Lionfish - Phase IV

A Major Qualifying Project submitted to the faculty of
Worcester Polytechnic Institute in partial fulfillment of the
requirements for the Degree of Bachelor of Science in Robotics
Engineering

Lokesh Gangaramaney (RBE), Matthew Ferriera
(RBE), Cray Johnson(RBE), Leonardo Go(ME)

Advised by Professor Craig B. Putnam,
Professor Bradley A. Miller, and
Professor William R. Michalson

April 21, 2021

Abstract

Indigenous to the western Pacific, lionfish are an invasive species that have been wreaking havoc along the coasts of the western Atlantic. The rapid reproductive cycle of these fish combined with the fact that they have no natural predators in the Atlantic Ocean has made them a considerable threat to the local ecology. Current methods of combating the invasive lionfish have seen little to no results. This is the fourth year that students from WPI have collaborated to develop a robotic solution to help curb the exploding lionfish population. This year's focus was on developing a stereo vision system capable of detecting an object and determining its distance, an improved navigation system that incorporates P control, as well as a revised design for the harvesters' container.

Acknowledgements

First and foremost, we would like to thank our advisors for guiding us through the project and providing insight for our work: Professor Craig B. Putnam, Professor Brad A. Miller, and Professor William R. Michalson.

We would also like to thank the WPI recreation building staff and lifeguards for allowing us to test our robot in the WPI pool.

Finally, we would like to thank Blue Robotics for giving us a open source ROV for development.

Contents

List of Figures

List of Tables

1 Introduction

Indigenous to the western Pacific, lionfish (Pterois miles and Pterois volitans), is a genus of vibrantly colored fish, covered in venomous spines which have been wreaking havoc along the coasts of the western Atlantic. It is believed that lionfish were brought to this region sometime around 1985, likely having been dumped from home aquariums (NOAA, 2019). Being introduced to an entirely new environment so quickly has left the lionfish with no natural predators. This fact, combined with a remarkably fast reproductive cycle, has led to an explosion of lionfish in the Caribbean and along the eastern coast of the United States (Figure 1). In recent years, this species has been spotted as far north as Cape Cod, in waters previously thought too cold for this fish to tolerate, as it is native to more tropical climates (USGS, 2020). Lionfish have been particularly troublesome for the local ecosystems, as they eat upwards of 50 different species of fish, many of which are crucial for maintaining the health of the native coral reefs. This in turn negatively affects the coastal residents who rely on the health of their marine ecosystem (UCSD, 2019). Without a predator species to keep the lionfish in check, they will continue to expand along the western Atlantic, jeopardizing the well-being of the local species.

As of today the primary method of harvesting lionfish comes in the form of divers equipped for spearfishing (NCSU, 2019). In order to motivate as many divers as possible competitions are often organized, offering prizes for

Figure 1: Lionfish (Pterois miles and Pterois volitans)

capturing the greatest number of fish. To further incentivize these hunting excursions and provide a sustainable means to limit the lionfish population, a movement has been started to incorporate lionfish meat into local cuisines. The hope is to make lionfish hunting a lucrative business and encourage a reduction in the invasive population. This solution not only creates an industry motivated to capture lionfish but provides a financial offset to locals who have suffered economically from the devastation inflicted by the lionfish's invasive nature. There have also been attempts to get native predators to assist in hunting. In 2010 divers in Honduras endeavored to train local reef sharks to eat the foreign lionfish, with the goal of introducing a predator to finally keep the lionfish numbers in check (National Geographic, 2011). Unfortunately these current methods have seen unsatisfactory results; the number of lionfish harvested or eaten by other animals are simply not enough to counteract the species' rapid reproductive cycle.

This is the fourth year that students from WPI have collaborated to develop a robotic solution to help curb the exploding lionfish population. Robots provide a myriad of benefits that would be useful for this application. Capable of operating independently, AUVs (Autonomous underwater vehicles) offer a means of hunting lionfish that requires minimum human oversight. While the venom from a lionfish is not lethal to humans, the results of being stung are extremely painful. The use of a robotic platform therefore provides a safer method of harvesting this invasive species. More critically, these vehicles are capable of reaching depths, not possible for divers, which have up until now provided lionfish a dependable refuge from humans.

This year's team seeks to expand and improve upon the work of earlier iterations. The work done by previous MQPs has provided vital information from testing that provides insight into how former mechanical designs can be improved upon as well as how specific materials cope with being submerged in corrosive water environments. A complete overhaul of the navigation system will allow the AUV to traverse a much more complex and realistic surroundings, a critical requirement for a machine that will operate near fragile coral reefs. An improved computer vision system will also enable greater accuracy when it comes to identifying and targeting lionfish, and will provide additional safety to nearby divers.

1.1 Related Works

1.1.1 The Guardian by RSE (Robots in the Service of the Environment)

The Guardian by RSE (Robots in Service of the Environment) is a lionfish hunting robot that can descend up to 700 ft underwater. It requires a driver that controls the robot by a game controller to navigate and hunt lionfish. The robot has two stunning panels that run a low - voltage current to stun lionfish (Figure 2). The stunned lionfish are collected by the thruster capture lionfish system that can hold up-to 30 lbs of lionfish (Figure 2). The robot uses a 200 m tether for control and communication with the driver (Figure 2).

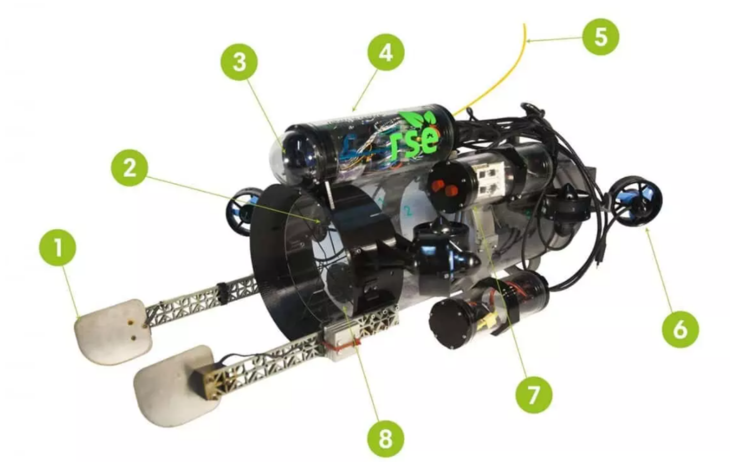


Figure 2: The Guardian by RSE

Label	Description
1	Stunning Panels
2	Capture System
3	Power Enclosures
4	Main ROV Enclosure
5	Surface Tether
6	6-DOF Navigation System
7	Stun Enclosure
8	Fish Retention Feature

Table 1: Description of different labels in The Guardian

2 Background

2.1 Lionfish

The term lionfish usually refers to members of the members of Pterois, a genus in the family Scorpaenidae, which are highly venomous and predatory marine fish. Members of the Pterois are well-known for their vibrant colors, striped skin patterns, and ornate fan-like fins. The most well-known species are Pterois miles, the Common Lionfish or Devil Firefish, and Pterois volitans, the Red Lionfish or Zebrafish. The appearance and behavior of these two species is nearly identical. Due to their beauty and popularity, these two species are often desired for both home and public aquariums; this introduction to the aquaria trade may have led to their invasion in Atlantic waters when captive specimens outgrew their tanks and were released or when tropical storms damaged public aquariums.

2.1.1 Invasive Species

Lionfish are voracious predators, capable of consuming prey up to 48% of their body length in a single feeding strike. Being slow swimmers, they cannot chase their prey and must approach their prey with stealth and surprise. Getting into position for a successful feeding strike requires precision and accuracy; to achieve this, a lionfish has modified swim bladder muscles that allow it to alter its center of mass, giving it the ability to maintain any angle it chooses to approach its prey. When in position, a lionfish fans out its large pectoral fins to disorient and corner a target animal as it slowly moves in. The striped pattern on the lionfish's pectoral fins also make it difficult for the target animal to judge how far the lionfish is, resulting in a delayed escape response. This fanning technique has been documented in several species of lionfish, in some cases with multiple individuals working together to corral a school of prey fish. Another hunting technique that has been recently discovered is to blow jets of water at the target animal. This results in the prey responding to the current of water, which is to turn and swim in the opposite direction of the flow, which happens to be the lionfish' mouth. Blowing water may also disrupt a fish's lateral line, rendering it unable to detect the disturbance in the water produced by the lionfish as it prepares to strike.

Even with their unique hunting methods, lionfish are still potential prey to even bigger predators. To avoid conflict, lionfish hide in crevices and caves along the reef for half of the day, and generally become more active at night. They may also employ camouflage during the day; the light stripes break up the outline of the body, making it difficult to spot from a distance, while many species stay close to reef walls and cave ceilings, blending in with the coral. Should a lionfish meet a potential threat, it can defend itself with the hard spines in its dorsal, pelvic, and anal fins. Each of these spines is a hollow needle with a venom gland at the base, which acts like a syringe when the skin surrounding the spine is retracted. This neuromuscular toxin results in severe pain, discomfort, dizziness, nausea, difficulty in breathing, and various other side-effects. The aposematic stripes of the lionfish may also serve as a warning to potential predators of their venom.

As with most bony fish, lionfish reproduce in large numbers. A fully-matured female lionfish can produce between 10,000 to 30,000 eggs per day, leading up to 2 million eggs per year (though only a fraction of the offspring will survive to adulthood). When mating, lionfish often group together during breeding sessions, with several females guarded by a few dominant males; gathering in numbers reduces the chance of predation, though it also leads to competition between rival males. Lionfish also reach sexual maturity within a year, and that allows their numbers to multiply very rapidly in a relatively short span of time.

2.1.2 Native Range

In their indigenous range in the Indo-Pacific, lionfish have evolved alongside the local ecosystems, which have remained stable despite their presence. Prey in the native range have evolved methods of evading capture, and are capable of recognizing lionfish as a threat. Predators in the native range, including moray eels, groupers, sharks, and bobbit worms, appear to have built up a tolerance to their venom or have found ways to avoid the venomous spines. The presence of even a few animals eating lionfish keeps their populations in check, despite their rapid reproduction.

2.1.3 Invasive Range

In their invasive range in the Atlantic and the coast of the Eastern United States, lionfish have become a serious threat to the local ecosystems. Prey in the non-native range have not evolved methods of avoiding lionfish, and they do not seem to recognize lionfish as a threat. This allows the invasive lionfish to quickly diminish the prey species' populations, in some cases down to 10% of the original population. This in turn has adverse effects on the rest of the reef ecosystem, especially when the species that the lionfish has eradicated plays an important role in maintaining the health of the coral or the health of other fish species. Likewise, predators in the non-native range may not recognize lionfish as a prey, and are not accustomed to the venomous spines that protect fish. Recent studies also reveal that lionfish are extremely adaptable and hardy; specimens have been found in the temperate northeastern coasts of the United States, in waters much colder than they would normally be found. With prey that makes no attempt to escape, no predators, an extremely fast reproduction rate, and the ability to adapt to a variety of marine climates, lionfish are able to flourish and devastate local ecosystems throughout their invasive range.

2.2 Current Methods of Capturing Lionfish

There are a couple current methods of getting rid of the lionfish such as scuba divers hunting them, and local predators eating them. Each of these methods vary in effectiveness and have their own problems for the environment.

2.2.1 Scuba divers hunting

The primary way to affect the lionfish population is scuba divers hunting them when they dive. When scuba divers hunt lionfish they bring a spear and a container down with them. Lionfish are slow which allows scuba divers to get close and easily catch the lionfish they encounter, and some places have even made competitions for catching lionfish (Kletou, Hall-Spencer, & Kleitou, 2016). Along with competitions some places have incentives for hunting and fishing lionfish usually during their reproductive season which results in massive removals of lionfish biomass (Kletou, Hall-Spencer, Kleitou 2016). Lionfish have been observed at a depth of 304m which is significantly more than the depth of 40m that scuba divers can safely dive (Gress et al., 2017). Due to the difference in depth scuba divers alone won't be able to solve the lionfish invasion.

2.2.2 Local Predators

The majority of fish won't eat lionfish due to their venomous spines, however there are a few fish that can consume them safely. Groupers are one of the fish local in the Caribbean that are able to eat lionfish safely however they alone would not be able stop the lionfish population growth. The reason they can't is that they don't eat enough nor are there enough of them to counteract the reproduction of lionfish (Mundy, Harborne, 2011). Sharks are another animal that have been found to eat lionfish. Natural predators of lionfish won't be able to stop the reproduction after the populations are established, but they could limit the population of them at the onset of an invasion (Hackerott, Valdivia, Green 2013).

2.2.3 Combining current methods

Due to the lionfish being a slow moving species of fish and there being scuba diving locations all throughout the Caribbean and Gulf of Mexico scuba divers are a good early detection of a lionfish invasion. Since lionfish at shallow depths can be targeted and captured by scuba divers the abundance of lionfish at those depths has decreased, however different methods will be needed for deeper depths. The deeper depths could use Groupers and other predators of lionfish, however since they aren't native to the Carribean and Atlantic there aren't many predators that naturally prey on them. To combine the efforts of scuba diving hunting and using local predators to the lionfish population, the populations of the predators has to first be restored (Kletou, Hall-Spencer, & Kleitou, 2016).

2.3 Ethics of using Automation for killing Lionifsh

With continued growth in Automation, AI, and Machine Learning there are increasing ethical and moral concerns that need to be addressed. This is especially true in our project, where the sole intent of our robot is to autonomously kill and harvest lionfish. Even though our robot does not affect or harm human life like other autonomous technologies such as Amazon Go (Appendix I), it still takes life from a living being. There were a lot of moral and ethical concerns that we considered while working on a project like these that are addressed below.

2.3.1 Human Intervention

The first question that we asked themselves was the need for human intervention; Whether it is our duty as humans to intervene in the Atlantic ocean's natural order. The answer is yes, it is not only our responsibility as humans to intervene, it is our obligation. Lionfish were never native to the Atlantic ocean, they were brought to the Atlantic through human intervention and therefore were never really meant to be in the Atlantic ocean's natural order in the first place. Humans put the native species in danger by introducing the invasive lionfish, therefore are ethically obligated to to save the native species that we endangered.

2.3.2 Killing Lionfish

The next component of the ethical ramifications is the killing of lionfish. Unfortunately, the only realistic way to undo the mistakes humans made is to kill and harvest lionfish. Even though it is morally wrong to take another life, the thousands of native fish that are killed by lionfish on a daily basis (IC), make killing and harvesting lionfish justifiable. This project will, if successful, create a robot that is able to autonomously kill, harvest and store lionfish that can be used later for human consumption. Lionfish damage ecosystems, and lead to the extinction of other fish and plants. It is an ethical dilemma to pick one fish over another. However, the problem becomes much simpler when you pick between one fish species (that didn't naturally arrive there) against the many flora and fauna species that are native to the Atlantic Ocean. Ethically, it comes down to the lesser evil: kill lionfish or let them wipe out other fish and plant species. It is easy to pick killing the lionfish with these arguments.

2.4 Platform for development (BlueROV2)

The underwater environment proves to be a harsh challenge and requires manufacturing a robot that can withstand the pressure at depth and the corrosive nature of saltwater. While building a custom underwater robot would have been an ideal situation, it is really challenging and expensive to manufacture. Therefore, an off the shelf ROV enables us to develop a harvesting mechanism, a navigation library and an identification system to harvest and store lionfish.

2.4.1 BlueRobotics BlueROV2

BlueROV2 (Figure 19), manufactured by BlueRobotics is an ideal off the shelf Remote-Operated Vehicle (ROV) designed for harsh underwater environments. Compared to the cost of manufacturing a custom underwater vehicle, the BlueROV2 is an affordable solution with many features and a flexible design for custom mechanical and software additions. It provides an open source python library with built in low-level functionality so we can focus more on building high level algorithms for path-planning. There are open sourced CAD models provided so we can build custom mechanical designs for our project, making it an ideal base system for building an autonomous harvesting robot.

Figure 3: BlueROV2 with heavy lift configuration (Top View)

2.5 Environmental Concerns

2.5.1 Corrosion

The ocean is an extremely harsh and challenging world for any mechanical device to operate it. Much consideration must be given to ensure that the AUV can work reliably in a marine environment for an extended period of time. One of the biggest issues of submerging many materials into water is the inevitability of corrosion. This is especially true in ocean waters which are considerably briny and contain reactive elements such as carbonic acid. Corrosion is the result of a chemical process called oxidation. In oxidation an element or compound gives up some of its electrons to another element or compound known as an oxidizing agent.

A very familiar example of corrosion is iron rust. Here the oxidation reaction takes place between iron (the reducing agent) and oxygen (the oxidizing agent). When iron comes into contact with water it gives up two of its electrons and becomes Fe^{2+}. These iron ions combine with hydroxide ions which naturally form in small amounts due to the dissociation of water. The result is $Fe(OH)_2$ which is called iron hydroxide. Two of these compound can bind together rearrange to form iron oxide hydrate, Fe_2O_3*H_2O. This is the full technical name of rust. Salt water is a very good conductor of electrons and oxidation is ultimately about the transfer of these particles. This is why salt water accelerates the corrosion process significantly more than fresh water. The presence of acid also increases the rate of oxidation. Carbon dioxide when dissolved in water forms carbonic acid, a process that has become more prevalent due to climate change, again increasing the caustic effects of the ocean.

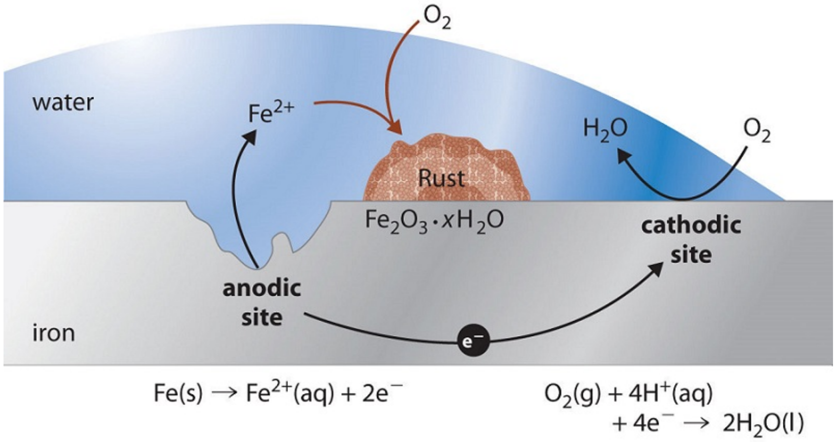


Figure 4: Corrosion Reaction

An important question to ask is why one would care about the formation of rust. Unlike with some metals like aluminum, which forms a protective outer layer when oxidized, and so stops further oxidation, the corrosion of iron facilitates further corrosion. This is because the formation of rust splits oxygen atoms from water molecules. This creates more oxidizing agents allowing for a continuation of the process. The fatal problem with rust is that it expands. A sample of iron that is converted to rust can have up to seven times its original volume. This extreme expansion will put stress on any component that is corroding, forming cracks which will accelerate the oxidation process. Often as the rust expands it will flake off, removing material from the component and exposing more metal to corrosion. If left to rust for too long the result will be structural failure.

It is for these reasons that the choice of materials used is very important for ensuring the durability of an ocean-worthy AUV. There are metals and alloys such as aluminum or stainless steel that have relatively high resistance to corrosion. However even these materials can corrode and lose structural integrity when operating in salt water. If possible it is best to use nonmetallic materials whenever possible. Processes such as galvanization can be used to increase a metal's resilience to erosion. Procedures can also be developed, such as limiting submergence time and after operations cleaning to extend the life of the robot.

2.5.2 Pressure

Another critical challenge of working underwater is being subjected to extreme pressures. While a mechanical device can handle pressures much greater than humans are able to, and does not have to worry about developing decompression sickness, pressure is still one of the greatest limiting factors. Decompression sickness occurs when nitrogen gas, which had previously been dissolved in the blood due to the hydrostatic pressure of the water at depth, comes out of solution as the diver ascends to the surface. If the gas comes out too fast it will form bubbles which can harm the human body. This condition is extremely serious, resulting in severe pain and potentially death.

For this reason divers must be very cautious and ascend slowly enough for the nitrogen to safely be removed from their blood. The deeper the dive the longer the ascent process must take, eating into the time when the divers can actively hunt for lionfish.Water is a fairly dense material and the result is that every additional 33 ft in depth results in an additional atmosphere (atm) of pressure. This means that at the expected working depth of 100 ft, the AUV would be under a relative pressure of 3 atm. At these pressures the issue isn't complete structural failure but rather a high risk of leaks. Due to the high electrical conductivity of salt water, a breached chamber would be catastrophic for the entire system.

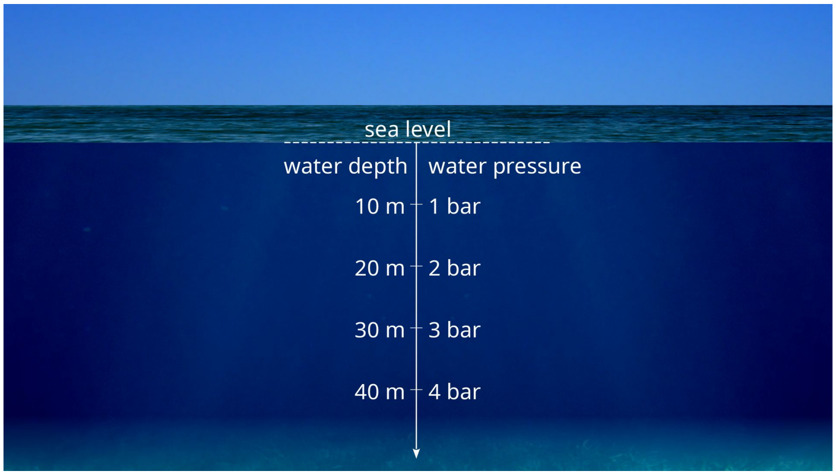


Figure 5: Ocean Depth and Pressure

2.5.3 Buoyancy

Buoyancy in water provides both an opportunity and a challenge. Making the AUV have a slight positive buoyancy can be very helpful. This guarantees that in the event of a power failure, the robot will float to the surface and can be retrieved instead of sinking and rusting away on the seafloor. The challenge arises when trying to balance the AUV. If not properly balanced, the thrusters will need to be more active to maintain the correct orientation and depth, increasing battery usage and limiting operating time.

2.5.4 Lack of Communication

One of the biggest goals of this multi-year lionfish project is to develop a robot which is autonomous. The capability of this device to operate independent of human input is critical since there is currently no adequate solution that involves a human operator controlling the robot in real time. As of now the robot functions as a ROV (remote operated vehicle) and as such uses a tether as a means of communication between itself and a computer on the surface. This setup cannot be used while lionfish hunting since the tether poses too much of a risk of getting caught up in boat propellers, swimmers, or nearby corals. The point of this project is to reduce ecological damage, not increase it. Wireless communication is also not a practical option. Water does a great job of absorbing radio waves, making communication via radio transmission impossible. Sound waves on the other hand do a much better job traveling through water. The previous year's MQP tried to take advantage of this fact by developing an acoustic modem. Unfortunately this modem had too small of a bandwidth to transmit much useful data. The purpose of this acoustic modem was only to transmit critical information about the robot such as its current status and to issue very basic commands such as return to the surface. The intention of the device was never to establish the communications necessary for real-time control of the ROV. Interesting, there has been research done on a system called Aquai-Fi which establishes communication between the surface and an underwater modem with the use of lasers. Work on this research is still ongoing and is far beyond the scope of this project, however this system allows for the creation of a Wi-Fi at a depth of more than 30 ft. While impressive, it is unlikely that this system can currently work at our presumed operating depth of 100 m (328 ft)These limitations make it necessary to turn the Blue Robotics ROV into an AUV (autonomous underwater vehicle).

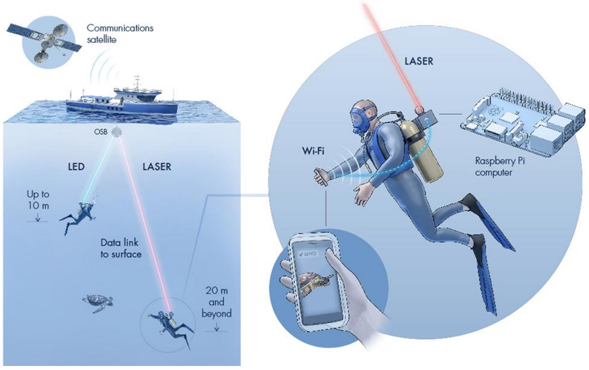


Figure 6: Aqua-Fi created by King Abdullah University of Science Technology

2.6 Previous Lionfish Major Qualifying Projects

There have been a couple of iterations of the lionfish project at WPI.

2.6.1 First Year

During the first year of the MQP, the lionfish robot, was a harvesting accessory meant to be attached to different autonomous underwater vehicles. The mechanism was designed to be operated completely independent from the AUVs with its own vision and lionfish detection system. The device did not use a typical spear and containment mechanism, it used a mechanism that targets a lionfish and shoots out a buoyant spearhead that floats to the surface with the harvested lionfish. To identify and target lionfish, the first year team used a deep learning neural network that determines if lionfish are in the camera's range and takes the shot (Yuzvik, Kelly, Lombardi, Uvarov, & Godsey, 2018). There were a couple of problems with this design as there is a high probability that the mechanism misses the shot and pollutes the Atlantic seabed with sharp metallic spears. Moreover, there is only a maximum range of 8 shots that are not really profitable to the fisherman in one run. Furthermore, with each shot, the robot loses weight and gains buoyancy, which can disrupt the magnetometer, accelerometer, gyroscope, calibration and stability of the underwater vehicle. A SolidWorks model of the prototype can be seen in Figure 7.

2.6.2 Second Year

During the second year of the MQP, the team worked together with Robot in Service of the Environment (RSE), where they improved and developed the company's computer vision and harvesting mechanism for their Guardian

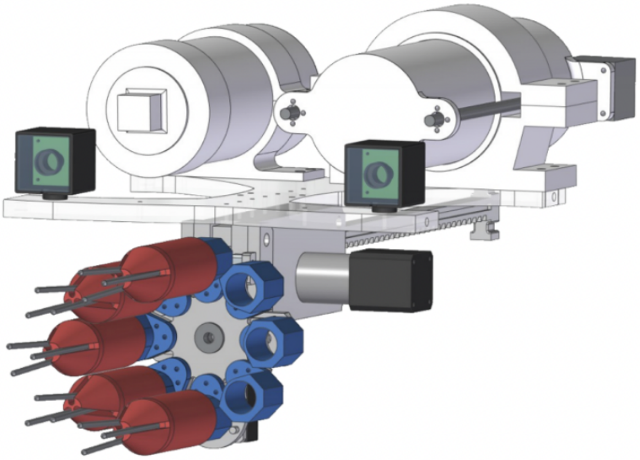


Figure 7: First Year MQP Iteration

Robot. The Guardian is a remotely operated robot that requires human input to identify lionfish, shock and stun the lionfish, store them, and bring them to the surface. The team created an object detection model to identify lionfish, an electrical configuration for the zapping mechanism and an intake system to store the lionfish (Antaya, Peterson, Conroy, & Ralph, 2019).

2.6.3 Third Year

The third year of the MQP, the team bought an off the shelf robot (BlueROV2) and built a custom electronic chamber for a high level CPU (Figure) and a harvesting mechanism for storing lionfish (Figure). They created a rudimentary navigation library but due to the COVID-19 pandemic were not able to make the test most of their code (Abadjiev, Chen, Ewen, Johnson, Olgado, Strickland, Whimpenny, Saperstein, 2020).

2.7 Containment System

In the first iteration of the project, there was a spear mechanism that was loaded with eight lightweight spear projectiles that surfaced after firing. The idea behind this concept was that the robot would incapacitate a lionfish and send it floating to the surface for pickup. However, due to the limited usage in a single run and the infeasibility of recovering the spears out in open water, it was decided that the robot should have an onboard containment system to store the lionfish it captured on patrol. This would also require a method of removing the fish from the spear to store it, and a method of extracting fish from the robot when it returns to base. While it is desirable to have the robot to have a high carrying capacity to reduce the number of trips necessary to clear lionfish, there are other limiting factors regarding the container's design: The container must not impede the robot's ability to move, especially when the container is full, and the fish inside must be easily accessible without injury to handlers.

2.7.1 Previous Iteration

In the first iteration of the project, there was a spear mechanism that was loaded with eight lightweight spear projectiles that surfaced after firing. The idea behind this concept was that the robot would incapacitate a lionfish and send it floating to the surface for pickup. However, due to the limited usage in a single run and the infeasibility of recovering the spears out in open water, it was decided that the robot should have an onboard containment

system to store the lionfish it captured on patrol. This would also require a method of removing the fish from the spear to store it, and a method of extracting fish from the robot when it returns to base. While it is desirable to have the robot to have a high carrying capacity to reduce the number of trips necessary to clear lionfish, there are other limiting factors regarding the container's design: The container must not impede the robot's ability to move, especially when the container is full, and the fish inside must be easily accessible without injury to handlers. The previous iterations of a container were constructed from plastic buckets, with slots cut into lids to allow the insertion of the lionfish via piston-operated retractable spear, and holes cut into the bucket walls for water to flow through. The incorporation of a bucket lid allowed for easy access to the lionfish when it was time to empty the container. However, the solid surfaces of the bucket caused the AUV to experience high amounts of drag, even with the holes cut into the sides. A later design tried to remedy the issue by replacing the bucket body with a net that was held open with a pair of curved wires, while still using a slotted bucket lid for the top. This reduced some of the drag, but the large flat bucket lid continued to impede the AUV's movement as it was a solid face positioned normal to the direction of travel. A new design was made to address these drag problems as well as a few other concerns regarding mobility.

2.7.2 New Design

This year's proposed containment system designs consist of a cage-like container and a pneumatically-operated opening underslung to the chassis. A cage was chosen over a mesh net to reduce drag, while still preventing the large-bodied lionfish from slipping out and reducing the chance of their fins and spines getting entangled. The cage fits closely to the rear chassis, hiding within the front profile of the robot instead of hanging below it. By doing this, the container minimizes both the front and lateral cross-sectional area, reducing the amount of drag experienced while travelling or turning. The cage is also easily detached from the robot, by removing the screws that hold it at the bottom, to allow rear access to the electronic systems.

The main opening of the containment system is the only part that increases the AUV's front profile. It consists of a pair of jaws that are underslung to the bottom, fitted with flexible mesh net to allow the jaws to open smoothly and a curved member to maintain the shape of the mesh. The jaws are powered by pneumatic cylinders to open and accept the spear after it impales a lionfish and rotates inwards. The jaws close around the fish, leaving a gap between them to let the spear retract and rotate back to its ready position. The lionfish, which is neutrally buoyant due to its aquatic nature, will be forced backwards and upwards inside the cage as the robot continues on its way. When the robot returns to base, the top of the container can be opened to allow people to pull out lionfish with a reach extender or fishing spear.

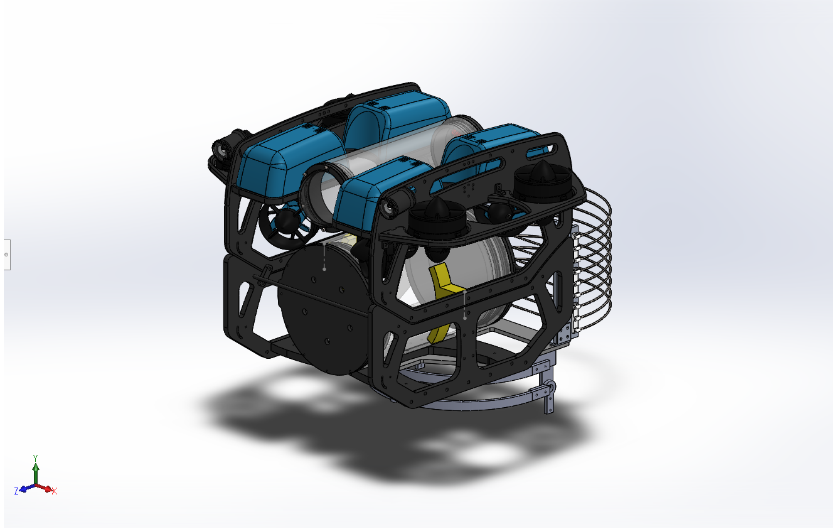


Figure 8: $\frac{3}{4}$ front view of robot with current container design.

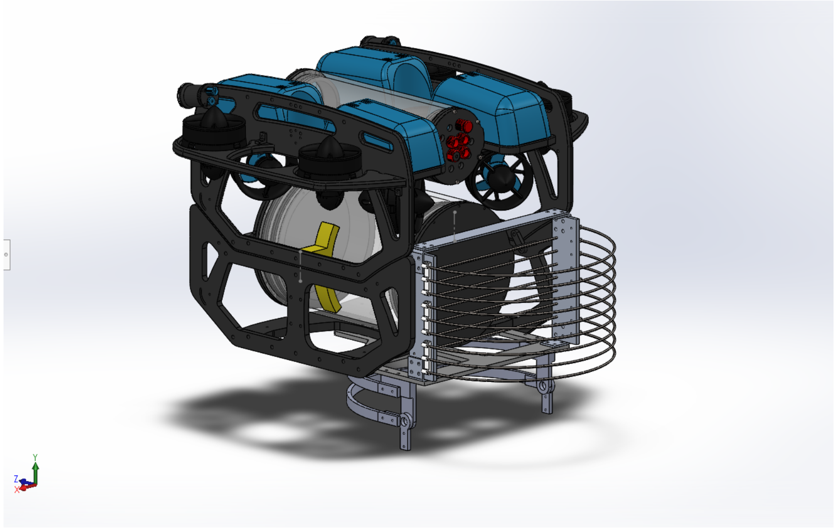


Figure 9: $\frac{3}{4}$ rear view of robot with current container design.

2.8 Stereo Vision

A pivotal need of an autonomous vehicle is the ability to perceive depth. The ability to determine the distance to various objects allows the robot to avoid obstacles such as fragile coral reefs. If a scuba diver is identified via the image classifier the robot can be programmed to maintain a specific distance away from the diver to assure safety. Very relevant to this project AUV is the need to identify a lionfish and position itself to spear the target. This can only be achieved if the robot can determine distances. Depth detection can also be used for 3-dimensional mapping, a critical element for localizing and navigating a real world environment. Here “depth” refers to the distance from the ROV to the object being measured, not to how far the object is beneath the water’s surface.

2.8.1 What is Stereo Vision

Stereo vision is the processing of deriving depth information by comparing the 2D information generated from two traditional cameras located at different positions. This process relies on the parallax effect, where the apparent position of an object for a viewer is different depending on the position of that viewer. This effect can easily be demonstrated if someone holds out their thumb at arm’s length and takes turns looking through either eye. The position of the thumb will change depending on which eye is used. Stereo vision can take advantage of parallax because the magnitude of the effect is dependent on how far away the object is from the viewer. If the switching eyes demonstration was repeated instead on a person across the street rather than a thumb at arm’s length, it would be observed that the difference in position from one eye’s point of view to the other would be much more subtle. This difference between the apparent position of the object is known as its disparity. The farther the object is from the two cameras, the smaller its disparity value will be. Using this principle, a mathematical model for a two camera system can be empirically derived which can be used to determine the distance of objects.

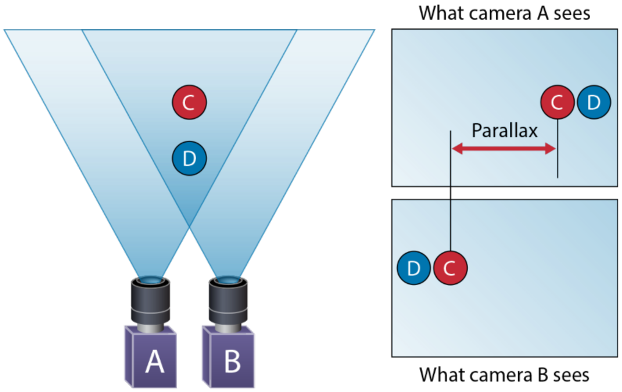


Figure 10: Stereo Vision Concept

2.9 Object Detection

Object detection is a branch of computer vision that involves locating and identifying different objects within an image. This ability is most often achieved by training a neural network on a large quantity of prelabeled images. The neural network is capable of extracting features from the data that can be used to identify these labeled objects in new, never before seen images. The applications for this technology are far-reaching and significant.

Autonomous cars and other machines require object detection in order to safely interact in their environment. For our robot the ability to identify lionfish and distinguish them from other species is paramount. It would also be extremely valuable to be able to identify and locate many other objects. The object detection system will need to be able to identify swimmers and divers, in order to ensure that the robot remains at a safe distance away. Motorboats must also be detectable since their propellers could do catastrophic damage to the robot. As such it must keep track of the surrounding boats as it surfaces to return to its support vessel.

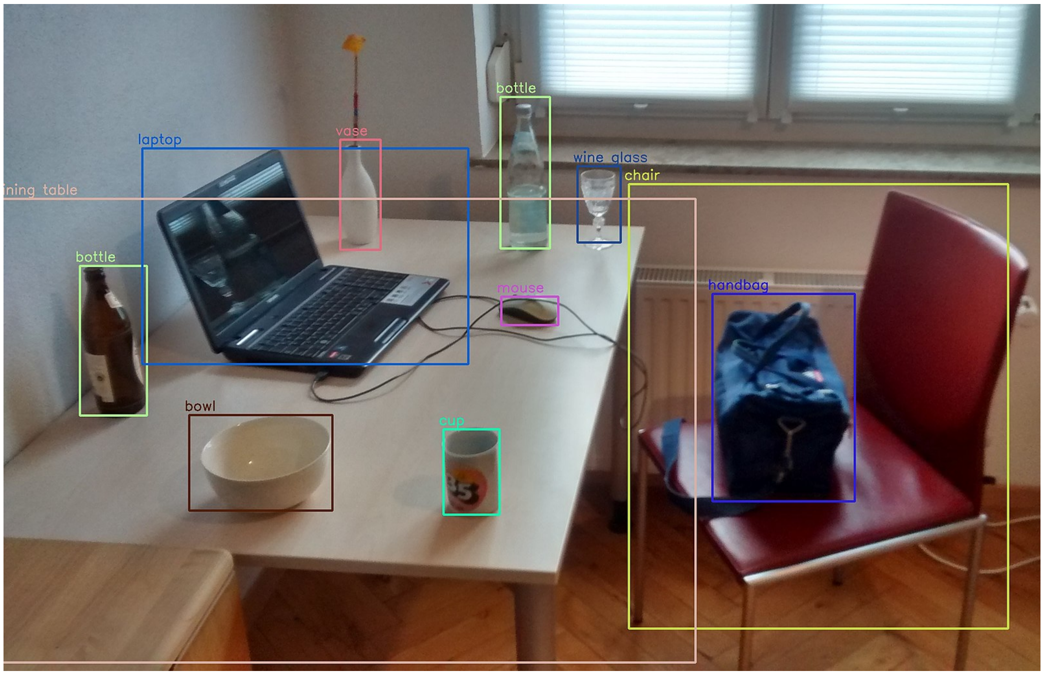


Figure 11: A Common Example of Object Detection

3 Methodology

To build an autonomous platform for capturing and harvesting lionfish, we had to implement the following functions in the robot :

- Create a navigation library
- Use stereo vision and object detection to identify and track lionfish
- Create better powering mechanism
- Build a Containment Mechanism for lionfish

We decided to split the above tasks into sub-teams, with every team focused with one objective: navigation, stereo vision, magnetic switch, and a containment mechanism.

3.1 Navigation

To be able to harvest lionfish, the robot has to be able to perform basic navigation functionalities such as:

- Driving straight
- Turning
- Ascending
- Descending

To achieve the above functionality, we utilized the built in low-level navigation capabilities by BlueRobotics, built high level code for driving forward, backward, ascending, descending. Finally we implemented PID control for reliable turning.

3.1.1 Pymavlink

BlueRobotics used the pymavlink communication library to send low-level PWM commands to different channels of the robot. There are 8 built-in channels that send different PWM Signals to the Pixahawk to control depth, thrust, yaw and more in the robot. The following channels map to their respective capabilities on the robot:

- Channel[2] - Dive Straight (Ascend/Descend)
- Channel[3] - Yaw
- Channel[4] - Thrust Forward/Backward

There are other channels that have different functionalities such as pitch, side to side movement, and more. But for the purposes of our MQP, we only used the mentioned channels as combining them can achieve the necessary path.

An example of how we implemented driving straight by utilizing the different built-in PWM channels is below:

```
def write_pwm(master, output_channel, output_val):
    rc_channel_values = [65535 for _ in range(8)]
    rc_channel_values[output_channel] = output_val
    master.mav.rc_channels_override_send(
        master.target_system,  # target_system
        master.target_component,  # target_component
        *rc_channel_values)


def drive_forward(master, val, time_to_drive):
    if val > 0 and val <= 100:
        output = (val * 5) + 1500
        end_time = time.time() + time_to_drive
        while time.time() < end_time:
            write_pwm(master, 4, output)


def main():
    # Create the connection
    print("======================")
    master = mavutil.mavlink_connection('udpin:0.0.0.0:15000')
    print("**********************")
    # Wait a heartbeat before sending commands
    master.wait_heartbeat()
    print("Waited a HeartBeat")
```

```
# Drive forward at half speed for 4 seconds
drive_forward(master, 50, 4)
```

In the above example, we use the main() function to establish a connection to the udpin:15000. It directly correlates to an ethernet connection to port 15000 that communicates from a host laptop to on board raspberry pi and pixahawk on the robot. After we establish a connection and we use the driveforward(master, 50, 4) to drive the robot in a straight line. This function takes in different parameters such as master, val, and timetodrive. Master parameter is the udp pin connection to the robot. Val is the thrust given to the robot from a range of 0 to 100. time to drive is the time needed to be driven by robot to cover the required distance. These parameters are used to control the thrust and time on the robot to cover the required distance in an underwater environment.

3.1.2 P Control

To implement reliable turning, we needed to use P Control (Figure 12). We created a custom P Control algorithm as we found it really difficult to integrate a PID library with BlueRobotic's PWM functions.

To create our feedback loop, we used an IMU that consisted of a accelerometer, gyroscope, and magnetometer. We tested the IMU by moving manually to different headings to check if the output heading was reliable for feedback

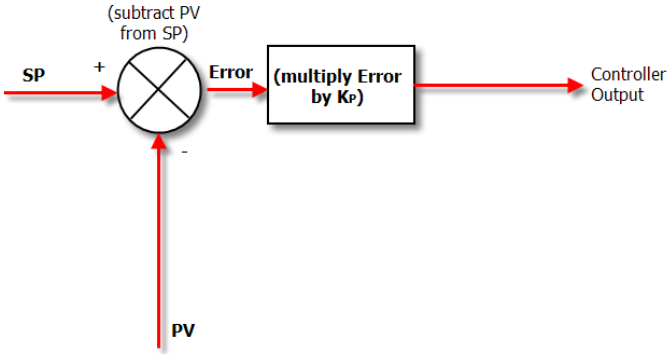


Figure 12: PID Controller

loop. We found that the heading results were accurate to use the IMU as the primary feedback mechanism for P Control turning.

To give PWM signals we normalized our heading to be in a range from -180 to 180. This helped us create an error function that checks the subtracts the current heading from the final heading, which is multiplied by a K_P gain.

With a lot of testing in the WPI Pool and checking the feedback mechanism we found that a K_P value of 0.65 gave us the best results for making turns reliably.

3.2 Stereo Vision

3.2.1 Why Use Stereo Vision

When it comes to depth perception, stereo vision isn't the only option for our underwater application. There exists underwater sonar devices such as the Ping360 Scanning Imaging Sonar from Blue Robotics which is capable of providing detailed depth information about the robot's surroundings. The big drawback with using this device is that it costs $1,975, which was beyond the budget of our project. There is a more affordable sonar option provided by Blue Robotics called the Ping Sonar Altimeter and Echosounder, two of which have already been integrated with our ROV by a previous year's team. These sensors cost $279 each, however their purpose is to detect very large objects such as the walls of a pool or the ocean's floor. The ping sensors are unsuited for the task of determining the distance to small objects such as individual lionfish. Stereo vision on the other hand only requires two traditional cameras. HD web cameras, like the ones we have used, can be purchased for around $20. This was much more appealing for our team as it would be far less expensive to implement. The low cost of these cameras also allows the possibility of putting multiple pairs of cameras all over the robot to provide depth information in all directions.

For this year's project we were simply focused on proving that stereo vision is a possible solution for underwater distance measuring and so only had one pair of cameras on the front of the robot. In addition to its affordability, a stereo vision system is more flexible, allowing for hardware upgrades in the future such as cameras with higher resolution or better low-light capabilities. Critically, stereo vision has the capability of being able to discern the distance of individual objects as opposed to returning one general distance value as with the ping sensors. As such our team decided to explore the efficacy of stereo vision due to the fact that it is an affordable method of getting depth information about specific targets.

3.2.2 Hardware and Software

In order for any stereo vision to work two cameras are required. Our team used two Sony IMX 322 cameras which had been purchased by the previous year's team. Often used as web cameras, these devices are small, require very little power, have a high resolution of 1080p and can work in low-light environments down to 0.01 lux. These characteristics make it appealing for a robot which is expected to operate at depths of 100 m, where much of the available light may only be from the vehicles own headlights When it comes to the backbone for the software of our stereo vision system we decided to use OpenCV. OpenCV is an open source programming library that focuses around computer vision applications.

3.2.3 Rectification

Before the stereo process can begin the cameras must first be rectified. The purpose of this rectification step is to digitally remove any distortions

Figure 13: Sony IMX 322 Camera

found within the cameras. There are two main sources for these distortions. The first are the lenses themselves, which by their shape or imperfections in their manufacturing, warp the incoming light so that the image projected on the digital sensors differs from reality. A common example of this optical distortion is called barrel distortion whereby lines that should appear straight are instead curved. Another source of distortion is the refraction of light. Light travels at different speeds depending on the medium it is in. If light travels from one substance to another at an angle, the change in the speed of light will cause the light to bend. This bending of the light is called refraction. The angle of the light's deflection depends on the two materials' refractive indices. The refractive index of a substance is the ratio of how fast light travels in a vacuum and how fast it travels in that substance. The larger the difference in the two refractive indices, the larger the angle the light will be deflected. It is this very effect which allows lenses to work.

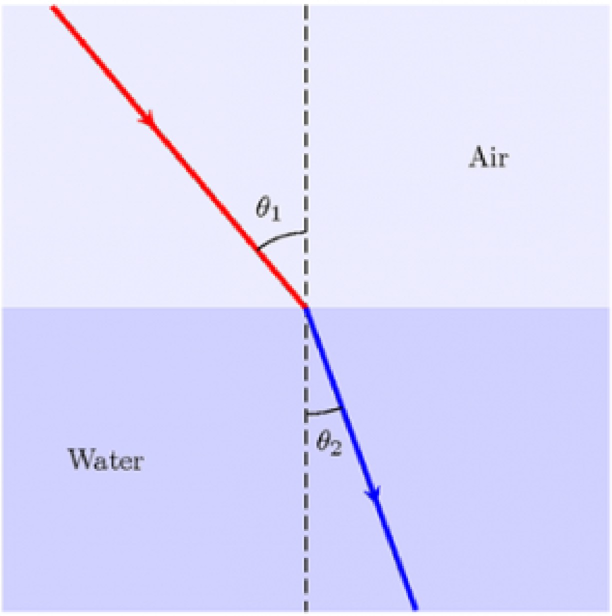


Figure 14: Refraction of Light

The issue for our project is that the cameras we are using are designed, like most cameras, to work in the open air, where light simply passes from air into the lenses. The cameras are built to compensate for this inherent distortion. On our underwater ROV however the light must first pass through the water, then through the transparent plastic of the waterproof camera case, then through the air trapped within the case, and finally through the lenses. The introduction of the water and the cameras case plastic will create additional distortion that the cameras were not designed to fix. This problem can be remedied with some software functions provided by OpenCV. The process begins with the printout out of a checkerboard pattern. The two cameras take pictures of this checkerboard in a variety of positions and orientations. This can be seen in Figure 15. The OpenCV program is capable of identifying the corners of the squares within the pattern and then draws lines between them. Since the program knows that the pattern is supposed to form straight lines, any change caused by distortion can be identified. Transformations can then be applied to the image so that the lines appear straight as they are supposed to. When this is done the cameras have been rectified. This process was repeated for both when the robot is out of the water and when it was submerged. This allowed the team to test the vision system without the need of taking the robot to the pool.

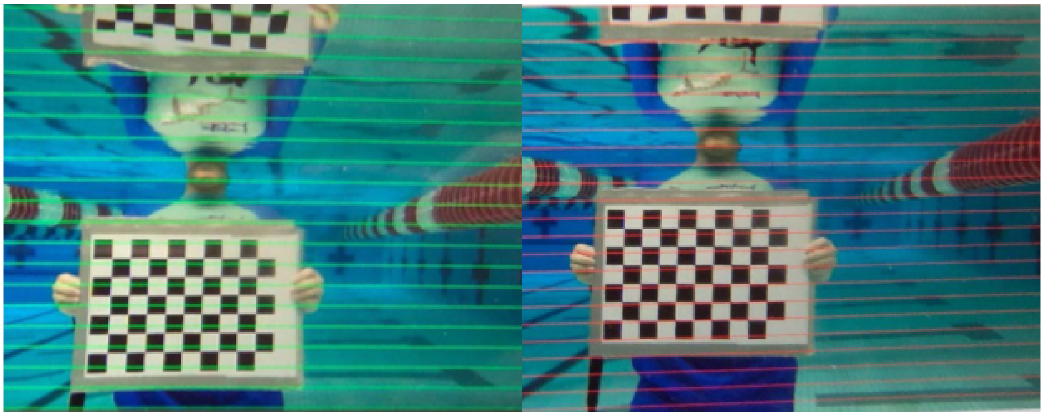


Figure 15: Before (Left) and After (Right) Rectification

3.2.4 StereoBM

Once the cameras are rectified it is time to build what is known as a disparity map. There are two functions in OpenCV that can do this, StereoBM and StereoSGBM. StereoSGBM was first attempted as it is supposed to produce more detailed results. Unfortunately, the performance of this method was too slow, having an average frame rate of 0.17 frames per second (fps). This was much too slow for our application and so it forced us to try the simpler StereoBM method. This function had a much better performance of about 12 fps. StereoBM is a block-matching algorithm and works by grouping a small number of neighboring pixels in one image, and then scanning through another image looking for as similar a block as possible. In our case the two images being compared were the frames being generated by the left and right cameras This is why the cameras need to be rectified, distortions can make it impossible for the block-matching algorithm to pair matching pixels. Once the block-matching process is done StereoBM compares how many pixels the block from the right camera has shifted along the x-axis compared to its left camera counterpart. The number of pixels it has moved is the disparity value for that block pair. By repeating this process for all the blocks of pixels in the frame, StereoBM creates a disparity map for that frame. Figure 16 shows a frame from the left, camera taken underwater, alongside its disparity map.

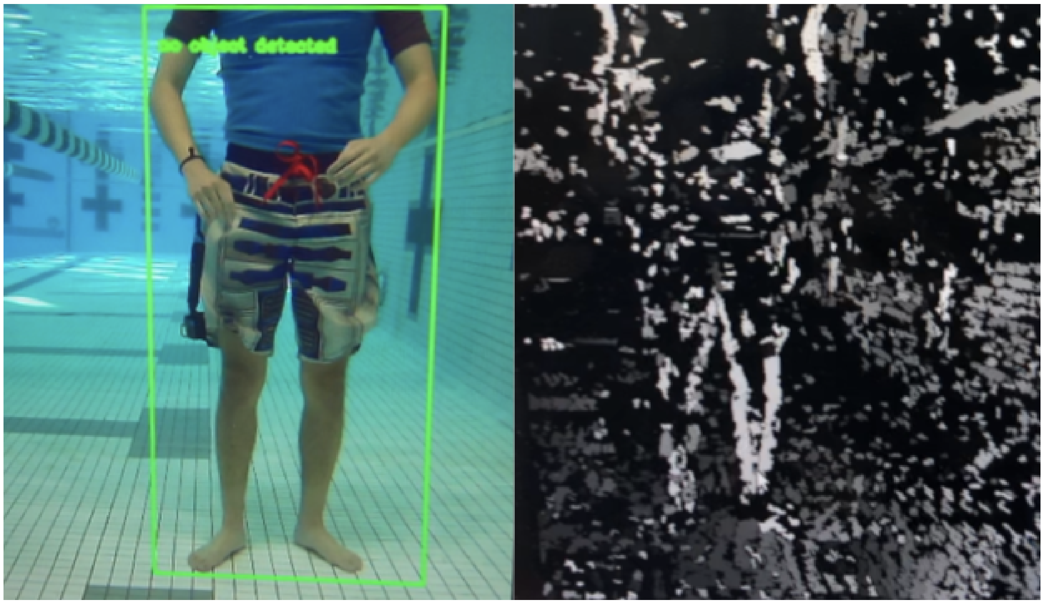


Figure 16: A Image (Left) and Its Corresponding Disparity Map (Right)

3.2.5 Tuning Parameters

There are numerous parameters associated with StereoBM, all of which were adjusted to determine what created the best results. With stereo vision there is a trade-off between increasing the probability that that disparity of a pixel can be determined and increasing the noise of the map. A system with very little noise will also see very few objects and a system that is able to see many objects will also be quite noisy. It is extremely difficult to identify setting values which yield acceptable results. Through much trial and error it was found that a few parameters had the biggest impact on the quality of the disparity map. For example having a uniqueness ratio of 15 was found to produce the best results. One of the biggest issues with StereoBM is that when it is attempting to match blocks of pixels between the two images it will often find multiple potential matches. Unless a single block can confidently be matched with the original, no disparity value is assigned. The uniqueness ratio defines how much more likely the best matched block must be compared to the other potential matches. The higher this value is the more confident we can be that the disparity value given is current. It also reduces the amount of noise. Unfortunately it also reduces the probability that pixels we are interested in are matched at all. Other useful parameters are those that filter out speckles. Speckles are small groups of disparity values that appear randomly throughout the map and can be considered noise. By tuning these parameters correctly it is possible to greatly reduce the speckle noise.

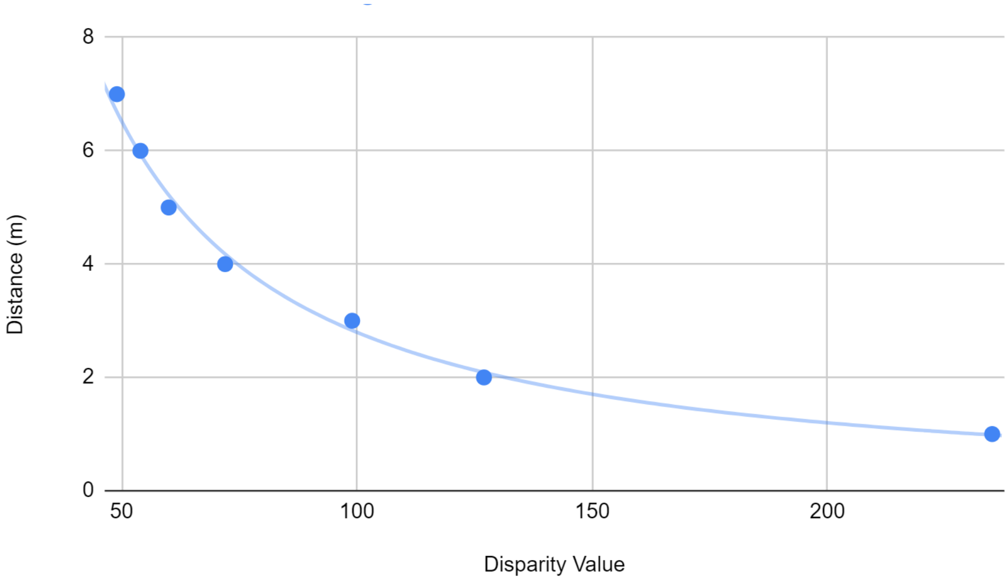


Figure 17: Disparity vs Distance Measurements

3.2.6 Calibrating Distance Measurements

After the disparity map is created and producing satisfactory results, the next step is to convert the disparity value of a pixel to a distance measurement. This is a very straightforward process. A simple program was written that would allow a user to read the disparity value of an object by clicking on that object once the user identified it on the disparity map. In order to minimize any possible noise the program averages the disparity value of the pixel selected by the user with its 8 adjacent neighbors. We would place an object 1 m in front of the robot and record its disparity value. This would be repeated, at a distance of 2, 3, 4, 5, 6, and 7 m. The data collected was then plotted in google sheets on a scatter plot. Multiple trend lines were tested and the one with the largest R2 was chosen to be the equation that would convert disparity values into distance measurements. We first completed this process dry and were happy with the results. We first completed this process dry and were happy with the results. The results of this test can be seen in Figure 17. However, when the robot was submerged in water it was found that the distance calibration no longer gave accurate results. This was expected due to the refractive properties of the water. As such a separate distance calibration was done underwater.

3.3 Object detection

One of the objectives of our project was to prove that data derived from stereo vision could be used for the purpose of navigation. With this being simply a technology demonstration we did not spend time training a neural network to identify lionfish or diver, something that would most certainly be needed before the robot is ready to go hunting in the ocean. Even if we had devoted time towards building such a model, we did not have access to lionfish to test it on, and we would have to have used another method for testing anyways. Instead, the team decided to use an already trained object detection program. After doing some research a model was found that was pretrained, was highly accurate, and worked in real time. The program was trained on a commonly used dataset called COCO (Common Object in COntext). This data includes a variety of everyday objects such as cars, dogs, cats, bicycles, people and sports balls. This was important because we wanted a model to be trained on people so that we could test how well the robot could react to the presence of swimmers. A sports ball such as a basketball could also serve as a decent model for a lionfish due to their similarity in size. Figure 18 shows us testing the efficacy of this program for the first time. It is important to reiterate that the majority of the object detection code was not written by the team and should instead be credited to Murtaza Hassan. The code uses OpenCV which made it convenient to integrate with our stereo vision. To make the program simpler the code was altered to only detect people and sports balls. The threshold values also had to be changed so that we could achieve consistent identification.

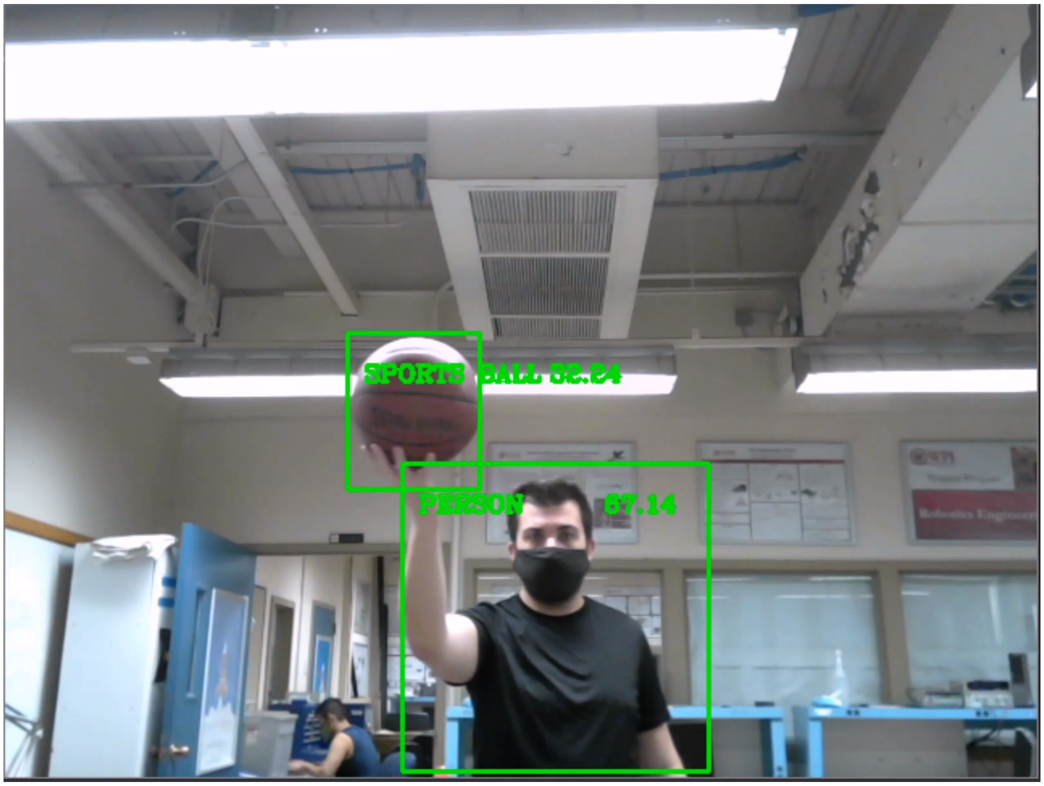


Figure 18: Object Detection

3.4 Combining Stereo Vision and Object Detection

With both stereo vision and object detection working the next step was to combine them. By doing so the ultimate goal was to have a vision system that could automatically identify and measure the distance to objects in the robot's path. This would allow the robot to get into better positions for lionfish harvesting or to avoid people or hazardous areas. The way the system works is that when the object detection program identifies a person or a sports ball it creates a bounding box, a rectangle that encapsulates the object. The stereo vision system then takes the location of this bounding box within the frame of the camera and looks at the disparity values within it to determine the distance of the object. This can be seen working in Figure 19. The stereo vision system does not actually search every pixel within the bounding box instead it searches a scaled down version of the box. The original reason for this was to minimize the effect of background noise. Since the shape of the bounding box is rectangle and the object is not, some of the pixels enclosed within the box will be of the background and not the object of interest. Reducing the size of the search area was a means of mitigating this issue. Another reason for scaling down the bounding box was discovered while testing. It was noted that the frame rate of the vision program was slower when the identified object took up a wider field of view. It was reasoned that the larger bounding box required that more disparity values had to be measured and so slowed the process. By making the search area smaller we could improve the frame rate, however there is a drawback to making it smaller. This is because of the nature of stereo vision. Since the block-matching algorithm only matches pixels between two images it is confident are the same, it only works well on high textured areas or where there is a large contrast. As a result it tends to only identify the edges of objects which often are clearly different from the background, while ignoring their centers. Therefore, if the bounding box is scaled down too much it will become unlikely to get any distance reading. It was experimentally determined that scaling the bounding box by 20% produced good results.

Once the stereo vision system has a list of all the disparity values within the scaled down bounding box it needs to make a decision on what disparity value to assign to the whole object. At first, we simply tried taking the average of all the values. This however proved to be too susceptible to noise. The distance return was always larger than the true value and the value could change pretty significantly from frame to frame as random outliers popped in and out in the background. Next we tried taking the median of all the values and while that did reduce the variability of the returned distances, the values were consistently too large. The next idea we had was to take the mode of the disparity values. The thought here was that the most common value is most likely the correct one. This works rather well and is less prone to variability. The situation sometimes occurred where there was more than one mode, which at first caused the program to crash. This happened most often when the object was not picked up well by the stereo vision and so no particular disparity value would become dominant. This was quickly remedied by creating a function that created a list of all the modes and then averaged them. Once the disparity value is assigned to the whole object it is converted to a distance measurement and displayed on screen.

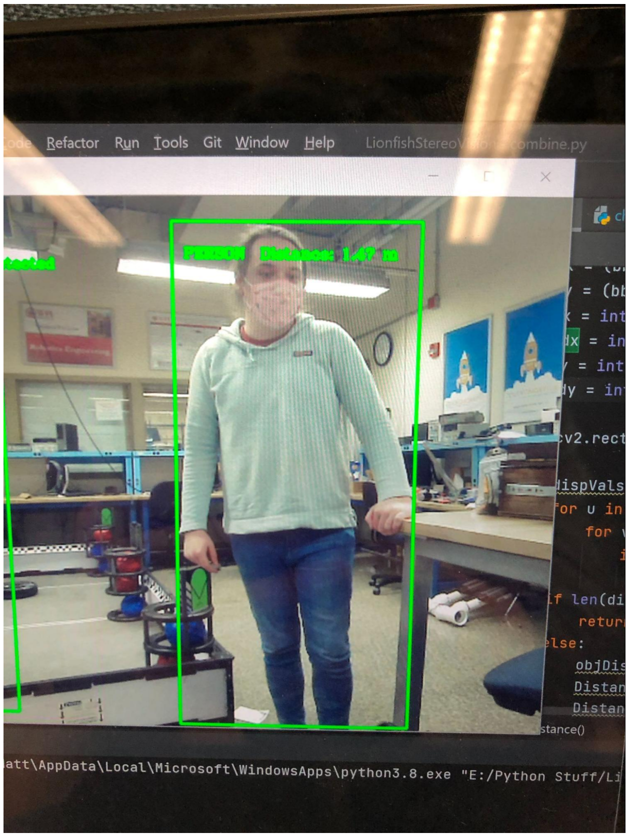


Figure 19: Stereo Vision Combined with Object Detection

3.5 Powering the robot

Previously the users of the project would have to open the lower pressure chamber to access the batteries and plug them in to turn on the robot. After plugging in the batteries the pressure chamber has to be closed, then make sure the computer is able to connect to the robot via the tether and check for leaks. This process can take anywhere from 30 minutes to an hour and then the robot is ready to go in the water. During this time the batteries are discharging which takes away from the time the robot could be hunting lionfish. This could be fixed by adding a power switch between the batteries and the rest of the electronics, however there are a couple problems with this.

The first problem with adding a power switch between the batteries and the rest of the electronics is the batteries can output up to 132A in bursts and 90A continuously which is higher than most switches can handle. A potential solution to this is using a relay. A relay has four pins, two are the coil and two are the contacts. The coil controls whether current should be able to pass through the contacts or if there should be an open circuit between the contacts preventing the current from passing through. When a magnetic field is applied an armature inside the relay connects the contacts; the magnetic field is produced dependent on the current going through the coil. The coil dissipates power equal to the voltage drop across the coil times the current (Zhai, Fan, & Wang, 2007). Since the coil doesn't need much current to provide enough power to activate the electromagnets, using a simple switch will suffice to turn on and off the power supply to the rest of the electronics as seen in Figure 20 below.

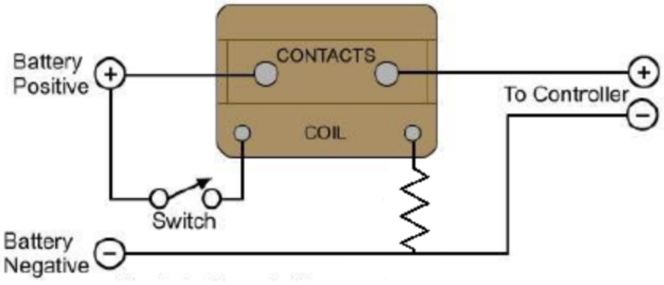


Figure 20: How relays are traditionally wired (RLY-24150)

The second problem with a power switch is that it needs to be able to toggle when the robot is sealed watertight and submerged in water. This can be achieved by a waterproof switch with all the wiring to it watertight, however this has its own problems. This switch could be mounted through and sealed on the bulkhead of the pressure chamber. There is a small chance a switch like this could be turned off if there is a piece of rock or fish bumps into it while underwater, and while this is unlikely a better option would be using a magnetic switch. Magnetic switches can work through the $\frac{1}{4}$" acrylic of the pressure chamber walls.

Combining the two solutions we can use a magnetic switch to trigger the coils on a relay to toggle the power to the robot. There are purely mechanical magnetic switches called Reed switches and there are magnetic switches called Hall Effect switches. Reed switches can handle the amount of current that is needed for the coil to toggle, however they also require the magnet to stay near the sensor at all times to remain on. This would mean the magnet would have to have a mounting mechanism to keep the robot on. Another option is a latching Hall Effect switch which toggles between letting current through based on the polarity of the magnet brought near it. Latching Hall Effect switches generally don't work with the amount of current needed to activate the coil. One way the current limitation of the latching Hall Effect sensors can be fixed is using a transistor to increase the current going into the relay coils.

3.5.1 Implementing the circuit

The relay we chose was a ZT662-12V-120A as it could handle the maximum continuous current output of the batteries (90A) and is easily available. We chose to use a latching Hall Effect switch and a transistor so we didn't need to worry about anything bumping the robot or a magnet falling off turning off the robot unintentionally. Calculating the power dissipated across a component can be done using Equation 1.

$$P = I * V_{\text{drop}} \tag{1}$$

Equation 1: Ohm's Law for power

The power required to activate the coil in the relay is 4.8W which means when solving Equation 1 for the current with a voltage drop of 12V the current required will be 400mA. To activate the coil we chose the NPN transistor ZTX694 which reaches saturation when the current to base (I_B) is 5mA which allows the current across the transistor from the collector to the emitter (I_c) to be up to 400mA. The latching Hall Effect switch we chose was US1881 which outputs a typical current of 6mA was enough to reach saturation in the transistor. The resistance need to have a specific current can be calculated using Equation 2.

$$R = \frac{V}{I} \tag{2}$$

Equation 2: Ohm's Law for resistance

To limit the current to a max of 8mA at 16.4V and a minimum of 5mA at 12V we added a 2kΩ resistor before the Hall Effect switch. The latching Hall Effect switch we chose only works if there is a pullup resistor across it. This means that there will always be a marginal amount of current flow across the switch and when the switch is closed the current won't be affected by the pullup resistor. We 75kΩ pullup resistor across the switch as this would mean that there would be a maximum of 200μA traveling into the transistor I_B when the latching Hall Effect switch is latched in the open position. Another thing needed for the circuit was a freewheeling diode to protect the transistor from reverse voltage spikes across the coil of the resistor. Before we put the circuit on the robot we wanted to test the circuit in Multisim and modeled it as shown in Figure 21.

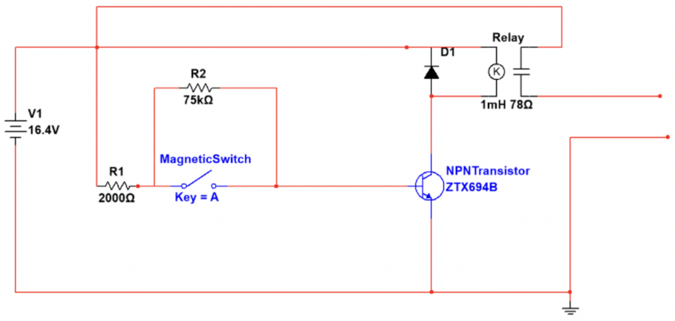


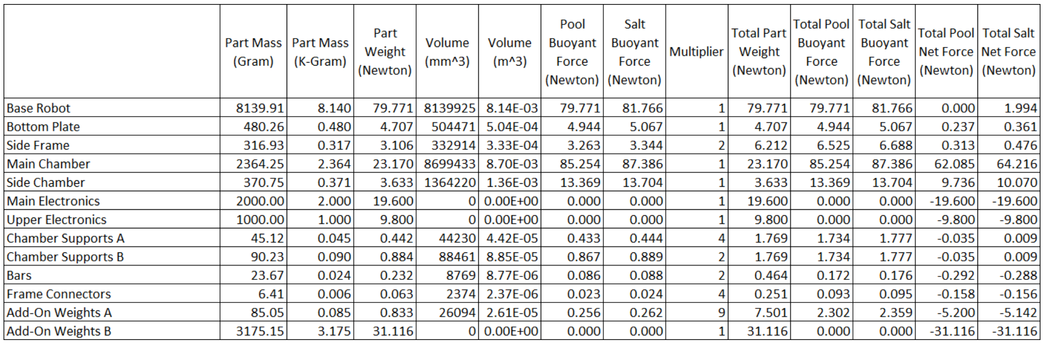
Figure 21: Circuit diagram of magnetic switch and relay

3.6 Containment System

The container design was developed in Solidworks, allowing for components to be virtually tested through simulations. The containment system is built with a large flat baseplate that is fastened to the robot's bottom frame. Since this baseplate is carrying the combined weight of the container and the weight of the opening mechanism, it needs to be strong enough to hold these parts sturdy while in operation. As well as this, there was the additional challenge of the robot being too positively buoyant, stranding the vertical-motion thrusters out of the water, where they would be rendered useless. The new parts would also have to be heavy enough for the robot to maintain neutral buoyancy, with or without extra weights applied. Finally, a CFD analysis is necessary to understand how the containment system interacts with the water around it.

3.6.1 Buoyancy Analysis

Using Solidworks to set material properties to parts of the robot and evaluate their volume and mass, it was possible to estimate the buoyant force that the robot was currently experiencing. The weight of each component was calculated by multiplying the mass in kilograms by the gravitational constant of 9.8 m/s2 while displacement force was calculated by multiplying the volume of a part by the gravitational constant and the density of the surrounding fluid; in this case, the density of water, which is 1000 kg/m3 for freshwater and 1025 kg/m3 for saltwater. Small, lightweight parts such as screws and wires were neglected due to their small mass. The net buoyant force of each part was obtained by subtracting the weight value from the displacement force value. For parts that appeared more than once throughout the robot's assembly, the net force was multiplied by that appropriate part count. Finally, the overall buoyant force of the robot was calculated by adding all part buoyant forces together.



	Part Mass (Gram)	Part Mass (K-Gram)	Part Weight (Newton)	Volume (mm^3)	Volume (m^3)	Pool Buoyant Force (Newton)	Salt Buoyant Force (Newton)	Multiplier	Total Part Weight (Newton)	Total Pool Buoyant Force (Newton)	Total Salt Buoyant Force (Newton)	Total Pool Net Force (Newton)	Total Salt Net Force (Newton)
Base Robot	8139.91	8.140	79.771	8139925	8.14E-03	79.771	81.766	1	79.771	79.771	81.766	0.000	1.994
Bottom Plate	480.26	0.480	4.707	504471	5.04E-04	4.944	5.067	1	4.707	4.944	5.067	0.237	0.361
Side Frame	316.93	0.317	3.106	332914	3.33E-04	3.263	3.344	2	6.212	6.525	6.688	0.313	0.476
Main Chamber	2364.25	2.364	23.170	8699433	8.70E-03	85.254	87.386	1	23.170	85.254	87.386	62.085	64.216
Side Chamber	370.75	0.371	3.633	1364220	1.36E-03	13.369	13.704	1	3.633	13.369	13.704	9.736	10.070
Main Electronics	2000.00	2.000	19.600	0	0.00E+00	0.000	0.000	1	19.600	0.000	0.000	-19.600	-19.600
Upper Electronics	1000.00	1.000	9.800	0	0.00E+00	0.000	0.000	1	9.800	0.000	0.000	-9.800	-9.800
Chamber Supports A	45.12	0.045	0.442	44230	4.42E-05	0.433	0.444	4	1.769	1.734	1.777	-0.035	0.009
Chamber Supports B	90.23	0.090	0.884	88461	8.85E-05	0.867	0.889	2	1.769	1.734	1.777	-0.035	0.009
Bars	23.67	0.024	0.232	8769	8.77E-06	0.086	0.088	2	0.464	0.172	0.176	-0.292	-0.288
Frame Connectors	6.41	0.006	0.063	2374	2.37E-06	0.023	0.024	4	0.251	0.093	0.095	-0.158	-0.156
Add-On Weights A	85.05	0.085	0.833	26094	2.61E-05	0.256	0.262	9	7.501	2.302	2.359	-5.200	-5.142
Add-On Weights B	3175.15	3.175	31.116	0	0.00E+00	0.000	0.000	1	31.116	0.000	0.000	-31.116	-31.116

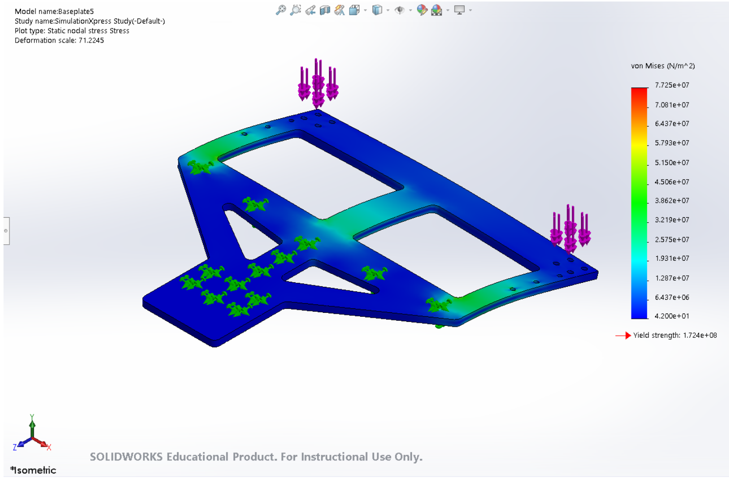
Figure 22: Calculations for the net forces on parts. Densities 1000 kg/m3 for pool and 1025 kg/m3 for saltwater.

3.6.2 Material Stress Analysis

The part of the container system that experiences the most stress is the baseplate that attaches to the bottom of the robot. It bears the weight of the cage and opening mechanism, and also experiences additional forces from drag as the robot moves through water. Due to this, it is necessary to ensure that this part was made of durable and lasting material that will not wear out or deform easily in a short span of time. An additional constraint to material choice is the fact that the robot will be operating in salt water, which is corrosive to most metals.

The materials choices selected for this plate were as follows: AISI 316 stainless steel, 5052-H32 aluminum alloy, 5052-H34 aluminum alloy, 6061-T6 aluminum alloy, high-impact acrylic, and high-density polyurethane or HDPE. The reasoning behind these materials is their corrosion-resistant nature, and common use in marine applications. In fact, some parts of the robot are made of these very materials, such as the frames (HDPE), the chamber walls (acrylic), and the chamber retaining bars (aluminum alloy).

The estimated max weight on the baseplate was the weight of the cage section, as the lionfish would be neutrally buoyant and thus not exert any weight. Major components such as the plates, bars, and ribs were made of aluminum alloy and stainless steel, while smaller parts such as the rib fasteners were made of 3D-printed ABS plastic. The result was a mass of 1.59 kg and therefore a weight force of around 15.58 N or 3.5 lb. It should be noted that this alone would be enough to sink the robot completely with additional ballast weights still attached. In the simulation, a force of 16 N was applied as a load to the rearmost screw holes, while the frontal pattern of screw holes we established as a fixed support. This process was repeated for all the material options for the baseplate.



Model name:Baseplate5
Study name:SimulationXpress Study(-Default-)
Plot type: Static nodal stress Stress
Deformation scale: 71.2245

Figure 23: Baseplate part in stress simulation.

3.6.3 CFD Analysis

Although the cage has a reduced front and side profile compared to the buckets to cut down drag, there was still concern on how the cage would affect mobility of the robot. The positioning of the cage meant that, at a certain height, it would lie in the streams of the robot's rearmost lateral motion thrusters. Though the front thrusters would not be blocked, this may ultimately hinder the robot's mobility, eventually impacting its power usage and ability to remain effective underwater.

Two similar designs were proposed, which share similar dimensions except height. One is 37 cm tall, reaching up to the top of the robot, while the other is 19 cm tall, reaching only halfway and stopping just below the thrusters. The taller one has an internal volume of around 34202 cm3 while the shorter one has an internal volume of around 17563 cm3. Assuming the average adult lionfish has a volume of around 662 cm3, the taller cage can carry up to 51 fish while the smaller cage carries up to 26. A standard-sized ZooKeeper container for human divers to collect lionfish ranges between 18 to 45 lb. Assuming the average adult lionfish has a mass of 2 lbs, the larger standard-sized ZooKeepers can carry up to 22 fish. Therefore, both the tall and short variations of the cage are valid, as they are greater than or equal to the amount of fish a human diver can harvest.

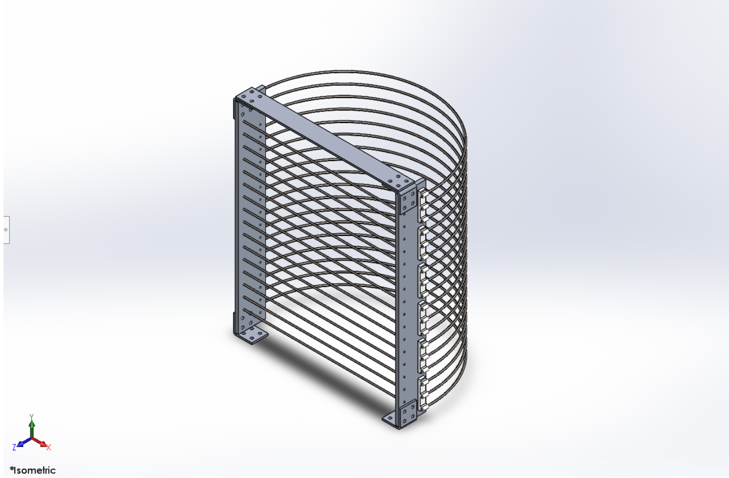


Figure 24: Tall cage design.

A series of CFD simulations was run to test how these designs affected the flow of water from the rear thrusters. The first simulation had a mockup of only the thrusters by themselves to evaluate how the flow from a robot without the cage or with the short cage would behave. The second and third simulations had the thrusters and cage bars to evaluate the flow from a robot with a tall cage. The third included multiple simplified models of

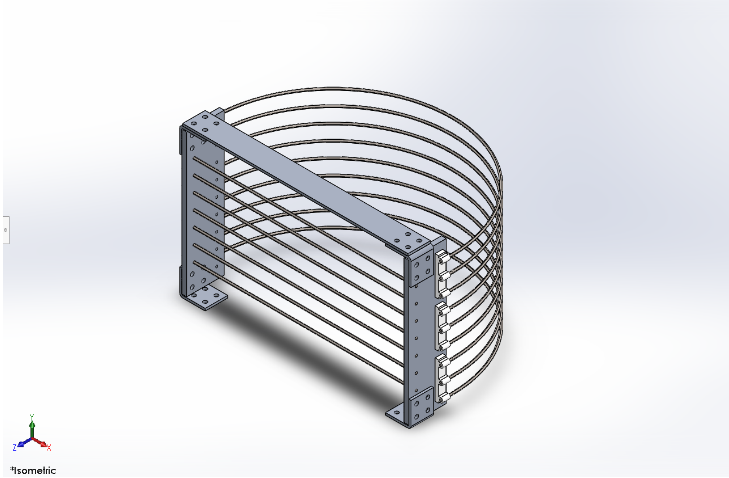


Figure 25: Short cage design.

lionfish bodies to simulate a full load.

4 Results

4.1 Navigation

We were able to achieve what we wanted and built a solid navigation library for the next year's team to build off on. We can reliably drive straight, hold a certain depth, ascend a certain depth. Using a custom P-Control function we were able to turn with an of error of +/- 4 degrees. To give the next year's team a better starting point for navigation, we documented our code and have instructions on how to find more resources for an even more advanced navigation system to do pitch, side to side movement and more.

4.2 Stereo Vision

The visual system did not perform as well as we had hoped. The biggest issue is that the combination of the stereo vision and object detection programs greatly reduced the frame rate of the system. When either of the two programs run independently they run at close to 20 fps. However once the two are combined the frame rate drops to 2-3 fps when running on the Jetson Xavier. This frame rate is highly dependent on what computer is running it. For example, on one of our member's laptops the frame rate was as low as 0.3 fps. 2-3 fps is very low for any application that is working in real time. With this frame rate it would be possible to use distance data to inform navigation decisions if the robot was moving slowly enough. However, it would be unlikely that this performance would be adequate to allow the robot to successfully track a moving lionfish and get into position to harvest.

We also discovered how unreliable stereo vision systems are at producing a disparity map. The system is best at identifying large objects that are highly textured or have well-defined edges. It can persistently make out the outline of a person or the edges of a table, or the frame of a window. It however struggles with small mono-colored objects such as a basketball. We attempted to improve results by creating a striped pattern on the ball using colored tape. This did not appear to enhance the results. The ball could still be seen on the disparity map but only a small portion of it was visible. It was also observed that the ball could not be detected when it was closer than 1.5 m to the robot. Larger objects such as people can get as close as 1 m before the stereo vision struggles to see it. One thing that did surprise us with how well it worked was the system's ability to use disparity values to accurately measure distances. At a range of 1-3 m the calculated distances were nearly always within 3% of the actual value. At distances of 3-6 m that error increases to 10%. We figured that this was more than sufficient for hunting lionfish. It is convenient that the error of the distance measurement decreases as the robot gets closer since having an accuracy within a few centimeters is not necessary when an object is far away. Note that a 3% error at 1.5 m would result in an error of +/- 4.5 cm. We deemed this sufficiently accurate to inform the robot that it is within striking distance of a 12in lionfish.

The object detection system produces mixed results. It can nearly always identify a person when one is in front of the camera. This is regardless of what they are wearing, whether they are in strange positions whilst floating underwater, or only have a small portion of their body in view of the cameras. The program could easily identify an individual at a range of 8 m (the longest distance we were able to test). The system did not fare as well at identifying the basketball. When in front of the camera the basketball was only identified as such 60% of the time. Results were especially bad if the ball was being held by a person. In this situation the person would still be identified, but the ball would simply be ignored. The basketball was also never detected beyond a distance of 5 m. This poor ability to be identified combined with the fact that the stereo vision also has difficulty seeing the ball, meant that getting the robot to navigate to the ball and stop 2 m away would be extremely difficult.

4.3 Power Switch

When we were able to correctly run the simulation of the circuit for the relay, magnetic switch, and transistor we implemented it onto the robot using a breadboard. After getting the robot to turn on when it was wired correctly on the breadboard the circuit was rebuilt on a protoboard. When it was built on the protoboard we tested how long the robot would stay on for and if there would be any problems when the robot was running. We were able to leave the robot powered on for 7 hours without turning it off with no problems, the voltage of the battery decreased from 16.4V to 13.8V. The robot also experienced no problems when running the motors extensively underwater.

4.4 Containment System

4.4.1 Buoyancy Results

With only the essential components and navigation systems, the robot in the pool experienced a high positive buoyant force of 9.54 lb. due to the hollow, air-filled nature of the main electronics chamber. During testing at the pool, we weighed the robot down with ballast weights placed inside the main chamber and along the chassis frame, which reduced the positive buoyant force to just 1.38 lb. These were enough to keep the vertical-motion thrusters submerged, leaving just the top of the robot's floats breaking the surface.

In salt water, the estimated positive buoyant force is 10.63 lb. without the weights and 2.48 lb. with the weights. Therefore, more weight is required to keep the robot submerged in saltwater, either by adding on more ballast weights or ensuring that the weight of the mechanisms match or exceed these forces.

4.4.2 Material Stress Results

MATERIAL	YIELD (N/m2)	MAX (N/m2)	MASS (kg)	SAFETY FACTOR
AISI 316	1.72E+08	7.73E+07	2.082	2.23
5052-H32/H34	1.95E+08	7.31E+07	0.698	2.67
6061-T6	2.75E+08	7.31E+07	0.703	3.76
HDPE	2.60E+07	6.66E+07	0.248	N/A
Acrylic	4.50E+07	7.16E+07	0.312	N/A

Simulations indicate that the baseplate cannot be made of either HDPE and high-impact acrylic, as the stress experienced when the cage is attached exceeds the yield stresses of these materials. There is the option of increasing the baseplate's thickness to allow the usage of these lighter materials, but this would increase the front-view profile, and additional weight is necessary to keep the robot underwater anyway, negating the need for lighter materials. Therefore, the baseplate should probably be made of stainless steel or aluminum, as these materials can carry the weight of the cage and are heavy enough to weight the robot down in operation; assuming a stainless-steel baseplate is used, the total extra weight to the robot is 36.4 N or 8.18 lb. which is almost enough to completely submerge the robot without additional ballast weights in both fresh and saltwater.

4.4.3 CFD Results

The results of the first simulation show that the flow from each thruster consists of a smooth stream of constant velocity and pressure, with a widening cone where the two streams intersect.

With the cage bars added, the flow of each thruster is redirected around some of the bars, but continues smoothly afterwards at the same velocity and pressure. The same goes for the cone where the thrusters intersect. This indicates that the bars themselves have little effect on the thruster streams.

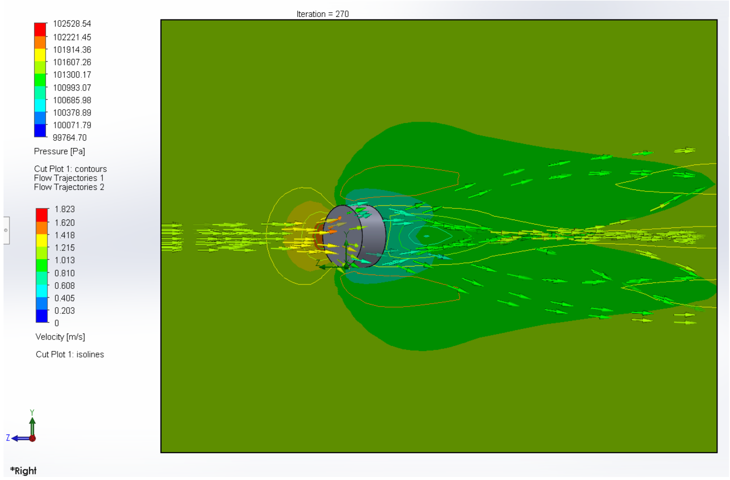


Figure 26: Flow of rear thrusters with no additional parts. Cutaway at single thruster.

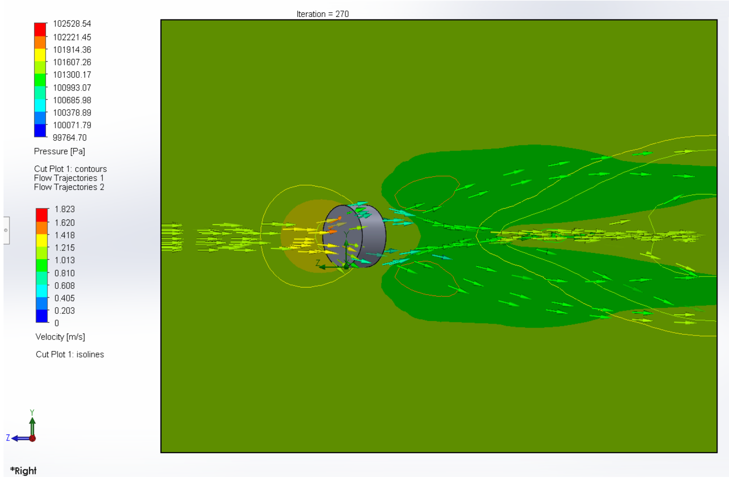


Figure 27: Flow of rear thrusters with no additional parts. Cutaway at thruster stream intersection.

However, when the cage is fully loaded and packed with lionfish, the flow of each thruster is redirected greatly from their original streams, and a "dead spot" of relative-zero-velocity generates behind the center of the full cage.

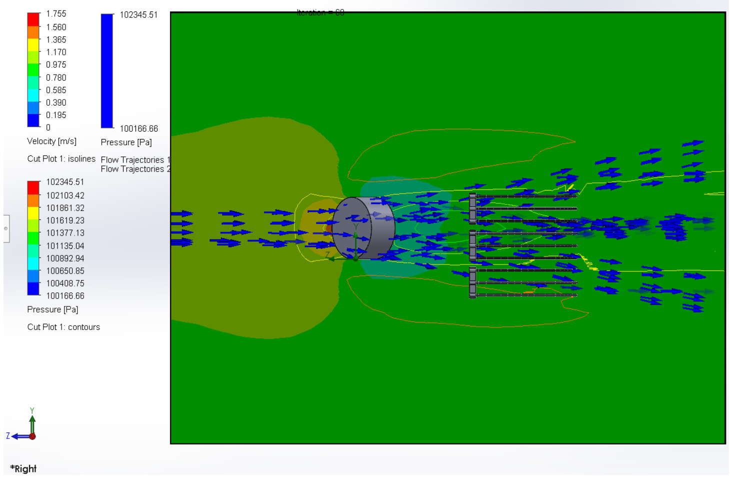


Figure 28: Flow of rear thrusters with empty tall cage. Cutaway at single thruster.

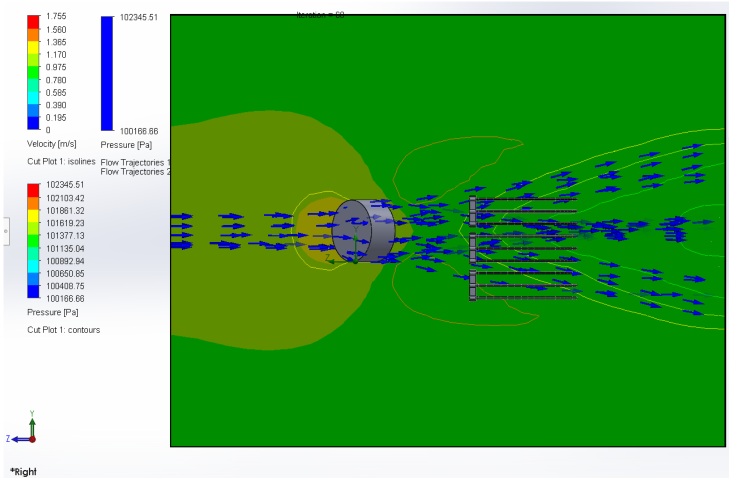


Figure 29: Flow of rear thrusters with empty tall cage. Cutaway at thruster stream intersection.

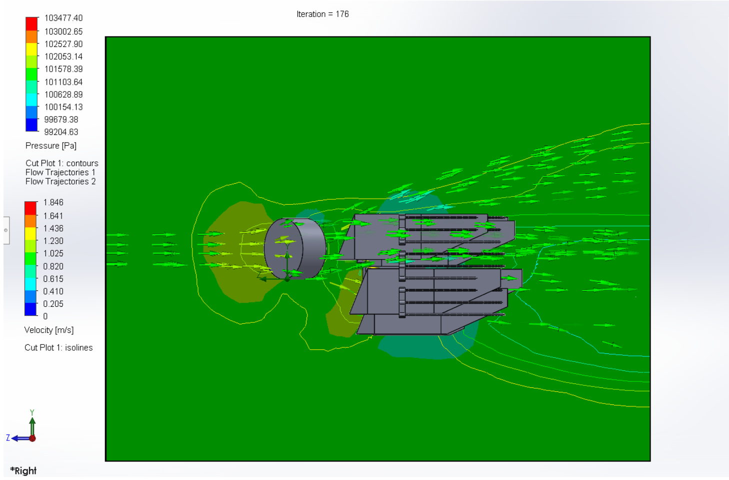


Figure 30: Flow of rear thrusters with full tall cage. Cutaway at single thruster.

This “dead spot” is an indicator of an area of low pressure, causing water to be pulled behind as the object moves through. This break in flow generates turbulence and increases the effect of drag. This indicates that a loaded cage

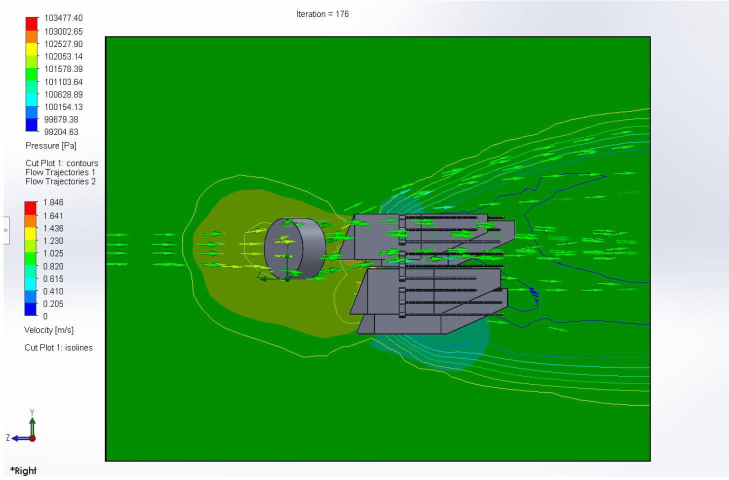


Figure 31: Flow of rear thrusters with full tall cage. Cutaway at thruster stream intersection.

would result in great impedance on the thrusters. In this scenario, the robot would only have its front thrusters providing effective forward motion, while the rear thrusters are rendered useless, wasting both battery life and time.

Given this information, as well as the estimated capacities of both containers, it is advisable to use the short cage design to avoid the problems of the tall one. The short design is still capable of holding more lionfish than a human-operated ZooKeeper container, and it has less impact on the mobility of the robot than the tall one. It is unlikely that the robot would find enough lionfish to make complete use of the tall cage anyway.

5 Future Work

With regard to the vision system there are a myriad of improvements that can be made. The chief problems at the moment are the slow frame rate and inconsistent distance readings. To address the first issue, the first recommendation would be to look to see if any optimization to the code can be made. The second option would be to spread the computation onto another machine, such as the currently unused Nvidia Jetson Nano from last year's project. It should be possible to have object detection run on the Nano and the stereo vision, which is more computationally expensive, run on the more powerful Jetson Xavier. For fixing the issue of inconsistent distance measurements, it would definitely be worth it to use a more advanced object detection system that is capable of drawing a bounding "box" around just the silhouette of the object. As it is now the bulk of the error for distances readings can be attributed to irrelevant disparity values. For objects that are vaguely rectangular such as a basketball or person standing straight with arms at the side, this problem is minor. However, if that person were to stretch out his or her arms the width of the bounding box would need to extend to the tips of the fingers. Now the majority of the pixels enclosed by the larger bounding box will belong to the background and not the person. This will lead to wildly inaccurate distance readings. Another way you can improve the distance measurements would be to use more advanced statistics. By looking at the distribution of disparity values for an object one could provide a confidence level for the returned distance value or produce a range of distances with a given confidence e.g. there is a 95% chance that a person is 2.3 – 2.8 m away from the robot.

Hardware improvements could also be made for the vision system. Currently, there is only one pair of cameras on the front of the robot. If pairs were to be added to the sides and back the robot would have a much greater chance of spotting a lionfish. With the development of a new mount it would also be possible to position the two cameras closer to each other. The advantage of this setup is that the stereo vision will have a shorter minimum range and wider field of view at shorter range. What would be sacrificed is the maximum range of measurable distance as well as a decrease in the sensitivity of the measurements. This may be a good system to have basic 360° depth perception around the robot, ensuring that it does not hit any large objects.

There is also the possibility of many additional software features for the stereo vision system. It would be possible for instance to determine the angle of an identified object relative to the robot using coordinate frame transformations and trigonometry. This information could be useful as a way to give the navigation system a heading. Another feature that could be added is the capability to estimate the distance of an object indirectly. One way of doing this would be to measure the width of the object in pixels when it is at a known distance and then measure it again when it is at another known distance. Using this information and along with some trigonometry it would be possible to determine the distance of the object based only on its current width in pixels. The problem with this approach is that it requires that the same side of the object is always facing the camera. Another approach is to estimate the distance by storing the average size of all the identifiable objects. Using the assumption that all objects identified have average measurements and by calculating its arc angle using its width in pixels along with trigonometry, an estimation can be made about how far away the object is. This method falls victim to the same issue as the previous one. If for instance we rely on the knowledge that the average lionfish is 12in long that means we not only need to view the lionfish from the side in order to see its full length, but we would also need a vision system with the ability to identify that we are indeed looking at the side of a lionfish. This would require a very well-trained neural network. The advantage in having these alternative methods for estimating distance is for the situation when the object is out of range of the stereo vision system or the readings are inconsistent due to noise.

References

Antaya, A., Peterson, E., Ralph, T., & Conroy, K. (2019). Lionfish Bot 2.0. : Worcester Polytechnic Institute.

Strickland, O., Whimpenny, C., Ewen, C., Abadjiev, M., Olgado, N., Saperstein, H., Chen, Q., & Johnson, N. (2020). Lionfish Harvesting Robot - Phase III. : Worcester Polytechnic Institute.

How Does Salt Water Rust Metals? (2018, May 5). Sciencing. `https://sciencing.com/salt-water-rust-metals-5150093.html`

Stereo-Vision. (2018). Github. `https://github.com/LearnTechWithUs/Stereo-Vision`

How Rust and Corrosion Work. (2020, January 22). Thoughtco. `https://www.thoughtco.com/how-rust-works-608461`

How to Write the Balanced Chemical Reaction for the Rusting of Iron. (2018, April 26). Sciencing. `https://sciencing.com/write-chemical-reaction-rusting-iron-8558862.html`

Structural Effects of Corrosion. (2016, August 3). Fprimec. `https://www.fprimec.com/structural-effects-corrosion/`

Scientists Create Underwater Internet. (2020, September 5). Sportdiver. https://www.sportdiver.com/scientists-create-underwater-internet?utmsource=internal\&utmmedium=email\&tp=i-1NGB-Et-QiI-16isP1-1c-15Ytj-1c-16ikur-l5541Azqch-KOpVG

Python OpenCV – Depth map from Stereo Images. (2020, June 23). Geeksforgeeks. https://www.geeksforgeeks.org/python-opencv-depth-map-from-stereo-images/

Depth Map from Stereo Images. (2013). https://Opencv-Python-Tutroals.Readthedocs. https://opencv-python-tutroals.readthedocs.io/en/latest/pytutorials/pycalib3d/pydepthmap/pydepthmap.html

Padierna, O. (2019, January 3). Tutorial: Stereo 3D reconstruction with OpenCV using an iPhone camera. Part III. Medium. https://medium.com/@omar.ps16/stereo-3d-reconstruction-with-opencv-using-an-iphone-camera-part-iii-95460d3eddf0

OpenCV and Depth Map on StereoPi tutorial — StereoPi - DIY stereoscopic camera based on Raspberry Pi. (2020). Stereopi. https://stereopi.com/blog/opencv-and-depth-map-stereopi-tutorial