CSUF AUV Technical Design Report

Tyler Groom, Henry Lin, Nate Ruppert, Douglas Shimizu, Kayla Lee

Abstract

This is a legacy senior design project that is in its third year. The purpose of this project is to design an autonomous underwater vehicle (AUV) that will use a trained machine learning model to detect objects and maneuver accordingly underwater. The vehicle contains parts from Blue Robotics, Amazon, and lithium ion batteries from an online battery store. Items from Home Depot and an art supply store were utilized in the process of putting the body of the autonomous underwater vehicle together and some of the waterproofing. The battery case was designed to be waterproof and each battery case can hold a total of twenty lithium ion batteries. The body of the AUV was assembled, tested, and deemed waterproof; the individual components including the camera, gripper, two LEDs, and six thrusters were also tested and verified to be working properly. Thus, the components left by the previous two teams and the components that were newly ordered by the current team were verified as working components and able to be utilized in the AUV system. The code used to test these components was Arduino code; however, in order to work with the Raspberry Pi, the code had to be converted to Python. In terms of the equipment used for software, the Nvidia Titan RX was used to train a YOLOv4-tiny model, and various toolkits/libraries such as OpenCV, OpenVino, and Tensorflow were used in the process as well as running the model.

Competition Strategy

Since we will be competing online and do not have access to an actual course similar to the one used in the in-person competition, we will be attempting to demonstrate a couple vehicle behaviors in our video. The main vehicle behavior we will be attempting to demonstrate is: maintaining heading and depth.

In order to accomplish this, we split the project up into various hardware and software tasks. We also did much testing along the way and tried to make sure that all of the individual components worked before testing the AUV as a system.

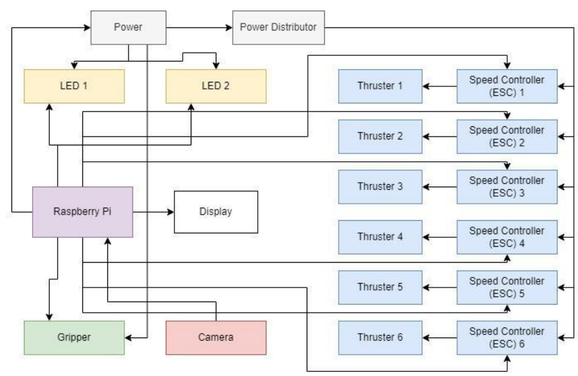

Fig. 1. Block Diagram

Fig. 1 shows the block diagram of our design. The power supply consists of the two batteries that were designed specifically for the AUV. This power supply goes through a power distributor to power the six thrusters, and the power supply also powers the two LEDs as well as the gripper. The Raspberry Pi will be used to control the six thrusters, two LEDs, and gripper; there is also a display connected to the Raspberry Pi. The camera will be used as an input and will be used mainly for machine learning and the autonomous side of the AUV.

I. Hardware

The key components used in the AUV include: 2 custom batteries, 2 custom battery cases, a BlueRobotics frame, a BlueRobotics 4” chassis, a BlueRobotics gripper, 6 BlueRobotics thrusters, 2 BlueRobotics LEDs, a BlueRobotics camera, a power distributor, electronic speed controllers (ESCs), an electronics tray, a Raspberry Pi, and the Intel Neural Compute Stick 2. Fig. 2 demonstrates the complete schematic diagram detailing how these components were to be connected together.

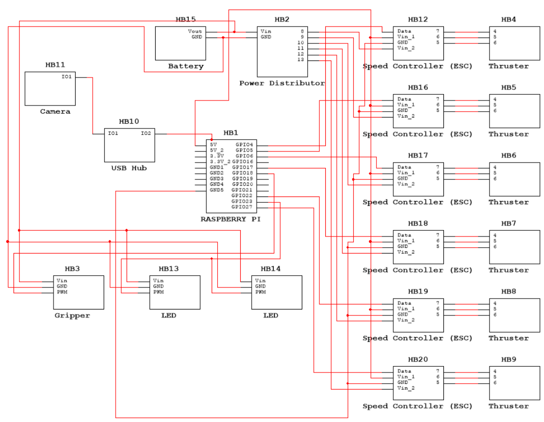

Fig. 2. Complete Schematic Diagram

The main goals on the hardware side were: to ensure that the AUV is waterproof, is balanced, and has proper buoyancy. This encompasses testing all of the individual components to make sure they are working properly and are waterproof before assembling the AUV. Much testing was done to ensure that each component worked on its own before being put together. For example, the thrusters, LEDs, camera, and gripper were each tested individually before being integrated into the AUV body.

Also, the AUV body or chassis needed to be tested thoroughly and verified to be waterproof before placing all of the electronic components inside. Thus, the AUV body had to be pressure tested multiple times as well as tested underwater to verify that it would not lose pressure and would not leak; even small leaks could cause short circuiting and possible irreparable damage. The battery cases also had to be tested multiple times to ensure they were waterproof before placing the batteries inside. Thus, they were tested underwater to make sure there were no leaks before integration and system underwater testing could take place.

Once the individual components passed the above tests and system integration was completed, the AUV also had to undergo much system testing underwater. This is to verify that the AUV as a system does not leak and does not have any hardware issues while going underwater; in addition, this could be used to see how long the system can function and adjust the system according to possible buoyancy issues.

Although waterproofing the AUV is the main priority on the hardware side, making sure the AUV is properly balanced and has proper buoyancy is also important; this was accounted for in the design by having one battery on each side of the AUV for balance and to act as some sort of counterweight to the foam to be added.

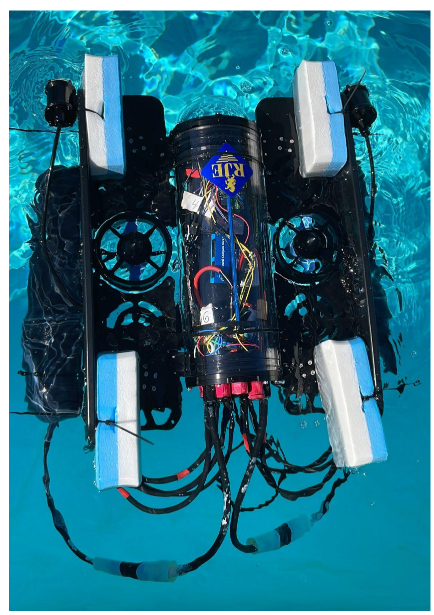

Fig. 3. AUV Design

Fig. 3 demonstrates the final AUV prototype that incorporates the above design elements as well as addresses a few of the issues left by the previous team, some of which will be explained in the following sections. The foam accounted for in the design was used not only to balance out the AUV, but also to provide some buoyancy and make the AUV positively buoyant. The reasoning for this is because positive or neutral buoyancy is preferable to negative buoyancy.

II. Software

The main focus regarding the machine learning side of this project was the object detection. This encompassed gathering a plethora of images for the machine learning database from different angles, training the machine learning model, and testing the model. OpenCV and YOLOv4-tiny were used in the making of the model; YOLOv4-tiny was specifically chosen because YOLOv4-tiny models are known to be faster and take less space than YOLOv4 models.

Regarding the software, Python was the programming language used. Scripts were written for controlling and testing the individual components such as the six thrusters, two LEDs, and gripper. Additionally, these were all incorporated into a program containing pseudocode for autonomous navigation. Essentially, this encapsulates taking input from the camera and object detection as well as the gyroscope to determine position, and then moving the AUV accordingly. Fig. 4 shows a flowchart of the overall process and logic used in the code.

The basic process of completing a challenge during the competition can be seen above in the flow chart. The AUV will power on, initialize Pigpio daemon, then use the ports to control all the I/O. The camera will receive the images and it will determine if the object in desire was detected. If the object was detected, navigate towards it; otherwise continue onward. Then, check if the object is close enough to grab. If it is, then grab it and return home, otherwise continue the path.

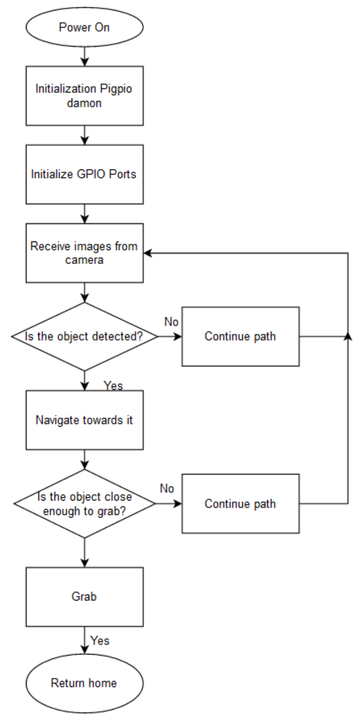

Fig. 4. Overall Process Flowchart

Fig. 5 demonstrates the sub operation flowchart, which details the flow of logic used during a certain operation or task.

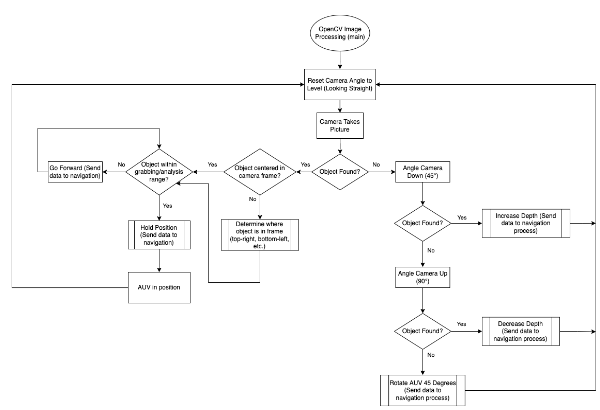

Fig. 5. Sub Operation Flowchart

Design Creativity

Many of the hardware components were custom designed and/or 3D printed specifically for our AUV. Taking a similar approach/concept that is used when building batteries in electric cars, we opted to put a set of four batteries in series, with five of those sets in parallel to build our 2 batteries. The battery design is shown below in Fig. 6.

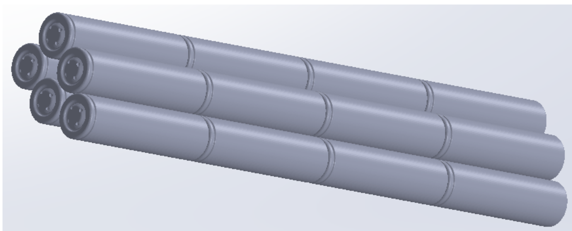

Fig. 6. Custom Battery Design

Overall this battery system outputs 14.8V with 27.6 Ah. Each battery holds a maximum draw of 100A, making this a robust solution for the entire system. Using these 2 batteries not only allowed us to fix the lack of power issue left by the previous team, but also it allowed us to fix the skewed weight distribution by putting one battery on each side of the AUV and balancing it out. We spent $290.81 total on 40 lithium ion batteries (including shipping costs). In comparison, one Lithium-ion Battery (14.8V, 15.6Ah) from BlueRobotics would have cost $330.00 before tax and shipping, and we would have needed at least 2 of these batteries to support our power consumption needs which would have resulted in $726 as our total battery cost (including tax and shipping) [1]. Thus, by custom making and designing our own power source, we were not only able to save money and make the AUV more cost efficient, but also we were able to customize the battery to handle our specific power needs.

In addition, the battery cases are also custom-made and are designed in SolidWorks, 3D printed, coated in 7 layers of epoxy, and sealed using waterproof sealant. A prototype of the battery and the battery case is shown in Fig. 7.

Fig. 7. Battery with Custom Battery Case

Furthermore, the electronics tray inside the 4” chassis that holds most of our electronic components was 3D printed as well. This is shown below in Fig. 8.

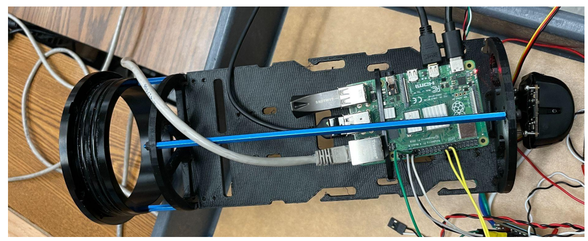

Fig. 8. 3D Printed Electronics Tray

Experimental Results

We were able to solve a couple of the issues that were left to us by the previous team. The first issue that was solved was the lack of power. In order to account for the lack of power, a battery was designed that contained five sets of batteries in parallel with each set containing four batteries in series. Two of these batteries were created to account for the power taken by the thrusters, LEDs, gripper, and other components of the AUV. Also, the final battery cases were 3-D printed, coated with 7 layers of epoxy, and sealed using clear waterproof sealant to ensure that they would provide a safe, watertight enclosure to hold our power supply. The second issue that was solved was the skewed weight distribution. By downsizing to a 4” chassis and having two batteries, with one on each side of the chassis, the weight should be distributed evenly. Also, the previous team had camera inaccuracy issues when working with low light; to account for this, we purchased two LEDs to incorporate into the design.

Much testing was done to ensure that each component worked on its own before being put together. For example, the thrusters, LEDs, camera, and gripper were each tested individually before being integrated into the AUV body. Since the thrusters, LEDs, and gripper were initially tested using Arduino code, this had to be ported to Python to be used with the Raspberry Pi. This was done to reduce the number of boards being used and to simplify the design, avoiding unnecessary extra boards. Each component was also verified to be waterproof before underwater system testing could be done. The AUV body and battery cases were specifically focused on as these housed all of the electrical components. The AUV body had to be pressure tested multiple times as well as tested underwater to verify that it would not lose pressure and would not leak; even small leaks could cause short circuiting and possible irreparable damage. The battery cases were also tested underwater to ensure they were completely waterproof before the actual batteries were placed inside them.

In addition, we took pictures of the images needed for the RoboSub competition from different angles for our machine learning training dataset. We were able to complete labeling of these images and train the machine learning training dataset, which resulted in a YOLOv4-tiny custom model. Furthermore, we were able to test the object detection. Since all of the individual components had been deemed waterproof, we were able to submerge the AUV and test the system as a whole. Thus, we laminated images of the RoboSub pictures provided and did tests to ensure that the AUV accurately detects objects underwater. The result, as shown in the image from Fig. 9. below, proves that the AUV was able to detect objects with accuracies ranging from 85% - 100% when working with a single image at a time.

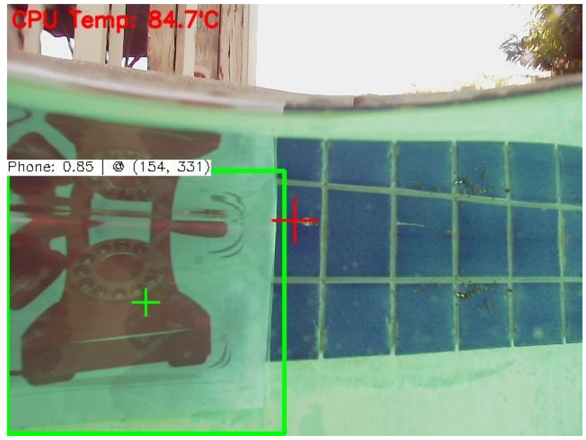

Fig. 9. Object Detection Example

In conclusion, the AUV was proven to be able to move in a line, rotate, and be controlled to move in any direction. The object detection was tested and verified, and the AUV was able to turn to face a detected image. It was also able to go underwater and dive in shallow water safely. We also have pseudocode for autonomous navigation. To summarize, we were able to improve on the last team's designs and fix much of the issues they left us. We also have the individual components needed in our prototype to autonomously navigate underwater.

Acknowledgements

Our team would like to thank RJE International for sponsoring this project. We would also like to thank our advisor Dr. George for his support of our work.

References

[1] “Lithium-ion battery (14.8V, 15.6AH),” Blue Robotics, 06-Jun-2022. [Online]. Available: https://bluerobotics.com/store/comm-control-power/powersupplies-batteries/battery-li-4s-15-6ah/. [Accessed: 06-Jun-2022].