Technical Overview of the Development of Autonomous Submarines LANTURN and BL45T015E

Aren Petrossian (ME, Team Captain), Katherine Bonomo (CE, Team Co-Captain), Jennifer Serrano-Perez (CS), Adrian Salgado Lopez (CS), Victor Solis (CS), Jordan Doose (CS), Paola Reyez (ME)

1. ABSTRACT

The previous designs of our submarine robots are carefully reviewed, and significant changes are made to our vehicle, LANTURN, while a new generation vehicle BL45T015E (BLASTOISE) was designed and manufactured. (1) First, maintenance and troubleshooting are prioritized as the time available at the competition is very limited. This improvement is achieved through the latch design on both submarine robots for immediate access to the hull and internal electronics. (2) Second, modifications are made to our electronics for better performance and a custom made power distribution board is designed and manufactured. (3) Third, the software team creates an architecture that supports easy expansion depending on the tasks outlined by the competition. (4) Fourth, a feature-rich graphical user interface is designed and implemented to provide real-time feedback on the robot's status. Simulations and experiments show that these design creativities meet our expectations, which will help the team stay in a competitive position for this year's competition.

2. COMPETITION STRATEGY

Our competition strategy for BLASTOISE focuses on module intercompatibility and ease of access for all mechanical and electrical components. By ensuring all components can be easily installed and removed, and accessed if necessary for maintenance and testing, we can make changes as necessary during the competition without affecting major functions.

In addition our software approach focuses on modularity, code reusability, and future software development. BLASTOISE software is separated into modules which can be developed and tested independently, and will allow the software in each module to develop while maintaining compatibility with one another. The modularity of our software allows us to efficiently make adjustments to different modules at the competition without disrupting other software components of our system. The computer vision portion of the software is of particular note. The competition requires heavy use of object recognition and our goal is to have a functioning system with the ability to detect distance of objects and the angle the object is facing in relation to BLASTOISE.

3. DESIGN CREATIVITY

A. MECHANICAL

Key design features of BLASTOISE are a suitcase-like hull with a removable electronics shelving unit. BLASTOISE's hull primary structure is formed by TIG welded sheets of AL 6061-T6. BLASTOISE is outfitted with forward and downward facing cameras. 180 degree snapping latches attached to the front and backplate which are cut in half horizontally allow for easy maintenance and troubleshooting.

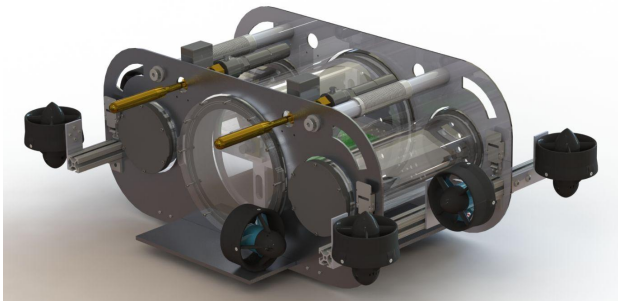

Figure 1. BLASTOISE Vehicle Isometric View

The electronics shelving unit is made of acrylic sheets laser cut to shape/size with mounting holes that span each sheet. An internal connector plate and connectorized electronics allow the shelving unit to be unplugged and removed from the hull.

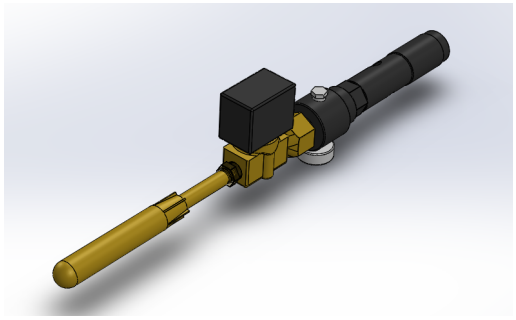

Figure 2. Pneumatic Torpedo Design

The torpedo design revolved around cold gas propulsion systems. Using small 12 gram carbon dioxide tanks, in an adapter for fitting purposes, this propulsion source could then be connected to our on/off switch. The torpedo was designed to slide over the barrel and then attach to an o-ring that would hold it in place until fired.

BLASTOISE's companion robot LANTURN, produced by the senior design component of RoboSubLA, uses an alternative hull design comprised of an isomorphic 6061-T6 aluminum block, milled to produce an internal cavity and mounting holes for penetrators and camera domes. LANTURN's single-component hull design produces a strong, spacious internal cavity, and will be further investigated for use in future robots.

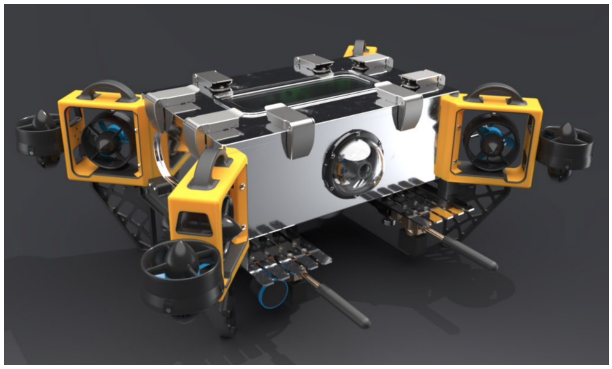

Figure 3. LANTURN Front 3-Quarter View

B. ELECTRICAL

The electrical system on BLASTOISE consists of a thruster signal routing board, power distribution board, microcontrollers, and a sensor suite, powered by a 14.8 V 20 Ah LiPo Battery.

After observation of previous robots, it was concluded that overheating is not an issue for AUV's and removed an active cooling system. BLASTOISE uses the system Jetson TX2 to deploy computer vision and machine learning applications. We tested that the desktop motherboard and GPU draws far more power and creates more heat than the Jetson TX2. With more internal volume than previous AUV designs we predict around 20 minutes before any signs of overheating issues. As a backup we explored a water cooling system, where an AFT bulkhead will isolate the components and attempt to keep the systems cooler for longer. This is also more attainable with the LiPo battery which rose from 10 Ah to 20 Ah.

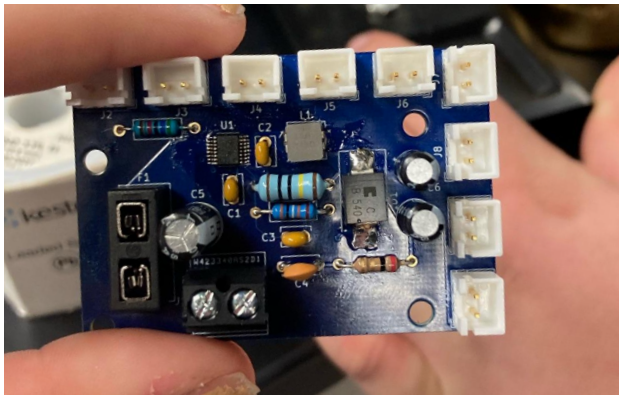

Figure 4. Physical Power Distribution Board

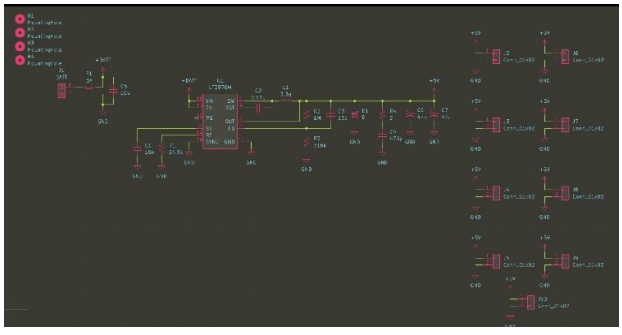

Figure 5. Regulator Schematic Diagram

In contrast to previous design, we selected a Jetson TX2 over the Raspberry Pi. We also selected the Teensy 4.1 over an Arduino, because it's faster, has a larger flash system, and we were able to add the cooling system above. This is also the first year we designed, printed, wired, and tested a custom made power distribution board.

C. SOFTWARE

The Jetson TX2 is designed with Autonomous Machines in mind and comes pre-flashed with Ubuntu 18.04. This operating system is used as the platform for ROS melodic, a software framework with a set of libraries and tools that allow for fast robot software development.

The software is broken up into three modules: mission planning, controls, and computer vision. Each module is a standalone program that communicates through the ROS API using the Publisher/Subscriber model to send and receive data. The sensor data is sent to the mission planning program running on the motherboard, and returns data to controls running on the microcontroller to control BLASTOISE's actuators. The communication system of the modules at the most abstract level is shown in Figure 6.

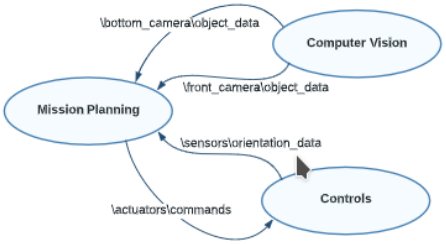

Figure 6. Communication System

A. Mission Planning

For mission planning, the software architecture that implements task execution is state machines. A python library, smach, can be used for fast prototyping of state machines along with having useful containers for different configurations of state machines. In BLASTOISE's case, there is a main state machine that holds multiple sub state machines in individual state machine containers. The main state machine has the competition strategy logic of which tasks will be executed and in what order. The sub state machines each hold the logic of one task execution. This structure has the benefits of allowing to test task transitions and individual task executions without running the whole program.

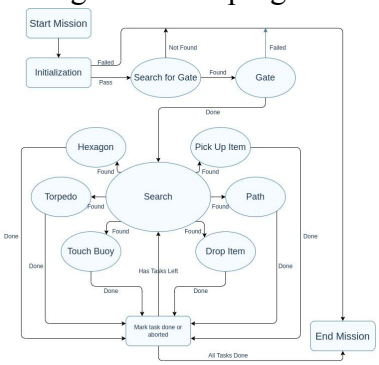

Figure 7. Main state machine

A sub state machine is entered when it has objects detected by the search state and it is the highest priority in the ordered list. Each sub state machine contains states that accomplish a small part of the task. The gate task state machine is illustrated in Figure 8. The gate sub state machine has states that are common for many other sub state machines, e.g. Align with [Object]. This modularity of the state machines allows for reuse of logic and code between state machines and therefore faster development, testing and debugging.

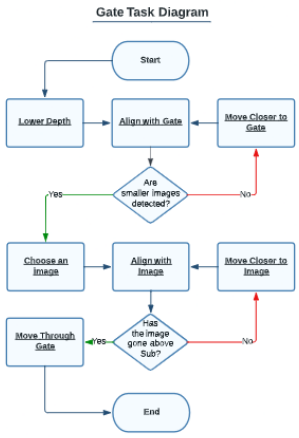

Figure 8. Gate Task Sub State Machine

B. Computer Vision

For Computer Vision, object detection is used and accomplished by our 4 step process: dataset creation, labeling, training, and implementation. With this the submarine is able to successfully detect, classify and localize objects.

We constructed objects to the specifications provided in the handbook and filmed them underwater to simulate an environment most similar to those anticipated at the competition. The addition of the consideration of videos of the objects taken out of the water was to diversify and broaden our dataset which was integral to improve and substantiate the vision of our vehicle. The footage took an accumulative 10 hours and were filmed from different angles, depths and distances from the objects. Utilizing OpenCV's Video I/O module, a module primarily handling the processing of video or image sequences within the open-source library, and Python, we extracted frames from the aforementioned videos to be used to annotate or label. Ultimately, we were able to obtain 1,100 images per object, with the distribution being 1000 underwater and the remaining out of water. To further improve the efficacy of our submarine's vision, we elected to perform image augmentations on ~20% of each object's frames to again variegate our dataset using OpenCV built-in algorithms and Python. The techniques used were blurring, rotating, altering brightness, saturation, and contrast.

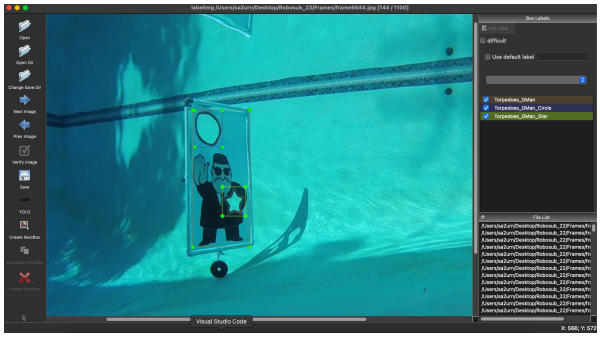

Figure 9. Labeling dataset

For training we used YOLOv4, You Only Look Once, a convolutional neural network (CNN) as our Object detection model. In order to generate the required annotation files, we employed the graphical image annotation tool, labelImg, to create bounding boxes, as shown in Figure 9, on our dataset. The annotation files were used to train Yolov4 on our dataset to produce custom weight files. Which are uploaded to the submarine's motherboard and implemented to detect, classify, and localize objects.

C. Controls

Running on an ARM Cortex-M7 microcontroller, BLASTOISE's 6DOF orientation and positioning control is provided by a custom software suite, integrated with movement commands from the state machine over ROS. The controls system consists of several individual PID controllers, implementing integrator anti-windup and circular error optimization, which are ultimately mixed to produce throttle values for each of BLASTOISE's eight thrusters.

In preparation for further tuning of BLASTOISE's control system, a two-part tuning ecosystem was developed using a smaller, six-thruster vehicle intended for early testing. The system consists of a feature-rich GUI, seen in Figure 12, providing real-time feedback on vehicle orientation, position, and thruster commands, and allowing live adjustment of tuning parameters on the active vehicle. Companion software, running on the vehicle's controls processor, responds to commands sent by the GUI and sends back vehicle telemetry at a configurable rate. This live tuning system significantly decreases the time required to stabilize a system, as effects can be immediately observed in thruster actions and telemetry outputs.

While our initial tuning testbed implements 5DOF control, implementing all axes of rotation as well as depth and forward/reverse translation, BLASTOISE includes two thrusters

perpendicular to the hull axis, which allows full 6DOF control and strafing movement. The similarity of thruster positioning, differing only in the two additional strafing thrusters, allows broad code reuse between the two systems and further reduces complexity.

Data collected throughout the remainder of vehicle testing and the competition will be used to develop and refine a more advanced controls architecture in future vehicles, including state estimation through the implementation of an Unscented Kalman Filter, true position control with the addition of a hydrophone array and doppler velocity log, and more advanced 6DOF control, which will allow translation in all axes regardless of vehicle orientation.

EXPERIMENTAL RESULTS

Hull

After many dimension changes were made for ease of manufacture, LANTURN's hull was analyzed using SolidWorks' static-stress analysis at a depth of 50 feet or 15 meters which corresponds to a pressure of 150,000 pascals. Under these conditions the smallest factor of safety found was 2.08, which was very satisfactory given the large overestimate of the depth.

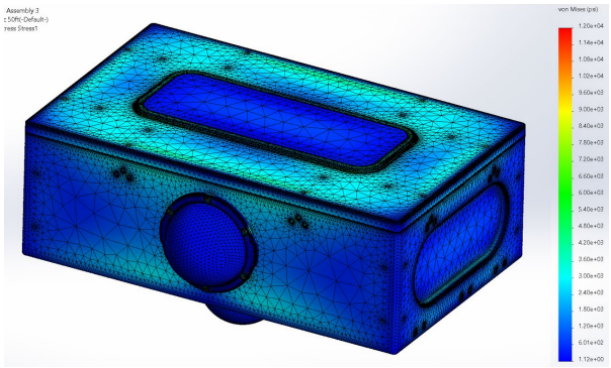

Figure 10. LANTURN 50ft Stress Analysis

Power Distribution Board

In agreement with the LTSpice simulation, BLASTOISE's custom regulator was observed to meet voltage ripple and startup timing parameters. Testing of the regulator was performed under three load conditions, including a 20mA LED, a 500mA microcontroller load, and a 2A shunt resistor, and all results indicate function within limits.

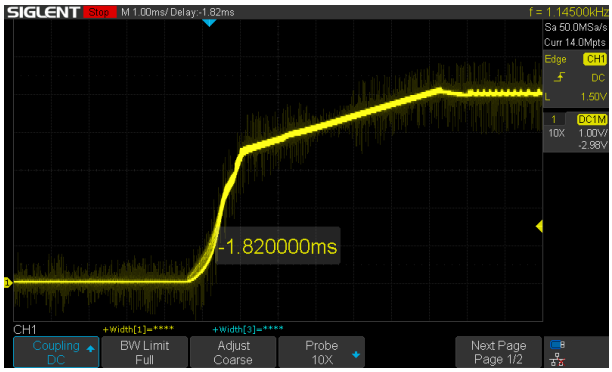

Figure 11. Regulator Startup Trace

Graphical User Interface

The custom UI developed for tuning our vehicles was tested with our 6 thruster testbed in preparation for tuning BLASTOISE. The inclusion of graphs helped greatly with observing any oscillations in vehicle state, and the ability to quickly load and modify tuning parameters in vehicle RAM, storing to EEPROM for future reuse in the controls software, allowed easy tuning over the course of one testing session.

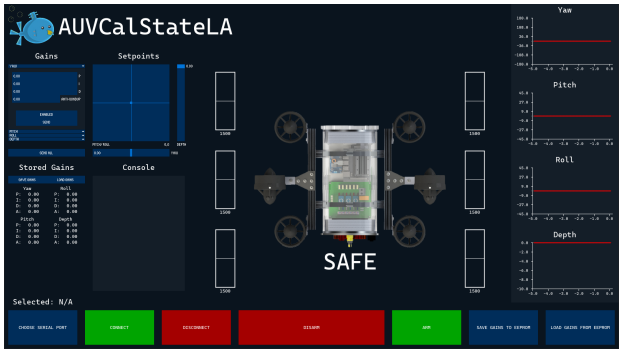

Figure 12. Graphical Tuning Interface

ACKNOWLEDGEMENTS

Our team would like to thank our faculty advisers, Dr. Mark Tufenkjian, Dr. He Shen, and Richard Cross for their valuable input to our project during review sessions and for their continuous support of the RoboSub team. We would also like to express our gratitude to the sponsors who supported the project: the CSULA department of ECST, Office of Naval Research (ONR), SolidWorks, and MathWorks. Thank you for making the development of our 2022 RoboSub vehicles possible!

REFERENCES

D. L. Poole and A. K. Mackworth, Artificial intelligence: foundations of computational agents. Cambridge: Cambridge University Press, 2018.

A. Anka, “YOLO v4: Optimal Speed Accuracy for object detection,” Medium, 16-Jul-2020. [Online]. Available: https://towardsdatascience.com/yolo-v4-optimal-speed-accuracy-for-object-detection-79896ed47b50. [Accessed: 10-June-2022].

J. Solawetz, “Breaking Down YOLOv4,” Roboflow Blog, 04-Mar-2021. [Online]. Available: https://blog.roboflow.com/a-thorough-breakdown-of-yolov4/. [Accessed: 10-June-2022].

Cengel and Cimbala, Fluid Mechanics, McGraw Hill, 2006.

Appendix A-1: BLASTOISE Component Specifications

Component	Vendor	Model/Type	Specs	COTS	Cost	Purchase Year
Buoyancy Control	N/A					
Frame	RoboSubLA	BRIEFCASE	N/A	Custom	$750	2022
Waterproof Housing	Blue Robotics	4” & 8” Hull	11.75” Length	COTS	$850	2022
Waterproof Connectors	TE Connectivity	SEACON	4 Pin	COTS	$600	2022
Thrusters	Blue Robotics	T200	5.25 Kgf max.	COTS	$1720	2022
Motor Control	Blue Robotics	Basic ESC	30A max.	COTS	$300	2022
High Level Control	PJRC	Teensy 4.1	ARM Cortex-M7 600MHz	COTS	$32	2022
Actuators	N/A					
Propellers	N/A					
Battery	Turnigy	20Ah 4S	12C, 336Wh	COTS	$150	2022
Converter	RoboSubLA	RAICHU	LT3976, 5V, 5A	Custom	$120	2022
Regulator	See Above					
CPU	NVIDIA	Jetson TX2	1.33 TFLOPS	COTS	$400	2022
Internal Comm Network	RoboSubLA	I2C Wiring Harness	N/A	Custom	$5	2022
External Comm Interface	Blue Robotics	Fathom-X Tether Interface	IEEE 1901	COTS	$250	2022
Compass	See Below					
Inertial Measurement Unit (IMU)	Bosch	BNO055	100Hz ODR	COTS	$35	2022
Doppler Velocity Log (DVL)	N/A					
Manipulator	N/A					
Algorithms	RoboSubLA	CROCONAW	N/A	Custom	Priceless	2022

Vision	OpenCV	YOLOv4				
Acoustics	N/A					
Localization and Mapping	N/A					
Autonomy	N/A					
Open Source Software	ROS	ROS Melodic				
Inter-Vehicle Comm.	N/A					
Programming Language(s)	Python, C++					

Appendix A-2: LANTURN Component Specifications

Component	Vendor	Model/Type	Specs	COTS	Cost	Purchase Year
Buoyancy Control	N/A					
Frame	RoboSubLA	XYLOPHONE	N/A	Custom	$300	2022
Waterproof Housing	RoboSubLA	BOX	N/A	Custom	$1200	2022
Waterproof Connectors	Blue Robotics	WetLink Penetrators	Various Cable Diameters	COTS	$400	2022
Thrusters	Blue Robotics	T200	5.25 Kgf max.	COTS	$1720	2022
Motor Control	Blue Robotics	Basic ESC	30A max.	COTS	$300	2022
High Level Control	Arduino	Mega	ATMEGA2560	COTS	$40	2022
Actuators	N/A					
Propellers	N/A					
Battery	Turnigy	20Ah 4S	12C, 336Wh	COTS	$150	2022
Converter	RoboSubLA	RAICHU	LT3976 Multi V	Custom	$300	2022
Regulator	See Above					
CPU	NVIDIA	Jetson TX2	1.33 TFLOPS	COTS	$400	2022

Internal Comm Network	RoboSubLA	I2C Wiring Harness	N/A	Custom	$5	2022
External Comm Interface	Blue Robotics	Fathom-X Tether Interface	IEEE 1901	COTS	$250	2022
Compass	See Below					
Inertial Measurement Unit (IMU)	VectorNAV	VN-100	800Hz ODR	COTS	$1000	2018
Doppler Velocity Log (DVL)	Teledyne	Pathfinder	±0.1 cm/s	COTS	$14000	2018
Manipulator	N/A					
Algorithms	RoboSubLA	CROCONAW	N/A	Custom	Priceless	2022
Vision	OpenCV	YOLOv4				
Acoustics	N/A					
Localization and Mapping	N/A					
Autonomy	N/A					
Open Source Software	ROS	ROS Melodic				
Inter-Vehicle Comm.	N/A					
Programming Language(s)	Python, C++					