RUMarino Technical Design Report

***Abstract*— This year, the RUMarino team comes with a new constructed AUV named "Hydrus". This AUV will be more advanced and modular than the previous AUV, Proteus, by expanding its physical capacities. The different divisions of the team were able to implement and combine their abilities so this new AUV can complete its tasks. This new structural design of the AUV has a more robust platform, a dual hull design, vectorial motors, dual torpedo launchers, and has the ability to charge its battery from the outside eliminating the need to open the cabins. Although the AUV offers new abilities that the team did not have with its previous AUV, it offers a difficulty for assembling and disassembling it, and accessing its interior components. As part of the submarine software, the team has decided to attack its difficulty by working all tasks independently by creating different modules from scratch. Manually managing this software comes with its disadvantages, and the team is managing the problem by using Robot Operating System (ROS). With this design, the AUV has the ability of gathering data and processing it concurrently. Another new addition was the integration of the Doppler Velocity Log (DVL), that can be used to keep track the AUV current state. There are still tests that need to be conducted, such as tests in the pool. Given the current situation of COVID-19, this test has been affected. Due to the situation, the team took other roads such as: studying, reinforcing, and providing documentation until we can conduct more physical tests of our new AUV.**

Keywords—AUV, DVL, design.

II. COMPETITION STRATEGY

In previous competitions, the RUMarino team has been limited by the physical capacities of our previous AUV, Proteus, in the sense that the platform was pushed beyond its limit for new additions and therefore established a solid "ceiling" of operational capacity for completing competition tasks. This year the team has constructed a new AUV, named "Hydrus", with the sole purpose of providing a more robust platform capable of expanding and adding new capabilities to the AUV through the coming years with modular attachment points.

To demonstrate these advantages, the Mechanical Team developed an AUV that brings a new capability that in the past was not able to be implemented, the addition of dual torpedo launchers to aid in the completion of more competition objectives. While it offers greater structural integrity, a disadvantage of this new design strategy is an added difficulty for assembling and disassembling the AUV and accessing its interior components. To combat this hurdle the new frame includes more hull penetration points and wet-pluggable connectors that will allow connection to the onboard computer and battery array without the need to disassemble the pressurized cabins.

Additionally, it has been pursued to complete different tasks quickly and efficiently. The capacity with which our AUV can return to the pool for more tests is of vital importance for all divisions that require precision to collect data and values for adjustments in programming, control systems and sensors. Therefore, the electrical systems division provided this facility so that our AUV can complete more hours of tests that provide us with the information and ability to complete the different tasks in the different areas in the competition.

Designing a software solution for a submarine is a complicated process with all the details, data processing and calculations it needs to perform to be able to control and efficiently synchronize every part of its hardware to be able to complete the necessary tasks assigned to it; in a simple case, traverse efficiently through a three-dimensional space. The decision our team made to design and develop the software was by splitting the different functions that the software will have to perform and solving each requirement or task independently of each other by creating different modules. Some examples needed for this type of system are the controller modules that manage the device's different components that control the position of the AUV, vision modules that can capture and process what the submarine is seeing, and the sensor module that manages the different sensors like depth and pressure and the logic module where the decisions are made based on the information gathered from all the modules mentioned before..

One of the disadvantages of developing software based on modularity is that every module used needs to be able to efficiently function and communicate with each other [1]. Using the competition as an example [2], badly implemented modules could lead to slower data processing making the submarine performance extremely poor and slow which is detrimental when participating in a competition. To help us solve this problem of managing the different modules efficiently we are using the Robot Operating System (ROS). With this framework we can easily implement independent modules where our team can focus on the desired module behavior and the heavy lifting of communication and synchronization is taken care by ROS. One of the great advantages of using ROS is that any information processed, or any information needed from other modules is made accessible to every module in the system by ROS making it a great option [3].

By maintaining the current design process the submarine will be able to use its modules concurrently providing the advantage of making logical decisions while gathering data and processing it at the same time to be able to alter its decisions while at the same time performing them. A very clear example where the ability to use all its modules concurrently is selecting the gate which is the first task the submarine will need to perform. The submarine can keep track of its current position with the Doppler Velocity Log (DVL) while at the same time find and locate the gate and its current position relative to the submarine, then proceed to pass the corresponding gate and make any necessary adjustments if for some reason the position of the submarine or gate where to change due to external factors and apply them accordingly, instead of performing all these actions one at a time. This type of task, tracking an object while positioning itself according to it, can be seen as simple, but failing to achieve it reliably and efficiently can cost the whole competition.

The control systems team built a Pyrunner library (code generator for python based on block programming software like Simulink and LabVIEW). In addition, a system independent communication architecture for the control system was designed and implemented using ROS. Currently, the code being tested using a simpler system where the dynamics and controller were already designed beforehand.

As for the embedded systems, the team did complete overhaul of the embedded code. A lightweight many-to-one communication interface between the microcontroller and the AUV's main computer was designed to reduce the overhead of purely using ROS to interact with these and to better integrate new devices into our system. The devices that are currently implemented are: the camera's servo, the thrusters (propeller system for the AUV), a gripper, and a pressure sensor. Additionally, a programmable testbed for the embedded was made to test the code in the microcontroller, the devices attached to the microcontroller, and other parts of the code in the AUV (like the control systems). With this, an adequate test plan is being developed to regularly test sensor functionality before and after performing in field testing in the pool.

III. DESIGN CREATIVITY

Ideas and alternatives were chosen in the identification of new ways to solve the problem of the previous models. Along the way, many creative ideas were implemented and rejected, which after several iterative attempts, generated what is the new design of the hull and structure of the AUV. The dual hull cabin and vectorial thrust arrangements turned out to be the most innovative. For the dual hull cabin design, we opted to use two standard six inches pressurized cabins joined in the middle by an original aluminum casing that serves as both access points for all interior/exterior connections and as a heat sink to better manage the heat generated by the power distribution system.

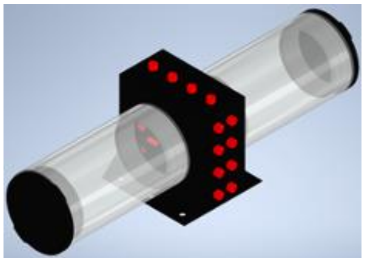

Figure #1: Dual cabin hull design.

The vectorial thruster arrangement utilizes a combination of machined HDPE panels and carbon fiber reinforced 3D printed components to align 8 T-100 thrusters in a way that allows for six degree-of-freedom movement underwater.

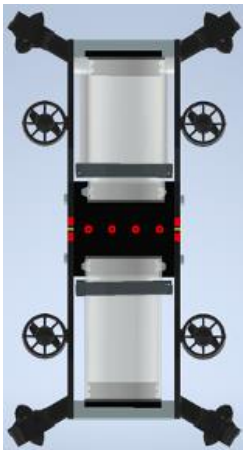

Figure #2: Top view of AUV, providing a clear view of thruster vectorial placement.

Similarly, in the search for new advances, the electrical power system is designed so that it can charge its battery without the need to open any of its cabins or remove the battery from its position inside the vehicle, in spite of the fact that it is supported through an external connection that can be supplied with power when needed. This is in order to reduce the time to maintain the AUV and to increase our ability to return to the pool and perform more tests. In addition, it has another external connection to provide an additional power source to the computer in case the vehicle's battery is in the process of charging, but it is required to update data in a vehicle. Thus, allowing various maintenance and troubleshoot tasks to be carried out without the need to disassemble it to make these connections directly to the devices.

Inside the AUV, the software system developed for the submarine is designed with modularity in mind. It will be divided into different modules each with their own function independent of each other. An example of modularity everyone can understand is “Legos”, different pieces are used to build something bigger, and any piece can be reused, used to create a different larger piece and moved to another place where it would be needed [4]. By maintaining this approach while developing the software it grants us many advantages in the long run, reduced maintenance costs, incremental upgrades without having to break the system, readability for future developers, cost efficiency, reusability of the modules and efficient testing of the whole system or individual modules, which is crucial when developing an object that must interact with the physical world, analyze it and traverse through it.

The Pyrunner package that was mentioned earlier was made for generating optimized Python code using a block programming interface. It provides an easier and more intuitive environment to create and evaluate calculations by creating objects that represent the calculations and connecting them. Behind the scenes, the package creates an executable object that can be called by other parts of the code. This was integrated to make the control system’s code structure that is platform independent, this code can be adapted and reused for any AUV, or more specifically, any other system that uses ROS for communications, and to make a system that is easier to test and modify.

Furthermore, a ROS library was used to communicate between our microcontroller and AUV’s main computer. Although this provides a way to transfer information from one place to another, this introduces additional overhead because the microcontroller is now tasked with doing more processing than is needed due to ROS’ structure for handling messages. A new communication protocol was designed and is now being implemented, which aims to minimize the amount of overhead by using a lightweight structure for synchronizing and delivering messages and using Protocol Buffers, made by Google [5], to provide ways of creating messages and processing them in a similar manner that ROS does.

The programmable testbed connects a device using python to the microcontroller. The testbed has callable objects that directly connect to specific functions in the microcontroller, so one could make a new test with these objects and verify if specific parts of the microcontroller’s code function appropriately. This lets the team test: the code in the microcontroller, the devices attached to the microcontroller, and other parts of the code in the AUV, by just creating scripts that use the callable objects.

The whole vision system is built upon a module-based mindset meaning that each component runs completely independent from the others. This means that complete changes to a module can happen if the expected outputs and inputs follow the documentation structure. The module flexibility is accessible through easy to modify server configuration files. What makes this flexible system so special is that the visual recognition pipeline can use all the provided modules interchangeably or at the same time, for example using multiple machine vision algorithms on top of each other.

IV. EXPERIMENTAL RESULTS

Many of the components utilized were manufactured before the onset of the COVID-19 Pandemic, but most were manufactured or acquired after, and all were assembled after. The team made full use of 3D printer technology in order to make test pieces and the final components from the safety of our individual homes. Pressure tests to ensure the rigidity of the seals and waterproofing were conducted with vacuum compressors by bringing Hydrus' internal pressure to 14.7psi vacuum, effectively simulating a depth of 33ft, and maintaining it overnight which is more than enough for the competition. Destructive tests were improvised with weights to confirm the ability of the 3D printed components to withstand the forces generated by the thrusters, and the HDPE chassis' rigidity. When it was allowed, the team was able to take Hydrus to the university pool in short squads of three members while taking the appropriate health safety measures to avoid contamination while also being supervised by lifeguards. In the future, we hope to be able to test Hydrus' capabilities more often and with greater rigorous tests.

For upcoming experimental tests in the pool, it is expected to be able to test the vehicle to mark its benchmarks of how much the vehicle can be with the battery charge and the time required to charge the vehicle before returning it to the test pool. Where the requirements to obtain the searched results are not necessary too many tests since although it is forecast to observe the functionality time limit this only requires approximately 3 tests, since in the consequent tests the time is not so high by which the searched goal is completely feasible.

Given the current situation with the global COVID-19 pandemic the progress and development of the submarine was greatly affected. The different teams had estimated the acquisition of certain components of the submarine and planned out time in advance to familiarize with the new additions and the time it required to assemble them together had to be rescheduled due to problems with supply, some others were delayed significantly due to the high congestion of the mailing system. The testing portion of the project was delayed due to the previously mentioned problems and the many restrictions issued by the government that made it harder to obtain facilities to test and for team members to gather and work on the submarine. To be able to deal with all these challenges some of the teams that were affected changed their efforts to planning and designing the software while also providing documentation and reinforcing previous documentation so that when the opportunity to retake the tasks that were put on hold, they could commence and approach them in an efficient and organized manner.

Since the implementation of the control system is platform independent, the code structure itself can be tested with any other well documented system that uses ROS instead of our own AUV. As for the controllers themselves, these can be tested numerically with a mathematical model of the system to verify if the controller works. Additionally, the control systems team has been working with Gazebo and UUV simulator to set up an environment to test out if the system sends the correct information and if the controllers behave properly when the AUV's 3D mesh is used to perform a more comprehensive simulation [7],[8].

The programmable testbed can be used to access tests that verify if all the devices are working properly. As mentioned above, the program is completely dependent on constant testing and designed around that. This means that before every mission be it a competition or just a test run, the machine learning models are tested and reviewed.

V. ACKNOWLEDGEMENTS

Our teams' success has had many contributing factors: our team members, our mentor Raul Torres, the University of Puerto Rico that provided us a space for us to work, and our sponsors. During the full trajectory of our project, we have received numerous support and financial backing, but during Hydrus's journey the companies that have provided the most financial support and technical advice where Naval Nuclear Labs, NAVSEA, Nvidia, Protocase, Teledyne, General Motors, L3Harris and Lockheed. Thanks to Teledyne, we now have a DVL in our submarine that has given us numerous advantages in this competition, and we also want to thank Margo Newcombe and William Meachum who have been our contacts in Teledyne. Along this line, we want to thank Jaime Calderón from NAVSEA, for putting us into contact with multiple resources to help us better understand how to use the DVL. NAVSEA is also responsible for a large part of the equipment used in Hydrus, which they have provided either directly given us the equipment or giving us monetary resources to acquire the equipment.

Covid-19 has brought many challenges, but in face of these we have prevailed and moved forward. Gene Terwilliger from Naval Nuclear labs (NNL) gave us guidelines on how to perform tests on our submarine during this pandemic, based on how NNL was performing their own tests. We want to extend our gratitude to three other employees from NNL: Sheelah Aulet, Frances Martinez and Jose Pabón, who have been part of multiple knowledge transfer meetings between RUMarino and NNL. Now, testing and simulations are a big part of our project and for that we need state of the art equipment, for which L3Harris provided the financial backing for us to get the equipment necessary to run the AUV simulations. Also, we had the opportunity to engage in a knowledge transfer meeting that allowed us to fine tune our vision departments' objectives. Finally, but not least we want to make an honorary mention of Omar Gonzalez from General Motors, who was the first captain in RUMarino and has continued to contribute to the team, with insights and support.

VI. REFERENCES

[1] University of Lucknow, *Software Design Principles*. Accessed on: Jun. 4, 2021. [Online]. Available: https://www.lkouniv.ac.in/site/writereaddata/siteContent/20200329162359626020200329162359626 0rohit_yadav_engg_Software%20Engg_2.pdf

[2] RoboSub, *24th Annual International RoboSub Competition Mission and Rules*, Mar. 26, 2021. Accessed on: Mar. 30, 2021. [Online]. Available: https://robonation.org/app/uploads/sites/4/2021/04/RoboSub-2021-Mission-and-Rules__V1.pdf

[3] Open Source Robotics Foundation, Inc, *Is ROS for Me?*, Accessed on: Mar. 15, 2021. [Online]. Available: https://www.ros.org/is-ros-for-me/

[4] *"Brick by Brick"*. Accessed on: May.9, 2021. [Online]. Available: https://static1.squarespace.com/static/55b7729ce4b00871cfc18b4a/t/57d0737f6a49639794889141/1473278847679/Rook_Chapter_Website.pdf

[5] Google LLC, *Protocol Buffers*. Accessed on: Feb. 20, 2021. [Online] Available: https://developers.google.com/protocol-buffers

[6] Open Source Robotics Foundation, Inc. *Gazebo Simulation*, Open Robotics Foundation, 2014. Accessed on: Jan. 27, 2021. [Online]. Available: http://gazebosim.org/

[7] The SWARMs Project, *Unmanned Underwater Vehicle Simulator*, Accessed on: Jan. 27, 2021. [Online]. Available: https://uuvsimulator.github.io/packages/uuv_simulator/intro/

[8] The SWARMs Project, *Unmanned Underwater Vehicle Simulator*, Accessed on: Jan. 27, 2021. [Online]. Available: https://uuvsimulator.github.io/packages/uuv_simulator/intro/

Appendix A: Component Specification

Component	Vendor	Model/Type	Specs	Cost (if new)	Status
Buoyancy Control	BlueRobotics	BAR30	**Electrical Supply Voltage**:2.5-5.5 volts I 2 C **Logic Voltage (SDA and SCL):**2.5 - 3.6 volts **Peak Current**:1.25 mA **Pressure Maximum Mechanical Pressure**: 725 psi* **Operating Pressure**: 435 psi **Operating Depth:** 300M	~$72.00	
Frame	N/A	Custom/Aluminum/HDPE/Carbon Fiber		~$1,000	
Waterproof Housing	BlueRobotics	6" series Cabin, 2" series Cabin		$317.00 / $145.00	
Waterproof connectors	BlueRobotics, McCartney	M10 Penetrators, Subconn Wet Pluggables	M10 Thread, 6pin connector, 10 pin connector	$4.00 ea, $828.05	
Thrusters	BlueRobotics	T-100	100 lbsf thrust.	$119.00 ea.	
Motor Control	SHENZHEN MODEL FANS ELECTRONIC CO.	F30A	**Continue Current:** 30A, **Burst Current:** 35A **Input Voltage:** 2-6S Lipo **Weight:** 12.95g **Size:** 43*40*8mm	N/A	

High Level Control	N/A	N/A	N/A	N/A	N/A
Actuators	N/A	N/A	N/A	N/A	N/A
Propellers	N/A	N/A	N/A	N/A	N/A
Battery	TURNIGY	6s LIPO	**Capacity:** 8000mAh **Voltage:** 6S1P / 6 Cell / 22.2V **Discharge:** 15C **Constant / 30C Burst** **Weight:** 1110g (including wire, plug & case) **Dimensions:** 170x69x48mm **Balance Plug:** JST-XH	~$114.95	
Converter	N/A	N/A	N/A	N/A	N/A
Regulator	DROK	Buck Converter Reducer	**Input voltage range:** DC 5.3-32V **Output voltage range:** DC 1.2-32V (input should be at least 0.8V higher than output) **Output current:** 8A for long-term stable work. If strengthen heat dissipation can reach 12A. **Output power:** natural cooling 120W (within 8A). If enhance cooling can reach 160W	~$16.89	
CPU	Nvidia	Jetson TX2	**AI Performance** 1.33	~$980.61	

			TFLOPS **GPU** NVIDIA Pascal™ architecture with 256 NVIDIA® CUDA® cores **CPU** Dual-core NVIDIA Denver 2 64-bit CPU and quad-core Arm® Cortex®-A57 MPCore processor complex **Memory** 4 GB 128-bit LPDDR4 51.2 GB/s **Storage** 16 GB eMMC 5.1 Power 7.5W | 15W PCIe 1 x1 + 1 x2 PCIe Gen2, total 30 GT/s **CSI Camera** Up to 5 cameras (12 via virtual channels) 12 lanes MIPI CSI-2 (3x4 or 5x2) D-PHY 1.2 (up to 30 Gbps) **Video Encode** 1x 4K60 | 3x 4K30 | 4x 1080p60 | 8x 1080p30 (H.265) 1x 4K60 | 3x 4K30 | 7x 1080p60 | 14x 1080p30 (H.264) **Video Decode**		

			2x 4K60 | 4x 4K30 | 7x 1080p60 | 14x 1080p30 (H.265 & H.264) **Display** 2 multi-mode DP 1.2/eDP 1.4/HDMI 2.0 1x 2 DSI (1.5Gbps/lane) **Networking** 10/100/1000 BASE-T Ethernet **Mechanical** 69.6 mm x 45 mm 260-pin SO-DIMM edge connector		
Internal Comm Network	N/A	N/A	N/A	N/A	ROS
External Comm Network	N/A	N/A	N/A	N/A	N/A
Compass	N/A	N/A	N/A	N/A	N/A
Inertial Measurement Unit	VectorNav	VN-100	Gyro In-Run Bias (typ.) : 5-7°/hr. Heading : 2.0° Pitch/Roll : 0.5° Accel In-Run Bias : < 0.04 mg		
Doppler Velocity Log (DVL)	Teledyne	Wayfinder	**Max Bottom Tracking Range** Frequency: 600 kHz BT Range:60m **Min Bottom Tracking Range**	$7,500	

			Frequency Min: 600 kHz BT Range:0.5m **Long Term Accuracy (Bottom Tracking)** Frequency 600 kHz BT LTA: ±1.15% **Velocity Range** ±10 m/s **Max Ping Rate**:16 Hz **Beam Angle**:30° **Standard Depth Rating (others may be available)**: 200m **Communications**:RS-232 **Average Power Consumption**: 3W		
Vision	StereoLabs	ZED		$349.00	
Acoustics	N/A	N/A	N/A	N/A	N/A
Manipulator	BlueRobotics	Newton Subsea Gripper		$439.00	
Algorithms:vision	N/A	N/A	N/A	N/A	MaskRCNN & Darknet YOLO
Algorithms: Acoustics	N/A	N/A	N/A	N/A	N/A
Algorithms:	N/A	N/A	N/A	N/A	Inhouse

Localization and mapping					pipeline (ZED, Darknet and OpenCV)
Algorithms: Autonomy	N/A	N/A	N/A	N/A	Inhouse system based on ROS
Open-Source software	N/A	N/A	N/A	N/A	Darknet, ROS, OpenCV, PCL
Team Size (number of people)	N/A	N/A	N/A	N/A	27 members
Expertise Ratio (hardware vs. software)	N/A	N/A	N/A	N/A	4:3
Testing Time: simulation	N/A	N/A	N/A	N/A	64 hours
Testing Time: in water	N/A	N/A	N/A	N/A	10 hours
Inter-vehicle communication	N/A	N/A	N/A	N/A	N/A
Programming (languages)	N/A	N/A	N/A	N/A	C++ & Python

Appendix B: Outreach Activities

One of RUMarino's priorities as a team is to be able to collaborate, educate, and interact with the community. Throughout 2020 to 2021, the team has had more than five outreach activities where almost 70 intermediate and high school students have participated. These activities help motivate them to study professional fields related to STEM. Also, they are informed about what RUMarino is, what it does, and the different aspects that make up the team. The students are from all over Puerto Rico between the ages of 13 to 17. In these outreach activities interactive dynamics are made, such as Kahoot, for children to learn and have fun. The winners of the Kahoot receive a 3D printed RUMarino keychain as a reward.

Currently RUMarino has been mentoring a group of five students from CROEV, a specialized school in science and mathematics in the town of Villalba, who have been participating in a research project. RUMarino has been advising the students on the design of their submarine, materials they should use, as well as teaching them Raspberry Pi. Our goal is to help these students keep motivated in such research and help them accomplish their mission.