Technical Design Report - Bears on Boats

Edward Ratliff
Electrical Engineering
US Coast Guard Academy
New London, CT, United States
edward.j.ratliff@uscga.edu

Ellie Hiigel
Electrical Engineering
US Coast Guard Academy
New London, CT, United States
ellie.k.hiigel@uscga.edu

Sierra Barbee
Cyber Systems
US Coast Guard Academy
New London, CT, United States
sierra.g.barbee@uscga.edu

***Abstract*- Our group is from the United States Coast Guard Academy's Electrical Engineering and Cyber Systems Departments and building the Autonomous Surface Vessel (ASV) serves as our senior capstone project. Our ASV is of a dual catamaran design, is battery-powered, has remote control capabilities, and utilizes the Robot Operating System 2 (ROS2) for autonomous decision-making. The propulsion system consists of two Blue Robotics T200 thrusters powered by lithium-ion batteries. The center of gravity of the ASV was adjusted by changing our vessel design to accommodate the weight of our new electronics and make the vessel more stable. Our team's strategy will focus on five specific tasks for the competition, "Navigate the Channel", "Follow the Path", "Docking", "Speed Challenge", and "Return Home". This will be accomplished using a Zed 2i camera, VectorNav GPS, and our YOLO (You Only Look Once) AI algorithm. The YOLO object detection algorithm is used to train the AI to identify different objects throughout the competition such as buoys and posters and adjust the power of each thruster accordingly.**

I. Competition Goals

Since the Coast Guard Academy has never brought a boat to the competition before, a major success for us is going to be competing in just the navigate the channel, follow the path, docking, speed challenge, and the return to home tasks. Our team currently consists of three (3) senior cadets, five (5) sophomore cadets, and two (2) freshman cadets. The three senior cadets having been taking a four (4) credit capstone course, whereas the underclass have each been involved in a one (1) credit directed study. Due to the number of people working on the project, the number of man hours put into the project each week has been limited. Additionally, the only inputs that we were able to integrate into our system are a ZED 2i camera and a VectorNav GPS, which creates a restraint on our mission capabilities. Because of this, we decided to forgo competing in the three tasks not named, and in doing so, allowed us to focus our efforts on completing those tasks within our capabilities and improve the reliability of the systems we were able to get in place.

Another one of our goals is to use a state machine to tackle the course. The in-depth workings of this state machine will be described later in the design strategy portion of the paper. For the qualifying rounds, our plan is to run each state individually by manually accessing and running the tasks state. For example, for the navigate the channel task, we will manually navigate the vessel to the start, set the state machine to navigate the channel, navigate through the channel, then turn the vessel back to manual mode and drive it back. For the finals round, we plan to have our vessel change between states autonomously using this state machine. Each state will have a different trigger that kicks the vessel from one state to another.

II. Design Strategy

Our system Design can be classified into Hull Design, Power and Propulsion, Control System, Image Detection, Machine Learning, Software Framework, and Autonomous Channel Navigation.

A. Hull Design

Our group is using a vessel handed down to us from the previous ASV capstone group. It is a dual-thruster catamaran design with no rudder. The power

input to the right and left thrusters controls the steering. The advantage of this design is its increased stability and maneuverability. A wide platform in the center of the two hulls provides ample room for mounting sensors and adding additional electronics. The body is three feet in length and width, making the vessel smaller than the competition requirements of no greater than six feet, by three feet. Which, again, allows our vessel greater maneuverability in the water. A central water-tight container located between the two hulls houses all the electronics.

B. Power and Propulsion

Two Blue Robotics T200 thrusters are the vessel's propulsion. A 14.8 volt, 18Ah lithium-ion battery powers each thruster. The thrusters are controlled by an electronic speed control which receives a pulse-width modulation (PWM) signal that determines how fast, and in what direction the thrusters spin. For example, a 1.5 millisecond width pulse puts the thrusters in neutral and 1.9 ms width pulse puts the thrusters at their maximum forward speed. The batteries are connected to the thrusters through two 25-amp circuit breakers and a relay module. The relay module is powered by a remote emergency stop switch, and functions through an input signal from our radio-controlled Fly-Sky remote controller. If the emergency stop switch is disconnected, or the radio-controlled remote is disconnected, the thrusters will not turn on. Our overall hull and power systems design can be seen in Fig 1 below

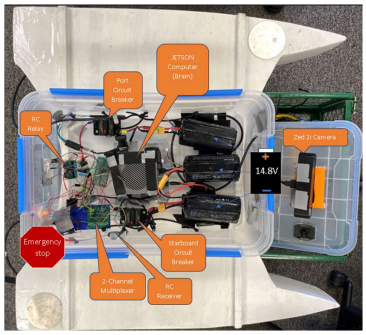

Fig. 1. – Placement of electronics within our system

C. Control System

The Jetson AGX Orin is used as the vessel's autonomous "decision maker", meaning it will control the vessel's thrusters when the vessel is in autonomous mode. The Jetson is powered by a 14.8V, 18Ah lithium-ion battery, the same battery used to power the thrusters. The vessel is controlled in manual mode using a Fly-Sky remote controller. To switch between the manual and autonomous control inputs, a Pololu 4-Channel multiplexer is used. Flipping the "SWA" switch down on the remote control will allow inputs from the Jetson to pass to the thrusters, while the switch in the up position gives control to the remote control. A block diagram of this system can be seen below in Fig. 2.

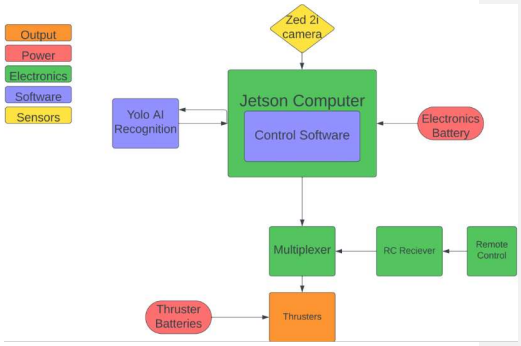

Fig. 2 Control System Block diagram.

D. Image Detection

To complete each of the tasks, we utilized machine learning to identify the various buoys in the challenge course. Using the YOLO algorithm to detect the objects we identify different buoys in a dynamic environment [1]. The algorithm is run on the Jetson AGX Orin, which takes information received from the Zed 2i 3d camera placed at the front of the vessel and identifies any buoys in the camera frame.

We are using machine learning for the image recognition for its high accuracy and scalability. As we continue to expand to complete additional tasks, we will be able to easily add additional buoys to the repository of trained objects. We are using YOLO for our machine learning specifically because it is a well-established object detection software system that has abundant support and documentation. When running YOLO with a trained model, the algorithm returns the identified objects to the Jetson user. These identified objects have specifications associated with

the pixel coordinates of the bounding box, which we use to decide how to adjust the thrusters.

E. Machine learning

We inherited a pre-trained YOLO model for the two types of buoys that will be seen in task #1 (Navigating through a pair of gated buoys) from the previous year's capstone group. To expand our database, we took about 300 photos of the Roboboat competition buoys for task 2 (navigation through a channel while counting ducks and avoiding obstacles) using the United States Coast Guard Academy waterfront boats over the course of two days to get different lighting and angles of the new buoys. We then applied filters to these photos which tripled the number of photos to train our YOLO algorithm. We then labeled all or our images using the pre-built YOLO Labeling Studio software.

The next step is to train YOLO models using our labeled dataset. We use Wandb, a software application that visualizes and tracks machine learning experiments as they run, to see how accurate our trained data is [2]. We are able to evaluate our model's performance by focusing on precision and recall. Precision is how many retrieved elements are relevant. In the context of our project, precision is how often the objects detected are actually what our model is identifying them as. This is crucial because if our model has poor precision, it will identify random objects as buoys which would confuse the navigation system. Recall is how many relevant items are retrieved. This is a ratio of how many objects in an image have been correctly identified out of how many there were in total, including the unidentified objects. We need high recall because in order to travel through a channel of buoys, the ASV needs to identify each buoy and any objects in the middle of the channel in order to both stay in the channel and not crash while travelling through it.

F. Software Framework

Using Robot Operating System 2 (ROS 2) in an ASV offers a powerful framework for seamless communication and integration among the diverse components of autonomous systems[3]. ROS 2's middleware facilitates real-time communication, allowing various sensors, actuators, and AI modules to exchange information efficiently. The modularity of ROS 2 enables the independent development and integration of different AI algorithms and functionalities. Its support for heterogeneous systems ensures compatibility with the diverse hardware components commonly found in ASVs. In our case this involves the integration of the VectorNav GPS, ZED 2i camera, and the YOLO package.

For VectorNav GPS integration there were already packages that were publicly available[4]. After discussing our project of ROS2 with the VectorNav company they directed us to a repository on GitHub they commissioned. These packages create six different nodes with publishers and subscribers pulling different raw data from the GPS and converting it to usable data. In order to access the data, one must establish a subscriber for the nodes. This simplicity makes it easier for other packages to access data in their own node.

There is a similar case for ZED 2i integration[5]. The ZED 2i camera has wrappers and interfaces created for the camera from StereoLabs. The wrapper creates a node to initialize the camera for ROS2 to acknowledge. The interface contains the message and action files, allowing data from the camera to be used in different nodes. Each of the different nodes corresponds to a different task in the competition.

The YOLO package runs in a congruent manner. This package was not created by the company in charge of it and instead by another GitHub user with similar goals to our competition group. They allow for the algorithms and detection scripts to be used in message files. Message files allow for data to be defined via their names and what type of data should be found there for ease of translation and replication between nodes.

The rest of the nodes, which include the publishers and subscribers, are being built from scratch. These nodes are currently still in development. Individual nodes such as Task 1 and Task 2 are operational on an individual basis, but not integrated into the greater ROS 2 infrastructure pictured below (fig 3). This means that with the training algorithms which are sourced from YOLO these nodes can run python scripts to identify buoys nearby associated with different tasks. When the

package associated with the task is being used there is both a publisher and a subscriber created to share data between each other and control the vessel according to the outputs.

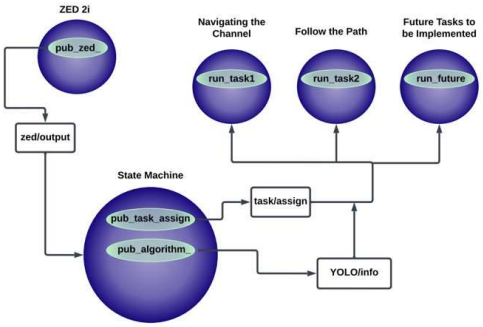

Fig. 3. Software framework diagram. ROS2 organization.

G. Autonomous channel navigation

With a trained AI algorithm that consistently identifies the competition buoys correctly, we can utilize the identified information to complete tasks. The first step in pulling the information from the identified buoys is integrating our algorithm with the Zed 2i camera. Using Stereo Lab's published documentation for their Zed 2i camera, we modified their "detect.py" script to run our AI model and pass identified Zed objects to our processing code. The Zed objects contain information such as position, distance, 2-d bounding box, and object classification. These objects are then passed from the modified "detector.py" script to our processing and navigation script.

Using a classification variable to determine which type of buoy the camera is identifying; a class 0 is a green can buoy, class 2 is a red nun buoy, class 1 is a red ball, class 3 is a green ball, class 4 is a yellow ball, and class 5 is a black ball. We do this by editing our .yaml script to add more classes. As we continue to increase our repository of trained models, we will be able to add additional images as a new class number. What is done with these identified objects depends on the task.

For task 1, navigate the channel, the distance variable is used to identify the closest green can and the closest red nun buoy. The 2-D bounding box variable provides x and y coordinates for all four corners of the 2-D box used to identify an object. So, we took the right corner x-coordinate from the red buoy and the left corner x-coordinate from the green buoy, added them together and divided by two to get the channel center pixel. We have the vessel drive towards this center pixel which effectively drives between the gated pair of red and green buoys using a proportional thrust controller.

For Task 2, follow the path, starts with the identification of the closest red and green ball buoys, also finding the closest yellow and black buoys. If there are no yellow or black obstacle buoys, the algorithm defines the center point between the closest red and green ball buoys and navigates to that point using the proportional thrust controller. If there is a yellow or black buoy in the channel, then it redefines that center point to the larger gap between the obstacle buoy and the channel buoys. It then navigates to this point, effectively avoiding all obstacles while staying within the channel. To accomplish object detection, as the algorithm identifies yellow ball buoys it adds to a counter. Once through the channel, the vessel spins the number of yellow balls that it counted in order to demonstrate accuracy.

For task 3, docking, we utilize a similar identification method to both task 1 and task 2 however, this time instead of acquiring buoys we will be identifying colored shapes. The goal will be to identify the correct shape and color based on the input of the day, then navigate towards that. Using the same proportional heading controller that has been used or the previous two tasks, and implementing the use of a speed controller, the vessel will drive directly towards the shape. As the vessel approaches the shape, it will use the speed controller to continue to move closer and closer to the dock until it is 3 ft away. At this point, it will stop and hold its position. Afterwards, it will reverse on its own and maneuver to the next task.

III. Testing Strategy

Our testing strategy focuses mainly on two types of test: laboratory tests and in the water tests. Our testing time was limited to 0900 to 1200 on Thursday mornings, due to schedule and equipment

availability. Because our time was so limited, our testing strategy focused heavily on maximizing the amount of things we were able to accomplish with the time our vessel spent in the water.

A. Water Test 1 plan, tank test

Our first in water test plan is inside McAllister Hall, our engineering building on campus. This tests is to be conducted in a pool in the naval architecture department to determine the watertight integrity of the hull, remote access, and manual operations mode. These tests will allow us to demonstrate proficiency in completing the competition prerequisites, such as a watertight hull with positive buoyancy.

B. Thruster measurement

Our next test is very similar in nature, a thruster measurement. We plan to take our vessel to the same pool in the naval architecture department and hook a scale up to our vessel's towing harness and check the thrust power. This test will allow us to ensure our vessel is towable, and allows us to maximize our points for the amount of thrust our vessel is able to produce.

C. River Tests

One large benefit of attending the Coast Guard Academy is our furnished waterfront center. The waterfront center, named the maritime center of excellence, allows us to go test whenever there is fair weather. We plan to take advantage of this with our first river test. For the river test, we will set up a singular gated pair of buoys and allow our ASV to autonomously navigate through them. This test will show us the capability of our vessel and whether it is able to identify the buoys on the river.

D. Specific River Tests

For our final waterfront test, we plan to take our vessel down to waterfront and set up a course of 3 gated pairs of buoys, implementing a turn and a narrow channel. These parameters will be two-fold. First, they will help us decide if we can accomplish the first task of navigating the channel. The second is they will allow us to test whether our heading controller is able to work for the follow the path challenge. Figure 4 below shows a setup we plan to use to test our robot on the Thames river.

Fig. 4. Task 1 & 2 river test.
Autonomously navigate through three pairs of buoys, including a turn.

E. Lab tests

Throughout the semester we plan to conduct laboratory tests for the new electronics parts we will be receiving throughout the semester. This will be conducted from the electrical engineering capstone laboratory that the team has access to or from other various locations throughout our campus. The main purpose of these tests will be to find where we need to improve our systems. As we find errors and ways to increase efficiency and effectiveness, we will tweak these tests accordingly.

F. GPS Test

The first lab test we plan to conduct is our GPS test. The goal will be to ensure that we can detect the outputted data, and then access and manipulate that data to integrate into our system. We will simulate the antenna offset that the receivers will have on our vessel on a lab cart, then bring the GPS outside to collect this data. This test will help us to set a waypoint at the start of the competition to navigate back to for the return to home challenge. It will also help us know our heading for when we spin after the channel portion of the Follow the Path challenge, to have an accurate spin count. Fig 5 is below.

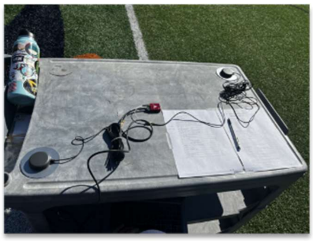

Fig. 5. GPS test setup.

Note the offset of the antennas.

G. Safety test

To ensure compliance with safety standards, we will conduct multiple mini safety tests. We test both our remote and local emergency stop switches. We kept the remote controller in one location and drove our vessel to a distant location to see how it reacted when it lost signal. If these tests fail, we will change and improve our system until it works.

H. YOLO Tests

We will also need to conduct many software laboratory tests. To ensure that our system can accurately detect every relevant object, we will run our trained objects through a testing algorithm to ensure we get at least an average of 90% accuracy. If we do not have the accuracy we want, then we will apply additional filters to our repository of buoy photos to get a better trained object. This testing will help with every task, as each task required the detection of a different buoy or a docking poster.

I. ROS2 Tests

We will also be testing our new ROS2 system often as it is being integrated later in the process. Again, as problems come up, our testing strategy will shift to determine the root of the problem, then solve whatever issue is at hand. This testing will help with the integration of the state machine so that we can seamlessly transition between tasks during the final challenges.

J. Follow the Path Lab Setup

For our final test this semester, we set up three buoys in the hallway of McAllister Hall to avoid having to go all the way down to our waterfront. On the left and right side of the hall, we put red and green nuns, and between them we put a ball buoy. We will position our vessel to face these buoys and shifted the center ball buoy left and right to simulate an obstacle at different locations in the path. The outputs our system gives us will allow us to test our ability to determine the best route to take to avoid an obstacle, and our ability to accurately count every yellow ball buoy we encounter. This can be seen in Fig 6.

Fig. 6. Setup of initial test for task 2, identify channel and obstacles, and maneuver through the optimal route.

IV. Acknowledgment

Without the help of the following individuals, we would not be competing in this competition today. Our first thank you goes out to our project advisor CAPT Seals. CAPT has kept us grounded throughout this project, reminding us of deadlines and to stay focused on the overall competition goals while completing our work. We would also like to thank the faculty in the Electrical Engineering and Cyber Systems department: CAPT Hartnett, Dr. Emami, LCDR McGarry, LT Williams, LT Quarry, and Mr. Gold. Along with these, our Laboratory Technicians: Chad Davis, David Fournier, and Jessica Rosekrans. Thank you for knowledge, experience, and guidance when it comes to building complex electrical circuits and software. Our staff down at the waterfront: Coach Randall and Coach Doolittle. Thank you for the help getting RHI qualified, and the ability to store buoys in the boathouse. Our IT Support: Cate Griffin an dJOn Mihalko. And finally the underclass cadets working as part of a directed study: Madeline Fontana, Nathan Harris, Hudson Holden, Tyler Bissett, Richard Barkley, Isaac Atkins, and David Ulmer. Our project is unique in the sense that we are entirely funded by the United States Coast Guard through the academy. We would like to thank you, the American taxpayer, for supporting us and our continual learning through this capstone project.

Commented [SB1]: Content needed: Acknowledgements detail supporting personnel and their contributions as well as resources. Sponsors and their contributions are acknowledged

References

[1] Glenn-jocher, and sergiuwaxmann (Github), "Comprehensive Guide to Ultralytics YOLOv5", updated Dec 2023.

[2] Wandb, "2023 Weights and Biases", updated Aug 2023

[3] Github, "ROS 2 Documentation," ROS Documenation. https://docs.ros.org/en/foxy/index.html (Accessed Dec 04, 2023)

[4] 2020 VectorNav Technologies, LLC, "VN-300 Development Kit Quick Start Guide" 2020

[5] 2023 Stereolabs Inc., "Getting Started with ZED" 2023 steroelabs.com/docs

Appendix A: Components List

Component	Vendor	Model/Type	Specs	Custom/ Purchased	Cost (Dollars)	Year of Purchase
ASV Hull Platform	Developed	Model 2	Plywood and insulation foam	Custom	Unknown	2023
Propulsion	Blue Robotics	T200 Thruster	https://bluerobotics.com/store/thrusters/t100-t200-thrusters/t200-thruster-r2-rp/	Purchased	200	2018
Power System	Blue Robotics	Lithium-ion Battery (14.8 V)	https://bluerobotics.com/store/thrusters/t10	Purchased	350	2018
CPU	Nvidia	Jetson AGX Orin	https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/	Purchased	2000	2022
Teleoperation	Fly Sky	FS-T6	https://www.flysky-cn.com/fst6	Purchased	Unknown	Unknown
E-stop button	EAO	61-6451.4247	https://www.mouser.com/ProductDetail/E	Purchased	75	2022
RC Relay	Pololu	2804	https://www.pololu.com/product/2804	Purchased	25.9	2023
Camera	Stereo Labs	Zed 2i	https://www.stereolabs.com/zed-2i/	Purchased	500	2022
GPS	VectorNav	Vn-300	https://www.vectornav.com/products/detail/vn-300	Purchased	5375	2023
Algorithms	Yolo	Object detection		N/A	0	2022
Open-Source Software	ROS	ROS 2	ROS 2 Documentation — ROS 2 Documentation: Foxy documentation	N/a	0	2023