Progress and Fabrication of UFRJ Nautilus' AUV: Lua

Victor P. Siqueira
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
victorprado@ufrj.br

Lidia G. Paúra
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
lidiapaura@ufrj.br

Augusto L. Rodrigues
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
lauande@ufrj.br

Felipe O. G. da Silva
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
felipe.oliver.gomes@poli.ufrj.br

Matheus R. de Souza
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
matheusrodrigues@poli.ufrj.br

Claudio M. de Farias
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
cmicelifarias@cos.ufrj.br

***Abstract*—This year the UFRJ Nautilus team has been dealing with the challenge of confectioning the Lua project. One of our greatest improvements is acquiring a DVL sensor and integrating it into the AUV system, which has been one of our goals since 2016. The team has been focused on going back to in-person work and building the Lua project which is technically ideal but has proven to be quite the challenge. Its new mechanical design based on an elliptical shape is hard for machining companies to accept the manufacturing order. The electrical work centered on the assembly of the PCBs designed last year, and the revision of the electrical monitoring and protection system. For the software, we have improved our control system and the integration of the mechanical arm and the DVL sensor.**

I. Competition Strategy

In last year's competition we had just started to develop the simulation for our new actuators that now can be included in our new AUV, this year, with the further developed robot and a new sensor, we intend to follow the sequence of tasks described below, choosing the Bootlegger side of the competition.

1) With Moxy (Coin Flip): According to the result of the coin flip Lua will be placed on the pool with the given orientation and guide herself towards the gate using data from the IMU, hydrophones, and DVL to confirm the correct heading.

2) Choose Your Side (Gate): Initially, our neural network detects the gate, hence, activating the gate state of our state machine, in this state, the AUV will align itself with the center of the gate and use the neural network again to find the bootlegger image. After that, the AUV uses the IMU and control system to align the z-axis with the correct path and go through the gate.

3) Style: While passing through the gate in the Choose Your Side task the AUV will use its control system fed mainly by our IMU and DVL to perform a 720º roll. This means it will rotate completely twice on the x-axis, which is possible due to the six degrees of freedom Lua has been designed with.

4) Path: To stay in the direction determined by the paths positioned in the RoboSub Pool, the AUV will use the image delivered to our neural network, calculate the center of it and use this value to correct the AUV orientation using our control system.

5) Make the Grade (Buoys): After passing through the first path, Lua will use its neural network to detect the Bootlegger buoy, calculate the center and use its control system to move and touch the chosen buoy.

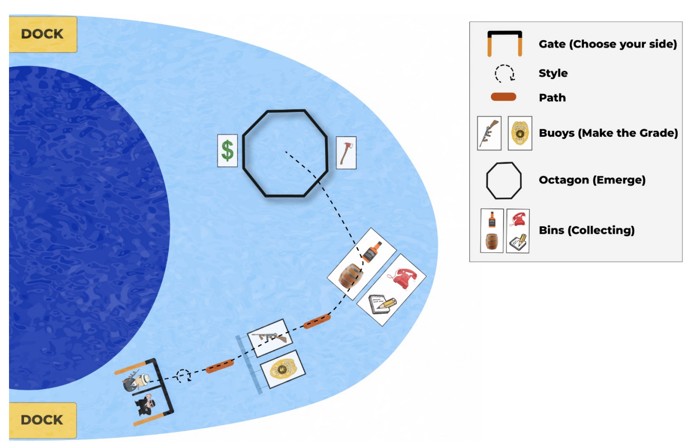

Fig. 1: 2022 Competition Strategy

6) Collecting (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-B2 and the algorithms at III-C2.

7) Cash or Smash (Octagon): Ending our course, we use the signals coming from the pinger in the bottom of the octagon to guide the AUV towards the bottles. When it gets to a certain depth, for each bottle it uses the gripper to catch it, moves upward, uses the neural network to find the bootlegger table, uses the control system to move towards the goal, and drops it. In the end, it emerges through the octagon using the signals from the pinger to align its orientation.

II. Design Creativity

A. Hydrodynamics and Mechanics

This year, we prioritized the manufacture of the AUV. We analyzed the best materials for each function, aiming for an overall low density and a hull adequately designed for final fabrication, while still following our initial goals. We iterated on each part of Lua to make the project more efficient and effective. In the same way, the alignment and position of the thrusters were finalized, in an arrangement that allows her to move in 6 degrees of freedom.

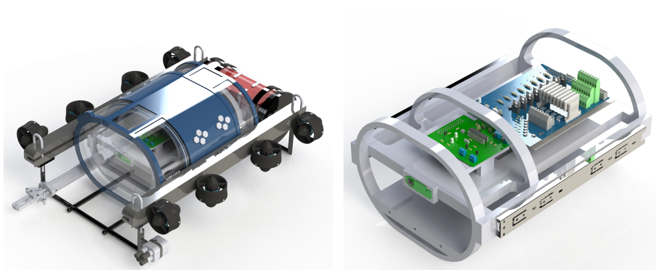

Fig. 2: Lua's SolidWorks [1] rendering

1) Frame and Support arms: The frame was our choice for coupling the actuators, a structure responsible for ensuring the best possible stability. It has a rectangular structural base that will wrap around the bottom of the Lua's symmetrically. Made with anodized aluminum profiles, this material was chosen for factors such as corrosion and mechanical resistance. The bars are fixed by aluminum angle brackets. The frame is connected to a structural arm, a polymeric component, which supports the entire weight of the Belt. The thrusters will be coupled to the arm, thus enabling the hydrodynamic organization of the AUV.

2) Main Hull: Acting as the main piece of our vehicle, our so-called belt was built to serve to connect the two acrylics and as an interface between the external part of the AUV and the internal compartment, through subsea connectors.Due to its high complexity, we chose to 3D print it in PET-G, and apply an epoxy layer to make the part watertight. This part together with the two acrylic lids and two exoskeletons constitutes our main hull. The exoskeleton function as a connection with the belt while strengthen the acrylic structure.

3) Internal Compartment: The main objective when modeling an internal compartment for Lua was to make practical any changes that might occur to the inner components during tests and to ensure that the PCBs remain organized. Therefore, we facilitate access to the AUV's electronic part through the use of slides between the internal compartment and the belt. Furthermore, we fixed our PCBs on acrylic sheets, which in turn are being supported by 3D printed PETG hoops.

4) Mechanical arm: To allow us to perform object manipulation tasks, a gripper/mechanical arm was designed last year to use a 3D printed linear actuator, controlling a claw, therefore being able to grab or release objects. This year the project has been slightly updated in terms of materials to accommodate current fabrication capabilities.

5) Markers Dropper: A stepper motor turns a screw engaged to a 3D printed structure that is able to store two markers. The screw moves a kind of "drawer" which aligns the marker with a release hole that allows it to fall due to gravity.

B. Electronics

All of the electronics have been modified from our previous AUV with the objective of advancing our systems and making an AUV that is more functional than the last one. Since last year, most sections have remained the same, with the exception of the BMS and other slight improvements. The projects were made using Altium Designer [2]. Lua's hardware diagram is

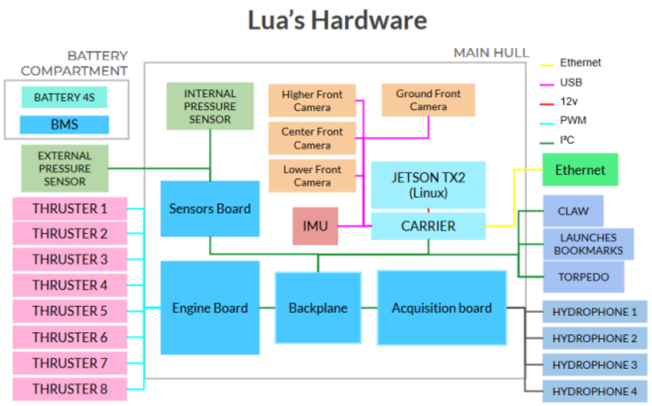

Fig. 3: Lua's electronics diagram

1) Backplane: The Backplane is the main change in the electronics of the AUV compared to the previous model. This board is responsible for the power distribution (through voltage regulators) of the Main Hull's internal components and the internal communication between the Jetson TX2 and the subsystems. The communication functions through the I2C and UART protocols.

2) Battery Management System (BMS): The BMS is the protection system connecting the battery to the robot, consisting of two parts: the protection circuit and telemetry. The former is responsible for controlling battery discharge and voltage. This system is uses MOSFETs to prevent the voltage level from oscillating. The telemetry is responsible for monitoring the battery level, measured with a voltage divider. Lastly, a killswitch allows power to be cut entirely.

3) Thrusters PCB: The thrusters board has been completely redesigned from our past robot, with improvements in the control and 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 is also included. The motor speed control is done through an ATMega 2560 sending a PWM signal to the ESCs.

4) Hydrophones PCB: The hydrophones board consists of four stages. The first of which is the pre-amplification of the signal to get to a better amplitude. 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 and sent to the Jetson.

5) Sensors PCB: The sensors board is responsible for the management of external and internal sensors and of the actuators. The central control is done through an ATMega 328P microcontroller which is is responsible for the board external communication through I²C. Included are: BAR30 (external pressure sensor), BMP180 (internal pressure), and the leak sensor. The marker dropper is driven by an L298N, driving two linear actuators. Lastly, the mechanical arm has a separate circuit, so this board only provides a PWM signal and 12V power.

C. *Software*

The software in Lua was extended to integrate our new DVL sensor and the Cash or Smash task. To achieve this goal, we improved the simulation environment, adding plugins that simulate the behaviour of our new sensor and integration tests to help implement our solutions of the tasks.

1) State Machine: The state machine is the part of the software architecture responsible for the top level decision making. In this year, we implemented the necessary states to accomplish the Cash or Smash task. The implemented algorithm consists in five states, the first four ones are responsible to pick each of the bottles and place it on the bootlegger table. The last one, manages to align the AUV with the center of the octagon and emerge it.

2) SLAM: Our Simultaneous Localization and Mapping system integrates data from the stereo odometry cameras, rtap-map [3], depth sensor, IMU and the beamforming algorithm. The addition of the latter into the SLAM system was motivated by the possibility of the rtab-map losing track of the pool bottom, leaving us with no source of odometry and leading to big errors being passed into the Kalman filter. By fusing the azimuth and elevation angles derived from the beamforming algorithm with the Kalman filter, we made the navigation of the robot much more reliable.

3) Control: Our control system, previously, did not have the linear velocity information from any of our sensors, in the past, we used the result of the linear integration of the linear acceleration as linear velocity, the problem with this approach is that the result did not had an accuracy that matched our needs for the competition. So, to solve this, we acquire and integrate it a DVL sensor into our project. In order to integrate it properly, we added a DVL sensor plugin, from the uuv_simulator project [4], which is a set of packages that include Gazebo plugins and ROS [5] packages for the simulation of underwater vehicles, in our Gazebo [6] simulation environment, with this, we were able to further tune the parameters of our control system and develop the Cash or Smash task.

4) Path Planning: The path planning algorithm was developed using the concept of the Bézier curves [7]. With this approach, every iteration of the ongoing AUV trajectory is sent to the control system, resulting in a much smoother and accurate trajectory.

III. Experimental Results

A. Hydrodynamics and Mechanics

The hydrodynamics and mechanics team test our projects in order to be possible to safely build our AUV, for that, we had to focus on simulations both on a structural and 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.

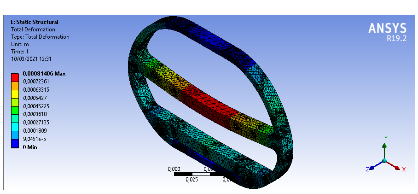

Fig. 4: Internal Compartment Deformation

- Belt Clamp:
  We tested the possible deformation at the clamp that joins our belt to our support arm in order to know what was the better size for the piece. The applied force here was the weight of the main hull.
- Support arms:
  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. 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 [8] program.

B. Electronics

Our tests were made using simulation software like Falstad Applet [9] and Proteus [10]. Only battery tests were made in the lab.

1) 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 is 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.

2) Markers Dropper: To simulate this behavior, 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.

3) Battery Monitoring: This system wasn’t simulated in 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%.

C. Software

In this year competition we had two challenges, the integration of our new DVL sensor and the development of the Cash or Smash task. In order to accomplish this, we improved our simulation environment by adding plugins to simulate the DVL sensor from the uuv_simulator project. Furthermore, we also developed two integration tests to simulate the response of a object classification neural network and our own beamforming algorithm. The environment of the tests below, is the latter described.

1) SLAM: The fusion between the beamforming algorithm and the Kalman filter was implemented using the azimuth and elevation angles obtained by the latter, which are used to create a unit vector in the direction of the pinger. To exactly create a transform from the hydrophones to the pinger position, it’s necessary to scale the unit vector by the depth divided by the cosine of the angle between a down vector and our facing direction vector. The down vector is calculated from the frame of reference of the hydrophone by rotating a (0, 0, -1) vector with the orientation given by the IMU and the hydrophones. This significantly improved our AUV’s location capacity.

2) State Machine: In our new states developed, meaning the Cash or Smash task, the AUV determines the location of the bottles by analysing the signals that comes from the pinger and using the YOLOv4 with images from the bottom camera. Furthermore, it determines the position of the Bootlegger table using the neural network and the proper position to emerge through the pinger signals.

3) Path Planning: The path planning system mas made capable of generating a accurate trajectory for the AUV to follow, between the vehicle original position and position and orientation goals. Also, the system takes into account the robot’s current orientation and position to make the trajectory even smoother.

4) Beamforming: Our beamforming [11] algorithm, developed to this year competition, takes into account the error rate of the beamforming in its angle of arrival estimation. We used a neural network to predict the beamforming’s error, and it went well on synthetic data, even with added noise, it managed to bring 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. We manage to bring the error down by 20% by clustering and calculating the median error in the synthetic dataset, Moreover, in the real world, this strategy manage to bring the error down 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 [12] 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

Now finally leaving dark times behind us, the Nautilus team aimed this year at the realization of many years’ plans. Lua has been a grail we’ve intended to reach for a long time, and with the help of our advisors and sponsors, particularly Claudio Miceli and Total Energies, it is now becoming reality. We can only thank all of those that allowed us to take part in something as big as Robosub, and wish all teams the best of luck this time around.

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] 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%2Foceans.2016.7761080
[5] Stanford Artificial Intelligence Laboratory et al., "Robotic operating system." [Online]. Available: https://www.ros.org
[6] Open Source Robotics Foundation, "Gazebo simulator." [Online]. Available: http://gazebosim.org/
[7] 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.
[8] "Autodesk cfd." [Online]. Available: https://www.autodesk.com/campaigns/cfd2020
[9] "Falstad circuit simulator applet." [Online]. Available: www.falstad.com
[10] "Proteus." [Online]. Available: www.labcenter.com
[11] 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.
[12] S. Adavanne, A. Politis, and T. Virtanen, "A multi-room reverberant dataset for sound event localization and uetection," 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	-	Aluminum Profile	-	-
Waterproof Housing	-	Custom 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	URUAV	LiPo Battery	10000 mAh 4S	$80.45 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	-	-	-
DVL	WaterLinked	DVL A50	0.05m – 50m, ± 0.1cm/s	$6 550.00 USD
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/µPa): -202±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	42	-	-	-