatives.

B. Electrical

While Græy is already a notable improvement from our 2019 AUV, there was more room for growth. Onyx's longer enclosure allowed for better cable management. Sensitive electronics were shielded and placed in locations isolating them from noise.

1) Power Distribution Printed Circuit Board (PCB). Onyx's custom PCB is based on Græy's and unifies the power distribution and conversion functions within one compact board. Combined with a microcontroller, the Unified Power Controller manages hot-swapping between batteries, intelligent error reporting/datalogging, control from external computers, and temperature monitoring. The hot-swapping system gives Onyx over 1.5 hours of runtime on both batteries, and allows us to install new batteries without turning off any part of the sub. Additionally, swapping between batteries allows us to maintain battery health for future tasks.

2) Hydrophone PCB. We integrated our previous hydrophone designs originally planned for Græy into Onyx due to the larger hull capacity. The motherboard and daughter boards are the result of industry best practices that are adapted to work for our team. For instance, we followed standard signal routing requirements and created wire traces with acid traps in mind. The designs were verified in design reviews before being manufactured.

The hydrophone sub-system went through several iterations. We offloaded as much filtering as possible to custom designed hardware, saving resources within the microprocessor. Each hydrophone has its own daughter board containing noise-isolated circuitry. This includes the pre-amp, a custom amp, a custom variable frequency filter, adjustable between 10khz to 40khz, a digital potentiometer for variable gain, and a custom active high-pass filter to ensure the signal is as clean as possible. Signals are analyzed with custom algorithms, utilizing arrival time and signal phase to triangulate a heading for the pinger.

This setup removes the need for computationally expensive Fast Fourier Transforms (FFTs) on the microprocessor, a Teensy 4.1. The Teensy can also change the gain using a digital potentiometer over Serial Peripheral Interface (SPI) to equalize and contain the signal within an analyzable state.

The Teensy transmits pinger heading to the main processor in degrees relative to the hydrophones' positions. The Xavier can also send commands to Teensy over the same serial protocol to select which frequency to locate.

3) Underwater Communication. We built custom software for the Succorfish Delphis modem because of its low latency, high bandwidth (512 bits/s), and long range. We worked with our mentor and sponsor, Alan Kenny from Kenautics, to use these modems.

4) DVL. After continued unit tests and communication with the company, we realized that our WaterLinked DVL had failed and switched to a Teledyne Explorer DVL, provided to us by Alan Kenny.

5) Sonar. We added a Ping360 sonar from Blue Robotics for its increased image scanning capability. Previously we used the one dimensional ping sonar, which only allowed us to receive distance values in the direction the sonar was pointing. The new 360 degree sonar allows us to get a snapshot view of our whole environment so we can analyze where the mission objects are and which direction we should head in.

C. Control Systems/Software

We focused on creating a baseline software platform to allow our algorithms to more easily interface with basic AUV motor control. Previous years' autonomous performances showed that reliable basic movement was key to having a working navigation system and making vision and ML algorithms integrable. We continued to use ROS for our inter-process communication framework as its ability to receive and publish data from multiple different sources makes it modular and scalable.

1) Navigation Algorithms

a) Localization, Perception, and Waypoint Navigation. For AUV localization we use cameras, barometers, hydrophones, Inertial Measurement Unit (IMU), DVL, and image scanning sonar in tandem.

Currently, our navigation algorithms use distance and direction inputs from sensors to reach a target. To prevent overshooting due to inertia, we implemented a universal backstopping function. The function dynamically reacts to the AUVs current thrust vectors, reversing them for the appropriate duration and speed to bring the vehicle to a perfect standstill.

We developed our own Proportional Integral Derivative (PID) controller, which reads pressure from a barometer, to implement depth hold capabilities.

b) Simplified Mission Planning and Inter-Sub Communication. Our planner has a set route of specific missions for each AUV. Intersub communication provides the opportunity for each AUV to be aware of which missions were successful. If one AUV fails, the other will finish its own missions first and then attempt the remaining missions of the lost AUV.

c) Simplified Mission Execution. Each mission is set in sequence per the mission planner. The mission software controls the actions necessary for the robot to get from one mission to another. The appropriate transitory action between missions is determined by the nature of the mission.

2) Perception. Our experience in RobotX exposed us to an extremely diverse array of options to pursue a robust camera-based perception system. Within the subdomain of computer vision (CV), we explored Simultaneous Localization And Mapping (SLAM), stereo-perception, machine- learning, genetic algorithms, context, and feature tracking.

Constrained by processing power, we offloaded our ML processing on Onyx to two internal (forward and downward facing) OAK-D cameras, and one external (payload facing) OAK-D PoE camera with ML capabilities. Both AUVs still feature multiple standard RGB cameras for OpenCV processing, allowing for basic navigation and position tracking.

To track our AUV position, we developed a custom vision-based SLAM package for 2D-space, using IMU data and over 1,000 high-contrast points from the downwards-facing camera. This gave us accurate station keeping and location approximation at low speeds. We are working on using forward and downward cameras to create a 3D fix for the AUV.

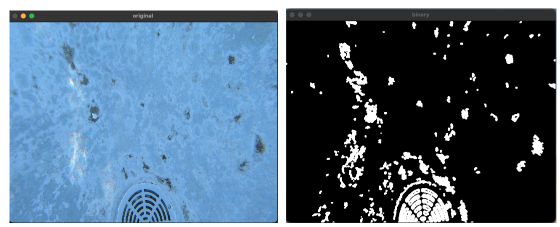

Fig. 2. Original Image vs High Contrast Points

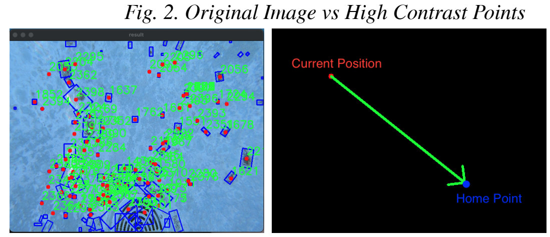

Fig. 3. Points are tracked, creating an estimated position

With various cameras and algorithms running in parallel, we redesigned our camera perception architecture for more efficient data management and integration. Our backend supports automatic camera detection, enumeration, stream handling and forwarding, algorithm/model selection, and data collection.

To identify larger game elements, we used OpenCV and a variety of filters to isolate images within certain parameters, such as hue, as seen in Fig. 4. Additional steps, such as multi-frame context and pattern recognition, were then used for validity detection.

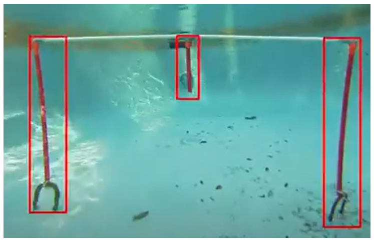

Fig. 4. *The process of OpenCV algorithms outputting bounding boxes with color detection of the gate.*

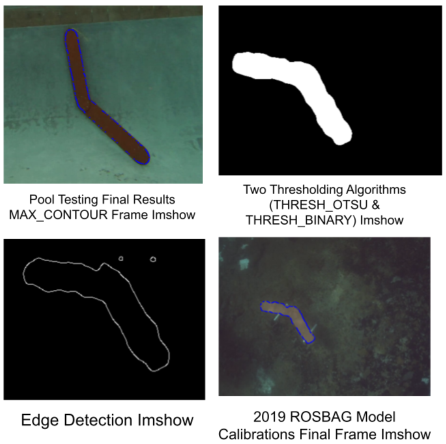

Fig. 5. *The process of OpenCV algorithms outputting edge detection, thresholding, and contour results through HSV color detection of the path.*

III. Testing Strategy

We believe quick prototyping and iterative testing is key to effective learning and integration. At the beginning of the season, we went through our legacy code and sensors to test their capabilities. With a tight schedule, we designed and tested in tandem, scheduling progressive design reviews.

One of the most important parts of our design process was creating detailed test plans (Appendix B). This not only ensures that the tester knows a test's goals and procedures, but also allows us to support data analysis post-execution. We have had many failures with testing components and we have documented each of them on our website. If a test fails, we identify the root cause of the failure and work on fixing and preventing it in the future.

We divided system testing into four key areas: bench and unit tests, CAD/FEA analysis, simulation, and in-pool testing.

A. Bench and Unit Tests

To validate our updated sensors and electronics configuration, we conducted a full bench test before assembling Onyx. We also unit tested individual sensors to collect data without the time and resource overhead of water testing the entire AUV.

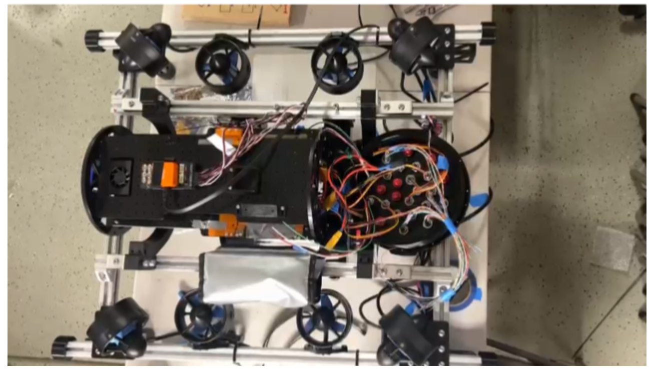

Fig. 6. Unit testing thrusters in final configuration on frame

B. CAD/FEA Tests

Our team makes sure to CAD and review each component of our AUV with mentors before beginning the physical construction. This allows us to iterate in the virtual environment, saving time and resources during the assembly process. Using CAD also allows us to test the durability of each part through stress analysis to understand potential points of failure.

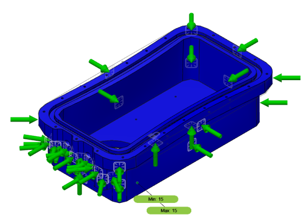

Fig. 7. *A CAD model of our DVL enclosure in a pressure simulation to determine the thinnest wall thickness permissible at 0.1 MPA*

C. Simulated Tests

To facilitate simultaneous development of programs, team members simulated tests on data from previous competitions and on competition images in team members' home setups.

1) Computer Vision. We converted a variety of ROSBAGS from 2019 missions to videos to recreate the competition environment. This data let us fine-tune blurring, thresholding, and morphology sequences. The computer vision code was then integrated onto the AUV, where it went through further tests in the real world.

D. *In-Pool Tests*

We performed various in-pool tests to verify the functionality of our AUVs as we developed each mechanical and software component.

1) Baseline Test. We emphasized the importance of setting up a functional baseline software to work off of, solidifying depth hold, compass turning, and general movement. This eased the integration of other mission software.

E. Modem Tests

We tested many different iterations of modem software to determine the most efficient way to acoustically transfer the data with minimal loss.

The first iteration of this program was a system in which each transmitted packet needed to be acknowledged before proceeding to the next packet. This system was accurate, but yielded an 11 minute transmit time for a 6KB file.

After discussing with mentors, we opted for a modified system in which the acknowledgement is expected after sets of x packets. Then, the packets that were lost are filled in. We also reduced the header size to maximize the amount of information in one packet. After testing values of x for the packet sets before acknowledgement, we were able to get the transmission time down to 2.5 minutes with an x value of 50.

III. Summary

We learned the importance of planning, scheduling, and testing. We learned the value of working closely with team members, vendors and especially mentors. RoboSub data sharing via RoboNation and supporting local RoboSub teams helped us even more.

IV. Acknowledgments

Team Inspiration would like to thank the sponsors below:

Diamond level: Gilman Charitable Fund.

Platinum level: Qualcomm.

Gold level: Northrop Grumman, tinyvision.ai, Kenautics, Leverett Bezanson, Medtronic, and UC San Diego.

Silver level: Blue Trail Engineering, Blue Robotics, Nvidia, ModalAI, Hologic, San Diego Foundation, and HP.

Bronze level: ePlastics, Intuit, Manna's Martial Arts, RJE International, MathWorks, SolidWorks, and JetBrains.

We received engineering support from Aquarian [1] and Blue Trail Engineering [2]. We are grateful to our mentors who help review our work and provide guidance on team management: Jack Silberman, Alex Szeto, Amit Goel, Ravi Iyer, Alan Kenny, Pat McLaughlin, Venkat Rangan, Teresa To, Kenzo Tomitaka, Kevin Bowen, Andrew Raharjo, John Roscoe-Hudson, Debra Kimberling, Cris O'Bryon, and Damon McMillan. We appreciate the RoboSub organizers for putting the competition together to challenge us. We thank Porpoise Robotics for the partnership and opportunity to design low-cost underwater remotely-operated vehicles and their desktop development systems for STEM education. We also thank Team Helix for offering their pool for water testing. Lastly, we are indebted to our parents for their continuous support.

References

[1] Aquarian Audio & Scientific, https://www.aquarianaudio.com (accessed Apr. 15, 2021).

[2] Blue Trail Engineering, https://www.bluetrailengineering.com/ (accessed Apr. 23, 2021).

[3] Team Inspiration, RoboSub 101, https://docs.google.com/document/d/1yuWbiR0lFIkHwIBEXjY4m2Inh_yhhFkF2ClS6DIUXPY/edit?pli=1#heading=h.h6eeeq92ukvw (accessed Jun. 13, 2023)

Appendix A: Component List

Component	Vendor	Model/Type	Specs	Custom/ Purchased	Cost	Year of Purchase
ASV Hull Form/Platform (Frame, Waterproof Housing)	80/20 Inc.	1010 aluminum extrusion	Weight: 25 oz Size: 22 in x 7.9 in	Purchased and machined in lab	$234.73	2019
	ePlastics	Acrylic tube, 8 in. series	Size: 8" diameter	Donated	$343	2019
Waterproof Connectors	Blue Trail Engineering	10 cobalt series dummy plug 10 cobalt series locking sleeve 4 cobalt series cable termination kit	https://www.bluetrailengineering.com/product-page/cobalt-series-dummy-plug https://www.bluetrailengineering.com/product-page/removable-cobalt-locking-sleeve https://www.bluetrailengineering.com/product-page/cable-termination-kit	Purchased Purchased Purchased	$201.75	2020
Propulsion	Blue Robotics	T200 thrusters	Full Throttle FWD/REV Thrust @ Maximum (20 V) https://bluerobotics.com/store/thrusters/t100-t200-thrusters/t200-thruster-r2-rp/	Legacy	$200 apiece	2019
	Blue Robotics	T200 propellers	Max thrust: 49.82 N	Legacy	Included with thrusters	2019
Gripper	Blue Robotics	Newton Subsea Gripper	Grip Force: 28 N Jaw Opening: 2.75 in https://bluerobotics.	Legacy	$329.00	2019

			com/store/rov/bluerov2-accessories/newton-gripper-asm-r2-rp/			
Torpedos	Blue Trail Engineering	Underwater Servo SER-20XX	https://www.bluetrailengineering.com/product-page/underwater-servo-ser-20xx	Custom	$215	2023
Marker Dropper	Blue Trail Engineering	Underwater Servo SER-20XX	https://www.bluetrailengineering.com/product-page/underwater-servo-ser-20xx	Custom	$215	2023
Power System (Battery, Converter, Regulator)	Blue Robotics	Lithium-Ion Battery	4s 14.8V 18Ah	Legacy	Legacy	2019
	Blue Robotics	5V, 6A power supply	5V, 6A	Legacy	Came with BlueROV setup; Legacy	2019
	Mouser	Murata MYBSS054R6EBF	54V Power supply https://tinyurl.com/inspirationRegulator	Purchased	$22.10	2023
	AliExpress	100A Ideal Diode	100A Ideal diode https://tinyurl.com/idealdiodeinspiration	Purchased	2 x $36	2023
	Custom	Unified Power Controller	Three: 12V 3A, 5V 3A, 3.3V 1.2A Power supplies	Custom	$400	2023
Motor Control	Blue Robotics	Basic ESC	30A brushless ESC https://bluerobotics.com/store/thrusters/speed-controllers/besc30-r3/	Legacy	$36 apiece	2019
CPU	Nvidia	Nvidia Jetson Nano	1.4 GHZ clock speed 4 GB RAM	Purchased	$99.00 on Græy	2021

	Nvidia	Nvidia Jetson AGX Xavier	https://developer.nvidia.com/embedded/jetson-agx-xavier-developer-kit	*Donated by a sponsor*	On Onyx	2021
Teleoperation	Blue Robotics	Fathom-X and Fathom-X Tether Interface (FXTI)	Communication: USB 2.0, Ethernet 10/100 https://bluerobotics.com/store/comm-control-power/tether-interface/fathom-x-r1/	Legacy	Installed	2023
Inertial Measurement Unit (IMU)	Pixhawk	Invensense® MPU 6000 3-axis accelerometer/gyroscope	32-bit ARM Cortex M4 core with FPU 168 MHz/256 KB RAM/2 MB Flash 32-bit failsafe co-processor	Legacy	Included with Pixhawk on Græy (compass)	2019, 2022
	Dampener	XTORI Pixhawk dampener	Materials: plastic and rubber Weight: 17 g	Legacy	$7.99	2021
Doppler Velocity Log (DVL)	Teledyne	Explorer	Velocity Range: ±12 m/s Long Term Accuracy: ±0.3% ± 0.2 cm/s https://www.uniquegroup.com/wp-content/uploads/2022/10/Explorer_DVL.pdf	*Loaned by a sponsor*	*Loaned by a sponsor*	2023
Camera(s)	Blue Robotics	Low Light HD USB Camera	Pixel count: 2MP 1080P Onboard H.264 compression chip 32x32mm	Legacy on Græy Purchased for Onyx	$198.00 (1x) on Græy (2x) on Onyx	2019
	Luxonis	OakD-POE	12 MP Resolution 60 FPS max frame rate Focus: AF 8 cm+	Donated	$309.00	2023
		OakD Lite (x2)	13 MP Resolution 35 FPS max frame	Donated	$149.00	2022

			rate Focus: AF 8 cm+			
Hydrophone s	Custom	Hydrophone Circuit	100kHz sample rate for DTOA analysis. Variable gain control. 1-40kHz frequency lock range. Sub-watt and sub-degree precision under ideal conditions.	Legacy	$314.77	2020
	Aquarian Audio & Scientific	AS-1 Hydrophone	Linear range: 1Hz to 100kHz ±2dB Horizontal Directivity (20kHz): ±0.2dB Horizontal Directivity (100kHz): ±1dB Vertical Directivity (20kHz): ±1dB Vertical Directivity (100kHz): +6dB -11dB	Legacy		2019
Algorithms (acoustics)	Custom	Fast Fourier Transform (FFT)	Redundant		Free	2020
Vision	Custom	Open Computer Vision	Color isolation, binary thresholding, contour approximation, erosion and dilation, area thresholding, and Contrast Limited Adaptive Histogram Equalization (CLAHE)		Free/ Open Source	2019
	PTC Inc.	Vuforia/Vuforia License	Vuforia Engine version 8.6		Free	2019
Localization and Mapping	In-house	Custom	DVL, Hydrophones, CV		Free	2021

Mission Planner	In-house	Custom	Mission planner		Free	
Open source software	Open-Source (n/a)	OpenCV, Robot Operating System, Python, C++, Linux	Computer Vision, Inter-process communication, programming, computer operating system		Free	2019

Appendix B: Test Plan & Results

I. Bench Testing

The objective of the bench test is to make sure that ethernet connection between the ground computer and the Jetson Nano works and if servos and all sensors can be used with the flight controller. While servos were tested beforehand, thrusters were also used. Dry operation of thrusters during testing led to vibration and noisy rotations, despite running at low power for short durations. The successful connection between the ground computer and the Jetson, and the movement of all eight thrusters using the Pixhawk Cube confirmed a subpart of the system interconnect diagram.

Fig. 1. Bench testing with T200, Jetson Nano, and Pixhawk Cube

II. Unit Testing

A. *IMU Test*

Working with Alan Kenny, we were able to procure an OEM SBG Systems Ellipse 2A. However, due to a lack of support from the decade-old hardware and a realization the new Pixhawk Orange IMU was satisfactory, we dropped the pursuit and focused on using perception as a localization means instead.

B. *DVL Test*

Our original design included the compact WaterLinked m64 modems. We developed a bench test utilizing a Raspberry Pi and a hobbyist power divider/regulator for power and serial communication. These tests demonstrated that the modems were unreliable and 64 bits per second was insufficient to send images.

C. *Modem Test*

We worked with Alan Kenny, to develop our acoustic file sending process. We began testing by using a USB serial interface with the Delphis modem. The sending side code was run through a Python script on a computer and was received on an Android tablet. This setup was used to verify that we could communicate with the modems, send the packets, and reconstruct the file on the other side. Once this was verified on the tabletop setup, we integrated it into Alan Kenny's DiNIS system to be used by a diver to take a screenshot and send it to the surface via the modems. The code that was ported onto the DiNIS was also ported onto our two AUVs so that they could communicate.

D. *Torpedo Launcher Test*

Our initial design of the torpedo was too buoyant and therefore could not fly straight during our pool tests. To fix this issue we increased the weight of our second model with some improvement in trajectory. We measured accuracy and precision by firing repeatedly at distances: 0.3m, 0.45m, 0.6m, at thirty-three times each. During measurement for 0.3m, the 3D-printed backplate of our launcher shattered from repeated firing, during which the ejected spring became a safety hazard.

E. *Gripper Test*

The gripper was unit tested off the sub to verify its capabilities before integration. Testing was performed using a Teensy 3.1 USB development board powered by a 14V Lipo battery. We observed the gripper's interaction with the Chevron game pieces in multiple possible orientations to confirm that the manipulator was able to reliably pick up pieces. Then, the gripper and the sliding rail were mounted onto the sub to test the magnetic release latch. The angle of the magnetic release latch underwent revision to improve the interface between the magnets in the latch and the gripper in the fully retracted position.

F. Marker Dropper Test

We created dry and water unit tests using an Arduino for performance verification of the dropper mechanism and mounting solution before integrating it with the entire system. We adjusted designs for our servo arm, lid, mounting solution, and barrel based on the results. First, we enlarged our servo arm design after finding it unreliable because the legs of the arm were too short, allowing the markers to push past them and fall prematurely. Secondly, our lid design initially required four screws to access the dropper receptacle, which made for time-consuming reloads. We fixed this by creating a rotating locking mechanism. Next, we found that the bending radius of the wire of our servo was too large to fit under the dropper mount, so we adjusted the mount's position. Finally, after our first water test, we lengthened the barrel of the dropper after realizing that it was too short, causing the marker to drop inaccurately.

Fig. 2. Underwater unit testing the Marker Dropper mechanism

G. Hydrophone Test

A sponsor-furnished RJE International ULB-364/37 Underwater Acoustic Location Beacon was placed in a 2L tub and turned on. Then, hydrophones were submerged in this water tub to test their functionality.

H. Ping360 Test

The Ping360 can be tested for functionality with the Blue Robotics' ping viewer. To integrate the Ping360's data with the AUV's script, we can modify existing code that utilizes the Ping360's communication protocol to fit our needs. Modifications must implement filtering of certain objects such as buoys versus other AUVs. The Ping360 will be mounted on the Onyx and tested underwater to differentiate between a buoy, Græy, and a wall.

I. Thruster Test

To test if the thrusters can run on the Pixhawk Cube, servos were initially connected and run using QGroundControl to avoid dry testing of the T200 thrusters. Further testing involved dry testing of our older T200 models at low power in short durations to verify control of thrusters by the flight controller.

J. Camera Test

The OAK-D Lite and OAK-D PoE are initially tested with a provided test script on PC and later on the Jetson NX. The script helps verify that the cameras are in working condition and can be run with object detection capability. Finally, object detection models are trained and utilize a modified version of the roboflowOak library to run edge models on the cameras. Underwater tests were performed to ensure the accurate detection of objects such as the bins, the gate, etc.

K. Flight Controller and Single Board Computer Test

The Pixhawk Cube was first tested on the Jetson Nano for our drone and Mini-me prototype boat while preparing for RobotX. QGroundControl ran on a ground computer connected to the Nano via Ethernet cable, with remote connection to the Pixhawk Cube via radio. The water and air time spent preparing for RobotX proved that the new flight controller was more accurate than our older Pixhawks, validating our final implementation with the Pixhawk Cube connected to the Jetson NX and the tether from the ground computer to the sub for PoE.

L. Integrated Test Plan

With lessons learned from RobotX, Team Inspiration implemented test plans (example in Table B1) for the integrated Onyx and Græy to plan objectives beforehand, prepare accordingly, record data, and get together afterward for a “hot wash” where team members discuss successes, points that need attention, and lessons learned.

Table B1 Sample Test Plan and Validation Document

Approval Authority	
Eesh Vij (Team captain)	
Colin Szeto (System Lead)	

Date: 2023-06-11	**Mission Title:** Lab Prequalification	
Test #: Græy 1	**Location:** Team Inspiration Lab	**Risk:** Medium
Software Version: Baseline 0	**Hardware Sensors Mounted: Barometer, Camera, IMU**	**Hardware Sensors Used: Barometer, IMU**

Scope: **Primary:** Practice run of Pre Qualification Mission
Secondary: Record full length Prequal Run for submission

Roles		**Walkie Talkies/ Cell Phones**		**Times**		
				Event	**Time**	**Actual**
Test Conductor	Parth	**Ground**	N/A	**Packup**	5:10 pm	5:15 pm
Sub Launcher	Parth & Ashiria	**Pool Deck**	N/A	**Go to Pool**	5:20 pm	5:25 pm
Ground Control	N/A	**Lifeguard**	N/A	**Test**	5:30 pm	5:40 pm
Data Collector	Parth	**Tether**	N/A	**Cleanup**	8:00 pm	8:00 pm
Photographer	Aarrush			**Leave Pool**	8:10 pm	8:10 pm

<table>
<tr><td colspan="2">Safety Checker/QA</td><td>William</td><td></td><td></td><td>Put Away</td><td>8:20 pm</td><td colspan="2">8:20 pm</td></tr>
<tr><td colspan="2">Tether holder</td><td>Ashiria</td><td></td><td></td><td>Hot Wash</td><td>8:30 pm</td><td colspan="2">8:30 pm</td></tr>
<tr><td colspan="9">Status</td></tr>
<tr><td colspan="2">Sub</td><td></td><td colspan="2">GO/NO GO</td><td colspan="4"></td></tr>
<tr><td colspan="2">Ground</td><td></td><td colspan="2">GO/NO GO</td><td colspan="4"></td></tr>
<tr><td colspan="2">Pool Deck</td><td></td><td colspan="2">GO/NO GO</td><td colspan="4"></td></tr>
<tr><td colspan="2">Tether</td><td></td><td colspan="2">GO/NO GO</td><td colspan="4"></td></tr>
<tr><td colspan="2">Attendance</td><td colspan="7">Test Notes</td></tr>
<tr><td colspan="2">Parth
Ashiria
Aarrush
Eesh
Mia
Noah</td><td colspan="7">Resources Needed: Græy with fully charged battery, laptop with QGround control, tether, gate, camera
Environment: pool in the back of the lab, irregular shape that is 6 meters in length with variable depth
Expected Results: Navigates course successfully 15 times
Test procedure:
- Followed AUV set up using the baseline checklist
- Test the keyboard control feature using WASDKL to make sure AUV is able to autonomously move
- Set AUV in start position behind gate
- Run prequal.py
    - One complete working run out of 15
        - AUV started 1.5 meters behind gate
        - AUV moves forward through left side of the gate
        - AUV continues past the gate 4.5 meters
        - AUV turns right to go around marker
        - AUV moves forward 6 meters to go back through the right side of gate
        - AUV Returns to start position
- Followed post-testing checklist from baseline checklist
Risk Management: Our IMU occasionally drifts so we have to have at least one team member (William) to prevent AUV from hitting the wall. Python disarm file is called when code needs to be killed immediately.</td></tr>
<tr><td>#</td><td colspan="3">Validation Step Description</td><td>Expected Result</td><td colspan="3">Reference to Data</td><td>Pass/Fail</td></tr>
</table>

1	Test run prequalify mission in small pool	Pass	C:\\RoboSub\Prequalify_Run\1	Pass
2	Test run prequalify mission in small pool	Pass	C:\\RoboSub\Prequalify_Run\2	Pass
3	Test run prequalify mission in small pool	Pass	C:\\RoboSub\Prequalify_Run\3	Pass
4	Test run prequalify mission in small pool	Pass	C:\\RoboSub\Prequalify_Run\4	Pass
…	…	…	…	…
15	Test run prequalify mission in small pool	Pass	C:\\RoboSub\Prequalify_Run\15	Fail

Results and Reflections

- Compass readings become unreliable after 5+ runs due to combined sensor error
- Fixed heading issues by having AUV not automatically set but by manually setting start heading
- The set distance to go around the marker was too large so the AUV hit the wall; reduced set distance in the code
- Used keyboard control to set AUV back to same starting position after each run
- Get underwater recordings with the gopro
- Need enhance data logging