The Design of Team Inspiration’s 2020 AUVs

Colin Szeto (team lead), Ashiria Goel (deputy), Aditya Mavalankar, Shruti Natala, Ashika Palacharla, Raina Shapur, Pahel Srivastava, Mabel Szeto, Noah Tang, Rishi Veerepalli, and Eesh Vij

Abstract—Team Inspiration focused on perpetually learning and improving. We set out to improve the navigation of our Autonomous Underwater Vehicle (AUV) and expand our mission capabilities. Our second-year team of 12 middle and high schoolers designed our 2 AUVs, named Græy and Orange (a modified version of last year’s AUV), for the 2020 RoboSub competition. In designing Græy, we learned to use Robot Operating System (ROS), hydrophones, simulation, machine learning, Doppler Velocity Log (DVL), modems, and design Power Distribution Boards (PDB). Our 2-AUV strategy allowed learning of intersub communication and having point advantage based on competition rules. After the onset of COVID-19, our team learned effective virtual collaboration, remote configuration, command, control and operation of the AUV.

I. Competition Strategy

Last year, we aimed to be in the top half, and exceeded our expectations by ranking 12th overall and 3rd in static judging. The systems engineering process and K.I.S.S. (keep it simple, silly) worked for us, so we continued following those practices and added the Agile process.

We continued using commercial off-the-shelf products, and custom designed hardware and software to supplement as needed. Last year we used the MAVLink communication protocol. Now, we use ROS framework for interprocess communication, along with MAVROS (a ROS wrapper of MAVLink). With ROS, we can properly integrate input from multiple sensors. To enhance navigation accuracy, we added a DVL and hydrophone capabilities.

Computer Vision (CV) is used to identify the buoys and differentiate whether the tasks correspond to the G-man or Bootlegger. We enhanced our vision capabilities by focusing on determining the position of the image in relation to the AUV. Hydrophones are used to triangulate the pingers that mark the torpedoes and surfacing tasks. A DVL is used to navigate close to the target so the sonar and CV can be more effective to accomplish the task. We use multiple sensors for navigation as fail-safes to minimize the impact of individual sensor failure. Additionally, each sensor, like the DVL [1], has its own internal error correction and filters.

We emulated Orange’s simple and modular construction when designing Græy. The design allows easy expansion by simply increasing the cylinder length and diameter, accommodating for the increased amount of electrical components and the additional ports for the added equipment.

We decided to compete with two AUVs because of the strategic advantages in the rules. For example, using two AUVs nearly doubles our time underwater, as the run only ends when both AUVs have surfaced. Also, only the highest points earned at each attempted task will be counted. In addition, we can gain points through the intersub communication task. Overall, the point benefits of two AUVs far outweigh the cons presented in having weight penalties.

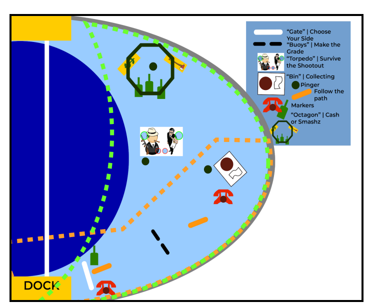

Fig.1. Diagram of the tasks completed by Græy (outlined in green) and Orange (outlined in orange).

Testing is always important to our team. With COVID-19, we were limited in our ability to meet and test in-person. We distributed equipment to team members and utilized family members to assist with testing at home. We set up an environment to collaborate and test remotely resulting in parallel development without disruption. We tested AUV movement on virtual simulations, CV systems on last year videos, and the hydrophones and flight controller on benchtop setups. We integrated different subsystems in Computer Aided Design (CAD) and simulation modeling. We learned how to write detailed Interface Control Documents (ICDs) and established more frequent reviews to ensure clear communication. Each subteam had multiple pivots as they learned more about their respective topics such as neural networks, machine learning, signal processing, PDB design, and kill switch design.

II. Vehicle Design (Novel Aspects)

A. Mechanical

We focused on creating a mechanical design optimized for completing all the tasks. To do so, we used the following design process: identify requirements, design, review, and model. This allowed the team to cross check designs before committing to sponsors for production of parts. This also ensured efficient assembly of the AUV, as we were able to catch design flaws before manufacturing. In making our designs, we utilized trade studies to explore the benefits of each possible design in a quantitative way. Last year's 6" enclosure limited our design, so we expanded to a 30% larger enclosure. We realized the importance of detailed modelling and simulation during the construction of Græy.

For the frame, we continued to use the 80/20 extruded aluminum as it is a robust, simple, modular, and proven design. This design allowed us to shift sensor positions, as all faces of the AUV are made of slotted, adjustable extrusions.

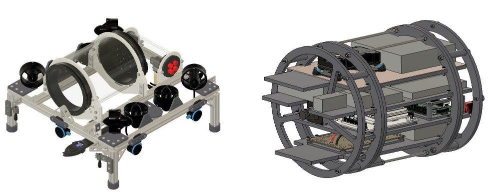

Fig. 2. Græy and its electronics board model.

B. Electrical

While Orange met the basic requirements from last year, we fixed some lingering issues with our electronics, and expanded our abilities.

We suspected electrical and electromagnetic interference is what caused our compass to be inaccurate last year, as the heading would drift up to 0.5º per second after moving, which severely affected Orange's ability to move reliably. Thus, we focused on creating a cohesive electronics board that would minimize electromagnetic interference. This included twisting wires [2], utilizing a copper shielding for all of our more sensitive components [3], and laying out the board so that the noisier components were separated from more sensitive ones [4]. We learned the importance of thoughtful planning and constant communication when designing the electronics board with the mechanical team.

We experimented with different board designs in CAD and put all of them in a trade study to determine which one fit our requirements the best. We chose to implement a tiered board design as it provided ease of maintenance and the greatest surface area. We got a 316% increase in surface area over last year to accommodate additional components.

We also collaborated with electrical engineering students from UCSD to create a custom PDB [5] to replace the exposed power terminals previously used. While the PDB took up more surface area, it reduced the amount of wires, consolidated the connection points, and shielded the terminals. We also made our PDB available publicly on our website [6] for other teams to reference and use.

We focused on improving the killswitch as last year's magnetic limit switch had a 3-5 second latency when triggered. Through research, we decided to create a switch that utilized a Metal Oxide Silicon Field Effect Transistor (MOSFET). In order to protect the AUV's software from a hard shutdown, the MOSFET killswitch only cuts power to the thrusters to immobilize our AUVs. The AUVs can be stopped by a diver with a latency of under

0.2 seconds or by the software if it detects a malfunction for self-protection.

C. Computer Vision (CV)

We explored three different methods of computer vision, taking advantage of each of their strengths: Vuforia for image recognition, along with Open Computer Vision (OpenCV) and neural networks for identification of objects with a distinct shape (e.g. gate, buoy).

1) Vuforia: Seeing the prevalence of image recognition this year, we decided to leverage our experience with Vuforia, an industrial-strength image recognition library, to identify the relative position of the RoboSub task targets. We used our previous Android-based Vuforia code for prototyping and tested with live video captured on-land and underwater. To easily integrate the Vuforia system with our AUVs, we ported our code to emteria.OS [7] for running Android on a Raspberry Pi. We modified event loops and hardware maps to obtain video feed and coordinates of images seen through a webcam plugged-in to the Raspberry Pi. The Vuforia program output was sent to the navigation program running on the Jetson Nano via Android Debug Bridge (ADB). Initially, we tried interprocess communication through a shared file, but realized that ADB logcat would be a more effective way to receive data, as it avoids the file I/O latency.

2) OpenCV: We identified that OpenCV would be effective in detecting objects with contours, like a gate or buoy. We used a series of algorithms (listed in Appendix A) in order to isolate the coordinates of objects like the gate. Furthermore, we used bounding box coordinates to locate tasks and recycled the same program for each task, only changing the Hue Saturation Value (HSV) parameter. We used HSV values as they work best in irregular lighting [8]. We applied failsafes to guarantee maximum reliability, including a size threshold of 10 pixels^2 ensuring that we only target recognized tasks.

We used pan-angle detection to aid navigation in aligning to tasks. Knowing that rectangles distort at different angles, we used this information to calculate the angle between the camera and the object. We verified our calculated result with pixel lengths of edges.

3) Neural Networks: We experimented with Artificial Neural Networks (ANNs) to detect objects. Since we never utilized an ANN before, we decided to create a basic ANN to determine whether or not there was a task target in the frame, and extract the region of interest. We used underwater recordings, sliced into thousands of frames, to collect data and preprocessed the images using filters mentioned in the OpenCV section. Each image was then made accessible by the Convolutional Neural Network (CNN) model. Through testing, we decided that a CNN would be the most accurate model to use.

Fig. 3. OpenCV processing a frame of the gate; Vuforia identifying features of the bootlegger.

D. Software and Navigation

1) Navigation Software Architecture: Our ability to navigate in an underwater environment and complete tasks autonomously required several processes, as seen in Fig. 4, to work in a streamlined and consistent manner. Each process employs various algorithms, detailed later in the paper, to complete the planned task. The processes exchange data with each other over ROS, via the publisher-subscriber model.

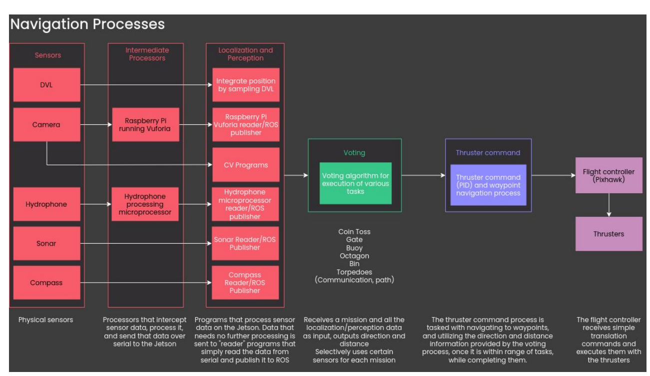

Fig. 4. Navigation software architecture

2) Navigation Algorithms

a) Localization: Græy has several independent ways to estimate its position in the pool. These include: a process that estimates the location of Græy using a DVL, computer vision programs that output AUVs' direction and distance to visible tasks, a hydrophone program that outputs the direction of the pinger, a compass for heading, and sonar for distance.

The default method of localization is the DVL, which samples velocity over an interval of time and numerically integrates the value every 1kHz to find displacement. Erroneous velocity values that are impossible or unlikely are filtered out. For samples when no values are received, the program extrapolates until a real value is received. Additionally, we use sonar to identify the relative position of the AUV to tasks like the buoy or torpedo in which the reliability will be high as they are distinct, wide targets.

b) Voting: The voting process reads the position estimates from the localization algorithm, and acts as a failsafe if one of them fails. The voting algorithm chooses sensor information it deems most appropriate at the time, and sends it to the thruster command process. If CV cannot see any tasks, the voting algorithm will default to using the DVL as a source of position information. Otherwise, it will cross reference all localization data with each other to make sure that everything aligns. If there is a disparity between CV and the DVL, the algorithm applies heuristics to identify potential problems with either sensor. The voting algorithm also calculates the angle between the last few DVL positions and compares it with the heading from the compass and the hydrophones.

c) Thruster Command: The thruster command process aims to move the AUV to a desired waypoint. Using the localization algorithm, the process first tells the AUV to orient itself towards the target using the compass. Then, it uses the position from the voting algorithm and moves until the AUV is estimated to be within 0.1m of the target. If our estimated coordinates of a task are incorrect, we can use sensor input to dynamically realign and complete the task. The thruster command process then moves on and repeats this process for a new task.

E. Intersub Communication

We created a trade study to narrow down which methods of communication would be easiest for us to start learning. We determined that acoustic modems would be the best course of action due to their reliability and optimization for underwater communication purposes. They also provided a well-known base to learn about underwater communication. The demo scenario is to send data between the two AUVs and complete synchronized yaw maneuvers after data is transmitted. In the future, the data from the modems can be used in the voting algorithm to optimize our time in the water by avoiding unnecessary task overlap.

F. Hydrophones

Since several tasks involve localization to a pinger, we realized precisely identifying audio sources underwater would be essential. In testing, we found that a greater distance between hydrophones maximizes the precision we get when attempting to locate the sound source. Thus, we placed three hydrophones around Græy in the vertices of the largest equilateral triangle that would fit in our design. We used a Teensy microprocessor for our hydrophone calculations to offload the strain of processing from our main control board. By sampling at nearly 200kHz at a 12bit resolution, we have nearly sub-degree precision without sacrificing the clarity of the signal received. To maximize the efficiency of the microprocessor, the signal goes through a series of custom bandpass filters to remove any unwanted frequencies so the microprocessor can focus solely on the signal emitted by the pingers rather than undesired noise. The signal passes through a custom low noise amplifier built with a NE5532 chip and with a digitally adjustable gain allowing the microprocessor to adjust as needed. The direction of the sound source is calculated by using time of arrival and triangulation. Once the microprocessor calculates the direction of the sound relative to the AUV and confirms it is an acceptable frequency through a Fast Fourier Transform [9], it sends the data via serial interface to the main control board.

III. Experimental Results

A. Unit Tests

To accurately simulate the performance of our subsystems in a competition setting, amid social distancing, we ran CV tests on underwater footage from last year and the target images from this year. In OpenCV, we achieved an error of ≤5º in the angle between the camera and object. For hydrophones, we tested our theories with three standard microphones, with frequencies in the audible range to learn how it functions. We then recreated the setup with our hydrophones in the assigned RoboSub frequency and a bandpass filter. In testing, we found our hydrophone's algorithm has an error of 3-5º. We also tested our compass in an isolated environment using the built-in compass in the Pixhawk flight controller observing a drift of 0.5º per second.

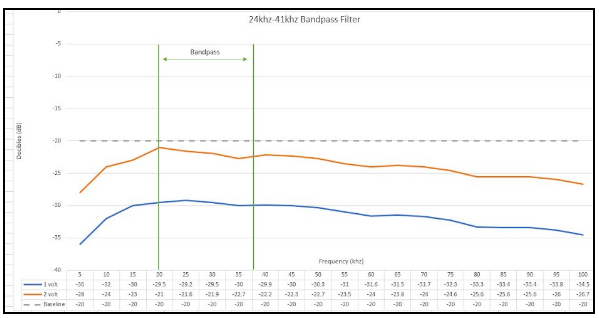

Fig. 5: Testing the dB output of a second-order bandpass filter at various frequencies from the hydrophones. The area within the green lines is the competition frequency range.

B. Thruster Command, Navigation Testing

Our thruster command program receives all the data from the sensors to orchestrate movement. We simulated our flight controller alongside a 3D model of an AUV in the Gazebo robot simulator. This simulation setup accounted for many of the sensors on our AUVs, giving us position and velocity data.

We were able to perform a range of tests using this setup. When we tested our DVL localization program, we used the velocity from the simulation and added an error value of ±0.001 m/s, the published margin of error of our DVL, to make it more realistic. When testing, we compared the computed position with the actual position of the virtual AUV to verify our results.

Our localization system also depends on data from hardware like the camera and hydrophones, which are not included in the simulator. We tested our thruster command program by writing programs to feed hypothetical data to the AUVs. We simulate CV by calculating if an object is in our line-of-sight, and provide relative positioning if it is. We also calculated the angle between the AUV and the pinger coordinate for the hydrophones. We observed how the virtual AUVs responded to the data, and if it behaved as expected, the trial was considered successful.

Although we implemented the capabilities to simulate vision and hydrophones without real hardware, we were unable to collect enough data from trials that incorporated vision and hydrophones. However, we were able to translate to any specified position, and specified a tolerance margin of 0.1m. In the span of 20 trials, the program to move to a desired position only failed once, giving us a 95% reliability.

Acknowledgements

Team Inspiration would like to thank the following sponsors: Northrop Grumman, Qualcomm, Water Linked, Medtronics, Forecast 3D, and Roboctopi. We received engineering support from Water Linked [10], Aquarian [11], and RoboSub teams from Carnegie Mellon [12] and National University of Singapore [13]. We are grateful to our mentors: Jack Silberman, Alex Szeto, Amit Goel, Pat McLaughlin, Venkat Rangan, Teresa To, Kris Chopper, Kenzo Tomitaka, Rory Miller, Kevin Bowen, Andrew Raharjo, and Cris O'Bryon. We appreciate 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. Lastly, we are indebted to our parents for their continuous support.

References

[1] Water Linked, “Waterlinked WL-11007 DVL A50,” Aug. 3, 2020, https://waterlinked.com/wp-content/uploads/2020/08/W-MK-20001-3_DVL_A50_Datasheet.pdf (accessed Aug. 6, 2020).

[2] T. Harada and H. Sasaki, “Printed circuit board for prevention of unintentional electromagnetic interference,” US Patent 5,966,294, Oct. 12, 1999.

[3] “What Materials Are Used for Electromagnetic Shielding?,” Jan. 15, 2018, https://matmatch.com/blog/materials-used-for-electromagnetic-shielding/ (accessed March 7, 2020).

[4] A. Ruffino, “Proven Robotics 2018 Technical Documentation,” Purdue University, West Lafayette, Indiana, USA, Aug. 2018.

[5] J. T. Salcedo, D. Renfrow, K. Alrabiah, M. Odaymat, and J. Silberman, “ECE 191 Capstone Project,” University of California San Diego, San Diego, California, USA, June 10, 2020.

[6] Team Inspiration, https://team11128.wixsite.com/main (accessed Jun. 2020).

[7] Emteria, https://emteria.com (accessed May 9, 2020).

[8] Caldera International, “The RGB and HSV color models,” Feb. 11, 2003, http://osr507doc.xinuos.com/en/GECG/X_Color_AbRGBandHSV.html (accessed Feb. 8, 2020).

[9] E. Condes, “arduinoFFT,” Jan. 26, 2020, https://github.com/kosme/arduinoFFT (accessed Apr. 7, 2020).

[10] Water Linked, https://waterlinked.com (accessed May 7, 2020).

[11] Aquarian Audio & Scientific, https://www.aquarianaudio.com (accessed Apr. 23, 2020).

[12] E. Wang, “Tartan Autonomous Underwater Vehicle Design and Implementation of TAUV-19: Albatross,” Aug. 2019, https://robonation.org/app/uploads/sites/4/2019/10/CarnegieMellon_RS19_TDR.pdf (accessed Apr. 4, 2020).

[13] H. Huan, “Design and Implementation of BumbleBee AUV8,” Aug. 2019, https://robonation.org/app/uploads/sites/4/2019/10/NUS_RS15_Paper.pdf (accessed Mar. 7, 2020).

Appendix A: Component Specifications

Component	Vendor	Model/Type	Specs	Cost (if new)
Buoyancy Control	Blue Robotics	LAST-A-FOAM® R3318 2. Stainless Steel Ballast Weight 3. 2” series enclosure, 6” series enclosure	1.9in x 3.4in x 4.9in 2. (200 g, 7 oz) 3. Refer to Waterproof housing specs	Legacy
Frame	80/20 Inc.	1010 aluminum extrusion	Weight: Size:	$234.73
Waterproof Housing	Blue Robotics	Acrylic tube, 8 in. series	Size: 8” diameter	$343
Thrusters	Blue Robotics	T200 thrusters	Full Throttle FWD/REV Thrust @ Maximum (20 V)	Legacy on Orange $1,432 on Græy (For 8x T200 thrusters)
Motor Control	Blue Robotics	Basic ESC	30A brushless ESC	Legacy on Orange $216 (For 8x basic ESCs)
High Level Control	Blue Robotics	Fathom-X and Fathom-X Tether Interface (FXTI)	Communication: USB 2.0, Ethernet 10/100	Legacy on Orange $169.00 on Græy
Actuators	Blue Robotics	Newton Subsea Gripper	Grip Force: 28 N Jaw Opening: 2.75 in	Legacy
Propellers	Blue Robotics	T200 propellers	Max thrust: 49.82 N	Included with thrusters
Battery	Blue Robotics	Lithium-Ion Battery	4s 14.8V 18Ah	Legacy

Converter	Blue Robotics	i^2C level converter	Operating Voltage 3.3v - 5v Max Current @ 3.3v Vout 150 mA Output Connector 4 pin 0.1” header Input Connector 4 pin 0.1” header + JST-GH + DF13	$15.00
Regulator	Blue Robotics	5V, 6A power supply	5V, 6A	Legacy
CPU	Nvidia	Nvidia Jetson Nano	1.4 GHZ clock speed 4 GB RAM	Legacy on Orange $99.00 on Græy
Internal Comm Network	ROS	ROS 1/Kinetic	N/A	Free
External Comm Interface	N750 Wireless Dual Band Gigabit Router	TL-WDR4300	Simultaneous 2.4GHz 300Mbps and 5GHz 450Mbps connections for 750Mbps of total available bandwidth	Legacy
Programming Language 1	C++	N/A	N/A	Free
Programming Language 2	Python	Python 3	N/A	Free
Compass	Pixhawk	ST Micro LSM303D 3- axis 14-bit accelerometer / magnetometer	3-DOF Magnetometer (on the Pixhawk)	Legacy on Orange $139.00 on Græy
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 on Orange Included with Pixhawk on Græy (compass)
Depth Sensor	Blue	Bar30	Operating depth: 300m	Legacy on

	Robotics	High-Resolution 300m Depth/Pressure Sensor	Supply voltage: 2.5-5.5 volts	Orange $72.00 on Græy
Supporting hardware for Pixhawk	Dampener	XTORI Pixhawk dampener	Materials: plastic and rubber Weight: 17g	$7.99
Doppler Velocity Log (DVL)	Waterlinked	A50	Transducer frequency : 1 MHz Transducer setup : 4-beam convex Janus array Transducer beam angle : 22.5 degrees Ping rate : 4-26 Hz (adaptive to altitude) Sensor assist: Integrated AHRS/IMU (Yost Labs TSS-NANO) Min altitude: 5 cm Max altitude: 50 meters (> 35 m is dependent on seabed conditions, salinity levels etc.) Max velocity: 2.6 m/s Velocity resolution: 0.01 mm/s Long term accuracy: ±1.01 % or ±0.1 % (Performance version)	$7,000.00
Modems	Waterlinked	Modem M64	- Two-way communication -64 bits/second net data link, both ways -Latency: 1.5 – 2.5 sec -Range: 200 meter	$3,000 (2 modems)
Camera(s)	Blue Robotics	Low Light HD USB Camera	Pixel count: 2MP 1080P Onboard H.264 compression chip 32x32mm	Legacy on Orange $198.00 (x2) on Græy

Sonar	Blue Robotics	Ping Sonar Altimeter and Echosounder	- 30m distance @ 15cm resolution - 300m depth rating	Legacy on Orange $251.10 on Græy
Additional hardware to support the Hydrophones		Teesny 4.0 Aquarian PA4 preamp Custom NE5532 Op-Amp	Sample Rate: 200kHz, 12bit Gain: 50db or 0.005% THD @1kHz Gain: 21db or 0.002% THD @ 1kHz	$314.77
Hydrophones	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
Algorithms: Computer Vision, OpenCV	Custom	OpenCV	Color isolation, contour approximation, erosions, dilations, gaussian blur, area thresholding, Contrast Limited Adaptive Histogram Equalization (CLAHE), and 2D convolution,	Free
Algorithms: Computer Vision, Vuforia	PTC Inc.	Vuforia/Vuforia License	Vuforia Engine version 8.6	Free
Algorithms: acoustics	In-house	Custom	Fast Fourier Transform (FFT)	Free
Algorithms: localization and mapping	In-house	Custom	DVL, Hydrophones, CV	Free
Algorithms: autonomy	In-house	Custom	Voting algorithm	Free
Open source software	Open-Source (n/a)	Open Computer Vision, Robot Operating System,	Computer Vision, Inter-process communication,	Free

		Python, C++, Linux	programming, computer operating system	
Team size (number of people)	12	Student	Middle and high schoolers	Priceless
HW/SW expertise ratio	9:7	Hardware subteams: mechanical, electrical, payload Software subteams: software architecture, navigation, computer vision, payload	Colin, Noah, Ashika, Shruti, Pahel, Eesh, Mabel, Raina, Shreyas Aditya, Rishi, Ashiria, Pahel, Shruti, Ashika, Eesh	Priceless
Testing time: simulation	25 hours	Software subteams: software architecture, navigation, computer vision	Aditya, Rishi, Ashiria, Pahel, Shruti, Ashika, Eesh	N/A
Testing time: component testing	90 hours	Hydrophones, OpenCV, Neural Networks, Vuforia	Eesh, Shruti, Pahel, Ashiria	N/A

Appendix B: Outreach Activities

On Team Inspiration, we put an emphasis on sharing what we learn with our local and global community. We truly believe robotics should be shared with the world, especially to those who are less fortunate or lack access to it. In the past year, we have mentored over 25 robotics teams and organized/hosted robotics scrimmages and competitions locally and globally. We have created sustainable robotics programs in over 4 different countries, taught robotics camps and classes, and reached out to youngsters at science museums and all of the major San Diego STEM fairs. In the past year alone, Team Inspiration has participated in over 1,500 hours of community outreach and service. To spread our knowledge of RoboSub, we have brought our AUV to many different public and private events, sharing our AUV and our RoboSub team's journey with thousands of people.

This season, we grew our existing STEM outreach programs and expanded to new horizons. Last August, three of our team members travelled to Benin, a West African country, to co-host Benin's first robotics competition. There, we met with government figures like the U.S. Ambassador and the Beninese Ministers of Education, Technology, and Economics Development. As a result of our continued involvement, the government will integrate STEM education into all of the schools in the country.

Fig. 6. Team Inspiration and Team Spyder meeting with the U.S. Ambassador in Benin.

We replicated our Benin's success in Paraguay, a South American country. In November 2019, we co-hosted a robotics competition with National University of Asunción. We met with the chancellor and dean of engineering to plan engagement to form a robotics region. We also got to share details about RoboSub to the university students.

Additionally, we put an emphasis on learning from industry professionals. We consulted with industry experts in addition to those outlined in the Acknowledgements from different companies, like Northrop Grumman, Amazon, and Qualcomm. Last December, we had the privilege of sharing our team's journey with the Chairman, CEO, and President of Northrop Grumman. We presented our team's successful application of systems engineering process in RoboSub at Northrop Grumman and at an International Council on Systems Engineering's conference.

We collaborate with local universities. For example, we worked on designing a custom PDB with electrical engineering students from University of California San Diego. Additionally, we reached out to a local robotics team, Team Roboctopi, to utilize their laser cutter and access to delrin, a durable material, to make parts of our AUV.

We share our knowledge with others. Nearly all information about our team, robots, program, and organization is published on our website, Instagram, GitHub, and other relevant social media websites. Additionally, prior to social distancing mandates, we hosted monthly open houses to share our knowledge and lab with anyone in San Diego (along with some good food!).

We strive to apply our experience in RoboSub to benefit the local and global STEM education community. Thus, we began collaborating with Porpoise Robotics, a team of former submarine engineers, to design low cost underwater remotely-operated vehicles (ROV) so it can be more accessible. The ROV is designed to be an affordable robot which can be used to teach STEM, especially robotics and mechatronics in schools. Sitting well within a consumer price range, the ROV gives schools the opportunity to immerse their students in underwater robotics. In addition to developing the ROV, we have been working with Porpoise Robotics to create a 2-week curriculum for schools to use with the ROV which will aim to teach students Python, underwater mechanics, and electrical integration.

The Porpoise Robotics underwater ROV comprises of a Raspberry Pi computer, LiPo Battery, servo driver, four Electronic Speed Controllers (ESCs) and Thrusters, camera and external lights, all connected via a 30-meter ethernet tether to a computer and joystick.

Fig. 7. Prototype ROVs developed in collaboration with Porpoise Robotics.

Porpoise Robotics's STEM courses reach Mira Costa College Girls STEM Camp; USC Upward Bound, UC Riverside University STEM Academy; Ocean Discovery Institute Computer Controlled Ground Vehicles; Rancho Minerva Middle School After School Robotics; Coral Reef Research Foundation, Palau; Micronesia Intro to Mechatronics.