

Design and Implementation of UFRJ Nautilus' AUV: Lua

Lara F. de Amorim

Federal University of Rio de Janeiro

Rio de Janeiro, Brazil

larafigueiredo@poli.ufrj.br

Vitor A. Pavani

Federal University of Rio de Janeiro

Rio de Janeiro, Brazil

vitorpavani@poli.ufrj.br

Lucas B. Alexandre

Federal University of Rio de Janeiro

Rio de Janeiro, Brazil

lucasalexandre@poli.ufrj.br

Pedro H. de J. Teixeira

Federal University of Rio de Janeiro

Rio de Janeiro, Brazil

pedroteixeir@poli.ufrj.br

Claudio M. de Farias

Federal University of Rio de Janeiro

Rio de Janeiro, Brazil

cmicelifarias@cos.ufrj.br

Abstract—The UFRJ Nautilus team focused this year on the making of a brand new autonomous underwater vehicle (AUV) called Lua. It comes with a completely new mechanical and electronic part and brings to the table our previous software algorithms with improvements. The mechanical work was focused on a lighter AUV with higher mobility and the addition of the mechanical arm and the ball dropper while the electrical work centered around the use of a backplane to guarantee a more organized and safer power distribution. For the software we brought our previous control system to the new AUV, now with 8 thrusters, enhancing the localization method and adding object manipulation to our set of skills. That enabled the team to do not only the tasks that depend solely on the location but also the tasks that involve the use of the mechanical arm and that requires dropping the markers.

I. Competition Strategy

Last year we focused on performing successfully all tasks that didn't include object manipulation and for this year with a new AUV we intend to realize the following sequence of tasks (that can also be seen in Fig. 1) choosing the Bootlegger side of the competition:

- 1) Coin Flip
- 2) Gate
- 3) Style
- 4) Path
- 5) Buoys
- 6) Path
- 7) Bins
- 8) Emerge at the Octagon

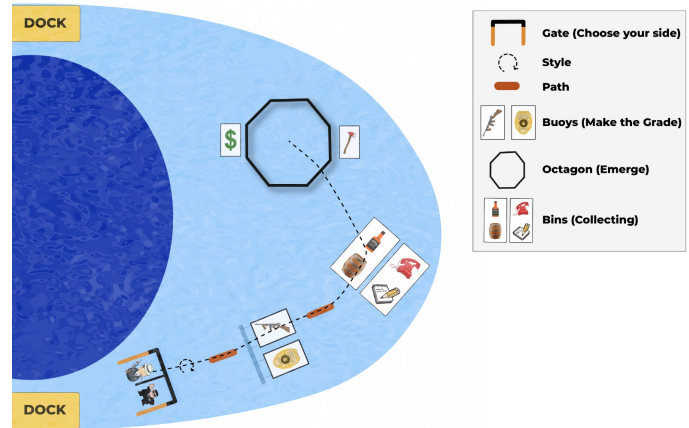


Fig. 1: 2021 Competition Strategy

Each task has its strategy, which will be described later in this section, but we also have general goals in terms of movement and localization. This year we made improvements to our SLAM method and implemented a new algorithm of path planning that enhanced our navigation performance between tasks. Those changes will be specified further onto this paper at II-C1 and II-C4.

1) *Coin Flip*: To have our vehicle lined up with the gate regardless of the result of the coin flip Lua takes the orientation from the IMU, which has an integrated compass, and uses the control system to set the AUV's roll pitch and yaw variable to zero.

2) *Gate*: Initially, our neural network detects the orange legs of the gate and then, by transitioning to the gate state, aligns all axis of the AUV with the center of the gate. When it reaches the midpoint, we detect the Bootlegger image and move Lua along the width of the gate centering it to the image and finally going forward to the next task.

3) *Style*: Our previous AUV only had 5 degrees of freedom and because of that, it was hard to recover its trajectory after successfully performing the style points. To fix that issue we aimed to improve our control system by adding another degree of freedom. That will be better explained at II-A.

4) *Path*: To follow the path the AUV stays in the center of it, which can be detected by our neural network, while moves forward.

5) *Buoys*: After passing through the first path Lua will use its neural network to find and touch the Tommy Gun buoys, as we chose the Bootlegger.

6) *Bins*: Coming close to the end of the course and passing through the second path, located between the buoys and the bins, the vehicle starts to look for the image of the bottle. The electronics from this task can be found better explained at III-B3 and the algorithms at III-C2.

7) *Emerge at the Octagon*: Ending our course we start looking for the pinger as you can see in III-C4. When the AUV gets to the octagon it emerges.

II. Design Creativity

A. Hydrodynamics and Mechanics

This year we aimed to design a lighter and more compact AUV and to achieve that goal we chose to use materials with lower density and designed a more organized hull so that we could effectively use our inner space. Lua also has the addition of 2 new thrusters that make it possible for it to move on all 6 degrees of freedom.

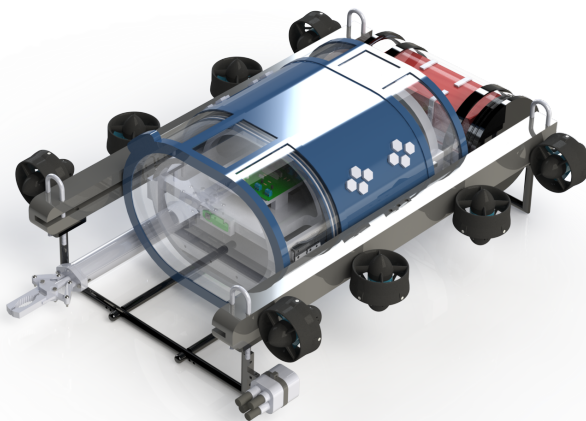


Fig. 2: Lua's SolidWorks [1] rendering

Each part of Lua was thoroughly thought so that it could enable the success of each one of the tasks from our competition strategy.

1) *Frame and Support arms*: One of the most important things when testing our control system is to ensure that the AUV has as much symmetry as possible. That requirement is caused by assumption of the mathematical model of the AUV that the vehicle has the same drag coefficient when moving forward and backward as an example. Another demand of our vehicle is space to attach our hydrophone support, our mechanical arm, our thrusters and our dropper. To fulfill both of those demands our hydrodynamic and mechanics team designed a frame and two support arms making it possible for Lua to collaborate with the work of our software team.

2) *Acrylic*: With this new geometry of acrylic hull we have more space to use two webcams as a stereo camera and more freedom to use whatever distance between them that we choose fit. That's crucial to our project given that during our physical tests we can adjust the best arrangement possible even if that differs from the one at the 3D project.

3) *Belt*: Acting as the main piece of our vehicle, our so-called belt was built to serve as an interface between the external part of the AUV and the internal compartment, through subsea connectors and to guarantee our AUV tightness through the use of three o'rings. This part is considered when added to two acrylics and two exoskeletons, our main hull.

4) *Exoskeleton*: When projecting the belt and idealizing its connection to the acrylic we realized that thinking in a long term the process of opening and closing the AUV could be detrimental to the integrity of our acrylic. And because of that, it was decided to build an exoskeleton to be used as a structural fortification and to reduce the effort caused on the acrylic.

5) *Internal Compartment*: During our time with our last AUV, we noticed that our PCBs were not organized at all and that occasioned several problems during our tests. With this new vehicle, we wanted to reduce as much as possible the number of setbacks, and to do that we built an internal compartment that would keep all of the PCBs always at the same place. Also, a problem that we were used to facing was to access our electronic part more

easily, and to bypass that issue, Lua has a rail built into the belt that can be pulled outside.

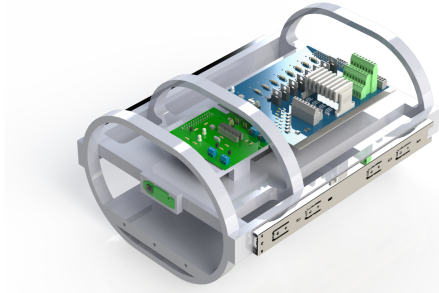
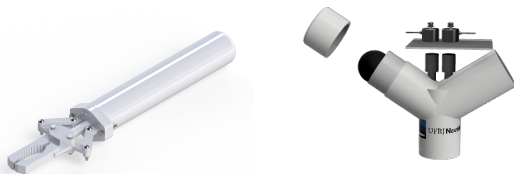


Fig. 3: Internal Compartment

6) *Hydrophone support*: Attached to the frame the hydrophone support is responsible for housing our hydrophones. This sub-project was initially modeled for our previous AUV but during the process of making Lua, how our software team couldn't test in real life the best angle necessary, we had to remodel it in a way that we could be prepared to any angle.

7) *Mechanical arm*: This year the team started developing two new subprojects that would enable our AUV to perform the remaining tasks and one of them is our mechanical arm/ gripper. Until this moment the mechanical and electrical project is ready where the main idea was that its functioning would be given through the use of a linear actuator.

8) *Markers Dropper*: Also using a linear actuator we get to our second subproject, our dropper. This subproject is already a part of our skills set and has a "Y" shape and two linear actuators to have the possibility of being used twice.



(a) Gripper

(b) Markers Dropper

Fig. 4: 3D Models generated with SolidWorks [1]

B. Electronics

All of the electronic part has been modified from our previous AUV with the objective to do our own systems and make an AUV more functional than the last one. The projects were made using Altium Designer [2]. Lua's hardware diagram is:

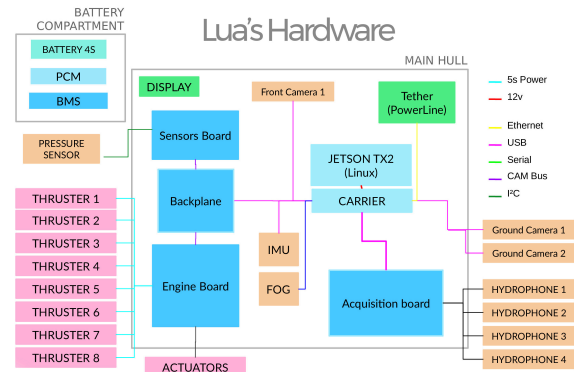


Fig. 5: Lua's electronics diagram

1) *Backplane*: The Backplane is the main change in the electronics area of the AUV compared to the previous model. This board is responsible for the power distribution of the Main Hull's internal components and the internal communication between the main computer (Jetson TX2) and the subsystems (propulsion system, sensor management). The power distribution system is composed of voltage regulators and its protection against some kind of surge, thus constituting a source similar to a computer. The communication part has an interface with a microcontroller communicating to Jetson with the UART protocol.

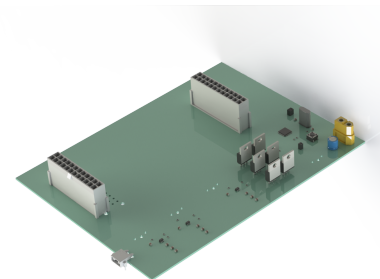


Fig. 6: Backplane PCB

2) *Thrusters PCB*: The thrusters board has been completely redesigned, with improvements in both the control and the electrical protection. The power is supplied using an ATX24 connector, due to its robustness against the high current required by this system. On this board, a system to read the thrusters current with ACS712 IC was also inserted. The control is done through an ATmega 2560 sending a PWM signal to the ESCs that will be mechanically coupled to the board. And it will be protected against overload and short circuits using MOSFETs, zeners and bipolar diodes.

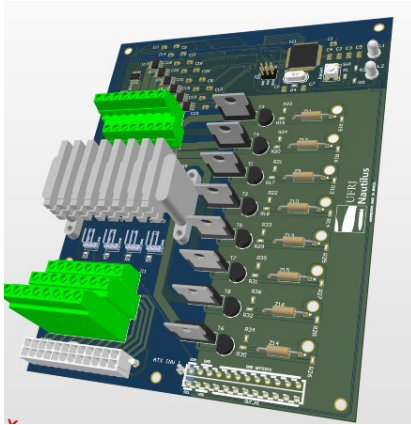


Fig. 7: Thrusters PCB

3) *Sensors PCB*: The sensors board is responsible for the management of the external and internal sensors and the management of the actuators (marker dropper, torpedo and mechanical arm). This board is modeled as an embedded system, where the central control is done through an ATmega 328P microcontroller and the whole peripheral system (such as bootloader and clock with a 16 MHz crystal) is responsible for the board communication with the central computer through I²C communication. The attached sensors are BAR30 (external pressure and depth sensor), BMP180 (internal pressure and temperature sensor) and the leak sensor. The pressure sensors are connected directly to the controller's I²C system, while the leak sensor has a switching system that sends a signal only when water is detected. The torpedo system is composed only of one BJT NPN transistor that works as a switch activating a solenoid pneumatic valve, where the digital signal just passes through the base, opening the circuit. The marker lance is driven by

an H-bridge, the L298N, which drives two linear actuators only receiving a digital signal at the input, determining motor's direction of rotation. And finally, the mechanical arm, as it has a separate circuit, the sensor board only sends a PWM signal and power to its own system.

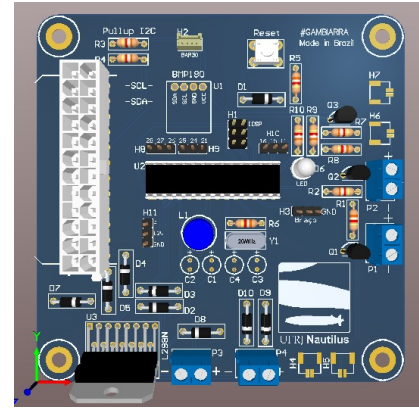


Fig. 8: Sensor PCB.

4) *Hydrophones PCB*: The hydrophones board consists of four stages. The first of which is the pre-amplification of the signal to get an amplitude compatible with the converter's full scale. In the next stage, the signal is filtered in a buterworth topology filter to eliminate any noise. Then an analog digital converter digitizes the signal, so that it can be processed by the controller and sent to the Jetson.

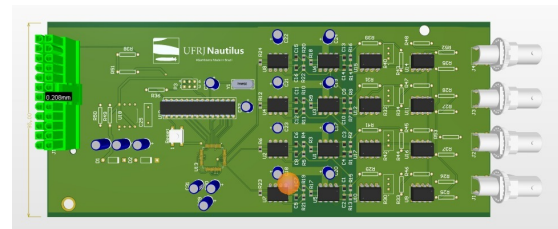


Fig. 9: Hydrophones PCB

5) *Battery Management System (BMS)*: The BMS is the protection system around the battery, and consists of two parts: the PCM (protection circuit module) and the telemetry. The PCM is responsible for balancing the battery cells during discharge. This system is basically composed of switches made with MOSFETs and relays and capacitors that prevent the voltage level from oscillating. The telemetry is the part responsible for

monitoring the battery level shown on an LED display. This circuit is quite simple, involving the use of an ATmega 328P microcontroller and three resistors establishing a mathematical relationship that makes it possible to read a high voltage from the analog pin. In the BMS is also found the main killswitch that allows the disconnection of all the electronics of the AUV.

C. Software

The embedded software in Lua was made to be more reliable and stable than our last vehicle software while improving the AUV navigation and autonomy capabilities. To achieve these goals the software team made several improvements in the simulation environment, writing a new state machine and adapting our control system to work with two more thrusters. The Lua software stack is the following:

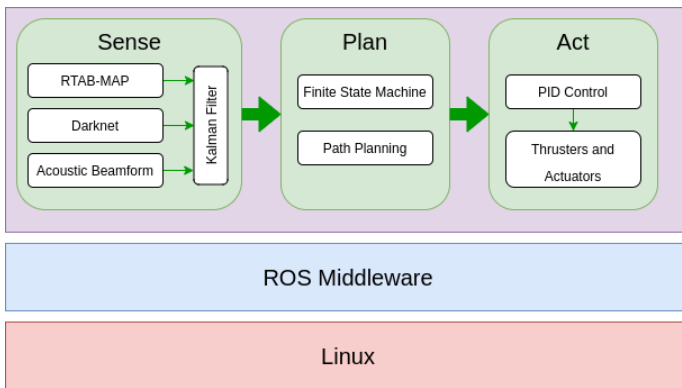


Fig. 10: AUV software stack.

1) *SLAM*: In the previous Robosub event, we managed to implement sensor fusion using stereo odometry from cameras with Rtab-map [3], depth altitude from the pressure sensor, and orientation from the IMU. The biggest issue with this system was that if Rtab-map loses tracking of the pool bottom we lose the only source of odometry available, leading to big errors being fused in the Kalman filter. To address this issue we implemented an algorithm to fuse the data from the previously made beamforming algorithm to the Kalman filter. With the azimuth and elevation angles obtained by the beamforming algorithm, we can easily find a unit vector in the direction of the pinger, but to exactly create a transform from the hydrophones to

the pinger position it's necessary to scale that vector by the depth divided by the cosine of the angle between a down vector and our facing direction vector. To calculate this down vector from the frame of reference of the hydrophone we simply rotate a $(0, 0, -1)$ vector with the help of the orientation given by the IMU and the hydrophone orientation relative to the IMU. Having this information we can navigate with some degree of accuracy even if the Rtab-map lost track of the pool bottom.

2) *AI*: The artificial intelligence of the robot is responsible for the top-level decision making of the software architecture of the AUV. This year we focused our efforts on making new states for the state machine embedded in the robot. Following the improvements in the localization capabilities of the vehicle using visual odometry from Rtab-map, we started to implement the necessary states to accomplish our first actuator task, dropping a marker in the bins. To do that, the implemented algorithm uses the image of the bottom camera to recognize the bins using the YOLOv4 [4] neural network. With the neural network output our state machine aligns the robot with the bin and then drops the marker.

3) *Control*: One of the biggest challenges faced when implementing a new control system for the AUV is modeling and simulation of the system, in order to do that we updated our simulation environment in the gazebo [5] simulator to utilize the uuv_simulator [6], a set of packages that include gazebo plugins and ROS [7] packages for the simulation of underwater vehicles. After that we started to estimate the vehicle hydrodynamic parameters from Fossen [8] robot-like hydrodynamic model implemented in this package. To do that our hydrodynamics and mechanics team started to simulate the robot utilizing CFD software to estimate the robot parameters for the simulation. To implement the control system in our new AUV we used the previously implemented PID controller from our last vehicle in this new simulation environment, tuning the PID gains and addressing the new thrusters in order to get the new vehicle running in all six degrees of freedom. Also we slightly modified the control output, to utilize the thruster manager from uuv_simulator to generate the control signal to command each thruster.

4) *Path Planning*: The path planning algorithm we developed was one of the biggest improvements we made this year. Previously, our control system takes a pose as a goal, and then he moves the robot to goal without carrying on how to do this in terms of the robot orientation. For Lua, we developed a new path generation software using Bézier curves [9] to solve this issue. Using this software we can pass each iteration of the curve to the control system so the submarine can navigate in a smoother and more predictable way. This approach made moving the AUV between tasks in the pool much easier, as we can simply put the desired position and orientation and the path planner will do all the heavy lifting for us.

III. Experimental Results

A. Hydrodynamics and Mechanics

Given the COVID-19 scenario, the hydrodynamics and mechanics team had to find a way to test our projects in order to be aware of the problems that can appear whenever it becomes possible to safely build our AUV. Because of that, we had to focus on our simulations both on a structural and on a hydrodynamics matter.

1) *Finite element analysis(FEA) simulations*: The structural simulations were made to choose our materials and to understand how they would behave in the long term knowing the forces that would be applied to them. Three parts went through a finite element analysis simulation to find the deformation caused and in all of them we simulated the UHMW polymer and the aluminum response .

- **Internal Compartment:**
This simulation was made considering the weight of the electronic part.

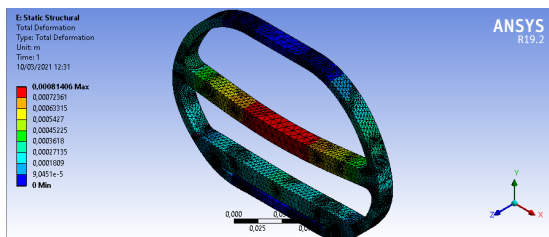


Fig. 11: Internal Compartment Deformation

- **Belt Clamp:**
In this part, we wanted to see the possible

deformation at the clamp that joins our belt to our support arm in order to know what was the better size to the piece. The applied force here was the weight of the hull, (that includes the weight of the "belt" the electronic part and the combined weight from the acrylic and the exoskeleton).

- **Support arms:**
In this part, several simulations were made with different width dimensions for the lateral arm and with the force applied in different places to analyze the total deformation and the equivalent stress in each case, and thus, dimension the arm in the best way. The forces here were the weight of the main hull and the resultant forces caused by the thrusters.

The UHMW case showed less deformation in all of the parts listed above and because of that it was chosen as the main material of our AUV.

2) *Computational fluid dynamics(CFD) simulation*: Focusing on enabling our software team to reach a better control system, we needed to know some important information about the project such as the drag coefficient of our AUV. To obtain that data, a computational fluid dynamics (CFD) simulation was made using the Autodesk CFD [10] program.

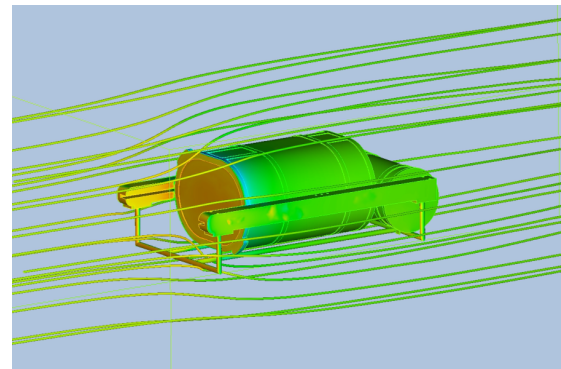


Fig. 12: CFD simulation

B. Electronics

Our tests were made using simulation softwares like Falstad Applet [11] and Proteus [12]. Only battery tests were made in the lab.

1) *Torpedo Electronic System Simulation*: It was simulated in the Falstad application, where the actuation current of the pneumatic valve to be used was

measured with an expected value of 0.54 A. In the simulator, a resistor with a value approximating the electrical resistance of the valve ($R = 22.15 \Omega$) was added. The current found was 0.533 A, showing a satisfactory result with an error of 1.3% in relation to what was expected.

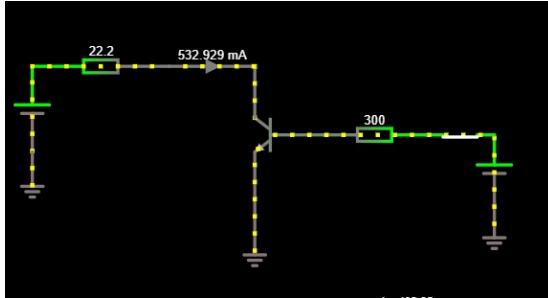


Fig. 13: Torpedo simulation

2) *Hydrophones PCB*: For the simulation portion of the hydrophone acquisition board, we used Falstad's applet online circuit simulator to check the performance of both, the bandpass filter and the two amplification stages. Regarding the former, we used a Bessel topology with a lower cutoff frequency of 19kHz and an upper bound of 46kHz; it also had an expected gain of 4V/V. As for the results observed, the filter responded as expected, rejecting frequencies outside the bandwidth, while amplifying the desired frequencies by a little less than 4x, which is to be expected, as some losses prevent the gain from reaching the theoretical value. As for the two preamplifier stages, the first one being a differential amplifier (needed due to the nature of the signal received from the hydrophones), which did, according to the expectation, amplify the signal tenfold. The second stage consisted of a simple 2x amplifier, also working as expected in the simulation.

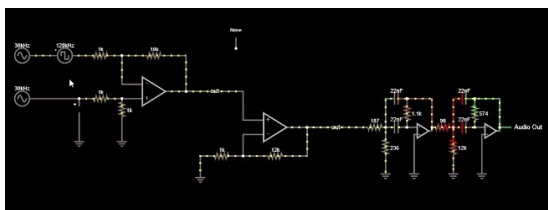


Fig. 14: Hydrophones simulation

3) *Markers Dropper*: To simulate this behaviour, we've written a code that periodically emits a signal,

as if the AUV were receiving the stimulus required to release a ball. That signal was "received" by the ATMEGA, which emitted the HIGH and LOW signals for a few seconds, and then turned them off. Consequently, the h-bridge activated a motor, which stood for the solenoid in the real robot, and after a moment, turned it off.

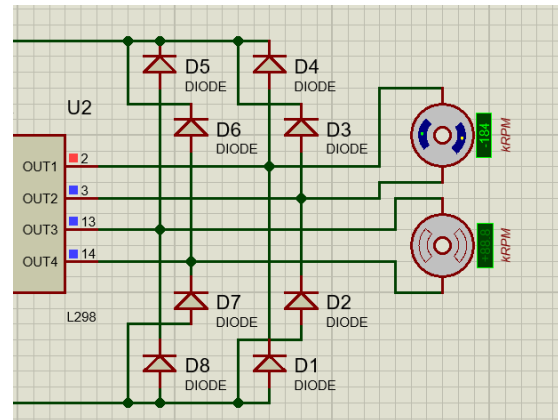


Fig. 15: Markers Dropper simulation

4) *Battery Monitoring*: This system wasn't simulated in a software, it was tested in the lab. We used, an Arduino Uno, nine resistors and one 5S battery. The expected result was 20.1 V and the measurement showed 19.91 V. This result was satisfactory with an error of 0.94%.



Fig. 16: Battery monitoring

C. Software

One of the biggest challenges of this new AUV is the addition of two more thrusters, to make possible the simulation and test of such vehicle during the COVID-19 pandemic the software team started the development cycle of Lua updating the simulation environment to use the `uuv_simulator` packages also

took advantage of this new cycle to update all of our software packages, mainly the operating system that runs on our embedded computer but also the Python [13] and ROS distributions. All results presented in this section were made using this new simulation environment.

1) *SLAM*: This year we worked on expanding our SLAM capabilities in order to use the newly created path planning system. To use this new path planning system we need a reliable source of odometry. We are able to achieve this by fusing data from IMU, depth sensor, Rtab-map, and beam-form using an Unscented Kalman Filter from the robot_localization [14] package.

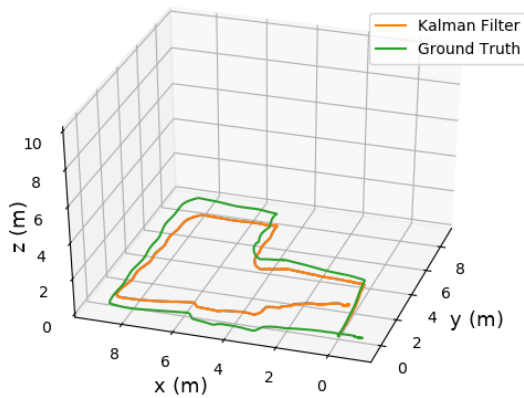


Fig. 17: AUV position estimaion.

So using the sensor fusion data from Kalman filter seen in fig.17 with the map produced by Rtab-map we are able to navigate in the pool.

2) *AI*: We developed and tested the new state machine to perform the bin task in our gazebo simulation environment, developing an integration test to validate this new state. Basically, we simulate the response from Darknet inside the camera feed of Gazebo, feeding the state machine. When the robot sees the Dropper label it starts to place itself over the bin, centralize the image camera feed, and after that release the marker.

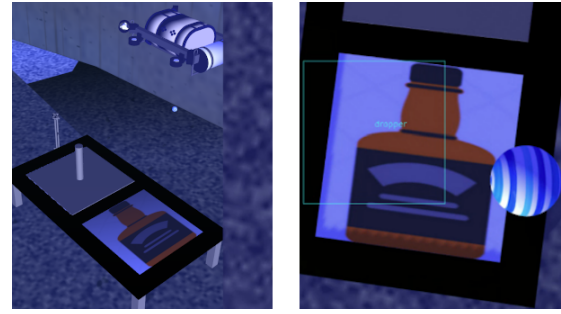


Fig. 18: Lua performing the bin task.

3) *Path Planning*: The path planning method using Bézier curves developed this year was able to generate a path for the vehicle to follow, between the vehicle position and the desired point, taking into account the AUV current orientation and the orientation desired for the vehicle in the final point.

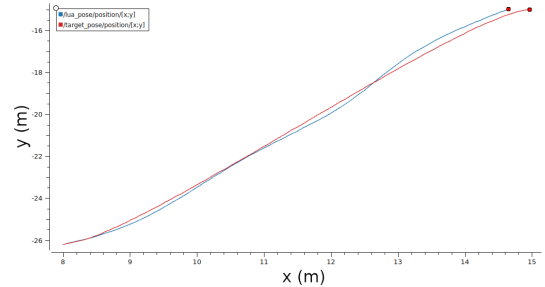


Fig. 19: Lua following the Bézier path.

Using the developed path planning software the AUV was able to navigate to the final pose maintaining a close distance from the desired path as seen in the fig. 19.

4) *Beamforming*: We kept improving the beamforming [15] algorithm developed for the last vehicle. We proposed a method that considers the error rate of the beamforming in its angle of arrival estimation. A neural network managed to predict the beamforming's error very well on synthetic data, even with added noise, bringing the error down by more than 60%. It did generalize well into a real-world dataset, however, not managing to bring the error down in any meaningful way. Grouping angles together (clustering) and calculating their median error proved to be a way to bring the error down by around 20% in the synthetic dataset, while in a real dataset it managed to improve the error rate by more than 10%. The results below are a comparison

between the mean absolute error of the original fast beamforming and the new versions.

Beamforming	Neural Network	Clusterization
9.88	12.18	9.36
0.00%	-23.28%	5.26%

TABLE I: SELD [16] Dataset Evaluation Mean Absolute Error - Azimuth

Beamforming	Neural Network	Clusterization
16.45	5.57 ± 0.38	13.18 ± 0.13
0.00%	66.14%	19.89%

TABLE II: Synthetic Dataset Evaluation Mean Absolute Error - Azimuth

Acknowledgment

The UFRJ Nautilus team was created in 2016 and since then we have been climbing a ladder where each step is taken by one of our members, pushed by one of our sponsors, and supported by our advisor Claudio Miceli de Farias. We would like to thank each and every one of them individually but words could never express our full gratitude and because of that, we finish our 2021 Robosub journey giving everyone the biggest love that you can imagine. And for the world, we wish better days.

References

- [1] "Solidworks." [Online]. Available: <https://www.solidworks.com/>
- [2] "Altium designer 19." [Online]. Available: <https://www.altium.com/solutions/academic-programs/student-licenses>
- [3] M. Labbé and F. Michaud, "Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of Field Robotics*, vol. 36, no. 2, pp. 416–446, 2019.
- [4] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection. 2020," *arXiv preprint arXiv:2004.10934*, 2004.
- [5] Open Source Robotics Foundation, "Gazebo simulator." [Online]. Available: <http://gazebo.org/>
- [6] 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*. IEEE, sep 2016. [Online]. Available: <https://doi.org/10.1109/Oceans.2016.7761080>
- [7] Stanford Artificial Intelligence Laboratory et al., "Robotic operating system." [Online]. Available: <https://www.ros.org>

- [8] T. Fossen, *Guidance and Control of Ocean Vehicles*. Wiley, 1994. [Online]. Available: <https://books.google.com.br/books?id=cwJUAAAAMAAJ>
- [9] J.-w. Choi, R. Curry, and G. Elkaim, "Path planning based on bézier curve for autonomous ground vehicles," in *Advances in Electrical and Electronics Engineering-IAENG Special Edition of the World Congress on Engineering and Computer Science 2008*. IEEE, 2008, pp. 158–166.
- [10] "Autodesk cfd." [Online]. Available: <https://www.autodesk.com/campaigns/cfd2020>
- [11] "Falstad circuit simulator applet." [Online]. Available: www.falstad.com
- [12] "Proteus." [Online]. Available: www.labcenter.com
- [13] "Python." [Online]. Available: <https://www.python.org/>
- [14] T. Moore and D. Stouch, "A generalized extended kalman filter implementation for the robot operating system," in *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*. Springer, July 2014.
- [15] S. Simplicio, H. J. dos S. Ferreira, G. Villela, F. B. Costa, V. Pavani, L. Rodrigues, and C. M. de Farias, "Development of the ufrj nautilus' auv: A multisensor data fusion case study," in *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, 2020, pp. 1–8.
- [16] S. Adavanne, A. Politis, and T. Virtanen, "A multi-room reverberant dataset for sound event localization and uetecion," in *Submitted to Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, 2019. [Online]. Available: <https://arxiv.org/abs/1905.08546>

Appendix A
Table of components

Component	Vendor	Model/Type	Specs	Cost/ If new
Buoyancy Control	-	-	-	-
Frame	-	UHMW(Ultra High Molecular Weight)	-	-
Waterproof Housing	-	Costum format Acrylic	-	-
Waterproof Connections	Bulgin	PX0931, PX0800, PX0931, PX0800, PX0805, PX0911	IP68, 6 pins	Donated
Thruster	BlueRobotics	T200	5.25 / 4.1 kg f	6 Re-used/ 2 new - \$360.00 USD
Motor Control	BlueRobotics	BasicESC	R3 30A	6 Re-used/2 new - \$54.00 USD
Controllers	N/A	ATMega 328p/ATMega 2560	IC MCU 8BIT 32KB FLASH 28DIP	\$43.44 USD
Actuators	-	-	-	-
Propeller	BlueRobotics	T200 Propeller	Diameter: 76.2mm	-
Battery	MaxAmps	LiPo Battery	5200 mAh 4S	\$57.71 USD
Converter	-	-	-	-
Regulator	-	Backplane Custom Made	-	-
CPU/GPU	NVIDIA	Jetson Tegra X2	Cortex-A57, 8gb 128-bit LPDDR4, 256 CUDA core	Re-used
Internal Comm Network	-	I2C/UART/TTL Serial	N/A	N/A
External Comm Interface	-	Ethernet	100Mbit	-
Programming Language 1	-	C++	-	-
Programming Language 2	-	Python	-	-
Compass	Xsens	-	-	-
Compass	Parker Lord Microstrain	-	-	-
Depth sensor	Blue Robotics	BAR30	Operation depth: 300m	\$72.00 USD
Inertial Measurement Unit(IMU)	Xsens	MTi-G-AHRS	Triaxial accelerometer, triaxial gyroscope	-
Inertial Measurement Unit(IMU)	Parker Lord Microstrain	3DM-CX5-10	Triaxial accelerometer, triaxial gyroscope	-
Cameras	Logitech	C270	720p /30 FPS	Re-used
Hydrophones	Benthowave	Bii-7141	Sensitivity@1kHz (dBV/ μ Pa): -202 \pm 2	-
Manipulator	-	-	-	-
Algorithms: vision	-	YOLOv4	-	-
Algorithms: acoustics	-	Beamforming	-	-
Algorithms: localization and mapping	-	Rtab-map and robot localization	-	-
Algorithms: autonomy	-	PID and FSM	-	-
Open Source Software	-	ROS and Linux	-	-
Testing Time: simulation	250h	-	-	-
Testing Time: in water	-	-	-	-
HW/SW expertise ratio	17:11	-	-	-
Team Size	43	-	-	-