Autonomous Lionfish Harvester

Joseph Lombardi William Godsey Brandon Kelly
Nikolay Uvarov Andrey Yuzvik

April 30, 2018

Advisors:
Professor Craig B. Putnam
Professor Kenneth A. Stafford
Professor Bradley A. Miller

A major qualifying project submitted to the faculty of Worcester Polytechnic Institute in partial fulfillment of the requirements of the degree Bachelor of Science

Abstract

The Lionfish Major Qualifying Project Team has developed a harvesting mechanism for the purpose of hunting lionfish, with the intent of attaching it to an autonomous submarine robot. The harvester functions as an independent mechanism capable of sensing lionfish, determining their location, and harvesting them. Thereby, when paired with an autonomous submersible robot with underwater exploration functionality, it acts as a sustainable solution for population control of the Caribbean's invasive lionfish. Performance of the system has been evaluated through testing the system's efficacy and robustness.

Contents

List of Figures

List of Tables

1 Introduction

Lionfish are a family of fish native to the Indian ocean and Polynesia. Unfortunately, due to some unintended events in the 1990's, lionfish have managed to populate a significant portion of the waters surrounding the Caribbean Islands [3]. Lionfish have spread across most of the Caribbean Sea, and have begun to spread northward, even reaching the shores of Rhode Island. They have no natural predators in the Atlantic or Caribbean, plenty of food sources, and a rapid rate of reproduction [19]. All of these traits make lionfish a formidable invasive species. Lionfish are indiscriminate predators, with a large stomach that expands up to thirty times its original size [19]. Even a single lionfish is capable of causing a sizable impact on its surrounding ecosystem, indicating that lionfish, with their rapid reproduction rate, pose a significant threat to the Caribbean biome. If action is not taken to control the situation, it may result in Caribbean reefs suffering catastrophic and permanent damage. Multiple species of indigenous fish to the region are already believed to be extinct due to this crisis, and more are expected to follow suit in upcoming years [19].

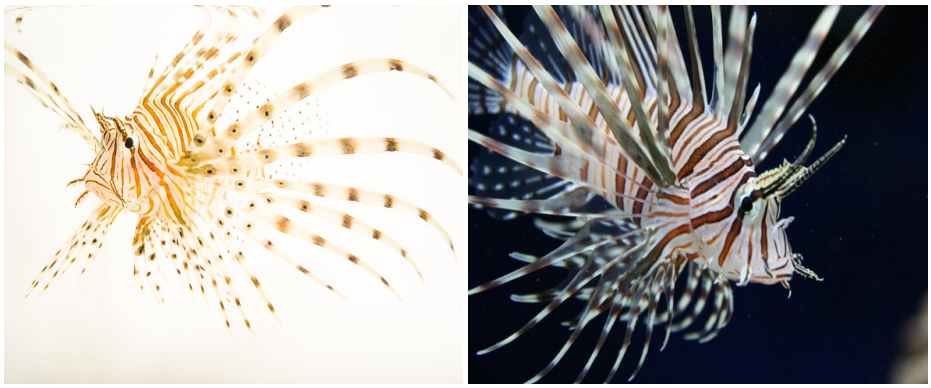
Attempts to eradicate the lionfish invaders to date have proven to be largely unsuccessful in many areas. Currently there are many organizations and groups putting their best effort towards fighting the invasion [19]. They work by raising awareness of the issue and encouraging divers and fishermen to focus on hunting lionfish. The lionfish is marketed for sport hunting, and as a newly popular food source. A few businesses even operate by large-scale harvesting of lionfish and selling them to local restaurants and stores. While these efforts have helped to curb the rapidly increasing population of lionfish, they have not been enough. More effective solutions are needed before the reefs and their inhabitants suffer irreversible damage.

The lionfish MQP team proposes a robotic solution to the problem of invasive lionfish around coral reefs. The ultimate goal for this project is to create an automated lionfish harvesting system that is capable of being integrated with a pre-built Autonomous Underwater vehicle, or AUV, for the purpose of hunting lionfish. The harvester will work in a standalone configuration. It will recognize lionfish with computer vision, be capable of harvesting multiple fish, and output the fish's location coordinates relative to the harvester, consisting of x, y, and depth data. By taking on this project, the team hopes to kick-start progress towards a sustainable lionfish solution.

2 Background

2.1 Lionfish

Pterois Volitans and Pterois Miles are two of the most commonly known members of the lionfish family. Both are highly popular in the pet trade, making them often found in home and public aquariums [29]. The two species grow to be over a foot in length, can weigh nearly 3 pounds, and as can be seen below in Figures 1a and 1b, they possess extravagant and venomous spiny fins for which their family is known. Their venomous spines and bright colorations serve as a warning to any would-be-predators, meaning that both P.Volitans and P.Miles have very few natural predators, even in their native Indo-Pacific region [41].



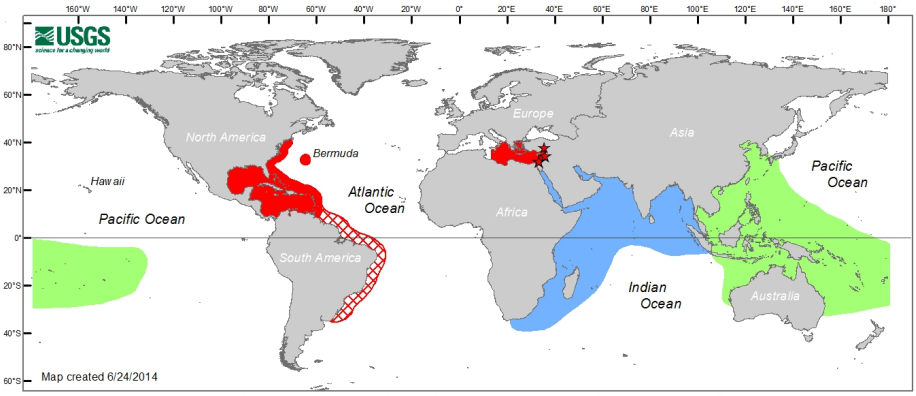
(a) Lionfish on white background [28] (b) Lionfish on black background [27]

Figure 1: Two images of lionfish

Lionfish typically prefer to inhabit coral reefs or rocky outcroppings, but will settle for almost any habitat that provides them with plenty of hunting opportunities and hideaways [31]. P.Volitans and P.Miles are ravenous predators meaning that they will prey on any fish or invertebrate that will fit into their mouths, including their own young, which they may produce a new brood of every few days [27].

2.1.1 Lionfish as an Invasive Species

Lionfish are both extremely well suited to ward off predators with their garish coloration and venomous spines and are still relatively new to the area, meaning that few of their possible predators in the region have adapted to prey on them [31]. In addition to having few to no natural predators within the region, the Indo-Pacific invaders have an abundance of food sources on the reefs of the Caribbean, thus devastating the populations of native reef fish and invertebrates [23]. Biodiversity within the region is under great pressure as the lionfish population and range, which can be seen in Figure 2, continues to grow, as many native species are being out-competed [23].



Map of native range of *Pterois volitans* (green) and *P. miles* (blue) adapted from Schultz (1986) and Randall (2005). Stars in Mediterranean Sea denote Lessepsian migration of *P. miles* via the Suez Canal (Golani and Sonin 1992; Bariche et al. 2013; Turan et al. 2014). Non-native range of *P. volitans* and *P. miles* in the Americas is shown in red (from Schofield et al. 2012). Predicted future distribution of lionfish along coastal South America is shown in red hatching (Morris and Whitfield 2009). Lionfish are continuing to expand westward in the Mediterranean Sea (red shading). See Azzurro et al. (2017) for a review.

Figure 2: Map detailing locations of lionfish [41].

2.1.2 Current Solutions

To address this rapidly growing problem, several organizations have taken steps to cull the lionfish population within the region [28]. The primary efforts at the moment consist mainly of divers using spear guns to hunt the lionfish en masse, which goes hand in hand with efforts to promote them as a delicacy, thus allowing divers and restaurants to profit off of saving the reefs [28]. Unfortunately, it is not currently feasible to entirely eradicate the lionfish population within the Caribbean through the work of divers alone [31]. The lionfish have begun to seek deeper, or more hidden refuges where the divers cannot follow, as a result of the recent culling efforts, subsequently requiring other methods to be investigated [31].

Logically if humans cannot hunt the lionfish well enough, then something else must take up this task. Efforts are underway to teach local predators, such as sharks, eels, and large groupers, to eat lionfish. There have been some small successes with particular individuals now hunting the invasive fish, but such behavior could take many years to spread amongst the region's predator populations [31].

Another solution is to have a robot hunt the lionfish, with several engineering firms attempting to tackle the problem with their own takes on a mechatronic predator. Unfortunately, at the moment, no company has managed to produce a design that meets the needs of the affected region in terms of both practicality, environmental safety, efficiency, and effectiveness [31].

2.2 Ethics

When it comes to discussing the ethical implications of the development of an autonomous entity with the purpose of hunting and harvesting lionfish the problem breaks down into two primary arguments: should the lionfish be removed from the region? If so, is the creation of an autonomous robot for the purpose of hunting and harvesting the lionfish an acceptable solution?

2.2.1 An Ethics Discussion of Killing Lionfish

The first question, being answered in a strictly consequentialist fashion, is easily answered with a yes, as the consequences of not eliminating the invasive lionfish from the region will drastically harm the native populations. In fact, several indigenous fish species are already believed to be extinct due to this crisis and others will follow suit if counter measures are not taken as soon as possible [19].

Protection of the indigenous species should be seen as a higher priority than the livelihood of an invasive, foreign population. Also the lionfish can then be used by the local people. Any harvested lionfish could be used as a potential food source, helping to feed the homeless, or for profit by restaurants and fishermen. The lionfish would not go to waste once killed, and would likely help local businesses offset the losses from the reduction in population of other fish species in the area.

2.2.2 An Ethics Discussion of a Fish-Killing Robot

After completing significant research on the issue, the group feels that the creation of an autonomous harvesting entity that hunts lionfish is a valid solution to the problem. Lionfish are extremely fast breeders that are capable of laying up to fifty thousand eggs every three days allowing them to spread throughout a region extraordinarily quickly [3]. Lionfish also possess venomous spines and bright colorations, which frighten many local predators and make them difficult and dangerous to be harvested by humans [39]. Some groups have attempted more creative solutions, such as training predators to hunt lionfish [30], but these methods have found little success, as many of the local predators do not recognize the invaders as prey. Given this information, it is unlikely that traditional methods of spear fishing, or natural predation will be viable solutions to controlling the lionfish populations, further highlighting the need for an effective large-scale solution.

Autonomous robots eliminate any risk for humans associated with hunting lionfish, and can be produced on a large scale. Many of the dangers of an autonomous harvesting robot can be mitigated with proper safety features, such as vision that only harvests lionfish, and will not shoot a human. Lionfish pose a significant environmental issue in the Caribbean that demands a solution. While an autonomous lionfish harvesting system does pose several ethics issues, it also offers a solution that reduces risk for humans involved, and may be able

to effectively manage the lionfish population enough to avoid an environmental catastrophe.

2.3 Harvesting the Lionfish

Herein the team details common methods for harvesting invasive lionfish. While there are many different implementations, the end result is effectively the same.

2.3.1 Pole Spear

Pole spears, like the one seen below in Figure 3 , are devices for underwater fishing that consist of a pole with a spear tip, and most frequently, a rubber loop on the other side of the pole to help the user avoid losing their grip on the pole spear once a fish is snagged. This implement is commonly used among divers, with the devices varying anywhere from four to ten feet long [22]. The tips of the spear vary depending on the type of intended harvest. An example of a detachable tip would be the "5 Prong Paralyzer" that the company Mako Spearguns markets for use against Lionfish. The tip is designed to grip the fish upon impact [22].

Figure 3: Example of pole spear [22]

2.3.2 Spear Gun

Spear guns are devices that fire bolt-like projectiles, or miniature pole spears, at fish. Such mechanisms can be either gas or elastic powered. Spearguns vary in sizes and power depending on the desired quarry and intended engagement distance.

2.3.3 Hawaiian Slings

A Hawaiian sling is a tube with an elastic band, and is reminiscent of an underwater crossbow. Hawaiian slings vary in sizes and power. There are several primary companies that make Hawaiian slings specifically for lionfish hunting. Those companies are "Lionfish Slayers", "Frappers", "AcuSpears", "SafeSpears" and "ELF" [21].

2.3.4 Containment Devices

Containment devices, or catch bags, come in a variety of shapes, materials, and sizes allowing for great flexibility in their use. Unfortunately, many of these devices lack an easy mechanism for the removal of venomous lionfish from the diver's spear. While these devices have proven to be effective containers, they still pose an unnecessary hazard to the user, as manually removing a lionfish from a spear tip places the diver at risk of being stung.

The "ZooKeeper" is a plastic tube that allows the wielder to push in lionfish and remove them from their spear without allowing the lionfish to escape due to a series of rubber teeth on the inside of the tube. As can be seen below in Figure 4, this device allows the user to safely deposit the venomous invaders inside, while also serving as an effective containment device.

Figure 4: Diver putting a lionfish into a Zookeeper containment device [21].

2.3.5 RSE Robot

Currently there is one robot that has been designed for the purpose of lionfish hunting. The nonprofit company Robots in Service of the Environment, or

RSE, made use of the crowd-funding website Kickstarter to raise funds for the development of an ROV to harvest lionfish [35]. The robot uses electricity to stun lionfish and a tube, much like the ZooKeeper, to draw them in, and keep lionfish contained within the robot [35]. The robot, which can be seen below in Figure 5, is tethered, has been proven to work, and the company has future plans to crowd-source the operation of the robot into a smart-phone game [35].

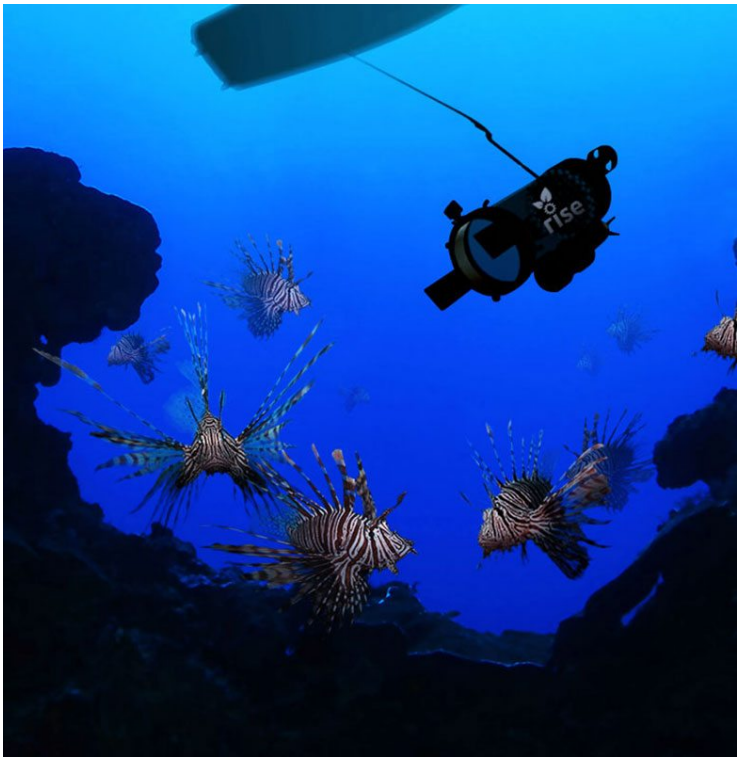


Figure 5: Concept art of RSE robot, note electrodes on front and tube-shaped body [35].

2.3.6 Lionfish Traps

Significant research and preliminary work has already been completed on traps for lionfish. Chief scientist for NOAA's Office of National Marine Sanctuaries, Steve Gittings, Ph.D, designed and published schematics for a passive lionfish trap, which can be seen in Figure 6 below [33]. The point of the trap is to act as an artificial reef, which lures nearby lionfish to make their home there. When the trap is pulled out of the water, two nets close in from the sides over the artificial reef, capturing any lionfish inside [33].

Figure 6: Trap closed with lionfish inside [33]

2.4 Robotic Harvester

The goal of the team's project is to create a harvester that will be combined with an underwater robot to hunt and harvest lionfish effectively and efficiently without endangering the environment or any surrounding divers. The design must be simple, effective, and adaptable enough to work on a variety of platforms.

2.4.1 Waterproofing

The main concern with any submerged electronics is effective waterproofing for a saltwater environment. Any leakage into the main electric hub or the various sensors of the system could be catastrophic. With the use of a professionally manufactured ROV kit the problem of waterproofing the main electric hub disappears. This leaves the issue of waterproofing any additional circuitry and sensors needed for the harvester, which can be accomplished in several different ways. The first possible solution would be to coat the electronics with a marine epoxy. This method can also be used to put wires through walls of a compartment, but at extreme depths, the pressure has potential to break any hidden weak spots within the epoxy-coating.

However, the harvester itself, and any additional sensors will inevitably require electrical components outside of the main hub. The team will have to ensure that these parts are well protected not just against water, but pressure as well. These parts must be able to withstand the pressure of four atmospheres, and function normally under these harsh conditions.

2.4.2 Battery Life and Autonomy

Due to legal restrictions and difficulties in acquiring permission for operating autonomous submarines near coral reefs, the robot may not be allowed to make use of a tether or power lines attached to a boat in some jurisdictions. Thus the system will have to operate using an on-board rechargeable battery. The amount of time that a system can spend underwater on one charge is an important factor. The harvester must be able to collect multiple lionfish per trip, and such a task requires the robot to traverse a reef in search of multiple targets. Movement will consume the bulk of the available power, meaning that the harvester will have to either share, or have its own power source.

2.4.3 Electrical to Mechanical Energy

Spear-based harvester designs would most likely spear the fish with a Hawaiian sling used in most other small fishing devices. Metal springs suffer from corrosion and eventually fail, while rubber is the mainstay for pole spears [22]. Regardless of the nature of the elastic part of the mechanism, the rearming of the harvester will be accomplished with an electric motor and transmission. After the mechanism is primed a simple release switch will launch the projectile forward releasing all of the stored elastic energy.

2.5 Environment of Underwater Robotics

The underwater environment poses many interesting problems for robots. In the following sections, some of those problems and their respective solutions will be discussed.

2.5.1 Buoyancy

The primary concern for submersibles that must maintain a near constant depth is the balance of vertical forces. Balance can be achieved through lateral movement with diving planes creating a hydro-motive force upwards, or possibly even a vertical thruster, but is most commonly done by reaching, or coming close to, neutral buoyancy.

Buoyancy is defined as the force that is exerted by a difference in pressure on an object, most commonly in water. A base whereby to start understanding buoyancy is in reviewing the Archimedes Principle: *"The buoyant force on a submerged object is equal to the weight of the fluid that is displaced by the object"*[4]. It is by balancing the force of the displaced water, and the object's force due to gravity, that neutral buoyancy is attained; net-zero force acting in the vertical plane [18].

2.5.2 Visibility

Visibility through water is subject to a higher rate of degradation due to the greater refraction of water as compared to air. As sunlight enters water, its

wavelengths are scattered and absorbed at varying depths [37]. Red wavelengths are greatly reduced at 5 meters, followed by orange at 10m, yellow at 20m, green at 30m, and blue at 60m, as shown in Figure 7. It is with such knowledge in mind that filters and artificial lights are employed to ensure adequate visibility underwater [13]. This becomes a concern when working with infrared sensors and cameras at various depths.

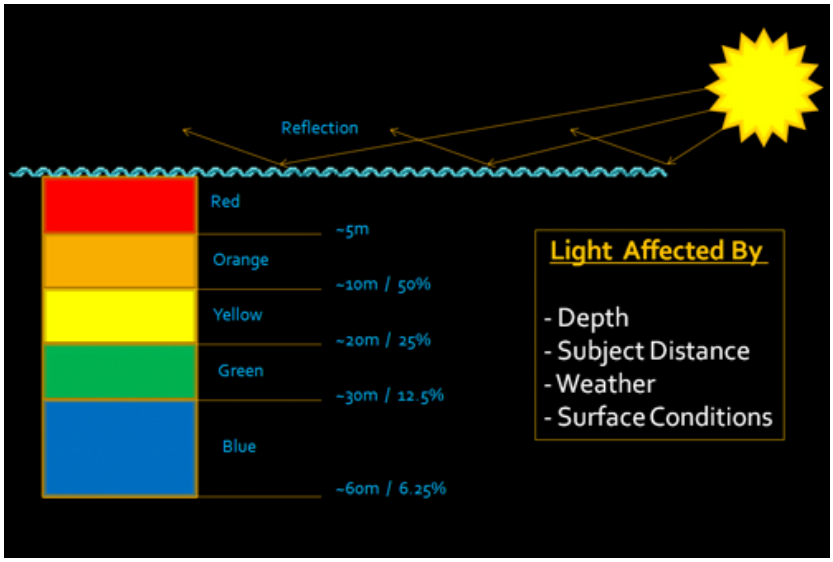


Figure 7: Diagram showing spectral drop as depth increases [13].

2.5.3 Wireless Signals

Sub-sea communication is a problem. Water is not conducive to the wavelengths on which many electromagnetic signals, such as optics, WiFi, and Bluetooth, operate. Such a conclusion is due to the high attenuation and scattering properties of water. For communication across any sizable distance, the choice is often to operate in the acoustic range [20].

Sound travels relatively well through water. Its propagation velocity is much greater in water as opposed to air, 1520 m/s versus 340 m/s. Still though, sound travels at a much slower velocity than its electromagnetic, or optical, competitors [20]. Lower frequency and travel rates translates, in communications-pertinent speak, to a lower bandwidth and greater latency. Basically, an acoustic signal will arrive later and convey less information. Despite these drawbacks, the acoustic signal is chosen most often due to its distinct advantage of actually being *able* to work underwater.

2.5.4 Pressure

Water, by its nature, is incompressible in its liquid form. That is, *ceteris paribus*, the volume of a body of water will not change significantly given changes in pressure or temperature. At sea level, the pressure amounts to one atmosphere, equal to 760 mm Hg. As an object, be it an animal, human, or robot, dives deeper into the depths of a body of water, the pressure exerted upon the object by the water increases in a linear fashion. The rate at which pressure increases in salt water is 1 atmosphere per 33 feet, or 1 atmosphere every 10.06 meters [30].

Generally speaking, there are two different methods of measuring pressure underwater: gauge pressure and absolute pressure [11]. Gauges and sensors are generally calibrated in the open-air environment of one atmosphere. Accordingly, a sensor calibrated on land will have an offset of an additional atmosphere for values underwater. The value read off a gauge or sensor directly will be the gauge pressure [11]. If one takes into account the additional offset of one atmosphere, the absolute pressure, that which the object is actually subjected to, can be ascertained. For practical uses, such as taking a reading of depth below the surface, the gauge pressure works perfectly fine. If one wishes to calculate a value from a mathematical formula though, the absolute pressure is necessary.

2.5.5 Leaks

The seepage of water into spaces where liquid is not desired can be a disaster for many submersibles. The main problem occurs when water creates an electrical short between contacts. The reason why water enters what one thinks is a closed space is due to a difference in pressure. If there is an imbalance, the liquid will try to flow into the cavity to correct the imbalance. As a dry-space continues a dive, water linearly increases its efforts to normalize the pressure within the cavity.

To prevent leaks the chamber must effectively act as a solid piece of material, while also remaining accessible in some fashion. Using O-rings to create a seal between a plug and its hole would be an excellent example. The same goes for filling an area with epoxy, or rubber pressure seals on cable penetrations. Both of them create a seal that requires extra force to break their bond.

2.6 Sensors for Robots Underwater

The job of a sensor is to translate a physical signal into an electrical one. Sensors are a type of transducer that reacts to outside stimuli [1]. Sensors can be designed to respond to changes in pressure, light, sound, vibration, electromagnetism, etc., and transform the signals into a form that computers can understand. In robotics, sensors are vital to the stability of control loops. A robot will sense its environment using the sensors installed, plan a course, then sense its environment again, and if necessary, adjust its course.

2.6.1 Leak Detectors

The role of a leak detector in an underwater submersible is to act as an alarm for the electronic components by signaling to the safety system if there has been a water breach. Often they are situated at the bottom of an enclosure as gravity would cause any water that has entered to flow to the sensor, making it more likely to detect the occurrence of a leak.

2.6.2 Pressure Sensor

Pressure sensors often take the form of protected strain gauges, a type of resistive measurement tool [14].

2.6.3 Inertial Measurement Unit

Inertial measurement units are sensors that measure angular and linear accelerations, typically using a multitude of accelerometers and gyroscopes [40]. Sensors such as these are extraordinarily useful for simultaneous localization and mapping underwater as they allow for the measurement of travel or rotation in any direction the robot could move.

2.6.4 Gyroscope

Gyroscope sensors measure the angular velocity of an object as it rotates making them quite useful for measuring rotations and keeping track of orientation for simultaneous localization and mapping purposes. These sensors are effectively included in inertial measurement units making a separate gyroscope sensor unnecessary.

2.6.5 Compass

A digital compass sensor, often featured in inertial measurement units, provides orientation information about how the sensor is oriented in relation to Earth's magnetic field. Such a sensor would allow the robot to always be aware of which direction it is facing [25].

2.6.6 Camera

Almost any camera can be used for computer vision purposes, but higher resolution cameras will offer more accurate results, as they can identify more key features in an image. The cameras also need an acceptable amount of lighting to acquire quality images, especially at lower depths.

RGB-D cameras, or often just known as depth cameras, could be useful for avoiding obstacles. These cameras are able to find the range, or depth, to any point of a captured image. They are quite expensive though, and normal cameras could be able to estimate the range of a target lionfish through simple computer vision programs.

The benefits of an RGB-D camera can also be offered by the use of multiple normal cameras as they could be set up to be able to discern depth through trigonometry and computer vision recognition, but would be much more complicated than a single camera method [10].

2.7 Controls and Computer Vision

2.7.1 Hunting Problem

The problem of hunting can be effectively broken down into three steps consisting of searching for the target, determining its location, and harvesting the target. The first step, searching for the target, requires the robot to navigate through a specified region which it attempts to map to a grid while looking for what it suspects to be the target. Next the target needs to be confirmed, which requires a more complicated vision test to identify that the target is correct. From the images, the confirmation and location of a target can be discerned for targeting purposes. If the target is confirmed, the location can then be given to navigation control systems which then initiate the navigation to and harvest of the target. This process then repeats until a given end condition, such as ammo limitations or a time limit [12].

This project's scope has been limited to the second and third steps of hunting, leaving all of the navigation actuation to the submersible vehicle. More specifically, the limited scope will require the team to implement functioning search patterns, target discrimination, targeting, and harvesting, but all of the navigation is expected to be handled by a platform that manages maneuvers and is capable of receiving commands from the harvester.

2.7.2 Control Systems

A robot with the combined complexity of being autonomous, submersible, and having a lionfish harvesting mechanism will need several complex control systems. The first system required will be a navigation system which will handle simultaneous localization and mapping, utilizing data from the vision systems, pathing, and automation of the robot. The second system required will be a vision system itself. It would periodically run a simple vision algorithm to determine if it thinks it has found a target and, if confirmed, run a more complex vision algorithm that determines the target and its position relative to the robot. Location data would be given to the navigation system. Finally, the robot will need a harvesting system that will handle firing and reloading the harvester, and tracking the remaining ammo. The code-logic flow chart below in Figure 8 describes how these control systems will interact with one another and the general logic flow of the robot.

The robot would start by being placed at the surface of the water at which point the robot would begin its fully autonomous functionality. The first task required is for the robot to try and record its location, most likely using a GPS, so it knows where it should attempt to navigate back to once resurfaced. The

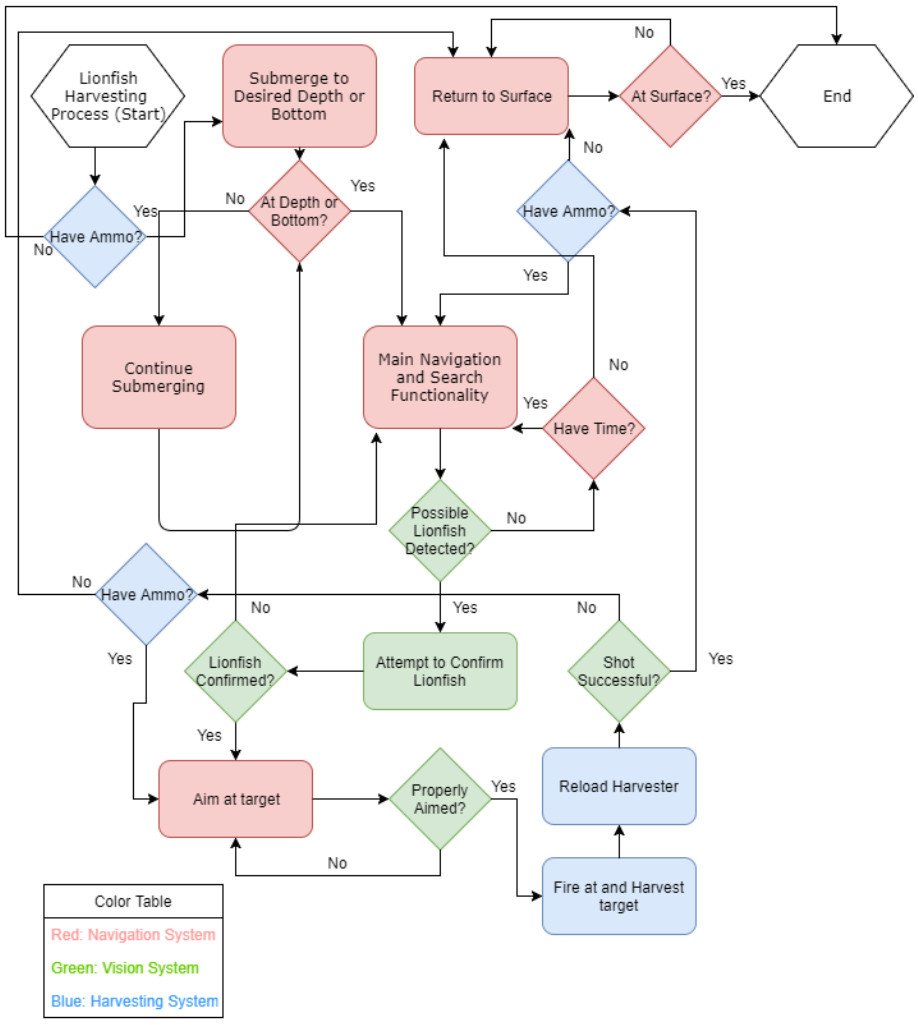


Figure 8: Lionfish hunter code-logic flowchart

next task required of the robot is to submerge. It would attempt to maneuver to a user-specified depth, most likely set to the most common depth at which lionfish are found, or until it reaches the sea floor, given that the waters are shallower than the optimal depth. The robot would then need to check its remaining mission length, as a preset time would be put in place to ensure the robot never stays submerged for too long. Another check would be for if it has ammo, since it would be unable to hunt without any harvester munitions remaining.

The next and most important function of the navigation system would be to have the robot navigate through its environment without colliding into any obstacles, all the while keeping track of its location, and constantly checking a simple vision program to decide whether or not it thinks there may be a lionfish in its field of vision. If the program does think it has found a lionfish it would then run a more exhaustive vision algorithm to confirm if the target is a lionfish. The algorithm would also project where the lionfish is located. If the target would be confirmed as a lionfish, the program would proceed to have the navigation system orient the robot to aim at the target. The loop would iterate until the vision system could confirm that the lionfish is oriented properly for the best shot and within the target range. If the target is not a lionfish, the program would return back to the navigation and search functionality.

Once the target is within harvesting range, the harvester is fired and promptly reloaded. The vision program checks to see if the shot was successful. Given that the shot was successful, the program checks if it has enough ammo to fire another shot and if so, the robot returns back to the navigation and search functionality. If the robot does not have any more ammo it would navigate back to the surface, attempting to surface as close to where the robot launched as possible. If the shot was unsuccessful then the program would check to see if it has any remaining ammo and if so, the robot would attempt to aim and fire another shot at the target. If the harvester does not have any more ammo, it surfaces. To ensure the safe return of the system, if the robot nears its battery limit, or encounters a problem, such as leaking or damage, while it is navigating and searching it will surface immediately. While this explains the entire programming logic of the robot, this project's scope is being limited to the creation of the harvester and targeting systems, with the navigation system, consisting of reef navigation and returning to the origin position, becoming a requirement of the submersible to which the harvesting mechanism is to be mounted.

2.7.3 Degrees of Freedom Underwater

Many underwater robots, such as the one that is intended to be the mount for the team's harvester, possess a total of six degrees of freedom. To develop common ground when discussing said degrees of freedom in a naval context, The Society of Naval Architects and Engineers standardized the following terms: Heave, sway, surge, pitch, roll, and yaw [38]. For the upcoming descriptions, a boat is assumed to have a coordinate system with x pointed towards the bow, y pointed starboard, and z down towards the seabed. In linear terms, heave

represents the z displacement, sway the y, and surge the x. In rotational terms, pitch is about the y axis, roll the x, and yaw the z [38].

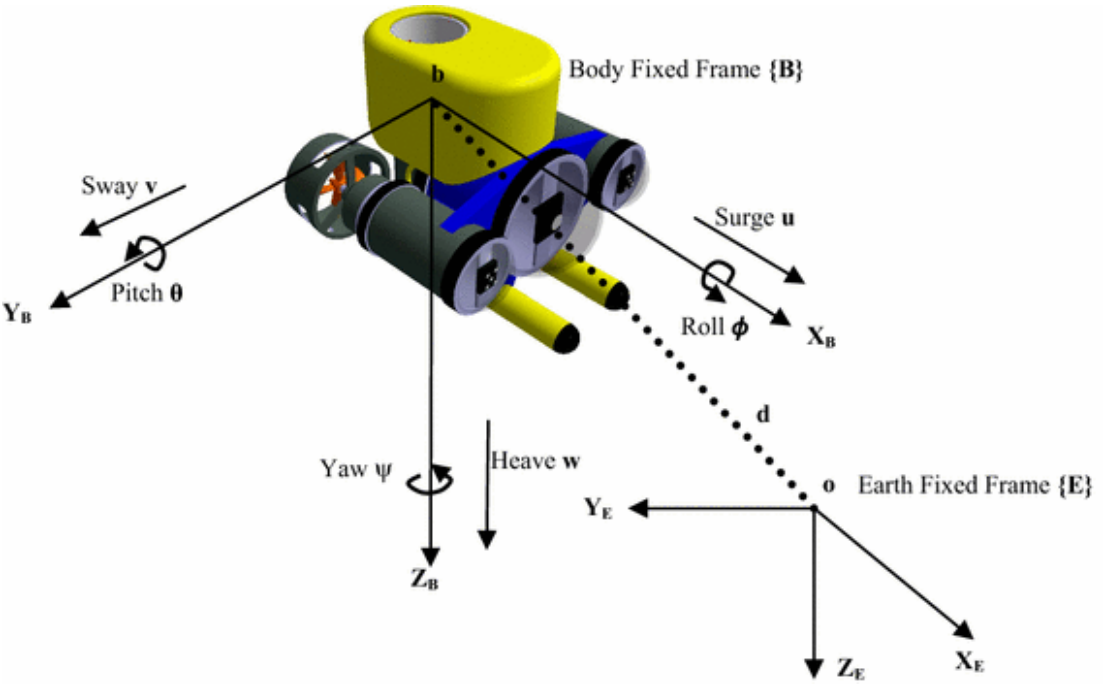


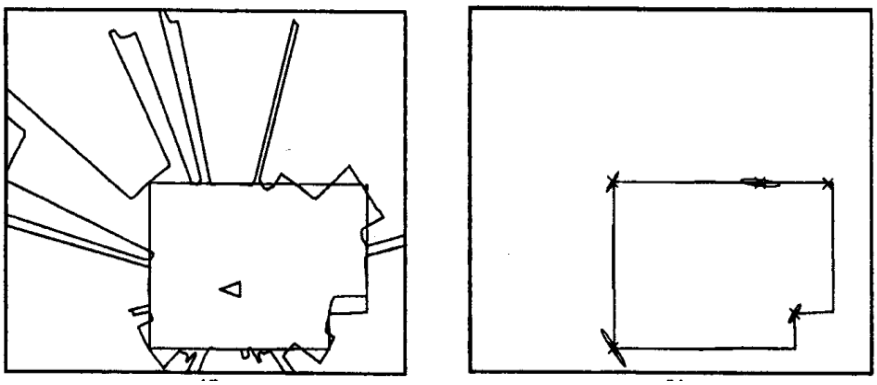
Figure 9: Degrees of freedom of a submersible robot [36].

2.7.4 Simultaneous Localization and Mapping

One huge problem in the study of robotics is having a robot sense its environment, make a map, move, and then repeat the cycle. This problem is called simultaneous localization and mapping, or SLAM [17], and is a crucial requirement of the submersible vehicle which the harvesting mechanism is intended to be attached to in order to successfully hunt lionfish. The value of SLAM is that it allows a robot to enter an entirely new space, map the environment to a data structure, and then have the ability to act in that environment with any number of algorithms.

The first step required in SLAM is for the robot to create a grid on which it assigns itself a base location and then is able to keep track of the locations of sensed obstacles and goals on said grid. For a robot to accomplish SLAM, it must be able to measure its movement within the system. Measurement allows the robot to estimate where it is located, and therefore the robot has a perspective of where it is in the data structure. The grid-data then can be used in navigation algorithms to direct the robot away from obstacles or towards a goal. All the while the robot would be repeating the SLAM process using the

initial grid as the robot navigates the terrain [17]. An example of the results of utilizing SLAM to map a room can be seen in Figure 10.



(a) Robot and sensor data shown (b) Data representation after filters

Figure 10: Example of SLAM sensing a room [17].

The problem becomes more complicated when working in an underwater environment, but there are many complex algorithms that are able to map and navigate through underwater environments. One such method is the use of a dynamically adaptive harmony search algorithm, an example of which can be seen below in Figure 11. The algorithm plans a path for underwater navigation by finding all likely poses to which the robot could travel. Then, the algorithm determines the lowest fitness value to be confirmed as the next location of which the robot is to travel. The process continues until no obstacles are in a given range. In such a case, the robot would travel straight towards its goal [36]. The robot can apply such an algorithm by choosing goals on its grid. The implementation would search within a pre-determined max range of exploration, and set goals in unexplored space. It would repeat the process until the entire region has been explored, the robot is out of ammo, the battery is almost out of power, or the robot has reached a mission time limit.

2.7.5 Theory of Computer Vision

The first step in working with computer vision is to break down how a computer "sees" an image. A digital image is effectively a series of color representations stored at different row and column values. Each of the data points represent a different color or shade, so when an image is displayed, all of the different values at their given row and column locations are shown. A program can use these values in order to look for certain objects in the image by checking though the data for certain patterns of colors.

An example of how this could be applied to searching for a lionfish would be by having the robot occasionally take an image, and then iterate through the image for a significantly sized cluster of colors commonly featured from known

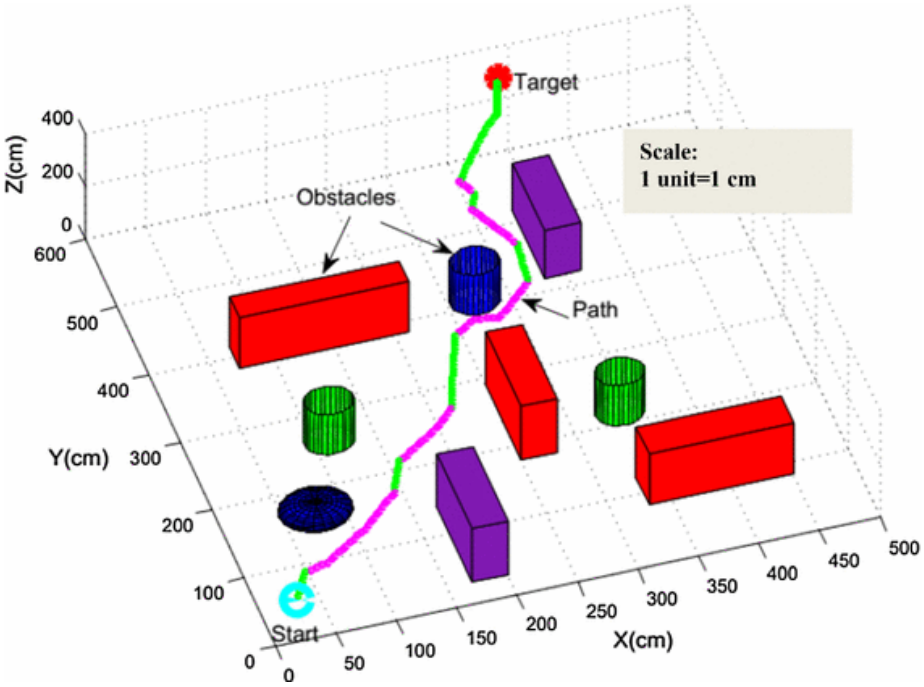


Figure 11: Path planning performed by a dynamic adaptive harmony search algorithm [36].

lionfish pictures. This method is fairly quick for a robot to process, but would not be able to say with absolute certainty that the identified target is in fact a lionfish, and not another object with similar coloration. This process can be adapted to check for shape, size, and other significant details, but the more complex the search, the more time it requires to complete. Computation time is a large concern when the robot needs to act on its results as fast as possible to successfully harvest lionfish.

To verify the target there is a more complex method that requires more time and memory. This method compares an image to other images of the target, and determines whether or not the target is in the first image. There are many implementations, such as key-point matching, histogram method, key-point plus decision tree method, a perceptual hash, and more, by which this type of computer vision algorithm can be implemented [26]. All have their own benefits and downsides.

Key point matching, being one of the most commonly implemented in a variety of methodologies, will most likely be included in the complex vision system implemented in the lionfish harvester. Key point matching effectively takes an image and finds important features based on a pattern of pixels that differs from its neighboring pixels. Then, it finds important features on the second image it is to be compared with, and proceeds to attempt to compare

key features between the two images. The preliminary comparison is to see if the images contain enough of the same features to say that they both possess the same object. Finally, based on the orientation of the features between the two pictures, key point matching can identify any movements of the object between the two images [26].

Computer vision would also be able to give the location of a target by analyzing the x and y coordinates of the object. The size of the target in the image can then be compared to the average size of similar targets in order to estimate how far away the target is from the robot. The algorithm would yield an inaccurate measurement if the target is of non-average size. More complex strategies of finding depth of objects using computer vision consist of using RGB-D cameras and stereo vision. RGB-D cameras are capable of providing depth data, but are quite a bit more expensive than normal cameras for similar quality, and have some problems operating underwater. Stereo vision, on the other hand, does not *require* expensive cameras, as it can be performed with any pair of normal cameras. It operates by comparing key features between two simultaneously captured pictures by the cameras, utilizing trigonometry to mathematically determine any individual feature's depth [10]. Computer vision can also be used to search for targets other than lionfish, such as reefs and other fish, for avoidance purposes.

2.8 Core Electronics

2.8.1 Raspberry Pi

The Raspberry Pi is a single-board computer. The most recent model is the Raspberry Pi 3 Model B. The small form factor of the Raspberry Pi has made it quite popular in the DIY community. The model 3B comes with an HDMI port, four USB ports, WiFi adapter, Ethernet adapter, and an SD card slot for memory, all on one board [34].

The reason to use one of these computers is for its huge computing power to size ratio. High-level computing can be completed by the chip, and then signals sent out through its integrated general purpose input/output pins. The Raspberry Pi can also run many different versions of operating systems; most of them are Linux based. Unfortunately, the Raspberry Pi is not an effective embedded solution, due to the thermal constraints of its CPU [34].

2.8.2 Arduino

Arduino is an open source electronics platform. The software and IDE support many embedded chips, and allow for 'ground-level' control of robotic systems. Arduino is also very popular in DIY and academic communities. Due to the memory and architecture constraints of the hardware that Arduino programs run on, high-level computing is near impossible at any sort of efficient speed [5].

Figure 12: Raspberry Pi Model 3 [34].

2.8.3 Ardusub

Ardusub is an open-source library for controlling undersea robots. The largest benefit of the Ardusub solution is that it has many control configurations and basic functions built-in. The library includes features such as data logging, depth holding, and inertial navigation based on an extended Kalman filter [6].

2.8.4 Brushless Motors

Brushless motors are the natural choice for use in an underwater situation. Outrunner brushless DC motors have the distinct advantage of not exposing any electrical contacts to the medium in which they are immersed. A properly constructed brushless outrunner will have well-insulated coils on the stator and a healthy tolerance for the permanent magnets around the rotor. As such, the normal operation of the motor switching the active electromagnetic pole will not be changed. However, there will be increased power draw due to the motor having to overcome the friction in water.

2.8.5 Electronic Speed Controller

An electronic speed controller, ESC, operates by translating a PWM signal to an output voltage. For brushless motor ESCs, the output is a three-phase alternating-current signal. An ESC often uses back electromotive force to sense the location, and thus correct electromagnetic firing order, of the rotor [39].

3 Methodology

The team decided to break down the problem of developing a lionfish harvester by limiting the myriad possibilities that could take form. The first limiting factor was choosing a robotic platform that could carry a payload. Early results are detailed in the section 'Robotic Submarine Platform' below. The next limiting factor was harvester design. The team decided that it would be most efficacious to approach the harvesting problem in the same way that divers currently do, detailed in the background section 'Harvesting the Lionfish' above. From these two limiting factors the team was able to design a harvester to meet the criteria listed in the requirements section and agreed to by the advising professors.

3.1 Robotic Submarine Platform

To objectively decide which robotic platform the team would pursue, the team used a weighted design decision matrix. Designs are placed along one axis and their traits are shown along the other. The traits are then given numerical weights in magnitude matching their importance to the team. The team iterated through the table and valued each design-criteria pair with a numerical judgment that can be either qualitatively or quantitatively supported. The last step was to make a summation of all the weighted criteria values for each design, and compare the total values. The decision matrix was applied to the following four submersibles, allowing the team to the platforms and determined the best solution:

1. BlueROV 2
2. Small build
3. Large build
4. OpenROV 2.8

3.1.1 Method

The team decided to use a scale of one through four, correlating to the number of options presented, as a way of valuing criteria for designs. The team's method uses a 'golf-score' style tracking system. In such a system one would be considered the best option, and four the worst. The golf-style scoring of criteria meant that the team would use a descending scale for the weight values. The initial numerical weighting took a form much akin to the criteria, having valuation ranging from the number of criteria themselves. The most important criteria has a weight of eight and the least important has a weight of one. The final score and acceptability of a design is determined by the lowest value, owing to the two facts above.

3.1.2 Weights

Price - Price was valued as the most important due to the static budget given to the team and the harvester's expected real-world use. The team was given a budget of $1,250 to work with at the start of the year. The difficulty of finding funding is thusly a huge factor in the success of this project. As well, considering the expected use of the team's harvester, a fisherman with a couple AUVs, the price factors in significantly to the real-world functionality of the robot.

Software Usability - Robotic systems are built upon layers and layers of abstractions. The team is going to be working at one of the highest levels, computer vision, to guide the robot to its prey. And yet, the team cannot forget the other levels at which the robotic platform functions, all the way down to the bits and bytes. The robot needs sensory data and a communication bus to communicate with the harvester. In commercial solutions, namely the BlueROV 2 and OpenROV 2.8 [7] [32], that problem is already solved and supported. If working on an entirely new system, a huge amount of time would have to be sunk into abstracting up to the level of computer vision, and this decided, for the team, the weight.

Payload Potential - Another major part of the platform is its ability to carry items underwater. A robot will need to maneuver with the harvester attached. The extra mass added to a vehicle could very literally mean, "sink or swim". Supposing that the harvester could be large and mechanically complex, the third highest weight went to payload potential.

Time Factor of Build - The time needed for the work is 4th most in importance. The reasoning for its weight is due to the thought of the team spreading itself too thin. The team supposed that it had two main, and rather large, goals; one for a harvester, and two for an autonomous lionfish hunting algorithm. The introduction of another goal, creating a vehicle, would introduce new uncertainties and time commitments.

Future Potential - The team's advisors made it clear to the team that the project would most likely turn out to be a multi-year undertaking. As such, the team desired to assure that they would not hinder future success with the project, earning future potential the 5th place.

Repair-ability - The expected areas of testing the robot, coupled with the possible chance for a total loss of electronics due to a leak, gave repair-ability the 6th most importance.

Mobility - After researching the nature of lionfish, and seeing the success of the RSE robot [35], the team decided that mobility would not be the most concerning feature of a potential design. Mobility is the 7th most important.

Initial Sensor Suite / Upgrade-ability - The initial sensor suite combined with upgrade-ability posed the least problems for the team.

3.1.3 Resultant Matrix

The resultant matrix can be seen below. The cells are color-coded from green, good, to red, bad.

		Criteria									
		Price	Software	Repairability	Mobility	Payload	Sensors	Future potential	Time		
Weight		8	7	3	2	6	1	4	5	**Score**	
										Sum	Weighted
Design	BlueROV 2	4	1	1	1	1	2	1	1	12	62
	OpenROV 2.8	2	2	3	2	4	4	4	2	23	92
	Build small	1	3	2	2	3	2	3	3	19	75
	Build large	3	4	4	1	2	1	2	4	21	90

Figure 13: Resultant graph of the team's robotic platform analysis

3.1.4 Discussion of Results

There are a few key points to look at in the matrix which might either be confusing, or add valuable information to the analysis. Those points and trends will be discussed in the following paragraphs.

Firstly, it is quite apparent that the BlueROV2 scores well in a majority of the areas. This is due to the team's method of benchmarking the designs against themselves. In many of the cases, the BlueROV2 was strictly better.

The mobility section has two ties. That fact may be a concerning feature, but the values were decided upon as such due to the method of weighting. Both the small build and the OpenROV 2.8 would have a 3-thruster setup. Meanwhile, the BlueROV2 and large build would have 6 thrusters, and a wider array of movement options. Those reasons backed the team's decision to grant the mobility section two ties.

The valuation for items may seem arbitrary, but there is a method to the madness. For example, the software usability section ranks BlueROV2 as the optimum choice. This is due to Blue Robotics providing a full software suite ready to be used when the robot comes out of the box. As well, the software is open-source, and documented. While the OpenROV 2.8 is open hardware, the software is less open to modification. The small build and large build are then based on complexity. While there are open source underwater robotics software platforms available, such as Ardusub [6], the team would have to research and implement it on a case-by-case basis.

Looking to the final weighted score it is interesting to see that the large build and OpenROV 2.8 are very close in value, and also the worst options. If the team was forced to decide between the two options, a more in-depth analysis would have to be conducted to find which would be the better design.

In conclusion, the team summed up the data and found the BlueROV2 to be the best option to move forwards with as the nominal platform for the

autonomous lionfish harvester.

3.2 Selected Platform: BlueROV2

In brief, the BlueROV2, as shown in Figure 14, is the flagship product of the company Blue Robotics, based in Torrance, California. The business started by developing thrusters and later branched out into ROVs with the creation of the first BlueROV model. The BlueROV2 is still highly supported by the company and has multiple planned upgrades in the future [7].

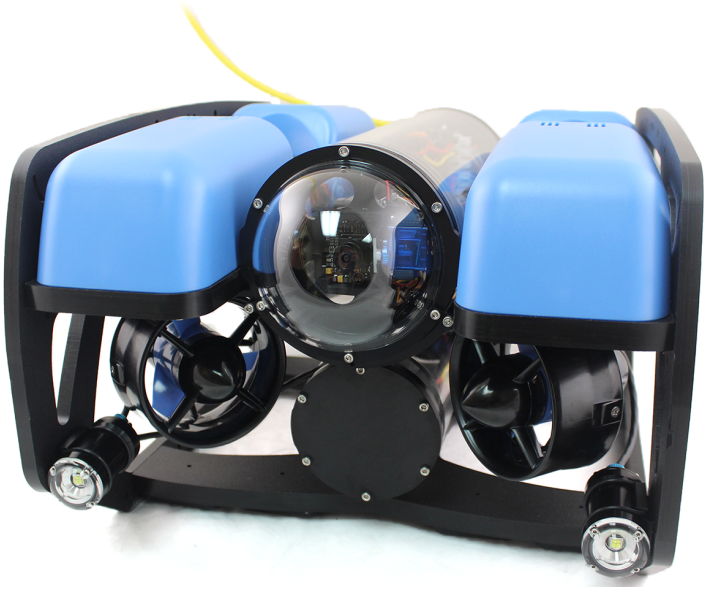


Figure 14: Front angle view of BlueROV2 [7].

3.2.1 Frame and Mechanical Interfaces

The frame, as shown in Figure 15, is made out of HDPE parts of varying thickness. It takes the form of a box with ends open bow and aft. Onto the frame, two cylindrical housings are attached. The cylinder mounted on the bottom of the frame houses the battery and the cylinder mounted on top houses the electronics. Located on either side of the electronics housing are three thrusters. At each corner of the top of the robot are blue shields, serving as housing for sub-sea foam [8]. Its physical dimensions once fully assembled are 457mm long, 338mm wide, and 254mm high[7].

The BlueROV2 has two primary mechanical interfaces. One is an optional skid that can be attached to the robot modularly. Blue Robotics designed the skid so that whatever the payload may be, it will fit its host vehicle and be removed easily. The second mechanical interface is included with the frame itself.

Figure 15: BlueROV2 partially assembled, note frame and battery housing [8].

There is an array of holes on the bottom of the robot that are normally used for attaching ballast, but provide an excellent means for attaching a mechanism directly to the bottom of the robot.

3.2.2 Ballast and Buoyancy

The BlueROV2 is neutrally buoyant in freshwater and positively buoyant in saltwater. It weighs 10kg in the air and 1.4kg in the water without ballast [7]. Adjustment of the buoyancy and trim can be completed by adjusting the amount and location of the robot's sub-sea foam under the fairings. The foam is R-3318 urethane. It is rated to 210 meters below the surface. The same goes for the robot's undercarriage, where six 200 gram weights are positioned for ballast. Figure 16 illustrates a possible configuration of the weights.

3.2.3 Thrusters

There are six thrusters on the BlueROV2 that provide locomotion. They are made in-house by the company. They are the newest Blue Robotics T200

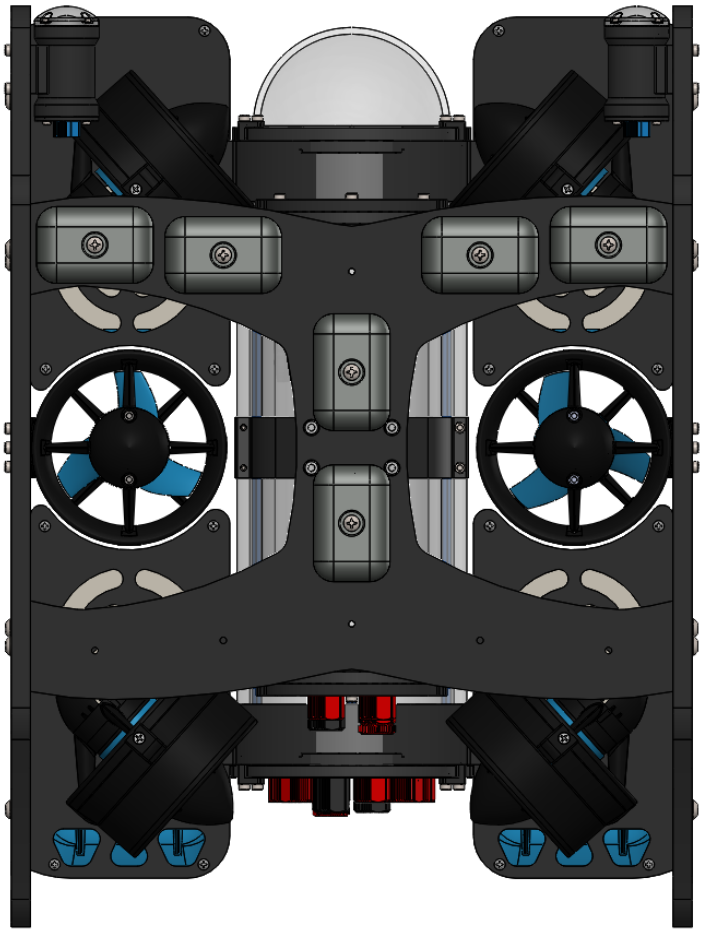


Figure 16: Undercarriage of BlueROV2 showing locations of ballast [8].

model. Brushless motors sit at the heart of the thruster and they are controlled with the company's 30A Basic ESCs.

The thrusters are configured in a vectored fashion, much akin to mecanum drive. Figure 17 illustrates how the six thrusters are angled. Four are used for forward and lateral motion, and two for vertical. They provide 14 kgf in both forward and lateral directions, and 9 kgf in the vertical direction. The robot has a maximum speed of 2 m/s, which equates to one knot.

3.2.4 Energy and Cable Interfaces

The BlueROV2 can operate for two to three hours on an 18Ah battery [7]. In the normal configuration there is only one battery and it is located within the battery enclosure.

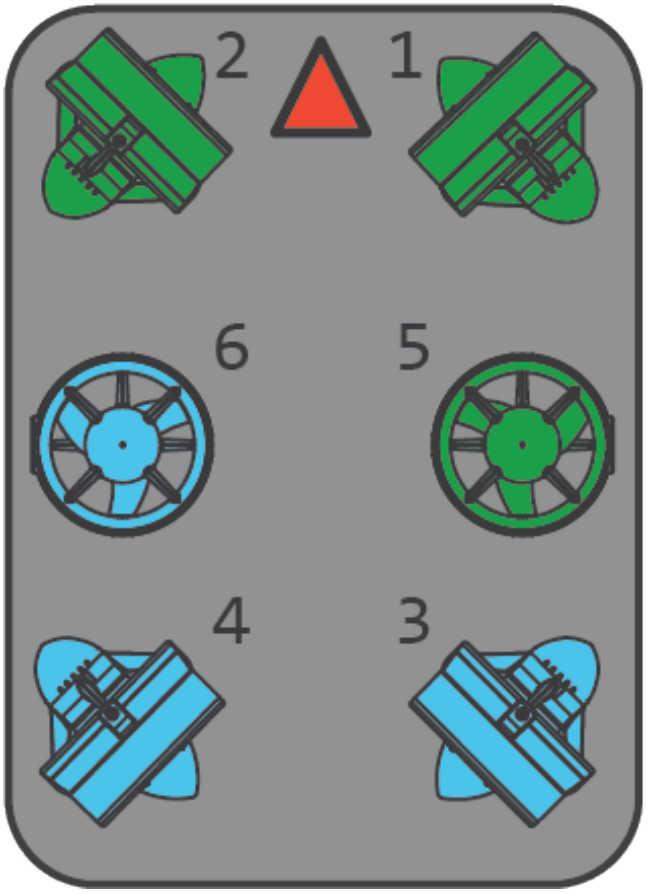


Figure 17: BlueROV2 vectored thrust. Green thrusters have CCW propellers, and blue CW [8].

Cables that transmit electrical signals and power penetrate the enclosures by way of Blue Robotics' aluminum hull penetrators. The cable is situated within an epoxy, surrounded by an anodized aluminum threaded bolt. The head of the bolt, which will face water pressure, has an O-ring that creates a seal between the surface of the enclosure and the penetrator. An illustration of this arrangement can be seen in Figure 18.

3.2.5 Electronics

The BlueROV2 is controlled by a Raspberry Pi 3. The computer is responsible for sensor data and communication. It delegates movement to a Pixhawk flight controller. The Pixhawk is responsible for PWM signal generation. Normal communication protocol is as follows: The topside computer sends a signal through the included Fathom-X brand tether. It is essentially an Ethernet ca-

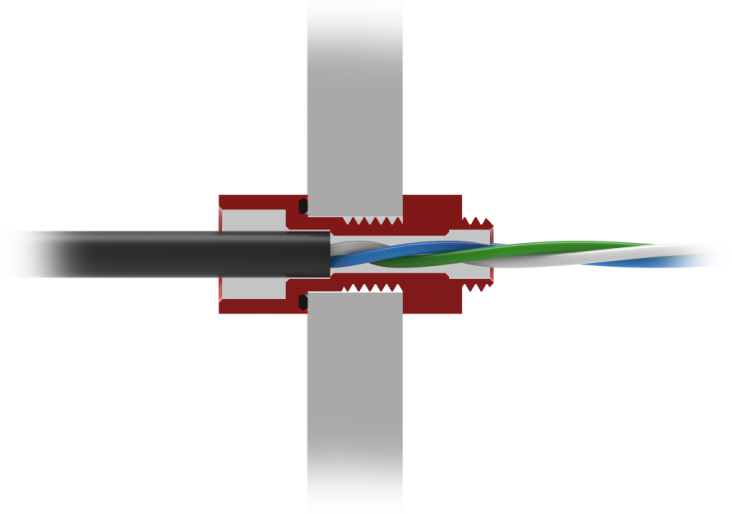


Figure 18: BlueROV2 cable penetrator [7].

ble that uses the IEEE-1901 standard for communications [7]. The tether ties into a Fathom-X signal conversion board, which outputs data through another Ethernet cord to the Raspberry Pi 3. The computer interprets the data, then initiates communication with the Pixhawk controller. The Pixhawk generates the aforementioned PWM signals, which in turn are received by the ESCs [8]. The BlueROV2's electronic system can be seen in Figure 19.

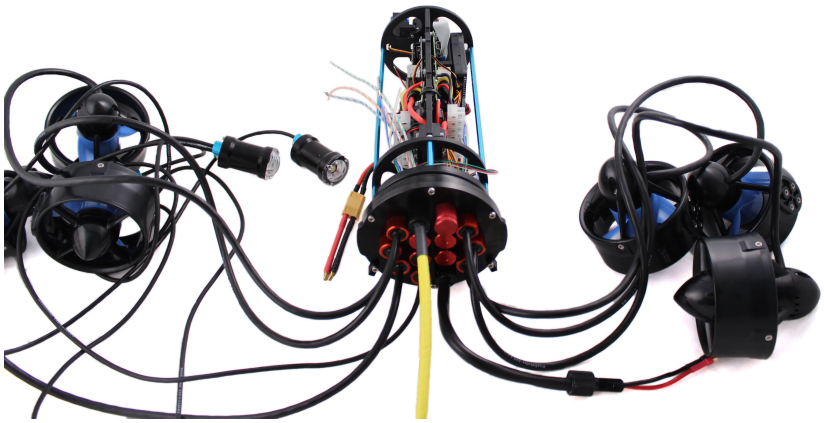


Figure 19: Electronics system of BlueROV2 [8].

3.2.6 Sensors

The sensors included on the platform are as follows. The first 3 items below are located internal to the Pixhawk. They allow the autopilot function of the Pixhawk to work for stabilization of the platform [8].

- 3-DOF Gyroscope
- 3-DOF Accelerometer
- 3-DOF Magnetometer
- 1080p Low-light Camera
- Internal Barometer
- Blue Robotics Bar30 Pressure, Depth, & Temperature Sensor [7]
- Current and Voltage Sensing
- Leak Detectors

3.2.7 Software Interfaces

The hardware on the BlueROV2 allows for multiple software interfaces. Specifically, by having all communication routed through a Raspberry Pi 3, the robot can run many different kinds of programs. The only limits are the processing power of the board, the skill of the programmer, and the sensors available.

3.3 Computer Vision

The computer vision component of this project consists of three major parts: recognition of a target lionfish, applying bounding boxes to the target lionfish, and determining the distance of the target lionfish from the harvester mechanism.

3.3.1 TensorFlow

The recognition requirement of this project is met through the use of TensorFlow, which is a large open-source machine learning library originally created by Google. TensorFlow allows for the running of a pre-trained neural network capable of recognizing a variety of commonplace objects, including lionfish. Running this recognition however takes quite a bit of time due to the sheer number of objects the model is trained to detect, so the team has retrained the final inception layer of the model on an assembled library of images compiled by the team in order to specifically detect lionfish, non-lionfish, and non-lionfish targets that when seen should prevent the harvester from firing. This allows for faster recognition of the desired targets, as well as a software safety system that would prevent the harvester from firing if a human or other targets of concern

are seen by the harvester. Once a lionfish target has been recognized to be in the harvester's field of view, a bounded box can be applied to the lionfish in the image. A bounding box gives the x and y coordinates of a given target in an image by a applying a box around the target with the coordinates of said box. These bounding boxes will be applied through TensorBox, which uses TensorFlow's libraries to perform image recognition on, and apply bounding boxes to the desired targets. The TensorBox program has been trained by the team to apply bounding boxes to lionfish and divers by feeding the program a library of images. In addition to images, this library contained the x and y data for two of the diagonal corners of the bounding boxes in each of the library's images. With the bounding box program running along with recognition, the computer vision program is able to determine the x and y coordinates of the target.

Figure 20: TensorFlow logo [15].

3.3.2 Depth, Detection, and Targeting

The prior functions allow the computer vision to recognize the location of a target lionfish in an image, but the distance to the target is still unknown. To determine the depth of the target, the team is using two stereo cameras, which allow for depth detection by comparing differences in two images captured by the two cameras simultaneously. The program then takes the x and y coordinates obtained from the bounding box step of the vision processing and finds the depth value of that set of x and y coordinates in the depth map. To perform this action the team is using OpenCV, which is an open source computer vision library with built-in stereo depth functionality.

3.3.3 Improving the Run-Time

These recognition and targeting processes are able to produce the coordinates of a desired lionfish target, but they are unable to achieve desired run times. This is because the robot needs to be able to reliably aim at a given target accurately and land shots on targets consistently. Given the slow nature of lionfish, a one second runtime should be sufficient to successfully aim at the

target. The reason for the slow run times currently is due to the limited processing capabilities of the Raspberry Pi 3 on which the computer vision processing is operating. The team found a solution to remedy this problem through acquiring a Movidius Neural Compute Stick from Intel, which is specifically for decreasing run times of computer vision programs that use neural networks, such as TensorFlow or Caffe.

The Movidius NCS, which can be seen in Figure 21, and its software development kit is capable of running on a Raspberry Pi, which thus allows the team to use it to run the TensorFlow functions being used for the recognition and bounding box functions at much faster speeds. The Movidius NCS however, is not meant to work with OpenCV libraries, so the team is investigating solutions to improve depth program run times.

The first solution is running the depth function only after the harvester has properly aligned its x and y coordinates with the target, reducing how often the depth must run.

The second option is to have the depth program only run on the section of the targeting region in the bounding box to improve run times, but requires more calculations to ensure the correct sections from each camera's captured image are the correct ones for finding depth at a desired location. Without these additional calculations, the program would be more prone to error as the sample size is decreased, while also still needing to be run frequently.

The third solution would be another open source library with similar stereo-functionality that works through TensorFlow and runs on the Movidius NCS.

The final option that was investigated is to have both of the cameras alternate between running the recognition and bounding box functions on the Movidius NCS and using the differences in the bounding boxes between the two camera's images and trigonometry to estimate depth. This, however, would require run times of at most 500 milliseconds per camera to keep vision functions from taking more than one second to produce coordinates of a target. The current recognition functions are able to provide run times of 250 milliseconds. Given some further vision program training, the running of the bounding boxes through the Movidius NCS, and the implementation of one of the listed solutions to the slow depth targeting run times, the vision program should be able to produce the desired result of at most a one second runtime for the entirety of the computer vision program.

3.3.4 Communications and Integration with the System

Once the program has determined the x, y, and depth coordinates of a lionfish, the program will then send this data over an Ethernet cable to an LCD readout which will interpret this data and display navigation commands to a diver who is manipulating the harvester. This is being done as the team will not have a submersible vehicle for the final demo of the project, but rather a diver who will act as the submersible. The commands are being sent over an Ethernet cable as this is how the BlueROV communicates navigation commands when tethered to a controller on the surface. These commands are also being

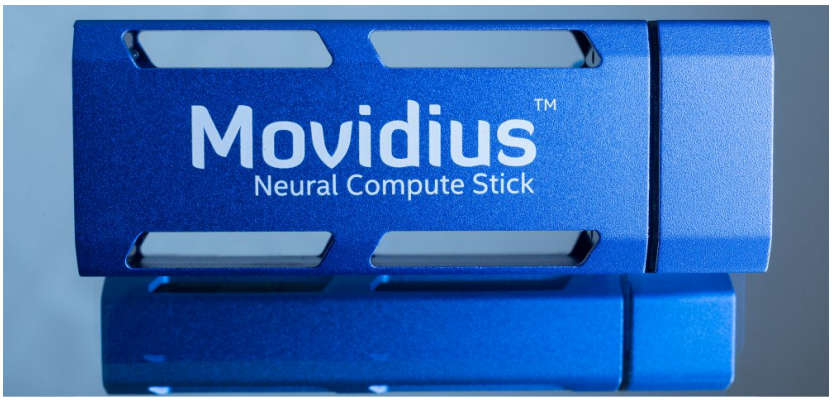


Figure 21: Movidius Neural Compute Stick [16].

sent in the same data type as the BlueROV would expect for manual control commands in order to act as BlueROV's default control program ArduSub. This will serve as a proof of concept that if the harvester were to be connected to a BlueROV, then it would be able to in theory cause the submersible to navigate to the target based on solely the computer vision as well as demonstrating that the computer vision targeting program is working as intended.

3.4 Lionfish Harvester

The overarching idea of the functionality of the lionfish harvester is detailed below. The harvester will be a mechanism that will attach to the payload skid of the BlueROV2. Power and communication lines will be fed to the harvester from the host's electronics compartments. When a lionfish is spotted the robotic harvester will feed location data to the host, and wait until the host robot has maneuvered into the proper harvesting orientation. An electro-mechanical trigger will fire, allowing a rod under tension by surgical tubing to travel towards the target. The rod will come in contact with a correctly oriented detachable spear-tip, and push it forward the rest of the distance into the target. The spear-tip has been designed to be positively buoyant, and will float the fish to surface for collection after the tip has detached from the rod. The trigger will reset into its natural state, and will act as a ratcheting mechanism as the rod is pulled back into the body of the gearbox, putting tension back into the system. Once the cycle is complete, a revolver on the front will index to the next spear tip and await firing.

3.4.1 Spear

The spear is simple in its design. There is a rack of one-way teeth on the bottom of a square rod. On the back end will be a groove or two to hold the surgical tubing for firing. On the front will be an interface that will slide into

the back of the buoyant spear-tips.

The team developed its first rod, and a picture is attached in Figure 22.

Figure 22: First attempt at making a spear that complies with design requirements.

3.4.2 Reloading Mechanism

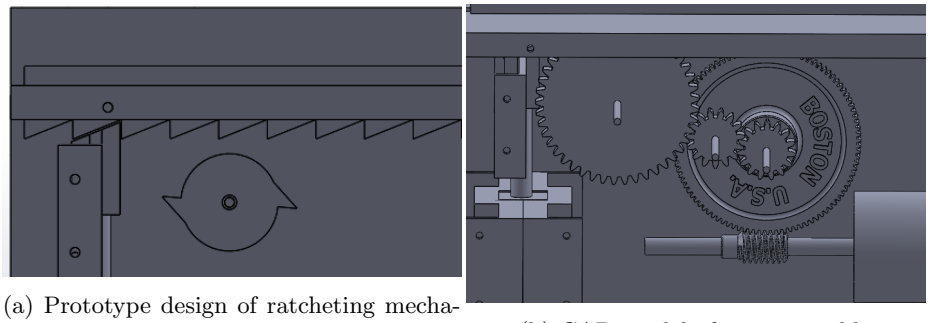
To reload the rod mentioned above the team will use a gear system that ratchets the spear into the firing position which can be seen in Figure 23a. A BlueROV electric motor will drive a transmission which in turn will drive a crank gear. The crank gear will load the rod back into the firing position tooth by tooth. Once the rod is loaded, the harvester will wait for a firing command. Once it has been received, a solenoid will release the ratcheting feature, giving the rod free movement. The rod will then shoot out delivering the spear tip to its target. The process then restarts and repeats until all of the spear tips have been used up.

To provide enough force to drive the rod back at a reasonable speed, the team implemented a transmission, see Figure 23b. The BlueROV M200 motor was chosen due to its availability and innate water resistance. The transmission consists of a worm gear, a worm and three regular gears. The resulting gear ratio is 1:300. The motor provides 0.5 Nm of stall torque and outputs 490 RPM/volt. To make sure the motor would be capable of pulling back the rod the team used a torque value of 0.4Nm. As a result, the maximum force that the gear system is able to apply to the rod equals to 2857N at approximately 30 RPM. With a rod length of 60 cm it would take the mechanism nearly 40 seconds to prime. When fully stretched the surgical tubing exerts 117.3 N of force, so the system is more than capable of fully loading the rod.

3.4.3 Revolver

To satisfy the requirement of harvesting multiple fish per dive the team has utilized the design of a Geneva mechanism. The robot will be capable of reloading up to eight times per dive. A Geneva mechanism was chosen due to the necessity of precisely rotating individual spear tips into the firing position. A front-on render can be seen in Figure 24.

One design consideration, since the spear tips are intended to be buoyant, is how to maintain equilibrium. To compensate for this factor, the Geneva mechanism will not rotate like a simple revolver, but will be programmed so that it loads spear tips in an indexed pattern. For example, in a four shot



(a) Prototype design of ratcheting mechanism

(b) CAD model of gear assembly

Figure 23: Reloading mechanism renders

revolver the pattern would be 4 indexes, 2 indexes, and 4 more indexes, or [4,2,4]. If expanded to 8 tips, two of the [4,2,4] sequences would be strung together with a [1] in between. Symmetry would be maintained during the first four shots, and then indexed to the next four. The pattern will insure that buoyancy is spread out symmetrically relative to the center of the robot after each shot, and prevent unnecessary keel.

To prevent the spear tips from falling out while the robot is maneuvering the spear tips will be held in place inside of the Geneva mechanism's drum by magnets. The magnetic system will prevent the accidental loss of any spear tips while still allowing the spear tips to be fired with minimal resistance.

The Geneva mechanism will be driven by a single electric motor that will rotate a crank which in turn incrementally rotates the drum with the spear tips. Once all spear tips are fired, the robot will rise to the surface to be reloaded and prepared for the next lionfish-hunting mission.

3.4.4 Buoyant Spear Tips

To successfully be able to hit the intended target as well as keep the spear tip embedded in the target, the team has investigated the existing designs for the spear tips currently in use around the world for hunting lionfish, and decided to use the standard 3-prong spear tips designed to fit and work with most of the standard spear fishing poles on the market. The spear tips selected for the purpose of this project needed to be modified to be capable of lifting the fish up to the surface. To accomplish this the team had to complete several tasks consisting of spear tip selection, material selection, buoyancy modification, and compatibility modifications.

There are several commonly used spear tips, with the primary spear tip types in use being the 5-prong spear tips and 3-prong spear tips. These spear tips are designed to hit the lionfish and limit its movement afterwards to keep the divers safe of venomous spines. The team's future robot may not need the safety that humans require, but it is important to prevent the fish from shaking

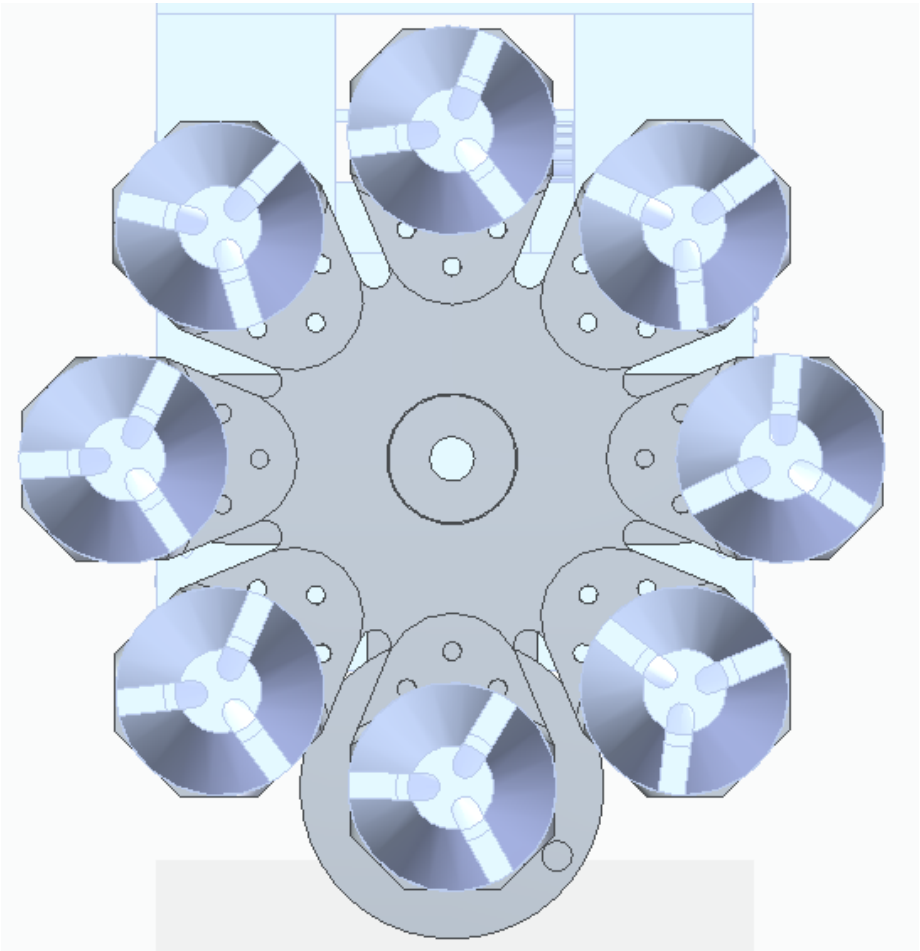
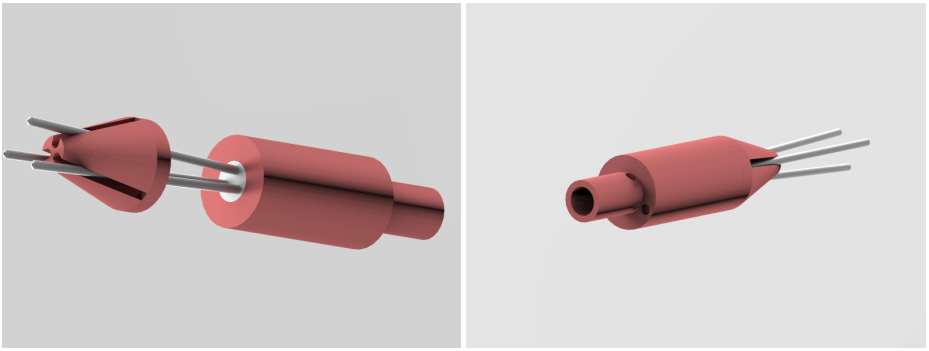


Figure 24: Render of prototype revolver

itself off the spear tip. The 5-prong spear tip unfortunately proved to be 1.5 times heavier, and thus the team chose to use the 3-prong spear tips as the base of the design.

The team chose PLA plastic as the material for the buoyant casings of the metal spear tips. The casing consists of 2 parts, the back and the front. Once attached to the spear tip they provide enough buoyant force to lift a fish to the surface. The exploded and the assembled views can be seen in the renders in Figure 25a and Figure 25b respectively. The front piece was designed to minimize drag while the back piece has a slot which the spear pole enters when being fired. The slot is necessary for the spear tip to stay in line with the spear pole while it is being fired.

The overall cost of such design is low relative to the other components. The



(a) Exploded view of buoyant spear tip. (b) Assembled buoyant spear tip

Figure 25: Spear-tip views

most expensive piece is the spear tip itself which costs on average around ten to twelve dollars depending on the shop. With PLA plastic being used as the material for the modification pieces, the overall price of the spear tip goes to about fifteen dollars per piece, which is the most economical solution the team was able to find thus far.

3.4.5 Electronics Design

The final product's success will hinge upon the the integration of the team's digital and mechanical systems, which is where electronics design comes into play. The system's processors must communicate and operate the various on-board mechanisms both with precision and in a timely fashion.

To facilitate this, the team has chosen to make use of several electronic speed controllers that will allow the Raspberry Pi 3, and the Arduino Nano to have control and use of the motors as needed. More specifically, the Raspberry Pi will be wired to the Arduino Nano, which will in turn be wired to two stepper motor drivers, and a Blue Robotics electronic speed controller. The two stepper motor drivers will be connected to the stepper motors that operate the buoyancy chamber, and the stepper motor that operates the Geneva mechanism.

To power the system, the team is making use of a battery pack. At the moment, the battery the team plans to use is a small Lithium Polymer battery rated for 1300 milliamp hours, which will feed into a small voltage converter, bringing the voltage input from 11 volts, to the 5 volts required by the Raspberry Pi.

The Raspberry Pi will also be connected via USB cables to the system's two HD cameras to facilitate the system's vision capabilities, and to the LCD readout which will give the diver operating the final product directional commands to find and harvest any lionfish that are detected.

One of the biggest factors for this system's electronic design was the need for waterproofing. To achieve this goal, the team must insulate the electronic

components and their wiring. Many of the components are contained within a sealed canister, with wires leading to the external components, i.e motors, sensors, and the LCD readout. The canister is made from PLA, and has been sealed with silicone and epoxy, and makes use of a bolt-on lid with an O-ring gasket to maintain a watertight seal. For the motors, the team has chosen to make use of stepper motors, and a motor commercially available from Blue Robotics, as they are inherently water-safe, reducing the complexity of the system as a whole from a design perspective.

3.5 Expected Budget

The team has been allotted $ 1,250 by the Robotics Department of Worcester Polytechnic Institute for the completion of the project. Table 1 below breaks down the expected expenses of the Lionfish MQP.

Table 1: Expected project budget

Sub-System	**Projected Cost**	**Description**
Computer Vision	$300	Two cameras, cables, and computer-on-board
Actuation	$200	3-4 brushless motors, motor controllers,
Sensing	$150	Assorted sensors for determining state of harvester
Structure	$400	Aluminum, screws, 3d printed plastic, etc.
Communications	$100	Out-going communication (cable, LEDs, screen)
Total	$1,150	Completed harvester

3.6 Team roles

The Lionfish MQP team is made up of five Robotics Engineering students. Each one brings their own specialization to the team, whether it be a focus on the mechanical, electrical, or programmatic function of a robotic system. Table 2 below identifies the roles of the members.

Table 2: Team roles

Member	**Major**	**Focus**
Andrey Yuzvik	RBE	Mechanics, Actuation
Brandon Kelly	RBE	Computer Vision, Communications
Joseph Lombardi	RBE	Electronics, Fabrication, Design, Organization
Nikolay Uvarov	RBE	Mechanics, Design, Outreach
William Godsey	RBE	Mechanics, Fabrication, Outreach

4 Requirements

This section will talk about the requirements and concerns for the harvester, that will be the key guiding points for the harvester-kit compatibility. The following requirements were considered to allow harvester to operate successfully at different depths and to be able to target and harvest lionfish of varying sizes.

Table 3: Project requirements

Requirement	**Value**	**Narrative description**
Maximum Depth	5 m	Testing depth of WPI pool
Neutral buoyancy	±1 N	Within host platform's ability to maintain buoyancy [7]
Target positioning	±15 cm	Half the size of a lionfish [28], high probability of harvest
Harvest distance	50 cm +/- 3	Distance where lionfish will not immediately swim away [28]
Operation time	3 hrs	2/3 of BlueROV2 maximum dive time [7]
Multi-harvest	4+	RSE robot harvests 4-6 [35]
Accuracy (CV)	>80%	Recognize lionfish in multiple orientations accurately
Communication	see desc.	Target location must be relayed to host/operator
Form Factor	BlueROV2	Attaches to BlueROV2, .5m x .4m (LxW), < 30cm over 1d[7]
Reliability	No failures	Test 4 complete harvest cycles a day for 7 days
Safety	see desc.	Mechanical safety lock and computer vision lock

4.1 Maximum Depth

For the harvester to be viable for the purpose at hand, and to successfully demonstrate its functionalities the minimum required depth of dive for the system will be 5m, which is the depth of the WPI pool, where the team will be performing tests. If the system is capable of operating at the set depth, adjustments could be made by future teams to make the system more durable.

4.2 Neutral Buoyancy

Since the harvester will be dependent on an outside vehicle, it must fit within the capabilities of a nominal underwater autonomous robot. In order to relieve the robot from additional work and balancing, the harvester design will account for buoyancy. The harvester will be made neutrally buoyant and the tolerance must be well within the robot's ability to maintain depth by thrust vectoring.

4.3 Target Positioning and Aiming

To successfully perform a harvest, the robot must determine the critical location of a lionfish within 15cm. This is due to the average size of a lionfish. If the team's harvester can determine the location within a sphere with a 15cm radius, the team is confident in a high chance of a harvest.

4.4 Harvest Distance

The distance of the harvest must be approximately 50cm (+/- 3cm). This is the suggested distance for the harvest since a lionfish will not react to the approaching robot at that distance. Most of the time they will swim away only when touched, therefore if the robot is within 50cm it will be able to successfully perform the harvest without scaring off the lionfish.Lionfish do not have natural predators in the region and therefore are not afraid of objects in their vicinity [28], thus 50cm is an appropriate distance from which to perform harvests.

4.5 Operation Time

The operational time of the BlueROV2 platform is 160 minutes [7], which limits the operational time of the harvester to that constraint. Due to the fact that the harvester does not require much power to function, the operation time can be easily achieved with a variety of batteries.

4.6 Multi-harvest

The goal for the harvester is to be able to complete a harvest of upwards of four lionfish. The repeated submerging and surfacing of the platform would consume energy and time. To compete with other harvesting methods in terms of efficiency, the platform must be able to harvest multiple lionfish in a single submergence. The more lionfish it is capable of harvesting per period of time, the more efficient the mechanism. The ultimate goal is to reach the efficiency of a diver, and then surpass it.

4.7 Target Discrimination

The goal for computer vision accuracy is to be able to recognize lionfish in multiple orientations. The robot must be able to distinguish between fish and rely not only on the color, but also on the overall unique features of lionfish in order to prevent by-catch. The vision will also be required to detect divers, which will prevent firing of the harvester.

4.8 Communication

The harvesting device will be required to communicate with a submersible on which it will be mounted, meaning that it will be required to have an externally accessible means of data transfer. The harvester will need to send coordinates, including depth, of a lionfish that has been detected by computer vision, to the submersible vehicle for navigation purposes, and inform the submersible if the harvester is out of ammo so the submersible knows to surface. The harvester will receive signals from the submersible telling it whether it is allowed to fire or not, as to prevent the harvester from firing in the wrong state. On the harvester itself, the system will be required to be capable of receiving data from

the dual mounted cameras, firing and reloading the harvester, and keeping track of remaining ammo.

4.9 Form Factor

The design of the harvester must fit the BlueROV2's frame, with an overhang of less than 30cm from the footprint of the robot. Failure to complete this requirement would yield an unwieldy system. The BlueROV2's footprint is .5m x .4m (LxW). The team suspects that due to the engagement distance for targets, there will be one protrusion outside the footprint, which led to the overhang caveat in one of the directions.

4.10 Reliability

The design must be reliable and durable. Due to the hostile nature of the work environment for this device, it must be structurally sound. As such, it will be tested for electrical and mechanical failure over a series of 7 consecutive days. Failure will be considered the inability to operate 4 complete cycles for one of the days. Testing will include at least 3 dry-fire days, and 4 in-water days.

4.11 Safety

Safety will be paramount with an automated hunter-killer robot. Moving parts that do not engage the target lionfish will be shielded from accidental human contact (such as in transport or mission readying). The team will include a mechanical safety that will prevent unintended firing, as well as a computer vision lock, which will disable the system when it detects even the slightest possibility of a human being present.

5 Results

During the last term of the project the team focused on assembly and testing of the prototypes the team had previously developed. To make sure that the overall system functioned as intended, the team tested each individual component in the lab, then tested each one underwater. After each system was confirmed to function as intended, the team synthesized the various subsystems. The team planned to test the entire system underwater, but did not accomplish this goal. The division of subsystems for testing purposes was: computer vision, control systems, LCD readout, spear mechanism, spear tips, Geneva mechanism, and buoyancy chamber. Pictured in Figure 26 is a render of the final design.

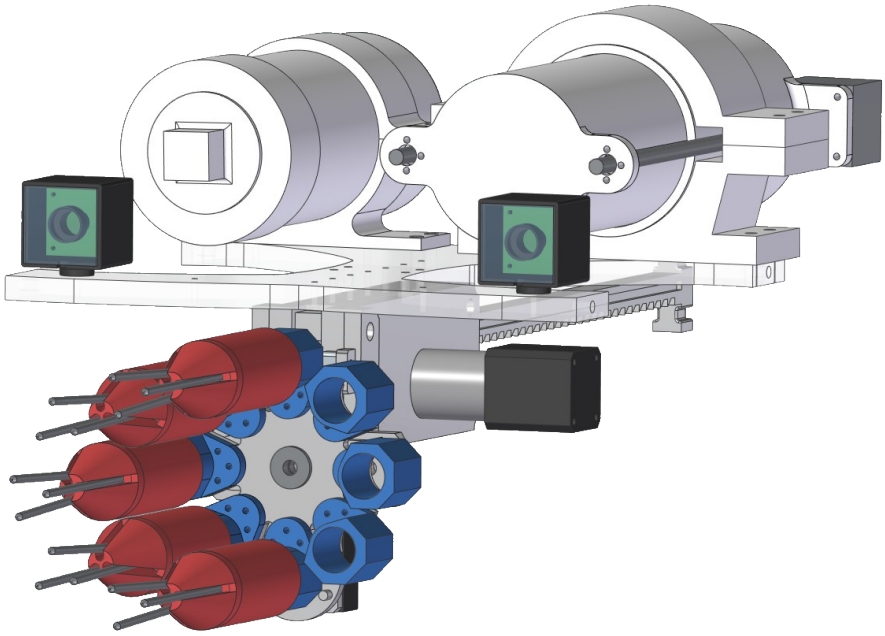


Figure 26: Render of final design

5.1 Computer Vision

The following sections will breakdown the individual key components of the team's computer vision implementation on the Raspberry Pi. Below is a flow chart in Figure 27 displaying how the computer vision system operates.

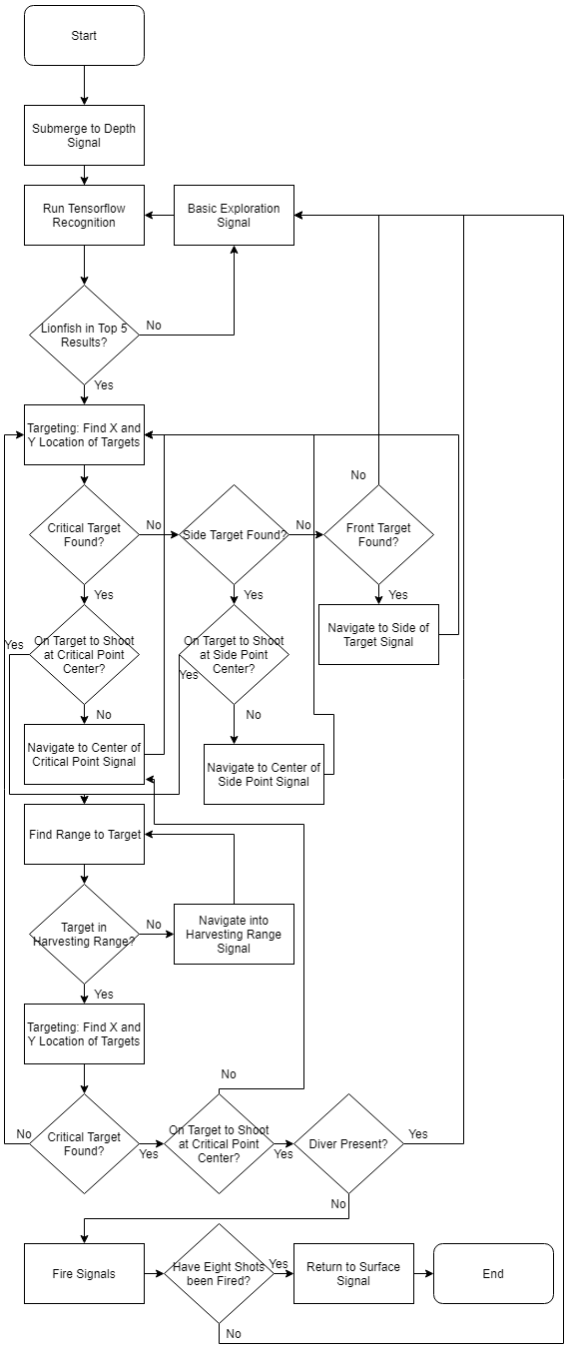


Figure 27: Computer vision code flow chart

5.1.1 Lionfish Recognition

When the computer vision program is started, the program begins by directing the diver or BlueROV2 to submerge. Once that is complete, the first computer vision operation, being the lionfish recognition, begins. The lionfish recognition functionality is performed using TensorFlow, a large open-source machine learning library. The team had retrained several models of TensorFlow's inceptionV3 final layer for specifically determining the likelihood that lionfish are in any given image. This was done by training it on two distinct categories of lionfish and non-lionfish. The team would then have the code proceed if the percent chance that there was a lionfish in the camera's sight was over a set percentage. This operation was taking far too long to reliably attempt to determine if a lionfish was present while navigating through a search grid despite the number of categories being reduced to two.

Due to this, the team had to find a way to speed up the running of the computer vision, which was accomplish by acquiring a Movidius Neural Compute stick, which was specifically designed to improve run times of neural network based vision programs, such as TensorFlow and Caffe. This improved the run times drastically to consistently produce results in around 100 milliseconds or often less on even TensorFlow's default inceptionV3 trained model which features hundreds more categories than the team's models.

Due to this the team decided to switch to using the default TensorFlow recognition model as it produced acceptable run times and was able to determine if a variety of other objects were present, which could be used in future versions of the lionfish harvester's computer vision operations. Due to this change, the team also began to have the computer vision code proceed not if the likelihood of a lionfish was above a given percentage, but rather if lionfish was in the top five things that are most likely present. This change was made as the percent chance that a lionfish was in the image was drastically reduced in the new model as it would also recognize a variety of other objects in the image interfering with the overall precision of the system. Below in Figure is 28 an example of the readout produced by the TensorFlow recognition program.

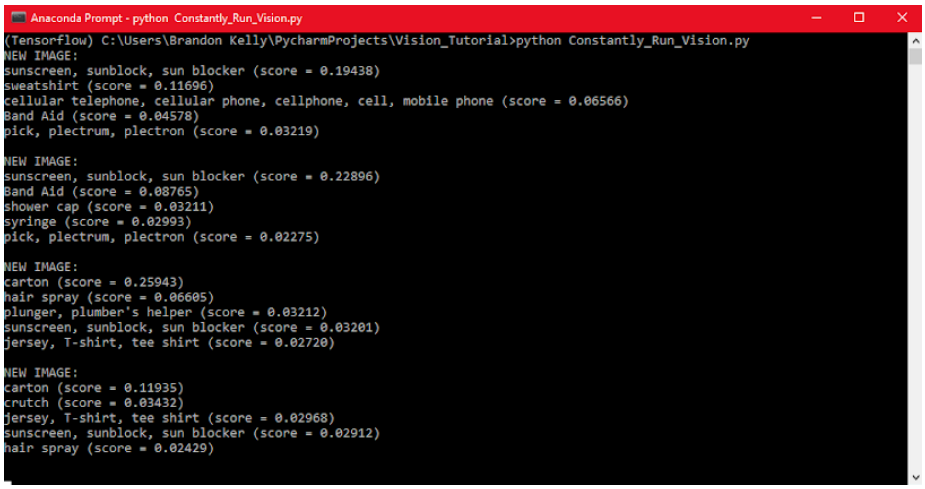


Figure 28: TensorFlow recognition program readout example

5.1.2 Targeting

Once the TensorFlow recognition determines that it believes there is a lionfish present in the field of view of the camera, the presence of the lionfish, along with its x and y coordinates, needs to be confirmed. This is done through the application of bounding boxes around targets in a captured image. The team's first attempt of doing this was through the use of TensorBox and later through TensorFlow's object detection library, but was changed to Caffe's MobileNet-SSD in the final iteration. This change was made due to the fact that implementing Caffe's MobileNet-SSD was similar to prior implementations of TensorFlow Mobilenets through its object detection libraries, except the Caffe version was supported in the Movidius Neural Compute Stick software development kit.

The team trained the bounding box program to target four different types of objects consisting of the front of a lionfish, the side of a lionfish, the critical targeting point on a lionfish, and divers. This training required the team to compile libraries consisting of hundreds of images of each of these categories and manually creating data files consisting of where bounding boxes should be applied to each of these images. Once the team's model was trained it could be run on the Movidius Neural Compute Stick to find the coordinates of any of the desired targets in an image. If the program found a critical target then it would prioritize navigating to the center of that target as that is where the lionfish should ideally be shot. If the program cannot find a critical point, but finds a side of a lionfish, then the program prioritizes navigating to the center of the side until it finds a critical target. If the program was unable to find a critical or a side point, but only the front of the lionfish, then the program attempts to have the robot navigate around the lionfish to the side. Finally if the program finds no part of a lionfish then either the target has left the vicinity or the

recognition has made a mistake, so if that occurs then the program returns to performing basic navigation and recognition.

The reason the program is able to target divers as well as lionfish is because if the program finds a diver then the robot will not fire, acting as a computer vision safety system. As a final note, since the camera performing the targeting is not aligned with the harvesting mechanism, but rather off to the side, thus it is important to know that the center of the camera is not where the robot attempts to get the center of the target it is trying to home in on, but is rather offset to the side of the camera's view. An example of the targeting program applying bounding boxes to an image of a lionfish can be found below in Figure 29.

Figure 29: Targeting program bounding box application example

5.1.3 Stereo Vision for Range

Lastly, once the targeting system has managed to properly align with where it would like to fire at the lionfish, the system needs to ensure it is within harvesting range, and if not then move into range. The team completed this task through implementation of two stereo cameras. Despite only one camera performing recognition and targeting, in order to find depth a second camera had to be included. Using the OpenCV libraries, the images from the two cameras

can be used to produce a depth map of the area in front of the harvester in which both cameras have visibility.

A depth map is effectively just an array of range values with a known height and width. This means that if an x and y coordinate is known, then it can be given to the depth map to find the range to whatever object is at that location. Since the depth maps produced have some areas where it could not accurately find a depth from the two images, those areas are given the value negative one. Should the range value at the desired x and y coordinates be negative one then that means the program must search around that coordinate for the closest known value to find the most accurate range value at that x and y location. The robot is told to navigate forward until within desired harvesting range.

Once it is within range the robot will refer back to the targeting to ensure that the target is still in the correct location and that there are no diver targets present. If the target is no longer in the desired area or a human is present, the program will then have to go back to either the targeting or search and recognition sections of the computer vision code respectively. If the target is in the correct x, y, and z coordinates to be harvested and no humans are present, then the mechanism fires and returns back to the search and recognition functionality. Currently the system is unable to verify if it successfully harvested the target. This occurs until the mechanism has fired eight times, after which it will tell the diver or robot to return to the surface and the executed code will be completed.

5.2 Control Systems and Electronics

The following sections detail the implementation, successes, and failures of the control systems and electronics the team created.

5.2.1 High-level Architecture

The team's high level architecture is a rather linear pipeline that starts at the Raspberry Pi. The Raspberry Pi is used for the vision, as mentioned above. The Raspberry Pi communicates with both the harvester and the LCD diver readout panel. Both the harvester and LCD readout are slave devices. They operate only on receipt of a command.

5.2.2 Harvester Control

The harvester's input is a digital I/O signal coming from the Raspberry Pi. This signal is for the final triggering of the mechanism. All other indexing and control is completed by the Arduino Nano which communicates with the motor drivers and relay breakout board. The code running on the Arduino Nano can be found in Appendix C. The harvester fires in the following sequence:

1. Trigger is primed, voltage applied to coil
2. Spear pole motor winds one tooth

3. Rod becomes free to move, trigger moves out of the way
4. Rod fires
5. Buoyancy chamber expands to account for loss
6. Voltage across trigger coil dropped
7. Rod is pulled back, trigger acts as ratcheting mechanism
8. Geneva mechanism indexes to next spear

5.2.3 LCD Diver Readout Panel

Since the team did not acquire the BlueROV2 platform the team created an LCD diver readout panel that displays navigation commands that would be sent to the host robot. The LCD readout displays navigational commands to a diver in the way of "<<<", ">>>" directional arrows to demonstrate in what direction the diver should rotate the robot to properly target and harvest a lionfish. The display is networking from the vision system to the receiving system through an Ethernet cable, the same way the harvester would be attached to the BlueROV2 platform. The data types are currently sent to the LCD readout, but would ultimately be sent to the BlueROV2 instead in the event of such integration. The data types can be seen below in Figure 30.

MANUAL_CONTROL (#69)

This message provides an API for manually controlling the vehicle using standard joystick axes nomencl

Field Name	Type
target	uint8_t
x	int16_t
y	int16_t
z	int16_t
r	int16_t
buttons	uint16_t

Figure 30: BlueROV2 manual control data type [24]

5.2.4 Underwater Motor Functionality

As mentioned in the background, the team decided to use otherwise unshielded motors underwater. Tests proved them to be functional as intended.

5.2.5 Electronics Chamber

All of the computational electronics are stored in a watertight chamber. This includes the Raspberry Pi, Arduino Nano, motor drivers, and relay breakout board. The pinout is featured below as a cross-reference between Figure 31 and Table 4.

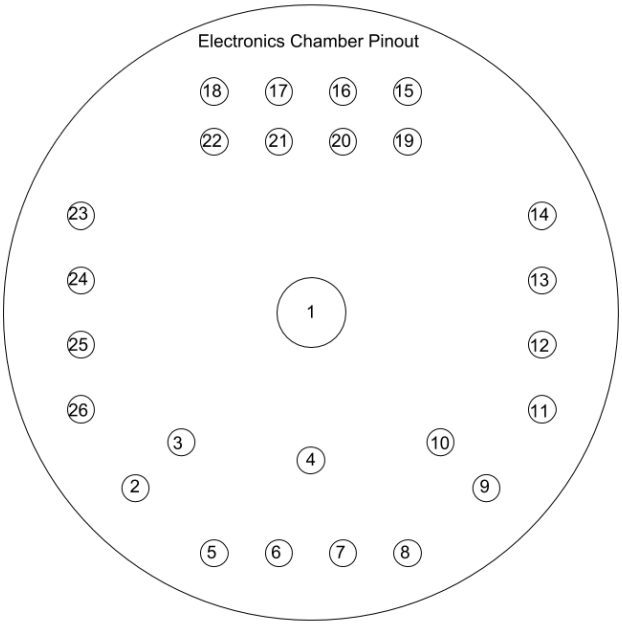


Figure 31: Diagrammatic pinout of electronics chamber

Table 4: Pinout cross-reference

Pin(s)	Function
1	Ethernet Communication
2-3	Trigger
4	Ground
5-8	Spear Reload
9	Photo-Gate Sensor Signal
10	+5 VDC
11-14	Geneva Mechanism
15-18	Right Camera
19-22	Left Camera
23-26	Buoyancy Chamber

5.3 Mechanical Design

5.3.1 Gearbox

During the production of the gearbox, the team encountered several issues with the prototype that was developed during the early stages of the project. The gearbox was overcomplicated and difficult to manufacture. As time and resources became scarce, the team decided to scrap that prototype and instead develop a far simpler system that would not make use of a worm gear.

The resulting gearbox consisted of a motor with a commercially available planetary gearbox with a 53 to 1 reduction ratio and a custom made gear for ratcheting the spear pole. A new housing box was also designed. The new box was far more compact and fit the inner mechanism better. This solution simplified the design and even though as a result it slowed down the reload time, it made the mechanism more reliable overall. Pictured below in Figure 32 is a render of the finalized gearbox. Note the 3-tooth gear in the middle, ratcheting mechanism towards the front, as well as the cylindrical interface at the tip of the rod.

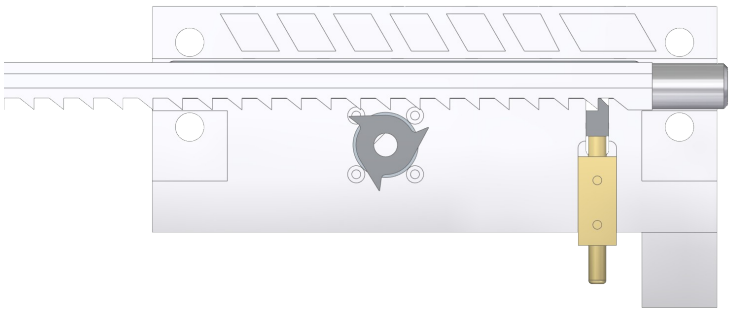


Figure 32: Side view of gearbox internals

5.3.2 Spear Pole

The team performed a calculation to determine whether the harvester would have enough velocity to pierce a fish. The results can be seen below.

Rod data:

$$Length = 80cm \tag{1}$$
$$Width = 16mm \tag{2}$$
$$Height = 16mm \tag{3}$$
$$Mass = .462Kg \tag{4}$$
$$Material = Aluminum \tag{5}$$

$$ToothHeight = 5mm \tag{6}$$
$$ToothLength = 10mm \tag{7}$$

$$k = 70\frac{N}{m} \tag{8}$$
$$x = 0.5m \tag{9}$$
$$m = 0.462Kg \tag{10}$$
$$PE = KE = \frac{1}{2}kx^2 = \frac{1}{2}mv^2 \tag{11}$$
$$v = \sqrt{\frac{kx^2}{m}} = \sqrt{\frac{70\frac{N}{m} * 0.5m^2}{0.462Kg}} = 6.1\frac{m}{s} \tag{12}$$

The final velocity of 6.1m/s being an ideal scenario, the team assumed some inefficiency, yielding the final velocity of 6m/s (inefficiency in the driving force). Averaged over the distance, this yields an average velocity of 3m/s. Physical tests in the WPI pool affirmed the 3m/s average value.

Using a video from National Geographic [28], the team looked at how fast the rod would have to travel to impale a lionfish. In the video, the sling takes 3 frames from release to impact. At 30fps this is a total time of 0.1s.

The rod travels half its length, and the full length as listed by Mako Spearguns [22] is 0.6m

Therefore, its average velocity is 3m/s, and the situation is quite comparable with the team's design due to the fact that the rod's travel distance in all the scenarios above is roughly equivalent.

In tests, the pole was able to penetrate a dead fish in water, proving the calculations and analysis above. The team is very confident that if put into the proper scenario, the team's spear would perform its job just as well as, or better than, conventional hand-powered Hawaiian slings.

5.3.3 Spear Tip Revolver

The Geneva mechanism proved to be an adequate solution to the problem of reloading a second spear-tip. The numerical design is featured below in Figure 33. The team used 5/8" square bar stock to manufacture the rod which passes through the Geneva mechanism's spear tip interfaces. To cut down on material waste and simplify design, the interfaces were designed with linear tolerances of 0.86mm, or a radial tolerance of 0.62mm. A combination of both offsets was expected to be in the final realization of the rod-interface synthesis. Exhaustive manual operation in addition to multiple powered tests proved the tolerances to function as intended. The team notes wear at the bottom of some of interfaces. This wear is due to improper assembly, and then operation. The team went back and reassembled the harvester after finding traces of debris on the rod's teeth.

The tolerance between the crank and the drive for the Geneva mechanism was intended to be 0.1mm, or roughly 5 thousandths of an inch. In practice,

the tolerance was a function of which way the drive motor was last pushed, due to the mounting plate holes being improperly fabricated. While no definitive measurements were taken on the side of direct contact, which stopped rotation of the mechanism entirely, the motor could be shifted enough to allow 15 thousandths (0.4mm) of distance between the drive and the crank of the revolver. Even at this extreme, the interface and rod performed as expected. Pictured is a diagram of the tolerances in Figure 33.

The indexing pattern was implemented (code can be seen in appendix C under the runGeneva function) and functioned as expected.

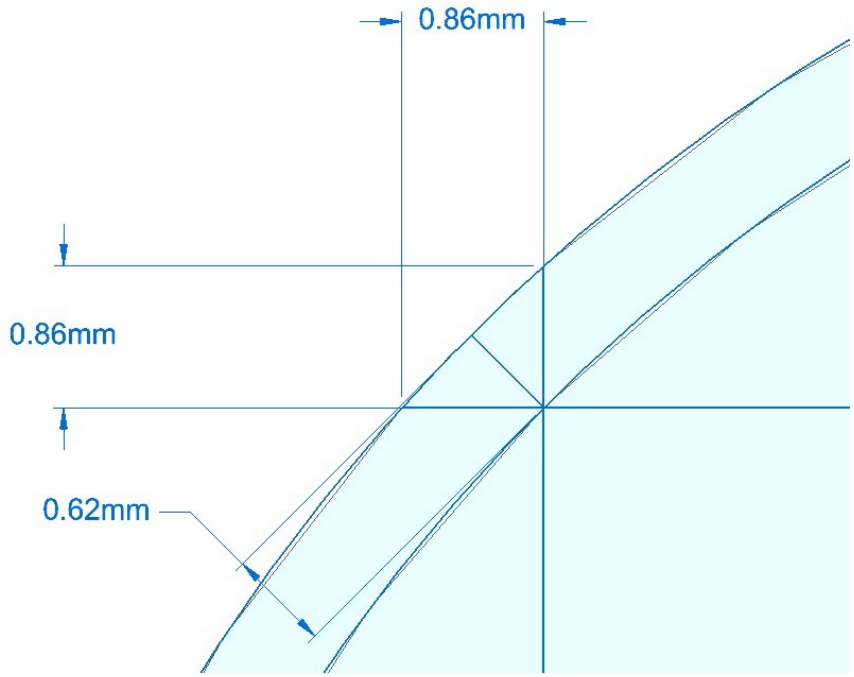


Figure 33: Dimensioning of rod/interface tolerances

5.3.4 Buoyant Spear Tips

After taking the design and material selection discussed in the methodology the team successfully created a batch of 8 spear tips that have been used for various testing in the system. The spear tips underwent buoyancy testing where they yielded 0.6N of buoyant force, which as expected and designed for, would deliver the fish to the surface. The testing was performed in the WPI pool,

although all testing involving actual fish analogs was done in other containers to avoid issues of contamination.

Next, the ability of the spear tip to pierce and stay lodged within a target was tested. For the purposes of this test the team purchased a dead fish similar in size to that of a common lionfish. The above mentioned fish was placed in a water tank and fired at from the prototype. The spear tip successfully pierced the fish without falling out unless an excess force was applied. While the test was a success there is still a possibility for improvement. Simply adding barbs to the spear tips can further reduce the chance of the spear tip slipping out.

Finally, the team tested the spear tip's ability to stay on the spear pole without sliding off during the shot. The initial test proved that the original simple solid cylinder design was not a viable solution. When fired underwater, during some of the shots the spear tip slid off the rod mid-shot. This was due to the water pressure in the spear tip's cylinder preventing the rod from fully entering it. This was remedied by making a new batch of spear tips with water vents. The vents at the base of the spear tip allowed the water to escape, reducing pressure and allowing the rod to fit all the way into the cylinder. During further testing the change was proven to be sufficient and no spear tip of the new design failed. The new design was proven to be a success.

The buoyancy calculations for the spear tips are as follows:

$$Density = \rho = 1\frac{g}{cm^3} \tag{13}$$

$$Volume = V = 146cc \tag{14}$$

$$g = 9.81\frac{m}{s^2} \tag{15}$$

$$Weight = .101Kg * g = 1N \tag{16}$$

$$BuoyantForce = g * \rho * V = 9.81\frac{m}{s^2} * .146Kg = 1.43N \tag{17}$$

$$Buoyancy = 1.43N - 1N = 0.43N \tag{18}$$

Experimental results led the team to the result of 0.6N of buoyancy, which is within reasonable deviation from the modeled result, especially considering the factors which may affect a spring scale (the device used to measure the buoyancy) underwater.

The drag calculations are as follows:

$$A = Area = \pi * 0.03^2 m \tag{19}$$

$$V = 3\frac{m}{s} \tag{20}$$

$$Cd = 0.3 \tag{21}$$

$$\rho = 998 \tag{22}$$

$$\tag{23}$$

$$Drag = 0.5AC\rho V^2 = (0.5 * 998 * 9 * 0.3 * (pi * 0.03^2) = 3.81N \tag{24}$$

The result led the team to ignore the rather small effect of the drag force as compared to the spring tension in the surgical tubing (70N).

5.4 Changes for Salt-Water Operation

The saltwater that the lionfish call home has posed a significant design challenge since the inception of this project. Salt water requires extensive electrical insulation to ensure the safe and reliable operation of the system, but also required the team to consider the corrosiveness of this environment. As a result special consideration was given to materials and maintenance routines that are suitable for this unique biome.

5.4.1 Materials

Aluminum was the primary material used for the system's mechanical components. The team chose aluminum as it was simultaneously lightweight, strong, and resistant to saltwater corrosion. This combination of properties allowed the team to create equally effective and durable components. The longevity of these components may be improved by future teams through the use of anodizing techniques that will further protect against scratching and possible corrosion.

Stainless steel was used for many of the other components on the system, particularly the mounting brackets and screws which held everything together. Stainless steel was chosen for its great strength and excellent resistance to corrosion and oxidation. For any future work the team would highly recommend the use of stainless steel for any components that require high strength and dimensional accuracy.

Low carbon steel was used by the team for several key components, namely the drive gear for the shooting mechanism and the trigger tooth. This material was chosen as an intermediary step, and not a final solution. The team needed to rapidly iterate to achieve a final design, and low carbon steel was the most easily available material that would be mechanically suitable. Unfortunately low carbon steel is highly susceptible to corrosion and oxidation, even when not submerged. The team would recommend that future teams replace these low carbon steel components with stainless steel versions.

Acrylic was chosen as the material for the platform on which all of the other components would be mounted onto. This choice was made given acrylic's relatively light weight, resistance to saltwater corrosion, and easiness to work with. In future iterations, teams may wish to choose a different material, as acrylic has proven to be somewhat prone to scratching and is only moderate in strength.

3D printed PLA plastic was chosen for the realization of several key components which fall into two primary categories. The first of these categories were the components which the team felt needed to be easily replaceable in addition to being saltwater resistant, which includes the "petals" of the Geneva mechanism, and the spear tip housings. The second of these categories would be components which the team was unable to create a second generation of, in

this case the buoyancy chamber. While PLA does meet the requirements for dimensional accuracy and saltwater resistance, it is somewhat porous, meaning our current buoyancy chamber has pinhole leaks, even with additional steps being taken to prevent such leaks. One key thing the team considered when using PLA was that it is biodegradable and thus should only be used for non-critical and easily replaceable parts unless further steps are taken to improve its lifespan.

One such method to improve both the lifespan and porosity of the PLA was the use of sealant materials, such as silicone, marine epoxy, or liquid rubbers. While these methods did prove effective for the team in preliminary testing, particularly in terms of waterproofing, silicone and liquid rubber cannot be recommended for use in more final iterations. These materials while waterproof and moderately resistant to corrosion are both susceptible to flaking and tearing when abused both physically and chemically.

Commercially available PVC piping was chosen for the electronics chamber as a result of its availability, strength, watertightness, and resistance to corrosion. The team made use of commercially available PVC welding compounds to seal the chamber's components, with a screw-on lid for ease of access. The threads of this lid were lined with Teflon tape to ensure a watertight seal. The team would recommend the continued use of PVC for such chambers as it has proven effective and easy to work with. The only caveat to this endorsement would be the longevity of Teflon tape in a saltwater environment, as with the team's limited testing abilities, the team is somewhat unsure of its lifespan in a marine environment.

5.4.2 Maintenance

As mentioned, the system will be operating in a very hostile and corrosive environment. There are procedures to prevent material corrosion and degradation. Each subsystem needs maintenance that will prevent it from breaking. The goal of this maintenance is to increase the life of the product. The details are listed below.

All the motors in the system (Geneva mechanism, gearbox, buoyancy chamber) must undergo rinsing with fresh water after the system is used in salt or chlorine water. It is *imperative* due to the fact that salt wreaks havoc on unshielded materials such as the aluminum housings of the stepper motors and the rotor's permanent magnets.

Several subsystems are covered with FlexSeal that prevents leakages. The subsystems mentioned are the buoyancy chamber, electronics chamber and cameras. After use, the sealed surfaces should be checked and more sealant applied to damaged surfaces.

The Geneva mechanism, rod, super gear and spear tips should be regularly checked for wear on critical points such as the spear tip interfaces, teeth of the rod, teeth of the gear and spear tips respectfully.

Since the platform is not designed to operate in high heat environments the platform should be stored in a room-temperature environment. This will

prevent damage to the FlexSeal and adhesives that are used throughout the harvester's components.

6 Conclusions

Through the duration of the project, the team has designed and assembled multiple prototypes of the desired system. The team has discovered multiple challenges which were not obvious during the planning stages. These challenges forced the team to iterate; change designs and re-manufacture parts. At the end of the process, the team is pleased with the result. The team successfully developed a prototype of a notional autonomous system that discovers, targets and harvests lionfish. The system has passed land-based testing and fulfilled almost all of the requirements for the project with the exception of operating in salt water.

References

[1] Agarwal, Anant. Foundations of Analog and Digital Electronic Circuits.Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2005, p. 43.

[2] Al-Amin, Hasanuzzaman, and N. B. Chowdhury, *"Design and Fabrication of a Submarine and Comparative Study with 6000m Driving Submersible Submarine"* International Journal of Novel Research in Engineering and Science, vol. 1, no. 1, 2014.

[3] A. Lowe, *"How lionfish spread against prevailing ocean currents"* Lionfish Hunting, 18-Mar-2017. [Online]. Available: https://lionfish.co/how-lionfish-spread/. Accessed: [9/23/2017].

[4] Archimedes, Heath, Thomas Little, Sir *The works of Archimedes*, ON FLOATING BODIES, 1897 pp.262-8 Available: https://archive.org/stream/worksofarchimede00arch#page/262/mode/2up [Accessed 10/2/2017]

[5] Arduino, *FAQ*, 2017, Available: https://www.arduino.cc/en/Main/FAQ [Accessed 10/2/2017]

[6] ArduSub, *ArduSub GitBook*, 2017, Avaiable: https://www.ardusub.com/ [Accessed 10/2/2017]

[7] Blue Robotics Incorporated, *BlueROV2*, BlueROV2 Specifications, 2017, Available: http://www.bluerobotics.com/store/rov/bluerov2/ [Accessed 10/2/2017]

[8] Blue Robotics Incorporated, *Assembly Instructions*, BlueROV2, 2017, Available: http://docs.bluerobotics.com/brov2/assembly/ [Accessed 10/11/2017]

[9] Boris Landoni, *Computer Vision with Raspberry Pi and the Camera Pi module*, Open Electronics, October 10, 2014, Available: https://www.open-electronics.org/computer-vision-with-raspberry-pi-and-the-camera-pi-module/ [Accessed 10/2/2017]

[10] Chris McCormick, *Stereo Vision Tutorial - Part I*, Nearist, January 10, 2014, Available: http://mccormickml.com/2014/01/10/stereo-vision-tutorial-part-i/ [Accessed 10/9/2017]

[11] C. Shilling, *The Underwater Handbook*. Springer US, Plenum Press, New York, 1976, pp. 85-86

[12] Daqi Zhu, Ruofan Lv, Xiang Cao, Simon X. Yang, *Multi-AUV Hunting Algorithm Based on Bio-inspired Neural Network in Unknown Environments*, International Journal of Advanced Robotic Systems, January 1, 2015, Available: http://journals.sagepub.com/doi/full/10.5772/61555 [Accessed 10/2/2017]

[13] Dive Photo Guide *Getting Started*, Basic Principles of Light Underwater, 2014 pp. Available: http://www.divephotoguide.com/getting-started-with-underwater-photography/basic-principles-light-underwater/ [Accessed 10/2/2017]

[14] eFunda, *Strain Gage: Materials*, N.d., Available: http://www.efunda.com/designstandards/sensors/strain_gages/strain_gage_selection_matl.cfm [Accessed 10/2/2017]

[15] Google LLC, *TensorFlow*, 2018, Available: https://www.tensorflow.org/ [Accessed 3/2/2018]

[16] Intel Corporation, *Order your NCS from one of our online partners*, 2018, Available: https://developer.movidius.com/buy [Accessed 3/2/2018]

[17] John J. Leonard, Hugh F. Durrant-Whyte,*Simultaneous Map Building and Localization for an Autonomous Mobile Robot*, IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS '9, Nov. 3-5, 1991, Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=174711 [Accessed 10/2/2017]

[18] Jonathan G. Fairman *National Aeronautics and Space Administration*, Buoyancy: Archimedes Principle, August 1996 Available: https://www.grc.nasa.gov/www/k-12/WindTunnel/Activities/buoyArchimedes.html [Accessed 10/2/2017]

[19] J. Wurzbacher *"The Lionfish Invasion"* Sailors for the Sea, Sep-2011. [Online]. Available: https://www.sailorsforthesea.org/programs/ocean-watch/lionfish-invasion. Accessed [9/23/2017].

[20] Lanbo Liu, Shengli Zhou, and Jun-Hong Cui, "Prospects and Problems of Wireless Communication for Underwater Sensor Networks," WILEY WCMC SPECIAL ISSUE ON UNDERWATER SENSOR NETWORKS, 2008. [Online]. Available: University of Connecticut, http://www.cse.uconn.edu/ jcui/UWSN_papers/wireless-overview_wcmc_2008.pdf [Accessed 10/2/2017]

[21] Lionfish Hunting, *Eradicate Lionfish Tool*, 2017, Available: https://www.lionfishhunting.com/lionfish-slings-c-14/elf-eradicate-lionfish-tool-p-50.html#.WdLYHmhSyUk [Accessed 10/2/2017]

[22] MAKO, *MAKO Spearguns and Spearfishing Gear*, 2017, Available: http://www.makospearguns.com/ [Accessed 10/2/2017]

[23] Mark A. Hixon, Stephanie J. Green, Mark A. Albins, John L. Akins, James A. Morris Jr., *Lionfish: a major marine invasion*, MARINE ECOLOGY PROGRESS SERIES, Vol. 558: 161–165, 2016, October 25 2016, Available: http://www.int-res.com/articles/theme/m558p161.pdf [Accessed 10/2/2017]

[24] *Sensors*, MAVLink, 2018, Available: http://mavlink.org/messages/common [Accessed 4/25/2018]

[25] *Sensors*, MaxCDN, 2017, Available: http://www.gsmarena.com/glossary.php3?term=sensors [Accessed 10/3/2017]

[26] M. Hassaballah, Aly Amin Abdelmgeid, and Hammam A. Alshazly, *Image Features Detection, Description and Matching*, Springer International Publishing, 2016, Available: http://www.springer.com/us/book/9783319288529 [Accessed 10/3/2017]

[27] Mindy Weisberger, *Scientific American*, Invasive Lionfish Arrive in the Mediterranean, June 28, 2016, Available: https://www.scientificamerican.com/article/invasive-lionfish-arrive-in-the-mediterranean/ [Accessed 10/2/2017]

[28] National Geographic, *Red Lionfish*, N.d., Available: http://www.nationalgeographic.com/animals/fish/r/red-lionfish/ [10/2/2017]

[29] National Ocean Service, *Why are lionfish a growing problem in the Atlantic Ocean?*, July 06, 2017, Available: https://oceanservice.noaa.gov/facts/lionfish.html [Accessed 10/2/2017]

[30] NOAA *How does pressure change with ocean depth?*, 2009-06-01 Available: https://oceanservice.noaa.gov/facts/pressure.html [Accessed 10/2/2017]

[31] Nsikan Akpan and Matt Ehrichs, *How do you stop invasive lionfish? Maybe with a robotic zapper*, August 24, 2016, Available: http://www.pbs.org/newshour/updates/robot-lionfish-invasive-species-rise-nekton/ [Accessed 10/2/2017]

[32] OpenROV CA., *OpenROV2.8*, OpenROV 2.8 Kit, Available: https://www.openrov.com/products/openrov28/ [Accessed 10/2/2017]

[33] Oren Lieber-Kotz, *NOAA and partners release new trap designs to corral invasive lionfish in deep water*, NOAA Sanctuaries, March 2017, Available: https://sanctuaries.noaa.gov/news/feb17/sanctuary-scientist-fights-invasive-lionfish.html [Accessed 10/7/2017]

[34] RASPBERRY PI FOUNDATION, *RASPBERRY PI 3 MODEL B*, N.d., Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/ [Accessed 10/2/2017]

[35] RSE, *The Lionfish Project*, Robots in Service of the Environment, 2017, Available: https://robotsise.com/lionfish-project/# [Accessed 10/7/2017]

[36] Shubhasri Kundu and Dayal R. Parhi, *Navigation of underwater robot based on dynamically adaptive harmony search algorithm*, Springer International Publishing, April 26, 2016, Available:

https://link.springer.com/article/10.1007/s12293-016-0179-0 [Accessed 10/3/2017]

[37] S. M. Luria and Jo Ann S. Kinely, *"Linear polarizing filters and underwater vision"* Undersea Biomedical Research, Vol. 1, no. 4, p. 371+, December 1974. [Online]. Available: The Rubicon Foundation, http://archive.rubicon-foundation.org/xmlui/bitstream/handle/123456789/2668/4469103.pdf?sequence=1 [Accessed 10/2/2017]

[38] Society of Naval Architects and Marine Engineers (SNAME), *"Principles of Naval Architecture"*, 1989, Vol. III, Pg.41, Section 3 - Ship Responses to Regular Waves

[39] S. Verkoetter, "An Electronic Speed Control Primer" Sailplane & Electric Modeler Magazine, September 30, 1997.

[40] University of Maryland Space Systems Laboratory, *Inertial Measurement Unit (IMU)*, Available: http://www.ssl.umd.edu/projects/RangerNBV/thesis/2-4-1.htm [Accessed 10/3/2017]

[41] U.S. Geological Survey. *Nonindigenous Aquatic Species Database*. Gainesville, Florida. 2017. Available: https://nas.er.usgs.gov/queries/factsheet.aspx?speciesid=963 Accessed [10/2/2017].

A Authorship

Table 5: Table of authorship

Section	**Authors**	**Editors**
Abstract	Andrey	William, Joseph
Introduction	Nikolay	William, Brandon
Background	Team	Team
Lionfish	William	Joseph
Underwater Robotics	Joseph	William
Notable Electronics	Joseph	William
Harvesters	Andrey, Nikolay	Joseph, William
Sensors	Brandon, Joseph	Joseph
Controls and Computer Vision	Brandon, Joseph	Joseph, William
Methodology	Joseph, Brandon	William
Requirements	Andrey, Joseph, Brandon	Joseph, William
Results	All	Joseph, William

B LCD Code

```
#include <LiquidCrystal.h>
#include <Ethernet.h> //Load Ethernet Library
#include <EthernetUdp.h> //Load the Udp Library
#include <SPI.h> //Load SPI Library

LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
int t = 19;
int q = 0;
int w = 0;
String x = "7";
String y = "7";
String z = "7";

byte mac[] ={ 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xEE}; //Assign mac address
IPAddress ip(169, 254, 1, 140); //Assign the IP Adress
unsigned int localPort = 5000; // Assign a port to talk over
char packetBuffer[UDP_TX_PACKET_MAX_SIZE]; //dimensian a char array to hold our data packet
String datReq; //String for our data
int packetSize; //Size of the packet
EthernetUDP Udp; // Create a UDP Object

void setup()
{

        lcd.begin(16, 2); //Initialize the 16x2 LCD


        lcd.clear();     //Clear any old data displayed on the LCD


        lcd.print("Directions:"); // Display a message on the LCD!
  lcd.setCursor(0, 1);
  lcd.print("Submerge!");

  Serial.begin(9600); //Initialize Serial Port
  Ethernet.begin( mac, ip); //Inialize the Ethernet
  Udp.begin(localPort); //Initialize Udp
  delay(10000); //delay

}

void loop()
{
  packetSize =Udp.parsePacket(); //Reads the packet size
  //Serial.print("here");
  if(packetSize>0) { //if packetSize is >0, that means someone has sent a request

    Udp.read(packetBuffer, UDP_TX_PACKET_MAX_SIZE); //Read the data request
    String datReq(packetBuffer); //Convert char array packetBuffer into a string called datReq

    Serial.println(datReq);
    x = datReq.substring(3,8);
    y = datReq.substring(9,14);
    z = datReq.substring(15,20);
    Serial.println(x);
```

```
  Serial.println(y);
  Serial.println(z);
}
memset(packetBuffer, 0, UDP_TX_PACKET_MAX_SIZE); //clear out the packetBuffer array

if (x == "7"){
  t = 19;
}
else if (x == "6"){
  t = 11;
}
else if (y == "6"){
  t = 2;
}
else if (x == "1"){
  t = 8;
}
else if (y == "1"){
  t = 5;
}
else if (x == "5"){
  t = 10;
}
else if (y == "5"){
  t = 1;
}
else if (x == "2"){
  t = 7;
}
else if (y == "2"){
  t = 4;
}
else if (x == "4"){
  t = 9;
}
else if (y == "4"){
  t = 0;
}
else if (x == "3"){
  t = 6;
}
else if (y == "3"){
  t = 3;
}
else if (z == "3"){
  t = 14;
}
else if (z == "2"){
  t = 13;
}
else if (z == "1"){
  t = 12;
}
else if ((x == "0") and (y == "0") and (z == "0")){
  t = 18;
}
else if (x == "8"){
```

```
  t = 15;
}
else if (x == "9"){
  t = 16;
}

      /*lcd.setCursor(0, 1);  //Set the (invisible) cursor to column 0,
                                                          // row 1.
int x = analogRead(13);
lcd.print(x);
delay(1000);
lcd.setCursor(0, 1);
lcd.print("                ");*/
delay(1000);
lcd.clear();
lcd.setCursor(0, 0);

switch(t){
  case 0:
    lcd.print("       ^^");
    break;
  case 1:
    lcd.print("      ^^^^");
    break;
  case 2:
    lcd.print("     ^^^^^^");
    break;
  case 3:
    lcd.setCursor(0, 1);
    lcd.print("       vv");
    break;
  case 4:
    lcd.setCursor(0, 1);
    lcd.print("      vvvv");
    break;
  case 5:
    lcd.setCursor(0, 1);
    lcd.print("     vvvvvv");
    break;
  case 6:
    lcd.print("<");
    lcd.setCursor(0, 1);
    lcd.print("<");
    break;
  case 7:
    lcd.print("<<");
    lcd.setCursor(0, 1);
    lcd.print("<<");
    break;
  case 8:
    lcd.print("<<<");
    lcd.setCursor(0, 1);
    lcd.print("<<<");
    break;
  case 9:
    lcd.print("                >");
    lcd.setCursor(0, 1);
```

```
    lcd.print("                >");
    break;
  case 10:
    lcd.print("               >>");
    lcd.setCursor(0, 1);
    lcd.print("               >>");
    break;
  case 11:
    lcd.print("              >>>");
    lcd.setCursor(0, 1);
    lcd.print("              >>>");
    break;
  case 12:
    lcd.print("       OO");
    lcd.setCursor(0, 1);
    lcd.print("       OO");
    break;
  case 13:
    lcd.print("      OOOO");
    lcd.setCursor(0, 1);
    lcd.print("      OOOO");
    break;
  case 14:
    lcd.print("     OOOOOO");
    lcd.setCursor(0, 1);
    lcd.print("     OOOOOO");
    break;
  case 15:
    lcd.print("Turn Right");
    break;
  case 16:
    lcd.print("Turn Left");
    break;
  case 17:
    lcd.print("     XXX");
    lcd.setCursor(0, 1);
    lcd.print("     XXX");
    break;
  case 18:
    lcd.print("XXXXXXXXXXXXXXXX");
    lcd.setCursor(0, 1);
    lcd.print("XXXXXXXXXXXXXXXX");
    break;
  case 19:
    if (q == 0){
      lcd.print("     OOOOOO");
      lcd.setCursor(0, 1);
      lcd.print("     OOOOOO");
    }
    else if (q == 1){
      lcd.print("Turn Right");
    }
    w = w + 1;
    if (w >= 10){
      if (q == 0){
        q = 1;
      }
```

```
        else{
          q = 0;
        }
        w = 0;
      }
      break;
  }
}
```

C Arduino Code

```
int bPin = 3; //buoyancy chamber pulse pin
int rPin = 4; //reload the rod pulse pin
int gPin = 5; //turn the geneva pulse pin
int tPin = 7; //trigger relay pin
int fPin = 9; //input pin that determines when the system fires
int sPin = 8; //unnecessary signal pin (rn, should put a sensor on it)

int oneRevGeneva = 200; //how many pulses for Geneva, etc.
int oneRevReload = 20000;
int indexBuoyancy = 350;

int gRate = 1500; //dwell between cycles in microseconds. ~390 is max.
int rRate = 400;
int bRate = 500;

int resetDistance = 0; //variable storing how far the buoyancy chamber must reset
char shotNumber = 0;  //how many shots have been fired
int revsToReload = 10; //revolutions for the rod to be drawn back.

void setup() {
pinMode(bPin, OUTPUT);
pinMode(rPin, OUTPUT);
pinMode(gPin, OUTPUT);
pinMode(tPin, OUTPUT);
pinMode(fPin, INPUT);
pinMode(sPin, INPUT);
}

void loop() {
  if(digitalRead(fPin)){
    fireShot();
  }

  else{
    delay(10);
  }
}

void fireShot(){
  digitalWrite(tPin, HIGH); //fire off the trigger relay
  runStepper(rPin,(oneRevReload/3),rRate);
  shotNumber++; //index the shot
  indexBuoyancyChamber(shotNumber); //move the buoyancy chamber to its proper position
  digitalWrite(tPin, LOW); //reset the trigger relay
  resetHarvester(shotNumber); //wind up the harvesting mechanism
  indexGeneva(shotNumber); //move the correct spear tip into position
```

```
  if(shotNumber > 8){
    shotNumber = 0;
  }
}


void indexGeneva(char shot){

  char toIndex = 0;

  switch (shot) {
    case 8:
      toIndex = 8;
      break;
    case 7:
      toIndex = 4;
      break;
    case 6:
      toIndex = 2;
      break;
    case 5:
      toIndex = 4;
      break;
    case 4:
      toIndex = 1;
      break;
    case 3:
      toIndex = 4;
      break;
    case 2:
      toIndex = 2;
      break;
    case 1:
      toIndex = 4;
      break;
    default:
      toIndex = 0;
      break;
  }
      runStepper(gPin,toIndex*oneRevGeneva,gRate);
}

void indexBuoyancyChamber(char shot){ //blocking function
    if(shot == 0){} //then reset
    else{
      runStepper(bPin,indexBuoyancy,bRate);
      delay(500);
    }
}

 void resetHarvester(char shotNumber){
  if(shotNumber == 0){}
  else{
  runStepper(rPin, oneRevReload*revsToReload,rRate);
  }
```

```
}

void runStepper(int pin, int cycles, int rate){
      for(int i=0;i < cycles; i++){
      digitalWrite(pin,HIGH);
      delayMicroseconds(rate);
      digitalWrite(pin,LOW);
      delayMicroseconds(rate);
      }
}
```