Amador Valley High School Robotics (AVBotz): Design of Nemo AUV 2022

Edward Ding (President); Mechanical: Sri Parasaram (VP), Aayush Gupta, Dylan Kwong, Isabelle Lo, Ishan Duriseti, Jacob Kadari, Kaustubh Lole, Meghana Kumar, Nathan Lin, Nikhil Bhargav, Rohan Panjikkaran, and Tejas Chugh; Electrical: Justin Yu (VP), Aarav Jain, David Li, Eaton Huang, Jessie Chan, Richard Bai, Sandilya Garimella, Tanusha Kolli, and Zachary Cheng; Software: Craig Wang (VP), Aarav Singh, Andrew Xiao, Angad Bhargav, Gregory Li, Jason Wu, Kalyan Sriram, Kush Nayak, Michael Li, Nikhil Sunkad, Ryan Yang, Serena Zhou, and Vincent Wang; Business: Leo Shao (VP), Ayushi Deokule, Declan McPherson, Ella Guan, Evan Tian, Gyan Bhambhani, Sarah Zhang, and Weston Zong

Abstract—**This year, the AVBotz team focused on refining and completing our systems for Nemo AUV, while developing a legacy Marlin AUV as a backup. Underscoring accuracy, simplicity, and efficiency, the mechanical team redesigned the hull to improve symmetry and stability, the electrical team focused on refreshing the wiring and components of the rack to ameliorate maintainability of sub components, and the software team upgraded the software system overall with vision system updates to run with more speed and efficiency and simulator updates to reflect competition conditions. With these changes, the AUV is sleeker, lighter, and faster, to run tasks with increased efficiency and speed.**

I. COMPETITION STRATEGY

In the latest in-person competition, our task accuracy was low due to untested mission code, a flawed hull, and convoluted electrical wiring. Based on this experience, we chose to focus on completing a smaller subset of tasks to the best of our ability.

To increase task accuracy, a top priority this year was to finish the construction and waterproofing of our new AUV, Nemo, to fix the shortcoming of our previous sub, Marlin. In previous competitions, because of Marlin's asymmetrical hull, the AUV veered off course when traveling at high speeds; thus, we aimed to finish Nemo this year, with a redesigned hull to increase symmetry and stability for tasks.

Because a significant amount of competition tasks rely on computer vision to detect the object and direct the submarine towards the object, we decided to focus our efforts on improving this central system to improve accuracy and speed for all tasks by extension. Using our machine learning (ML) models, we can detect competition objects through the sub's front and down cameras, generating position offsets that we can use to translate to that position. With this fundamental system, we can then attempt multiple tasks by repurposing fundamental building blocks for ease of mission code development.

Our two-sub strategy of using both Nemo and Marlin in our testing is a cornerstone of our development strategy this year. Most of our software and electrical subsystems are fully transferable between the two platforms, and being able to test in parallel allows us to iterate faster and more efficiently on our designs.

A. Gate

This year, we prioritized the vision system to be able to identify the gate. Using ML models inference on live camera images, we can calculate the angle and distance to the object. With this information, we send a command to the separate microcontroller to control the motors to move the sub to that destination. Further leveraging the vision system, we upgraded this system to repeat this process in a loop, to iteratively update the trajectory of the submarine to reduce errors and increase accuracy.

B. Style Points

With the design of Nemo AUV with eight motors in six degrees of freedom, Nemo can complete style points, which are planned to be yaw rotations. To optimize this process and avoid 90-degree blocks, we redesigned the spinning mechanism to instead spin continuously by continuously updating the yaw setpoint to a certain offset in front of the current yaw position, allowing the sub to save time during the competition run to gain greater amounts of points or save time for potential reruns.

C. Buoy

Because our software relies on functions that execute a fundamental movement such as moving, turning, or detecting the angle to an object, we can repurpose the same generic approach function and use it for the buoy task as well, enhancing reusability throughout the software stack.

D. Bin

Instead of only using the vision system to calculate position offsets and move to them once like in previous years which yielded inaccuracies in the bin drop, we now continuously repeat this process to readjust the sub's positioning multiple times in a loop, to increase accuracy levels.

E. Octagon

Because of monetary and hardware constraints this year, we were unable to pursue a traditional approach to the octagon task with hydrophones; instead, we turned to rely on our vision system for this approach. Because there are visible stool-like objects underneath the octagon, our software team focused on detecting the octagon stools with computer vision instead, allowing us to calculate the angle to the octagon visually as a substitute to the hydrophones method. With this angle information, we can then rely on the fundamental building block functions of our mission code to guide us underneath the octagon to the surface.

F. Path Markers

Because of the angles in between the tasks that vary between each competition task, we deemed it a top priority to introduce a path marker implementation to increase our overall reliability and error rate as we traverse through the course. For this method, we rely on OpenCV to detect the path marker, and we then reuse the same alignment system as with the bin to align the submarine towards the next prop.

II. DESIGN CREATIVITY

A. Mechanical Subsystem

1) Symmetrical Hull Design: In previous years, the asymmetrical hull of our previous submarine, Marlin, caused navigation issues of being unbalanced in the pitch and roll axes, as well as veering horizontally off course when traveling at high speeds. To remedy this issue, the mechanical team designed symmetrical cross-sections to improve stability, including two mounting planes that would evenly distribute the weight of the submarine. Because of these changes, Nemo's center of gravity is geometrically centered, increasing the sub's maneuverability and stability.

2) Mounting Plane Modularity: While the mounting planes fix instability issues, the mechanical team also designed them to increase the modularity of the submarine, to fix the restraints of having to redesign portions to fit around the irregular frame on Marlin. These modular panels can be reprinted at any team to fit any new set of components, allowing for greater flexibility and ease of system maintenance. This new feature has proved its worth several times this past year, such as when we printed a new hole to hold an updated down camera enclosure, or when we added a gap to clamp our new Doppler Velocity Log (DVL) after our previous DVL suffered damage. Thus, because of the new addition of the mounting planes, we can expedite redesign speeds and allow for new changes as necessary throughout the system.

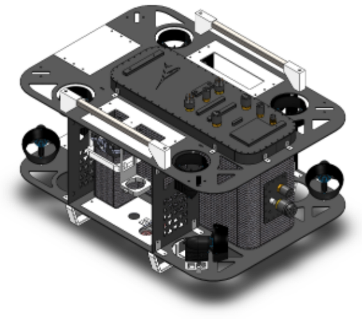

Fig. 1. Modular panels (white) on Nemo that allow for easy reconfigurations with different parts.

B. Electrical Subsystem

1) Refreshed Electrical System: While the mechanical team was water testing Nemo, electrical members also prepared Marlin's old rack to have both subs ready for competitions in case Nemo cannot be waterproofed in time. Throughout the year, electrical streamlined the power lines throughout Nemo, crimping and soldering a specialized XT-90 connector so that the subconns from the battery could be fed in one line directly to the LittleMega Fuse. Besides simplifying the power line, it also allowed members to refine their soldering skills, a necessary skill for all electrical members.

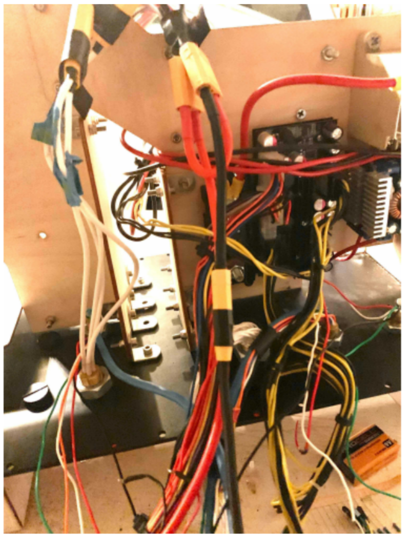

Fig. 2. Custom XT-90 connector on the rack.

2) GPU Upgrade: One of the main power draws in the AUV in the past was the GTX 1080ti. Through benchmarking various combinations of GPU and CPU configurations, we discovered that the GTX 1080 Ti was excessively power-hungry and limited our pool testing times. Luckily, we were able to replace it with Nvidia's GeForce RTX 3050, which consumes a fraction of the power of the 1080 Ti and provides similar performance. Additionally, the 3050 is also lighter and smaller than the 1080 Ti.

3) New DVL: Marlin's legacy DVL ran on our own custom suboptimal wiring and was prone to breaking down, with communication being a hassle over RS-232 transceivers. Thus, we decided to purchase a Cerulean DVL-75 with an upgraded IMU, to simplify the wiring system and communicate over UART. The upgraded IMU allows for a simpler and easier calibration process, perfect for a submarine the size of Nemo. The DVL-75 also minimizes external magnetic disturbances by separating the IMU sensor head from the electronics stack. With an extended Kalman Filter to compensate for drifts and offsets, the DVL-75 is the perfect option for testing in an unknown and new environment.

C. Software

1) Path Marker: To remedy issues of uncertain angles in between different competition props, we prioritized the ability to detect the path markers to avoid traveling off course. This year, we used OpenCV to implement our approach for detecting the path marker. One challenge we encountered this year with our AUV path marker detection was trying to separate detection of the path marker vs. other objects in the area—i.e. the bins and environment in the pool. We realized that the path marker had a different color than the other objects, such as the buoy's Tommy gun and the bin pictures. As we researched this more in-depth, we discovered a technique known as HSV filtering [1]. HSV filtering breaks down an image into three channels: hue, saturation, and value (brightness), similar to the RGB colorspace. With HSV filtering, it allowed for us to be able to differentiate—using specific values—between the path marker and other "noise" within the image, thus allowing us to hone in on the path marker and avoid detecting a different object. Building on HSV filtering, we also implemented a contour detection [2] — a built-in function in OpenCV—, which helped us to detect the edges of our path marker and decipher the angle needed to turn our AUV towards the center of the path marker. So, by combining contour detection with HSV filtering, we were able to detect the contours with the lowest hue and only pick up the path marker.

2) CVAT: Manual labeling of images to train vision models was often inaccurate and labor-intensive. To streamline the labeling process and increase the net output of datasets fit for training, a Computer Vision Annotation Tool (CVAT) [3] was installed and implemented to label sim data,

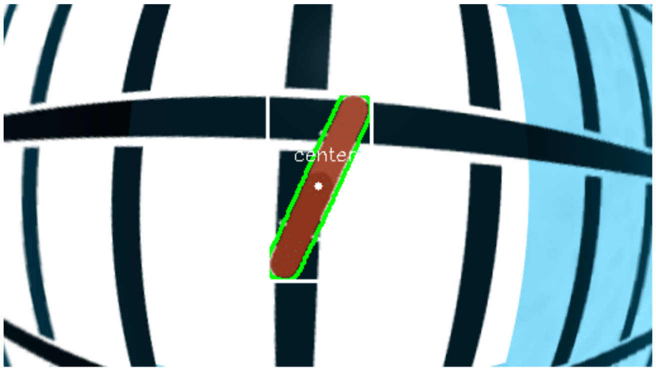

Fig. 3. Path marker detection in the simulator using OpenCV.

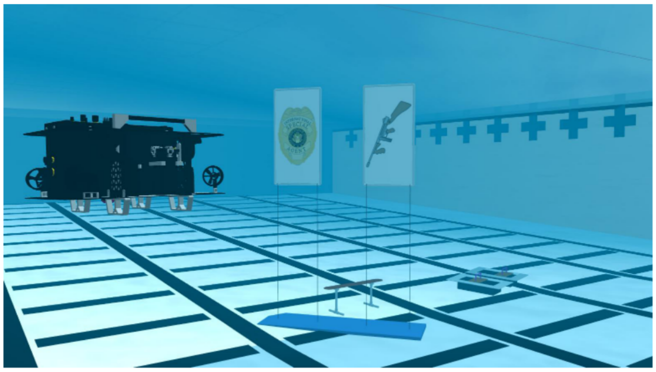

Fig. 4. Simulator with new props and more realistic fog to reflect the new UMD Natatorium pool.

which allowed for dynamic labels to be automatically placed between manual ones. Using their interpolation tool, we can label what surmounts to less than half of the images in a dataset and let the program automatically label the rest, significantly reducing the time in the competition where we mindlessly drone and label images. While these labels were seldom misplaced in series of images with greater visual noise, CVAT's flexible GUI allowed for easy adjustment. Relying on CVAT, we were able to label and train vision models for both the gate and buoy tasks.

3) Simulator Updates: Because of waterproofing struggles throughout the year, the software team developed alternative avenues to develop our motor control and mission code. This season, we updated the simulator based on Plankton, an open-source simulator based on UUV Simulator [4]. To reflect the UMD's natatorium and to mimic competition settings, we randomized the location of the props on each simulation run to deter developers from utilizing brute force methods for traversing through the course. We also noticed that the fog in the simulator looked unrealistic, so we implemented Gazebo's exponential fog model instead of the linear fog model to increase realism [5]. With the simulator, we can catch bugs in our vision code, our navigation mission code, and our lower level motor control code before testing in the water, streamlining software development to hopefully reserve pool testing time to validate approaches instead of catching bugs.

III. EXPERIMENTAL RESULTS

A. *Waterproofing*

As COVID restrictions lifted and in-person meetings resumed, we were able to undergo testing in a 10-foot pool. Because our designs were already completed, we dedicated most of our time to finishing construction and waterproofing. Observing preliminary results with worrying leaking issues, we determined that we would need to invest significant amounts of time to address this. The first step of testing was focused on external components, such as the battery and hydrophone boxes. By utilizing a GoPro inside enclosures and examining the resulting videos, we discovered that the battery box initially leaked through the screw holes and the corners of the lid. For the screw holes, we added sealing washers around each screw. For the corners, we epoxied 3D printed parts to both the lid and body which could be screwed together to create a stronger clamping force around the corner. These changes allowed us to fully waterproof the battery boxes.

Afterwards, we moved on to testing our MEB, the largest and hardest to waterproof. Our initial testing consisted of submerging the sub and pumping it with air. Bubbles and hissing noises would escape from holes, allowing us to pinpoint and resolve the issue by thoroughly applying epoxy, flex sealant, and marine adhesive. Later testing was done at greater depths, where we tied weights to the sub and allowed it to sit for an extended period. Currently, Nemo is still in the process of being tested for longer and at deeper locations.

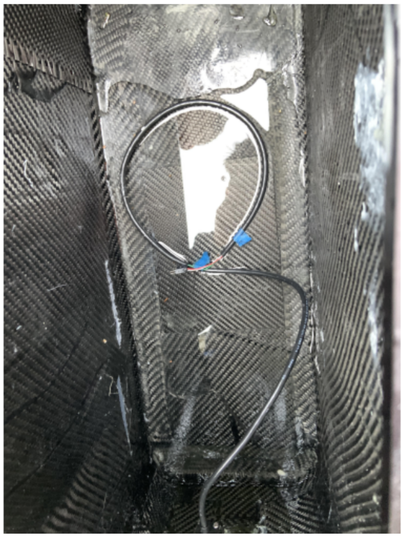

Fig. 5. An unsuccessful attempt at waterproofing Nemo. After this test, we concluded that the leak stemmed from a gap between the wall and the lid, which we covered with epoxy.

We registered both Marlin and Nemo to maximize our chances of being successful at competition with the option to ship either one. While testing Nemo, we concurrently utilized Marlin to test the electrical and software systems; this maximized our efficiency by limiting wasted time. For mechanical, time waiting for epoxy to cure went into further testing of Marlin and developing more efficient testing methods. For electrical, wiring Marlin became practice for future implementation of Nemo's electrical system. For software, we could move from using the sim to actual testing, much more accurately recreating the competition environment.

B. Simulation Results

We use the simulator throughout the year to test our mission navigation code; in sim runs, we have observed accuracy rates of 90% for gate, 90% for buoy, 80% for bin, and 70% for octagon. Granted, in-person testing will introduce a host of unforeseen errors that we will need to address. With the simulator, we have validated that our vision system can run at around 5 FPS, our new alignment system that continuously readjusts the sub's position increases accuracy levels (at least in the sim) for the bin task, our angle calculations are within +/- 7 degrees, and that the distance calculations are accurate within +/- 0.75 meters.

C. Lessons Learned

After missing in-person competition for two years in a row, the current team has minimal competition experience, marking this year an illuminating experience in attempting to recoup our abilities and re-learn hard lessons. The first and most important lesson is to start work early in the year, setting clear club-wide deadlines to get the sub in the water as soon as possible. Next, communication is essential, with weekly updates through the entire club to make sure that everyone is on the same page. Furthermore, documentation is extremely important to be able to transfer knowledge to the next generation. Finally, be careful with club hardware: rough handling of sensitive instruments (especially our old DVL which costed $10,000) can damage them and drain tens of thousands of dollars.

ACKNOWLEDGMENTS

We would like to thank our sponsors, who have made it possible for us to create our sub in the first place. Datron has helped us immensely by machining countless parts for Nemo, which has been essential to the construction of our new sub. Additionally, Theia has provided us with high-quality lenses to capture images of underwater objects, and Cerulean Sonar graciously offered us financial support for their DVL75, an extremely low cost and accurate DVL that has made it possible for smaller teams like us (and possibly other teams in need of cost-effective DVLs) to build a navigation system. In our school, we would like to thank our advisor Mrs. Bree Barnett Dreyfuss for having the patience to put up with our never-ending demands, as well as Amador Valley High School for allowing us to use their pool. Past members were also extremely instrumental, offering their insight and advice on how to run the club. Additionally, we are indebted to the organizers of Robosub, who have allowed us to overcome engineering challenges that we would never have been able to access at our school. Last but not least, we are especially grateful to our parents, who have continuously supported us throughout the year both emotionally and financially, guiding us through what was a stormy but transformative season.

REFERENCES

[1] O. S. C. Vision, “Thresholding operations using inrange.” [Online]. Available: https://docs.opencv.org/3.4/da/d97/tutorial_threshold_inRange.html

[2] ——, “Contours: Getting started.” [Online]. Available: https://docs.opencv.org/3.4/d4/d73/tutorial_py_contours_begin.html

[3] Intel. (2019) Computer vision annotation tool: A universal approach to data annotation. [Online]. Available: https://www.intel.com/content/www/us/en/developer/articles/technical/computer-vision-annotation-tool-a-universal-approach-to-data-annotation.html

[4] M. M. M. Manhães, S. A. Scherer, M. Voss, L. R. Douat, and T. Rauschenbach, “Uuv simulator: A gazebo-based package for underwater intervention and multi-robot simulation,” in *OCEANS 2016 MTS/IEEE Monterey*, 2016, pp. 1–8.

[5] D. Zdrojewska, “Real time rendering of heterogenous fog based on the graphics hardware acceleration,” 2004.

APPENDIX A: COMPONENT SPECIFICATIONS

Component	Vendor	Model/Type	Specs	Custom / Purchased	Cost	Year of Purchase
Frame	Custom	Aluminum 6061 - T6	90.50cm x 63.50cm x 33.34cm	Custom	Sponsored	2021
Main Waterproof Enclosure	In-House	Carbon Fiber Enclosure sealed with 2 EPDM O - Rings	62.23cm x 22.76cm x 35.24cm	Custom	$200	2021
Battery Enclosures (2)	In-House	Carbon Fiber Enclosure sealed with single EPDM O - Ring	10.50cm x 10.50cm x 23.95cm	Custom	$120	2021
Hydrophones Enclosure	In-House	Carbon Fiber Enclosure sealed with single EPDM O - Ring	13.00cm x 10.00cm x 19.62cm	Custom	$120	2021
Waterproof Connectors	SubConn	Circular Series	(Varies Based on Series) Micro-Circular Series, Power Series	Purchased	$1500	2015
Thrusters	VideoRay	M5 Thrusters	90mm Length	Purchased	$11,200	2015
Propellers	VideoRay	Standard Propellers	90mm	Purchased	Included with Thruster	2015
Motor Control	Rugged Circuits	Rugged MEGA	ATmega 2560 microcontroller, Arduino Protoshield	Purchased	$30	2017
High Level Control	In-House	PID Control	5-20 Hz, AVR	Custom	Free	2010
Actuators	Numatics	0438D01-04A	Bore Size: 7/16", Stroke Size: 4.0"	Purchased	$42	2016
Batteries	ZEEE Power	6S	6000mAh, 22.2V, 260Wh	Purchased	$200	2022
DC Converter	Cincon	Brick CFB600-300S	600W, 180-425V, 48V to 24V	Purchased	$15	2021
CPU	Intel	i7-4790T	4 Cores, 8 Threads, Base Freq: 2.7 GHz, Turbo Freq: 3.9 GHz, Cache 8 MB	Purchased	Sponsored	2017

GPU	Nvidia	RTX 3050	Memory Type: GDDR6, 8 GB RAM, CUDA Cores: 2560, Power: 130 W	Purchased	$300	2022
Motherboard	Jetway	NG9J-Q87 Mini ITX	4 USB 2.0 Ports, 2 USB 3.0 Ports	Purchased	$200	2017
RAM	Corsair	Vengeance 16GB	2x8GB DDR3 SODIMM RAM	Purchased	$80	2017
Storage	Samsung	ITB mSATA 860 EVO SSD	Max Seq Read/Write Speed: 550 Mb/s	Purchased	$150	2017
Internal Comm Network	ROS	ROS2 Foxy	Ubuntu 20.04	Custom	Free	2022
External Comm Interface	-	Ethernet	1 GB/s	Purchased	Included with SubConn	2015
Doppler Velocity Log (DVL)	Cerulean Sonar	DVL75	Type: Phased Array Transducer, Frequency: 675 kHz, Max Depth: 300m	Purchased	$2300	2022
Altitude Heading and Reference System (AHRS)	PNI Sensor	TRAX AHRS	Static Heading Accuracy: 0.3°	Purchased	$1000	2017
Pressure Sensor	Ashcroft	K-17	Accuracy: ±1%, Range: Vacuum to 20000psi, Gauge Range: 15 psig, Input: 10-36V (DC), Output: 1-5V (DC)	Purchased	$400	2010
Front Camera	FLIR	BFS-U3-200S6	Frame Rate: 30 fps, Resolution: 5472x3645, Megapixel: 20MP, Sensor Type: CMOS	Purchased	$750	2015
Front Camera Lens	Computar	VO828-MPY	8mm fixed lens, Resolution: 12MP, Horizontal Angle: 77.3°, Vertical Angle: 61.7°	Purchased	Sponsored	2015
Down Camera	FLIR	BFS-U3-13Y3C-C	Resolution: 1280x1024, Megapixel: 1.3MP, Frame Rate: 170FPS, Sensor Type: CMOS	Purchased	$540	2015

Down Camera Lens	Theia	SY125M	Focal Length: 1.3mm, Resolution: 5MP, Horizontal Angle: 125°, Vertical Angle: 119°	Purchased	Sponsored	2015
Hydrophones	Teledyne Reson	TC4013	Frequency Range: 1Hz to 170kHz, Depth: 700m max, -211 d13 ± dB receiving frequency	Purchased	$4500	2013
Signal Processing	Diligent	Nexys 4 DDR Artix-7	Block RAM: 4,860 Kbits	Purchased	$250	2019
Algorithms: Vision	Ultralytics	YOLOv5s, RGB equalizing filter	5 FPS	Open Source	Free	2022
Algorithms: Acoustics	In-House	MUSIC	Hydrophones	Custom	Free	2018
Algorithms: localization, mapping	In-House	DVL data, image calculations	DVL, IMU, CV	Custom	Free	2017
Algorithms: Autonomy	In-House	Linear instructions	ROS2 nodes	Custom	Free	2022
Open source software	Open source	ROS2, YOLOv5s, OpenCV	Node management, computer vision	Custom	Free	2022
Team Size (number of people)	44					
Expertise ratio (HW vs. SW)	23:13 + 8 Business					
Testing time: simulation	125 hours					
Testing time: in-water	90 hours					
Programming Languages	C, C++, Python 3					