Design of Suction Stabilized Floats for First Responder Localization via Ultra-Wideband (UWB) and Internet of Things (IoT)

by

Michaela Dye

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2020 by the
Graduate Supervisory Committee:

Sangram Redkar, Chair
Thomas Sugar
Bradley Rogers

ARIZONA STATE UNIVERSITY

May 2020

ABSTRACT

Suction stabilized floats have been implemented into a variety of applications such as supporting wind turbines in off-shore wind farms and for stabilizing cargo ships. This thesis proposes an alternative use for the technology in creating a system of suction stabilized floats equipped with real time location modules to help first responders establish a localized coordinate system to assist in rescues. The floats create a stabilized platform for each anchor module due to the inverse slack tank effect established by the inner water chamber. The design of the float has also been proven to be stable in most cases of amplitudes and frequencies ranging from 0 to 100 except for when the frequency ranges from 23 to 60 Hz for almost all values of the amplitude. The modules in the system form a coordinate grid based off the anchors that can track the location of a tag module within the range of the system using ultra-wideband communications. This method of location identification allows responders to use the system in GPS denied environments. The system can be accessed through an Android app with Bluetooth communications in close ranges or through internet of things (IoT) using a module as a listener, a Raspberry Pi and an internet source. The system has proven to identify the location of the tag in moderate ranges with an approximate accuracy of the tag location being 15 cm.

ACKNOWLEDGMENTS

I want to thank Dr. Sangram Redkar for allowing me to work with him on this thesis. He provided support for me to achieve every step of the work involved in this project and I was able to learn so much from him in both my classes with him and information towards this course. I am thankful for the time he has invested towards my education.

I would also like to thank Dr. Thomas Sugar and Dr. Bradley Rogers for being a part of my supervisory committee. I know they both have busy schedules and I appreciate their time they put into being a part of this project.

I also would like to thank Susheelkumar Cherangara Subramanian as he aided me with understand how the suction stabilized floats work. I would not have been able to do this thesis without his help and his initial work towards suction stabilized floats in his master's thesis.

I would like to thank Sandesh Ganapati Bhat for initially setting up the Raspberry Pi for the listener application of the tracking system. That was an area I did not have any expertise in, and I did not know much on how to approach that part of the project. He also helped teach me how to use the software on my own so I could alter the system when I needed to.

Finally, I would like to thank my family for their support during this entire project. I know they did not appreciate the 3D printer running all of the time while I was making prototypes of the system, but we all learned to tune it out.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

CHAPTER 1

INTRODUCTION

Project Outline

The initial problem statement for this thesis was to design and build a prototype system to help locate and assist airmen during a water rescue and recovery mission. This device is needed for the rescue of airmen who bailout in barren water environments with minimal visual location identifiers. The proposed system will help rescue crews identify the location of the airman while he is stranded on open waters.

The initial concept was based on using a suction stabilized platform with a GPS transmitter to help crews identify the airman's location and provide them with a float to help them survive on the open waters. However, this later developed into creating a network of suction stabilized floats equipped with ultra-wideband technology to help assist in the rescue of the airman by establishing a relative coordinate system to help the rescue crews identify the location of the airman. The airman could turn on his locater tag and the system of anchors could help identify his location. Another set up of the system could allow a searching entity to be equipped with a tag to help record areas that have already been searched. This technology would be ideal in helping identify locations and targets in GPS denied environments.

Suction stabilized floats are buoyant floats that have a chamber that holds liquid above the waterline of the body of water it is floating in which helps stabilize the float as it raises the metacentric height and increases stability. This concept is further discussed in the following sections as this thesis outlines the research on suction stabilized floats for

the floating platform and the GPS transmitters considered for the design of this rescue system.

Paper Outline

This thesis report will present the following information:

- Chapter 2 outlines the literature review conducted for understanding the background information and calculations required for the model of the proposed system. This includes the mathematical models used for analyzing suction stabilized floats, how the stability equations are found and the different types of GPS transmitters and real time location systems available for use.
- Chapter 3 describes the design and methodology used in creating the different models that were evaluated for the system. This section focuses on the material selection, the optimization of the float by analyzing the suction stabilized calculations and identifying the optimal locating system for this application.
- Chapter 4 outlines the testing and results of the prototype model. The testing focused on performing experiments to gather data to determine the stability of the float, how the device performs in water, and the range and restrictions of the locating system.
- Chapter 5 provides the conclusion of the project and discusses how the device should be used and what changes are needed to help the device to

better assist in water rescue. Additional applications for the use of suction stabilized floats are also included.

CHAPTER 2

LITERATURE REVIEW

History of Air Force Water Rescues

The methods currently used in water rescues for the Air Force are confidential so the most recent information about them is unknown. However, talking with former Air Force members can reveal some methods which can help provide some possible insight on what methods currently exists and what methods have proved successful in the past.

After interviewing a former Air Force member, it was discovered that when an airman ejects out of a plane, he is exposed to high G forces. The force causes the spine of the airman to compress approximately 3 inches. Due to this, it is essential for the airman to remain as still as possible to allow their back to recover from the trauma (Kwan, A. (2019, November 1), Phone interview with SME). Preexisting floats that have been used help keep the airman held in a certain position while allowing them to float on the surface of the water. These floats are also equipped with trackers to assist in locating the downed airman during a water rescue.

One such device is a raft used for water bailouts by airmen and can be seen in Figure 1. In the image, a sea dye that is being ejected from the raft can be seen and this helps the rescue personnel identify the location of the airman. The float helps support the airman as it allows him to float and remain in a single position until he is rescued from his location.

Figure 1: Air Force Captain Performing a Rescue Exercise with Inflatable Raft. (Gibson, 2011).

In addition to the raft, the airmen are equipped with radio devices. A Combat Survival Evader Locater (CSEL) can be seen in Figure 2. This device uses satellites to identify the location of the device and has an encrypted two way over-the-horizon communication method to have contact with the rescue personnel. This allows the device to be able to be used almost anywhere in the world where there is a satellite coverage. The encryption used within the device also helps reduce the chances of the airman from being located by potential enemies in the surrounding area.

Figure 2: A Lt. Commander Using a Combat Survival Evader Locator (CSEL) During a Survivor Extraction Exercise. (Fallon Naval Air Station, 2001).

The CSEL device performance is depicted in Figure 3. The system uses different methods of communications between various organizations. There is secured data from the device that is sent to the UHF SATCOM satellites and national asset to inform the proper bases and authorities. The system also has a line of sight voice transmission line that allows the downed airman to communicate with the rescue teams coming to their aid.

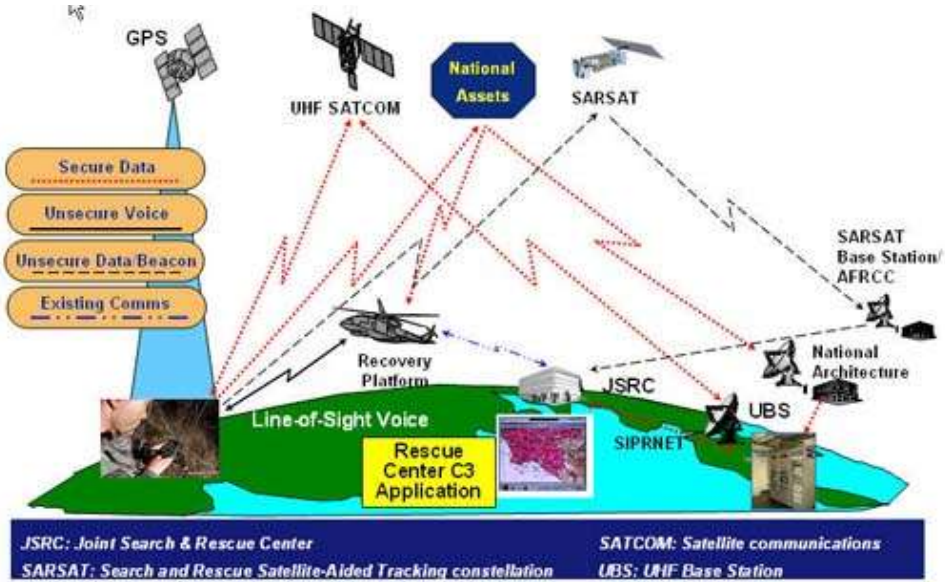

Figure 3: CSEL Device Performance and Components (Olive-Drab).

One of the most important things involved with rescue is identifying the location of the airman as quickly as possible. If the airman is injured or in enemy territory, it is essential to find them in a timely manner to remove them before they are further harmed. Due to this, it is important to design a device that can help the airman float in the open water while transmitting their location to the necessary rescue personnel.

Suction Stabilized Floats

Suction Stabilized Floats are used to increase the stability of a float in an open body of water by raising the metacentric height of the float through the inverse slack tank effect. A slack tank is typically an internal volume that is not enclosed or is a partially

filled tank within a buoyant object such as a float or ship. Since the tank is not enclosed where it can keep the same fluid volume and shape, the metacentric height is decreased with reduces the stability of the ship (Biran, 2003). The inverse slack tank effect uses a tank that is sealed and filled with a fluid to a point that is higher than the surface of the water. The water in this chamber has a volume that is held higher than the surface of the water. This chamber acts as an inverse slack tank which raises the metacentric height (Subramanian, 2014, pp. 4).

When a buoyant object tilts at an angle in a body of water, the center of buoyancy changes which can be shown as the metacenter radius because it is a point where the line from the buoyant force of the original position and the metacenter intersect. This is demonstrated in Figure 4. However, when an object with an inverse slack tank is tilted at an angle, the mass above the water plane area is different than a float without fluid inside. This causes the center of gravity to shift. However, the volume of the float below the water plane remains the same (Subramanian, 2014, pp. 35).

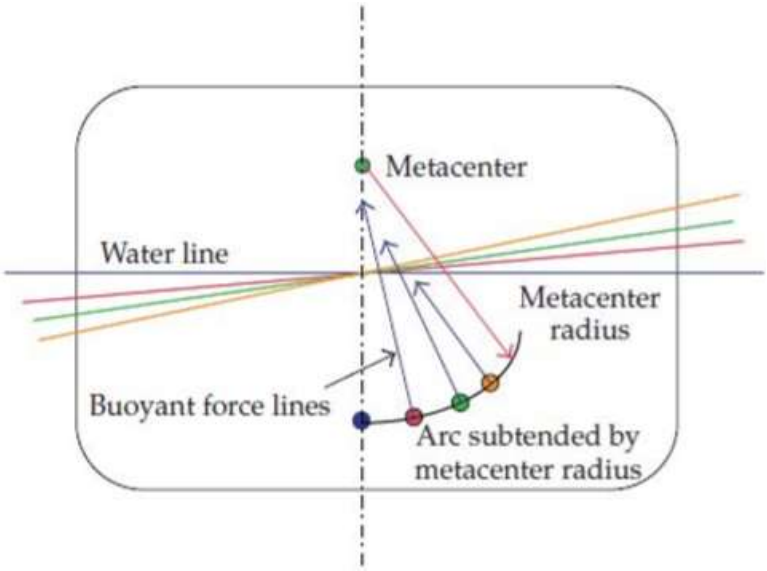

Figure 4: Metacenter and Buoyant Force Lines (R.A. Ibrahim, 2009).

The metacentric height for a float sitting at a static flat position can be determine by its size and weight of the float in Equation 2.1. In this equation I_{yy} is the moment of inertia of the float, V_i is the volume of the float that is immerged in the water, cog is the center of gravity height from the bottom of the float and h_i is the height of the immerged water from the bottom of the float.

$$\overline{GM} = \frac{I_{yy}}{V_i} - (cog - \frac{h_i}{2}) \tag{2.1}$$

The metacentric height for an SSF float can be calculated with Equation 2.2 where $\overline{GM}_{SSF}$ is the effective metacentric height with an SSF float, $\overline{GM}$ is the average metacentric height, I_{yy_w} is the moment of inertia of the water plane and V_{disp} is the displaced fluid volume.

$$\overline{GM}_{\mathrm{SSF}} = \overline{GM} + \frac{\mathrm{I_{yyw}}}{\mathrm{V_{disp}}} \tag{2.2}$$

When the object is tilted at an angle, the position of the center of gravity moves away from the center of the float. Due to this, a righting moment is created which is the righting arm multiplied by the weight of the ship or float. The righting arm is the horizontal distance between the center of gravity and the center of buoyancy which is represented as $\overline{\mathrm{GZ}}$ (Subramanian, 2014, pp. 11). This moment helps the float return to its stable, original position at a quicker rate than a float without the water chamber. This is due to the wedge of water inside the water chamber of the float that comes out of the water due to the change in the angle of the ship demonstrated in Figure 5. The equation for the moment caused when tilted is explained in Equation 2.3 and uses the weight of the ship represented by w and the righting arm.

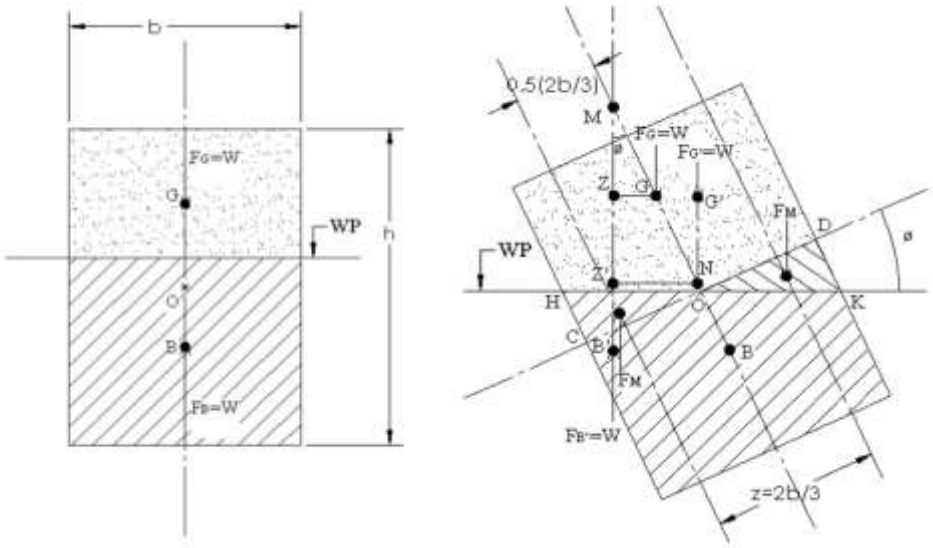

Figure 5: Float at Stable State and Tilted State (Subramanian, 2014, pp. 34-36).

$$M_w = w * \overline{GZ} \tag{2.3}$$

Some of the main applications for SSF include being used to improve ship stability and in platforms for offshore wind turbines. However, this can be applied to a smaller scale where it can be used for a single man raft for water rescues or as small floats to hold ultra-wideband devices to set up a rescue grid and mark positions that have been check. It also can be applied to a large platform that could potentially act as a landing platform for a helicopter on open waters.

The float can be made up of a buoyant material with a chamber for water or the entire float could consist of two different chambers where an outer chamber is filled with air and an inner chamber holds water.

Based upon the models to describe the motion of the float, a series of Lagrange equations can be created to understand how the system operates. The Lagrange equation represents the energy within the system and is shown in Equation 1.4. A combination of

differentials of this equation can be transformed to give the equations of motion of the system as demonstrated in Equation 1.5.

$$L = T - V \tag{2.4}$$

$$\frac{d}{dt}\left[\frac{\partial L}{\partial \dot{x}}\right] - \left[\frac{\partial L}{\partial x}\right] = 0 \tag{2.5}$$

For this specific application, the float with SSF can be represented with two springs, two dashpots, an oscillating mass and a pendulum mass and are represented as the variables demonstrated in Figure 6. The oscillating mass represents the movement of the float as it rides on the waves in the sea, the pendulum mass represents the mass of the SSF portion of the float. The linear dashpot and springs help demonstrate how the float moves on its own and the rotational dashpot help demonstrate the stability of the float due to the inverse slack tank within the float.

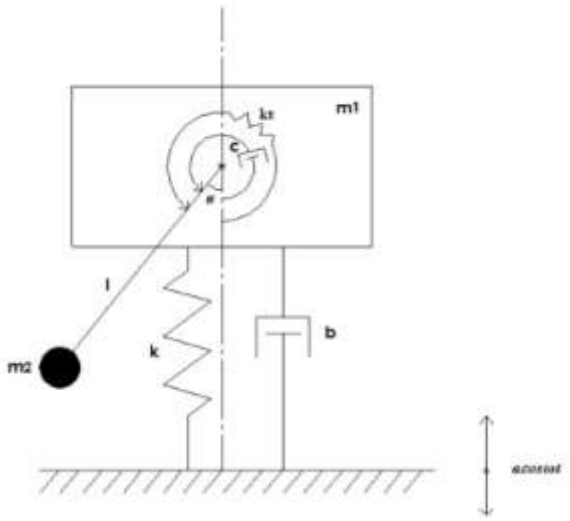

Figure 6: Model of an SSF Float (Subramanian, 2014, pp. 48).

Using this model, the two equations of motion can be derived for the model. These are shown in Equations 1.6 and 1.7.

$$(m_1 + m_2)(\ddot{z} - a\omega^2 cos(\omega t)) + m_2 l\ddot{\phi}\sin(\phi) + m_2 l\dot{\phi}^2 \cos(\phi) + kz + b\dot{z} = 0 \quad (2.6)$$

$$m_2 l\sin(\phi)(\ddot{z} - a\omega^2 cos(\omega t)) + m_2 l^2\ddot{\phi} + m_2 g l\sin(\phi) + c\dot{\phi} = 0 \quad (2.7)$$

These equations can then be checked for stability. The complete derivation of these equations is outlined in detail in Appendix A. After the derivation of these equations, they can be converted to a normal form to solve for the stability of the system. This process is described further in Chapters 2 and 3.

GPS Transmitters

There are a variety of available locating systems that are used for a range of applications. The operations of these systems are based off of ultra-wideband communications using anchors and tags, cellular signals, satellite signals, over the horizon communications and underwater acoustic transmissions.

The ultra-wideband systems use a combination of tags and anchors to establish a coordinate system. They require at least four anchors to properly identify the location of a tag within a certain range. The area where the tag can be identified can be increased with the addition of more anchors to the system to expand the relative coordinate grid. When the tag is within transmission distance of four anchors, its location can be displayed on an app on a tablet or laptop connected to the anchor tag system using triangulation as described in Figure 7. However, when using an app on a phone, the phone must be in range of the system as well as it uses Bluetooth to connect to the system. This restricts the range of use of this technology. This method of identifying location is optimal for GPS denied environments as this allows the user to set up their own locating system based off

their relative positions to each other. The only requirements is that each tag or anchor is within range of each other and have a direct line of sight to the other system components.

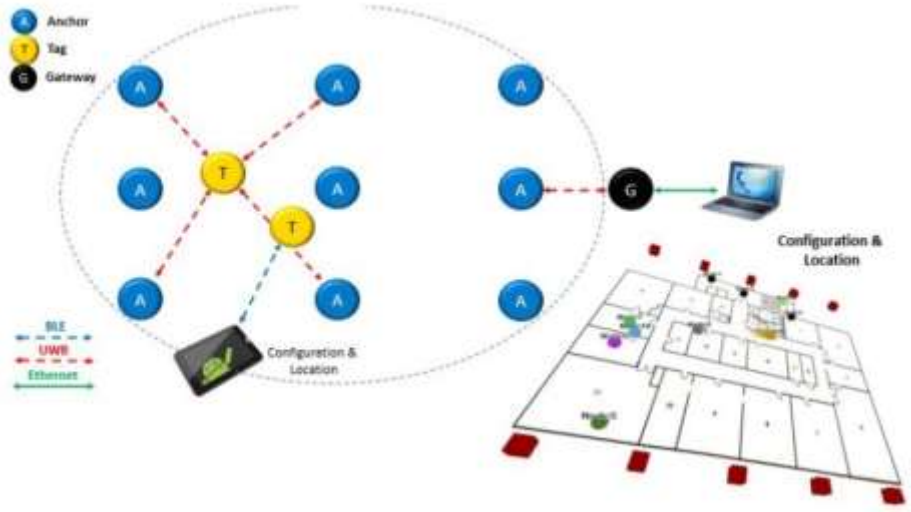

Figure 7: Tag and Anchor Based GPS System (Symmetry Electronics, 2019).

Cellular GPS transmitters are one of the most used types of transmitter when on land and at short distances offshore due to the numerous available cellular networks. In most cases, a location can be identified with high accuracy. However, this method is not available to open waters across the ocean because the signal range of cellular networks cannot be extended to cover open bodies of water (Dye, M. (2019, October 19). Personal interview with D. Dye.) In addition, in some terrains such as dense mountain ranges, deep canyons and remote areas on land, the cellular signals may not be able to reach the device.

Satellite GPS transmitter are one of the most reliable methods of transmitting locations across the globe due to the wide expanse of satellite systems circling the planet. It is free to use the satellites to obtain a position using a GPS but in order to transmit data using the satellite network, a subscription is usually required. The subscription often allows the customer to be able to program their device to send an emergency message

either to the proper authorities or family and friends when a button on their device is engaged. The device will then send their location and in some cases a message to their contacts.

There is also the Argos Satellite system which is an expansive network of satellites used to track wildlife and weather activities. Currently, this system is created from a partnership with multiple agencies spanning across dozens of countries in the interest of science (Argos). The system uses GPS transmitters on earth that are programmed to communicate with their network. The device data is logged periodically and is transmitted to a processing center where the results can be sent and analyzed by the specific user. A similar network could be used by the military to help with tracking down airmen and other military operations as the network could be encrypted for military use only. Figure 8 shows how the Argos Satellite system operates.

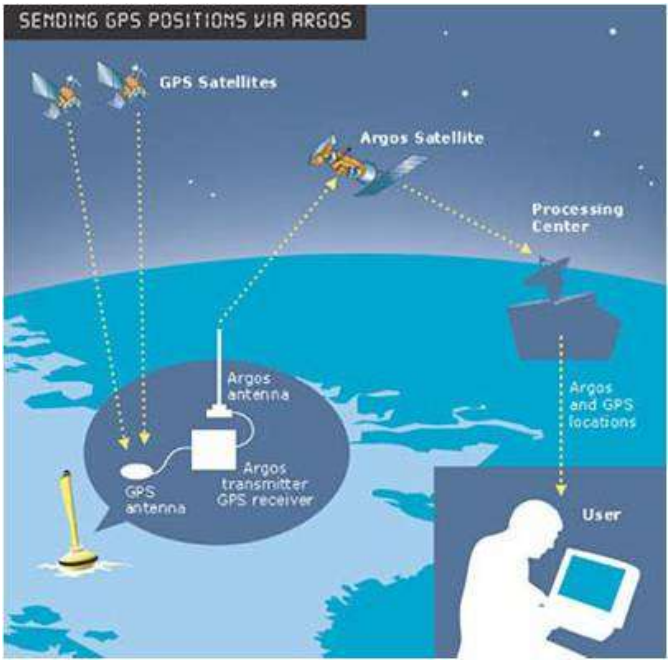

Figure 8: Demonstration of a GPS transmitter in using the Argos Satellites (Argos, 2008).

While satellite transmitters are reliable in most cases, they do not perform well under dense cloud coverage which is highly possible when out at sea due to the high content of moisture in the air (Kwan, A. (2019, November 1). Phone interview with P. Tracy.). The clouds can obstruct the signals between the transmitters and the satellites for an unknown amount of time which could be problematic if an immediate rescue is needed. However, due to limited accessibility to other communication methods at sea, such as cellular networks, this has a higher probability of being able to contact with others.

Another important means of communication is over the horizon transmissions. This is an effective way of contacting others within a certain range. This can be seen in the CSEL device previously discussed. The device primarily relies on satellite communications but also has a component that allows the user to talk with rescuers who are in the proximity of them. This means of communication is reliable in most terrains, but the over the horizon component of the device is not easily encrypted so others may pick up the transmissions which could endanger them if they are in hostile territories.

Long baseline underwater acoustic communication is another type of location and transmission device that is available. This method is like the tag and anchor-based system as there are beacons that are anchored in place on the surface of the water or anchored on the sea bottom (Gode, 2015, pp 2). The underwater device can then estimate its location using acoustic waves and time intervals to identify its location. The range of this technology is approximately 100 meters to 10 kilometers (Gode, 2015, pp 11). The cons of this system include ensuring the anchors are properly calibrated and the possibility of interference with other underwater soundwaves. There is also the possibility that other

ships or submarines within the vicinity of the device could pick up their soundwaves and identify their location. Like the over the horizon method, the signal cannot be easily encrypted so others could also identify the location of the airman.

CHAPTER 3

DESIGN AND METHODOLOGY

The goal of the thesis is to derive a water rescue technology to assist the Air Force in locating and rescuing a downed airman. The original design for this was to create a suction stabilized float that the airman could have access to when bailing out of a plane. The device was to be equipped with a GPS beacon that could be activated once in the water. However, with further interviews with subject matter experts and analysis of currently available solutions, the direction of design was changed. The proposed solution is to use a system of SSFs in a network that is connected to a device or rescue team that is scanning the area within range of the floats. This is to help establish a relative location coordinate system in a GPS denied environment to help rescue teams to help identify where the airman is to be rescued. This chapter outlines how the equations for stability of each float was determined, the materials for the float, the large and small scale application models, necessary components and the Matlab and Mathematica codes associated with the problem.

Material for the Float

The material used to construct the float needed to be able to withstand the environmental conditions faced out at sea. Research showed that the average temperature ranges for the surface of ocean water span from -2°C (28°F) to 30°C (86°F) (US Department of Commerce, & National Oceanic and Atmospheric Administration, 2013). Therefore, the float must be able to operate without restrictions for those ranges. In addition, the material must have a high tensile strength to prevent the material from being

punctured as that would render the float useless. Lastly, the material must have a low porosity and a low water absorption if it is to create a chamber that holds air that allows the rescue device to float. Another factor that helps increase the buoyancy of the float is having a less dense material.

Based upon this criterion, materials that are currently used for rafts, airplane slides, inflatables and wetsuits were researched to identify the best material for this application. PVC Sheet/Vinyl was ultimately chosen due to its low density, moderate tensile strength, high temperature range and it is a readily available material. Ideally, this could allow the systems to be compact for transport and so they are light weight and take up little space when not in use. The large scale model uses this material and Solidworks models were developed using this material to find its subsequent density and mass calculations. The mass density values are outlined in Appendix B.

For the purpose of demonstration of the float's abilities, a small-scale float will be constructed using a 3D printer with PLA filament. This material will be used instead of the PVC Sheet because the machinery and tooling required to properly craft a raft using PVC Sheets is unavailable and costly. However, using the 3D printed method will require extra materials to ensure the water chamber can be established and that the outer air chamber is waterproof. The mass density for the PLA filament is also outlined in Appendix B.

Calculations Using Excel and Solidworks

In order to begin designing the float, it was important to experiment with a variety of different shapes. Initial designs for the larger scale application consisted of three basic

float shapes that had a maximum diameter or side of 1.6 to 1.75 meters that would be able to hold a man and to understand how the stability is affected with different parameters. This dimension was originally tested because in the United States the average height is 1.75 meters (69 inches) (Fryar, C. D., Kruszon-Moran, D., Gu, Q., & Ogden, C. L., 2018). Therefore, the design of the raft should be able to fit this size of a man. All the designs had an inner platform 0.15 meters below the top surface of the float where the airman could sit or lay down on before being rescued. Each model was created in Solidworks with a vinyl shell, an inner water chamber and a chamber of air surrounding the sides of the water chamber. The models were used to get values for the mass of the float, the volume of the float and the center of gravity. These values were used to calculate a metacentric height value with and without an airman on the float in saltwater and freshwater. An in-depth chart of calculations required to calculate the metacentric height is detailed in Appendix C.

The value used for the average weight of a man in the United States was determined to be 89.8 kilograms (198 pounds) (Fryar, C. D., Kruszon-Moran, D., Gu, Q., & Ogden, C. L., 2018). The value used for the average volume of a man was 0.0652 m^3 (3980 in^3) (Nagao, N., Tamaki, K., Kuchiki, T., & Nagao, M. (1995). The mass and volume of the man were added to the float values in a uniform distribution across the float. The addition of the man to the float decreased the metacentric height by about 6.5 to 10.3 centimeters between all the floats for the full scale application.

Each of these floats have a small hole on the bottom of the float that is used to fill the water chamber. A vacuum effect can be made by placing another hole on the top of

the float and attaching a small pump. This will suck out all of the air to make room for the water that will enter in through the bottom hole.

Since the demonstration of this technology will need to be scaled down for fabrication purposes, the models were also designed. However, due to its size, a man is not able to sit on top, so the weight of the ultra-wideband tracker was used as a payload mass. The optimum float out of the three designs was chosen and a smaller scale model was developed in Solidworks. The metacentric height values were calculated with and without the weight and volume of the ultra-wideband tracker on the float in both saltwater and freshwater. The mass and volume of the tracker were measured to be 0.0568 kg and 74.381 cm^3.

Circular Float

The first float that was designed was a circular float that had a diameter of 1.6 meters and a total height of 0.5 meters as shown in Figure 9. The water chamber was a circle with a diameter of 1.2 meters and a total height of 0.33 meters. The metacentric height values are shown in Table 1.

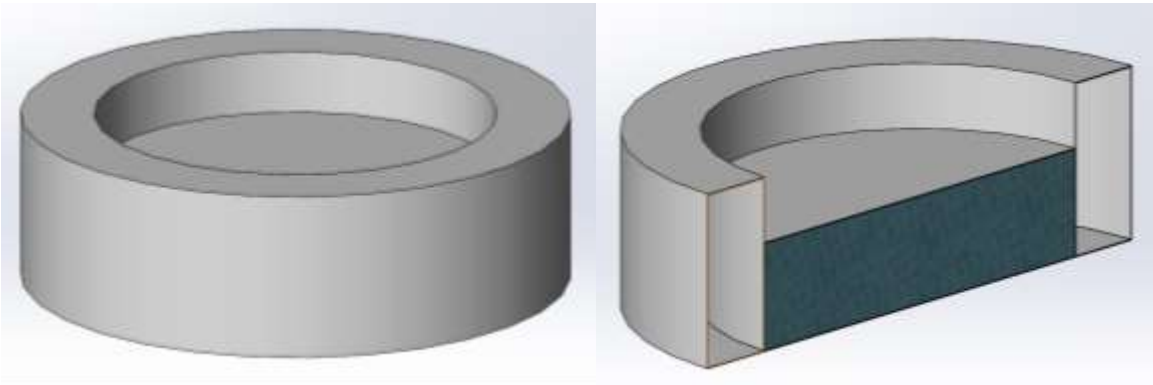

Figure 9: Isometric and Cut Views of the Circular Float from Solidworks.

Table 1
Metacentric Heights for a Circular Float

	Saltwater	Freshwater
Float	0.827 m	0.823 m
Float with Man	0.763 m	0.758 m

Square Float

The second float that was designed was a square float that had sides measuring 1.75 meters and a total height of 0.5 meters as seen in Figure 10. The water chamber was a square measuring 1.25 meters on each side and had a total height of 0.33 meters. The metacentric heights are given in Table 2.

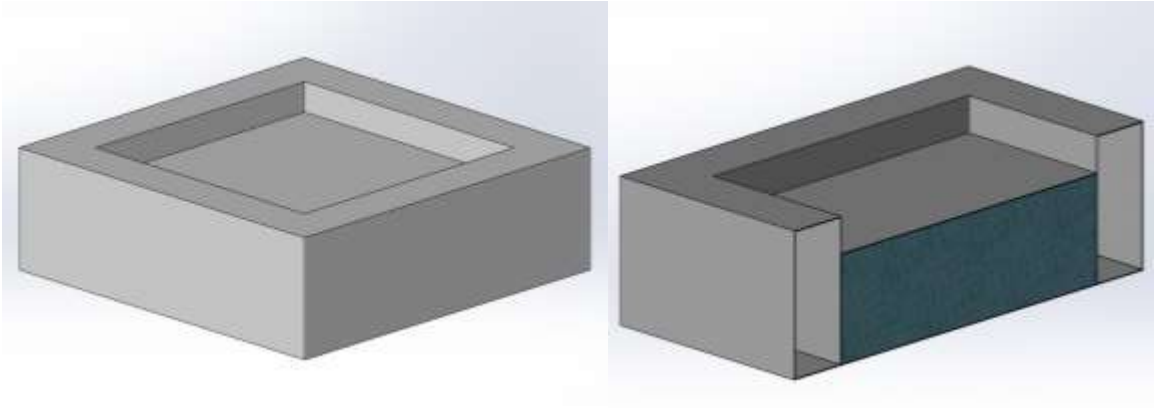

Figure 10: Isometric and Cut Views of the Square Float from Solidworks.

Table 2
Metacentric Heights for a Square Float

	Saltwater	Freshwater
Float	1.437 m	1.429 m
Float with Man	1.340 m	1.323 m

Triangular Float

The third float that was designed was a triangular raft with all sides measuring 1.6 meters and a total height of 0.5 meters. The water chamber was a triangle measuring 1.08 meters on each side and a total height of 0.33 meters. Figure 11 shows the triangular model. The metacentric heights are show in Table 3.

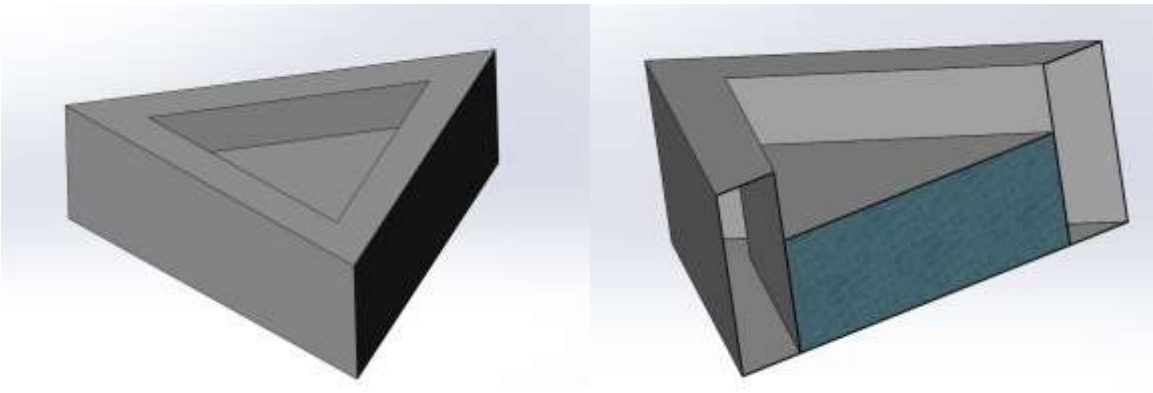

Figure 11: Isometric and Cut Views of the Triangular Float from Solidworks.

Table 3
Metacentric Heights for a Triangular Float

	Saltwater	Freshwater
Float	0.546 m	0.532 m
Float with Man	0.471 m	0.457 m

Observations of Metacentric Heights

Based upon the calculated values of metacentric height, the square will have higher values as there is more area that can be used to stabilize the float. The circular float also can have a high metacentric value with a similar diameter as the length of the square float but it is harder to manufacture. The triangular float requires a much larger

length in order to have similar metacentric values to that of the circle and square floats. It is also a shape that is not totally symmetric so that may cause some issues with stability.

Small Scale Float

Based upon the previous observations, the square float was pursued. To see if this method would work, a small-scale version was designed. To aid in manufacturing, the water chamber was enclosed, and two holes were placed on the top lid to completely fill the chamber and release any trapped air before being sealed. The smaller scale square float had a length and height of 16 cm and 5 cm, respectively. The water chamber had a length of 9 cm and a height of 3 cm. The metacentric heights are given in Table 2.4.

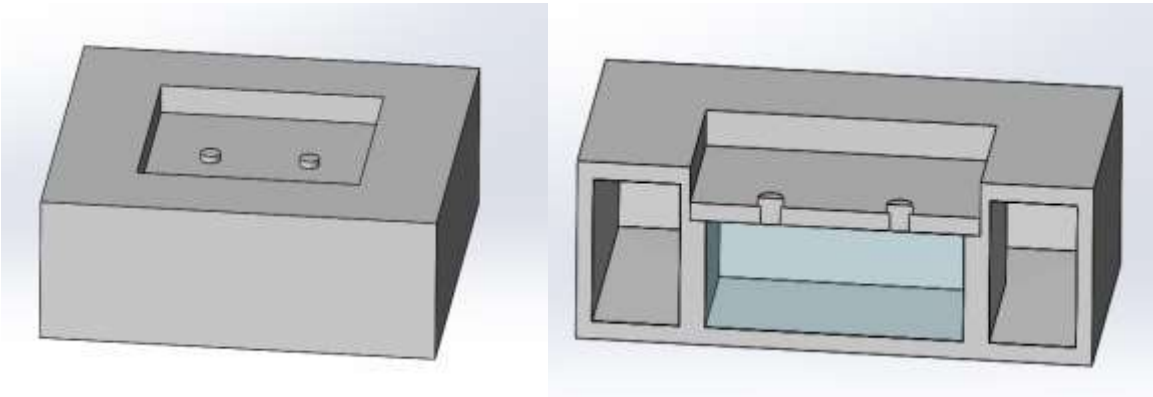

Figure 12: The First Design of the 3D Printed Float from Solidworks.

Table 4
Metacentric Heights for a Small Scale Square Float

	Saltwater	Freshwater
Float	7.883 cm	7.756 cm
Float with Tracker	7.848 cm	7.719 cm

The first print was a square float made from black 1.75mm PLA and was printed on a Tronxy X3 3D printer. Each layer was 0.2 mm thick and the walls were 1.6mm thick with a grid infill with 30% density. The support structure was also made with a grid

pattern at 30% density. The lid for the float was made with the same parameters except the support structure was not needed in the design. Upon completion of the print, there was an apparent error in float and lid. There was under extrusion of the filament due to an error in Cura, the software responsible for slicing the model and producing the g-code for the printer. The filament setting did not properly save to 1.75mm and reset to the default filament diameter of the software to 2.85mm. Due to this, the float was not watertight and there were a significant amount of light passing through tiny holes between the rasters in each layer.

In order to fix this print, the float was sprayed with a two in one primer and black paint spray paint. This did not seal many of the holes so three layers of a leak seal solution were sprayed on the float. This sealed most of the visible holes and an additional layer of paint was sprayed on for the final surface to make sure all the visible holes were sealed. Upon the completion of the waterproofing of the float, the lid was pushed into place as seen in Figure 13.

Figure 13: 3D printed Float that was Coated with Paint and a Waterproof Spray.

Then the water chamber of the float was filled and the holes to allow the water into the float were sealed after all of the air was removed from the chamber. Although the float appeared to be waterproof by a visual examination, it started sinking almost immediately after being placed in water. While the cause is most likely due to learning how to best adjust the settings on the 3D printer, the design was inspected to determine if there was a better design.

One major disadvantage of this design was how the water would have to be put into the water chamber. It would have to be manually filled and would have to be monitored to ensure it would remain completely full. Upon further research into this issue, it was determined that another float design was needed in order to ensure an easier way to maintain the level water in its water. To solve this issue, the float was redesigned to have the water chamber to have a cross-sectional shape of a T. The bottom of the float would be open so there would be constant contact between fluid inside of the chamber and the fluid the float is in. There would be a hole on top of the float where a pumping device with a valve is attached so the air can be sucked out of the chamber, creating a vacuum effect to completely fill it with water. For the purpose of testing in a small scale, this pump was replaced with just a standard ear plug and the float was turned upside down to fill it completely with water. The float was moved around in the water until air bubbles stopped coming out of the chamber. The float was then repositioned into its upright position and allowed to float to the surface of the water.

The newest design shown in Figure 14, has a height of 10.5 cm from the bottom of the float to the base of the tracker holder and a length of 16cm. The water chamber has an upper portion that is 2 cm high and 15 cm wide and a lower section that is 8 cm high

and 8 cm wide. The metacentric heights of this design are outlined in Table 5. Additional dimensions of the float are provided in Appendix C. While these values are lower than previous float designs, the design of the water chamber allows a greater moment to be achieved when the float is tilted as the water extends out farther from the center of the float.

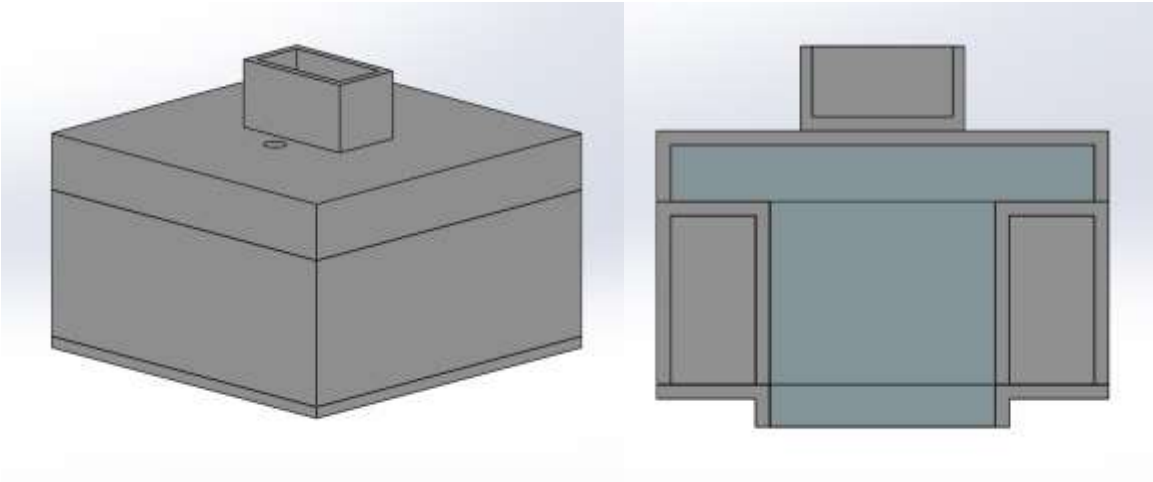

Figure 14: Final T-shaped Float Design from Solidworks.

Table 5
Metacentric Heights for a Small Scale Square Float

	Saltwater	Freshwater
Float	1.804 cm	1.723 cm
Float with Tracker	1.599 cm	1.533 cm

There is a chamber beneath the top section of the water that holds a chamber filled with air to make the float buoyant. To reduce the amount of support material used in the creation of the float and to reduce weight, the float was designed to be made of four separate pieces. These pieces would be glued together using a waterproof glue and then the entire float would be coated in a waterproof sealant to reduce the risk of water entering the air chamber within the float as seen in Figure 15.

Figure 15: 3D Printed Float with Waterproof Sealant and Plug.

GPS Transmitters

Based upon available funds and equipment, the tag and anchor-based system was chosen to help identify the location of a downed airman. The manual for the tracker system states that a combination of 12 anchors and tags can be connected in one network. The trackers shown in Figure 16 are 4.8 cm long, 9.4 cm tall and 1.5 cm wide. The protruding base is 4.8 cm long, 2.3 cm high and 2.25 cm thick at its thickest point. Each of the suction stabilized floats will house one of these trackers in a stand on top of the float. To prevent water from damaging each tracker, they will be placed inside of a sealed bag which will also help reduce the chances of the tracker from being dislocated from its stand.

Figure 16: DMW101 Module Used as the Anchors and Trackers in the Location System.

The trackers on the floats will act as anchors in this system as the tag can be given to a pilot or it can be placed upon a parasail or drone scanning the area or given to other rescue personnel in the range of the system. This will help identify what areas have been searched and pinpoint the location of the downed airman with respect to the established coordinate system when the searchers have identified the airman in the environment.

Equations for Stability

In chapter 1, equations were derived to model the system of the suction stabilized float. These included equations 2.6 and 2.7 which can be used to determine the stability of the float once they are converted into a normal form (Redkar, *Normal Forms*, 2020). To start the conversion into a normal form, a nonlinear transformation, also known as a near identity transformation, can be used to produce a linear (Nayfeh 2011). Then the equations can be used to solve for the stability of the system with each float shape. These equations can be used to test the stability of the equation using Mathematica and Matlab.

Mathematica Codes

After completion of a basic design for each type of float, the initial equations of motion for the system were put into a Mathematica code to try to solve for the normal form of the equation. The normal form method allows for a better representation of stability as the perturbation calculations for the stability for a small threshold interval and is more of an approximation. The Mathematica code used the equations of motion specifically designed to account for the shape of a circular float. To do this the mass of the wedge that appears above the waterline, m_2, was replaced to account for the volume and the density of the float as shown in Equation 3.1. Where R is half of the length of the top water chamber, h is the height of the wedge of water held above the water line, and ρ is the density of the fluid in the float.

$$m_2 = \frac{2}{3} R^2 h \rho \tag{3.1}$$

This equation can be further reduced as the wedge's height is equated to h=Rtanθ where θ is the angle that the float is tilted at (Subramanian, 2014, pp. 57). For small angles the value of tanθ can be approximated as θ which transforms Equation 3.1 into Equation 3.2.

$$m_2 = \frac{2}{3} R^3 \theta \rho \tag{3.2}$$

The new value for the mass of the wedge can be added to the Lagrange equations that describe the system and the equations can be linearized to be analyzed. This transforms Equations 2.6 and 2.7 into Equations 3.3 and 3.4 (Subramanian, 2014, pp. 101).

$$m_1\ddot{z} - m_1 a\omega^2 cos(\omega t) - \frac{2}{3}R^3\rho\theta a\omega^2 cos(\omega t) - \frac{2}{3}R^3\rho\dot{\theta}a\omega sin(\omega t) + kz +$$

$$kacos(\omega t) + \frac{2}{3}R^3\rho\theta g + b\dot{z} = 0 \quad (3.3)$$

$$-\frac{2}{3}(0.7)R^4\rho\theta a\omega^2 cos(\omega t) + I\ddot{\theta} - \frac{1}{3}R^3\rho a^2\omega^2 sin^2(\omega t) + \frac{2}{3}R^3\rho\dot{z}a\omega sin(\omega t) +$$

$$\frac{2}{3}R^3\rho gz + \frac{2}{3}R^3\rho gacos(\omega t) + \frac{4}{3}(0.7)R^4\rho g\theta + c\dot{\theta} = 0 \quad (3.4)$$

The above equations can then be rearranged to solve for the second derivatives shown in Equations 3.5 and 3.6. This is will be used to develop the matrix used to solve for the normal form of the system.

$$\ddot{z} = a\omega^2 cos(\omega t) + \frac{2}{3m_1}R^3\rho\theta a\omega^2 cos(\omega t) + \frac{2}{3m_1}R^3\rho\dot{\theta}a\omega sin(\omega t) - \frac{k}{m_1}z -$$

$$\frac{k}{m_1}acos(\omega t) - \frac{2}{3m_1}R^3\rho\theta g + b\dot{z} = 0 \quad (3.5)$$

$$\ddot{\theta} = \frac{2}{3I}(0.7)R^4\rho\theta a\omega^2 cos(\omega t) + \frac{1}{3I}R^3\rho a^2\omega^2 sin^2(\omega t) - \frac{2}{3I}R^3\rho\dot{z}a\omega sin(\omega t) -$$

$$\frac{2}{3I}R^3\rho gz - \frac{2}{3I}R^3\rho gacos(\omega t) - \frac{4}{3I}(0.7)R^4\rho g\theta - c\dot{\theta} = 0 \quad (3.6)$$

The normal form cannot be solved yet due to presence of the time variable portions in the equation. To eliminate these, a third equation, Equation 3.7, had to be introduced to help solve for the normal form equation. Its first and second derivatives are shown in Equations 3.8 and 3.9. These equation acts upon the time variable portion of the system that are caused by the time dependent oscillations of the system and presents it as an additional variable in the system.

$$y = cos(\omega t) \quad (3.7)$$

$$\dot{y} = -\omega sin(\omega t) \quad (3.8)$$

$$\ddot{y} = -\omega^2 cos(\omega t) = -\omega^2 y \quad (3.8)$$

Using the above equations, the time variant portions of the equations can be replaced with the variables based 'y'. The following 6x6 matrix shown in Equation 3.9 was created to solve for the normal form of the system. It is important to note that the variable "I" used to signify the mass moment of inertia is a designated variable in Mathematica, so this variable was renamed to "L" in the code. In addition, the variables z, $\dot{z}$, θ, $\dot{\theta}$, y and $\dot{\boldsymbol{y}}$ were replaced with x_1, x_2, x_3, x_4, x_5 and x_6 when solving for the Jordan decomposition and y_1, y_2, y_3, y_4, y_5 and y_6 when solving for the normal form of the equation.

$$\frac{d}{dt}\begin{bmatrix} z \\ \dot{z} \\ \theta \\ \dot{\theta} \\ y \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{k}{m_1} & -\frac{b}{m_1} & -\frac{2}{3m_1}R^3\rho g & 0 & a\omega - a\frac{k}{m_1} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ -\frac{2}{3L}R^3\rho g & 0 & -\frac{14}{15L}R^4\rho g & -\frac{c}{L} & -\frac{2}{3L}R^3\rho g a & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -\omega & 0 \end{bmatrix} * \begin{bmatrix} z \\ \dot{z} \\ \theta \\ \dot{\theta} \\ y \\ \dot{y} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2}{3m_1}R^3\rho a\omega^2\theta y - \frac{2}{3m_1}R^3\rho a\dot{\theta}\dot{y} \\ 0 \\ \frac{2}{3L}(0.7)R^4\rho a\omega^2\theta y + \frac{1}{3L}R^3\rho a^2\dot{y}^2 + \frac{2}{3L}R^3\rho a\dot{y}\dot{z} \\ 0 \\ 0 \end{bmatrix} \quad (3.9)$$

Before running the code, all the variables except those relating to the amplitude, a, and the frequency, ω, of the oscillations of the system were substituted into the equations. This was done because Mathematica cannot handle calculations using the number of variables required in this equation. To initially test that they system was working, all the variables were designated as having a value of 1. After running the code, a table contain the stability values can be generated for different values of the amplitude and frequency of evaluations. This table can be turned into a text file where all of the imaginary "I"

values can be manually replaced with “sqrt(-1)”. After doing this the file can be imported into Matlab and converted to a character string and turned into a format where the complex numbers can be read. The data can be rearranged to form a table with 3 columns. Using this new table, the stability of the system can be plotted. The Mathematica and Matlab codes are provided in Appendix D. The resultant normal form values and the stability plots of the code are outlined in Chapter 3.

CHAPTER 4

TESTING AND RESULTS

Results from Mathematica and Matlab Codes

The Mathematica code used the updated equations derived in Chapter 3 that substituted an equation for the mass of the wedge, m_2. The initial test of the system to ensure that the code works substituted the value of 1 for all variables except the amplitude, a, and frequency, ω, of the oscillations. These values have the units of m and Hz respectively. The plot of the stability used values in .5 increments from 0 to 10 for the amplitude, a, and frequency, ω of the oscillations. The results from the Mathematica code were exported to a text file where all of the imaginary "I" values can be replaced with "sqrt(-1)" as Matlab is unable to read the complex number forms from Mathematica. Then the text file data was run through Matlab to convert all the numbers into a readable complex number format. The Matlab code takes all the complex conjugate values produced from the Mathematica code and plots the values that have a negative real number. All negative real numbers whether they have a positive or negative complex number show the system is stable. (Redkar, *Stability Criteria by Eigenvalues*, 2020). Therefore, the graph has a plotted point for each combination of amplitude and frequency that leads to a stable system. As seen in Figure 17, the float is stable for almost all points that were evaluated except when the amplitude and frequency are 0 and 0.5 and in certain cases when the amplitude is equal to 1.

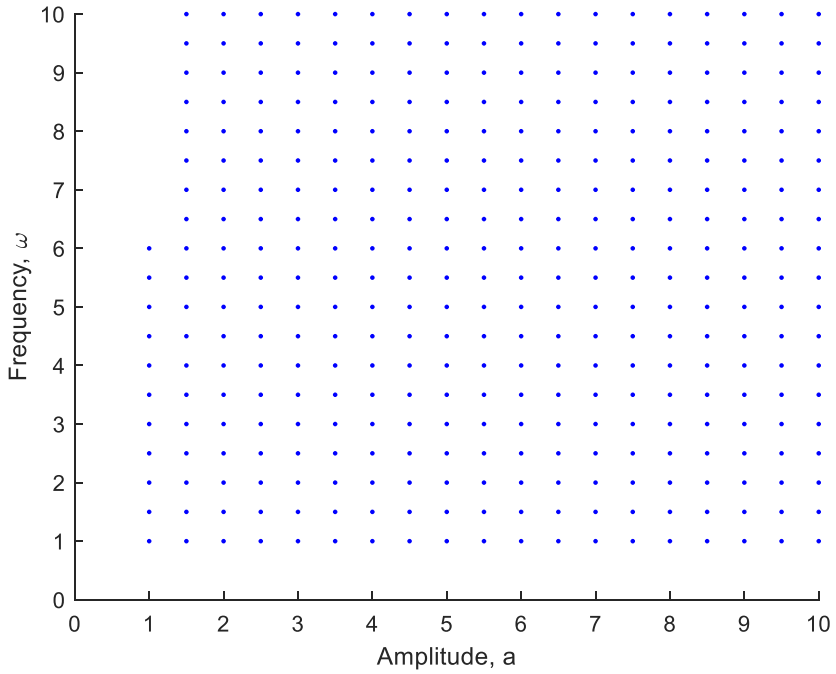

Figure 17: Test of Mathematica and Matlab Software for Stability Plots.

In order to plot the stability for the prototype floats, some tests were conducted to determine the two stiffness coefficients and the two damping coefficients of the system. To do this, a float was placed in a small spa filled with freshwater that had been treated with chlorine as shown in Figure 18. All the succeeding calculations are based off the freshwater calculations due to the unavailability of a large body of saltwater.

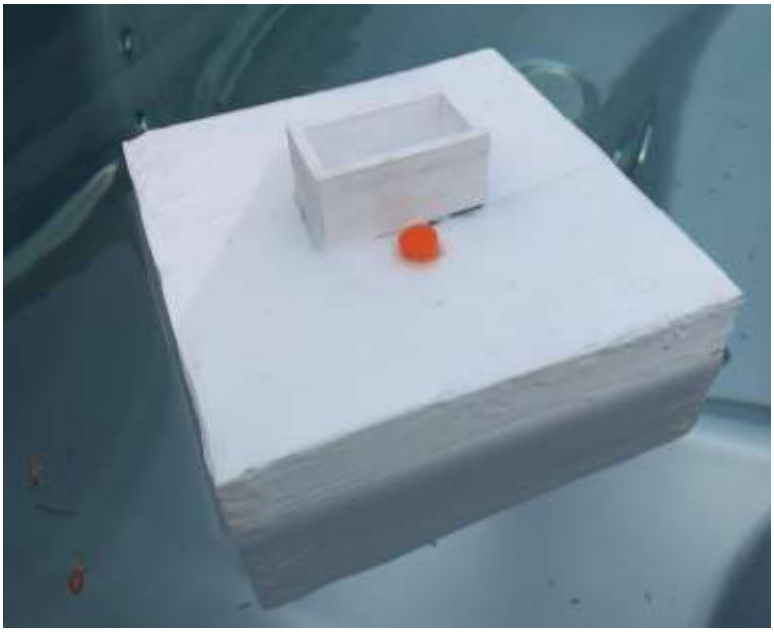

Figure 18: The Float Floating Inside of a Spa to Determine if it is Waterproof and to Gather Data.

After the water chamber was filled and the float stabilized itself on the water's surface, the depth of immersion was measured and recorded as 8.1 cm. The original calculations based off the weights and measurements derived in Solidworks estimated the depth of immersion to be 8.034 cm. This is a difference of .066 cm or a 0.8% error from the actual and predicted value. This error is likely due to the additional weight of the silicone coating and waterproof glue used on the float which increases the floats weight and lowers the center of buoyancy.

To determine the translational stiffness ratio, a brass weight of 121 g was added to the center of the top of the float. The additional weight lowered the float into the water and increased the depth of immersion to 8.7 cm. Using these two values, the stiffness coefficient was calculated using Equation 4.1 where m is the mass of the weight, g is the gravity constant and Δh is the change in the depth of immersion. The stiffness coefficient was determined to be 197.84 N/m.

$$k = \frac{mg}{\Delta h} \tag{4.1}$$

Although the rotational stiffness coefficient was derived through an experiment to provide the information about the float. A 50.6 g brass weight was placed 6.7 cm away from the center of the float along one of the center lines. The angle between the float at its stable equilibrium position and its new angle created by the moment of the brass weight was measured to be 14°. This information was plugged into Equation 4.2 where m is the mass of the weight, g is the gravity constant, x is the moment arm and θ is the change in the angle of the float. The rotational stiffness coefficient was determined to be 0.13611 N-m/rad.

$$k_r = \frac{mgx}{\theta} \tag{4.2}$$

In order to determine the rotational damping coefficient, the motion of the float when acted upon by a force needed to be derived. A Moto G6 Play android phone with the HyperIMU app was used to find this data. The app records data from the accelerometer, gyroscope and rotation vector from within the phone. To determine the rotational damping ratio, the phone was centered on top of the float and a small force was applied to the edge of the float which caused a rotational motion of the float. The data recorded from the y direction of the gyroscope was plotted and the values of the first two peaks were recorded as seen in Figure 19. The values were recorded as $\delta_1 = 0.216309$ and $\delta_2 = 0.125229$. The damping ratio was calculated using Equation 4.3.

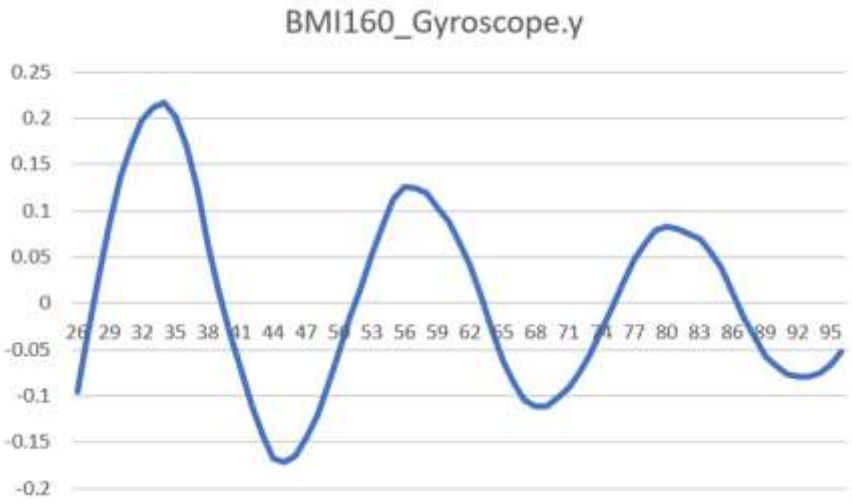

Figure 19: Gyroscope Y Plot to Determine the Rotational Damping Coefficient.

$$d_r = \frac{1}{2\pi} \ln\left(\frac{\delta_1}{\delta_2}\right) \tag{4.3}$$

This value was determined to be 0.086988. The damping ratio value was used to find the rotational damping coefficient using Equation 4.4 which includes the rotational damping coefficient d_r, the mass moment of inertia I and the translational stiffness k. The mass moment of inertia was found using the model in Solidworks. The value for the mass moment of inertia in the I_{zz} direction was used and determined to be 0.00556 kg-m^2. The rotational damping coefficient was calculated to be 0.18246 N-m-s/rad.

$$c = d_r * 2\sqrt{I * k} \tag{4.4}$$

The translational damping coefficient was calculated by using the same HyperIMU app. The phone was placed on top of the float and a force was applied to the center of the float which caused the float to move in an up and down motion. The data from the z-direction of the accelerometer was plotted and the first two maximum values were recorded as seen in Figure 20. These values were recorded as δ_1 = 11.06265 and δ_2 = 10.69275. The damping ratio was calculated using Equation 4.5.

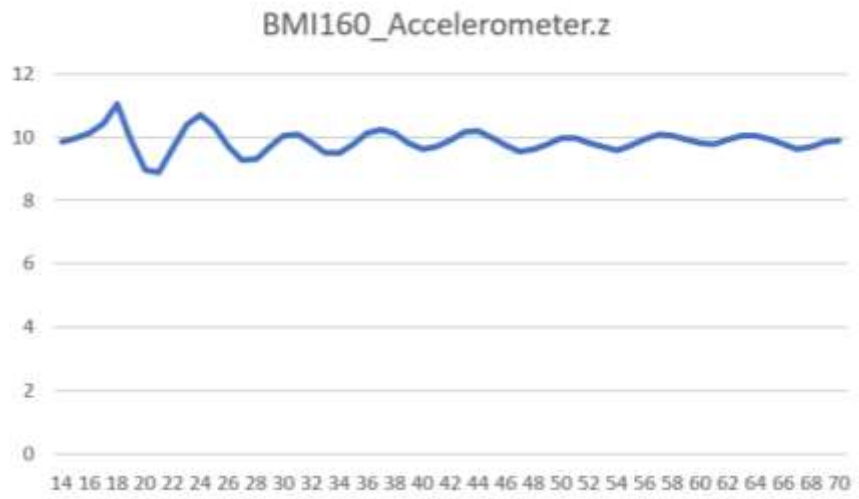

Figure 20: Accelerometer Z Plot to Determine the Translational Damping Coefficient.

$$d_t = \frac{1}{2\pi} \ln\left(\frac{\delta_1}{\delta_2}\right) \tag{4.5}$$

This value was calculated to be 0.005413. Using this damping ratio, the translational damping coefficient was calculated with Equation 4.6 using the translational damping coefficient d_t, the mass m of the float, and the translational stiffness, k. The final value for the translational damping coefficient was determined to be 0.201079 N-s/m.

$$b = d_t * 2\sqrt{m * k} \tag{4.6}$$

After the necessary tests were performed on the float, the float was left in the spa for 72 hours. There was no change in the depth of immersion for the float, indicating that all the seals were waterproof, and the float can be left in open water for an extended period.

Once the values for the coefficients were determined, all the variable values were substituted back into the Mathematica code to solve for the stability of the system. All the variables and their values are shown in Table 6.

Table 6

Variables and Values of the SSF System for Freshwater

Variable	Symbol	Value
Translational stiffness coefficient	k	197.63 N/m
Total mass of the float	m_1	1.74398 kg
Radius of the float	R	0.075 m
Density of the freshwater	ρ	1000 kg/m^3
Gravity constant	g	9.81 m/s^2
Mass moment of inertia	L	0.00556 kg-m^2
Translational damping coefficient	b	0.20108 N-m-s/rad
Rotational damping coefficient	c	0.18246 N-s/m

The Mathematica code accounted for varying amplitudes and frequencies of 0 to 100 with increments of 1 for both variables. This data was imported into a text file and converted into a readable format for Matlab to produce the graph shown in Figure 21.

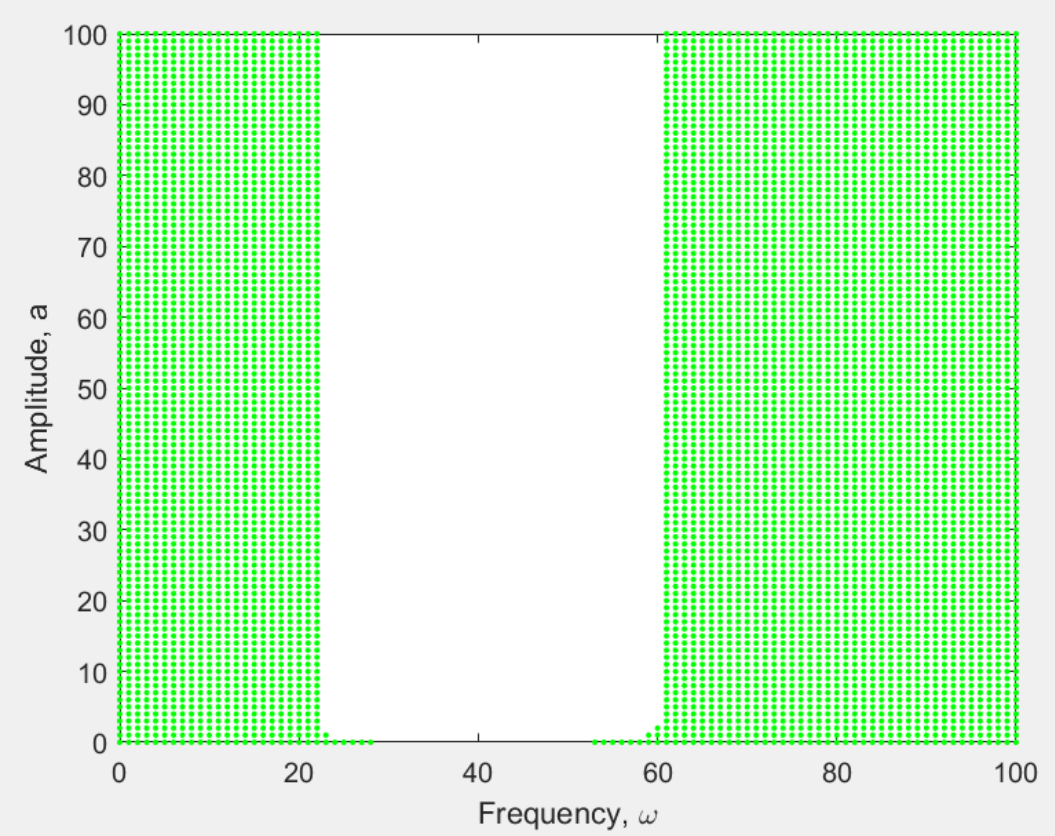

Figure 21: Stability Plot of the System Produced from Matlab

The system is stable for almost all cases when the frequency is between 0 to 22 and 61 to 100 and the amplitude ranges from 0 to 100. It is important to note the additional stable positions when the amplitude is 0 and the frequency ranges from 22 to 28 and 53 to 61, when the amplitude is 1 and the frequency range is 23 and ranges from 59 to 60, and when the amplitude is 2 and the frequency is 60.

The floats will be able to operate properly in the environments where the stability of the system is stable. It is not recommended to place the floats in environments where the system will be unstable as the float will not perform properly and has an increased chance of failing.

Testing the Ultra-Wideband System

For this project, Decawave's MDEK1001 development kit with DMW1001 modules was purchased. It is an anchor and tag based real time location system that can be used with an Android app or a PC with internet access.

For any arrangement of the anchors, the devices should remain vertical and at the same height to help reduce the chance of ultra-wideband communications being disrupted between the other anchors. However, the tag can be in any orientation in the line of sight of 4 of the system's anchors. In the network, one of the anchors is labeled as an initiator and provides the origin of the system and the other anchors are based off its location.

It was determined that one way to properly set up the network using the app was to set up a basic square grid using four of the anchors. These anchors could be auto-positioned using the app on an Android device and the estimated positions should have on average a 10 cm accuracy. However, it is important to note that when all anchors and

tags are set to the same height of 0 m, the system read varying height values for the tag when it was level with the other tags. These values typically ranged from 15-45 cm.

Once a system's 4 base anchors were established, additional anchors could be added at various distances in relation to the four original anchors. The manual for the tracker system states that a combination of 12 anchors and tags can be used on a single network. However, initial tests to add additional trackers beyond the 12 proved to be successful.

The system could also be set up by manually entering the locations of each of the anchors instead of using the auto-position option in the software as in introduces some minor inaccuracies in anchors' locations. This method is the only way that can be done when using a listener connected to the internet.

The tags can be switched on at any time and can be picked up by the system once device has turned on its ultra-wideband and Bluetooth communications. The tags communicate with the software on the app using a Bluetooth connection. An initial test with the phone was set up to test if the system would work as expected. The 4 base anchors were placed in a 1 m square and the tag was moved within the range of the anchors. Figure 22 shows the orientation of the anchors and Figure 23 shows a screenshot of the app displaying the system with the tag at three different locations. The lines in Figure 23 show which anchors the system is using to identify the location of the tag. In this case, since there are only 4 anchors, all the anchors are used to determine its location.

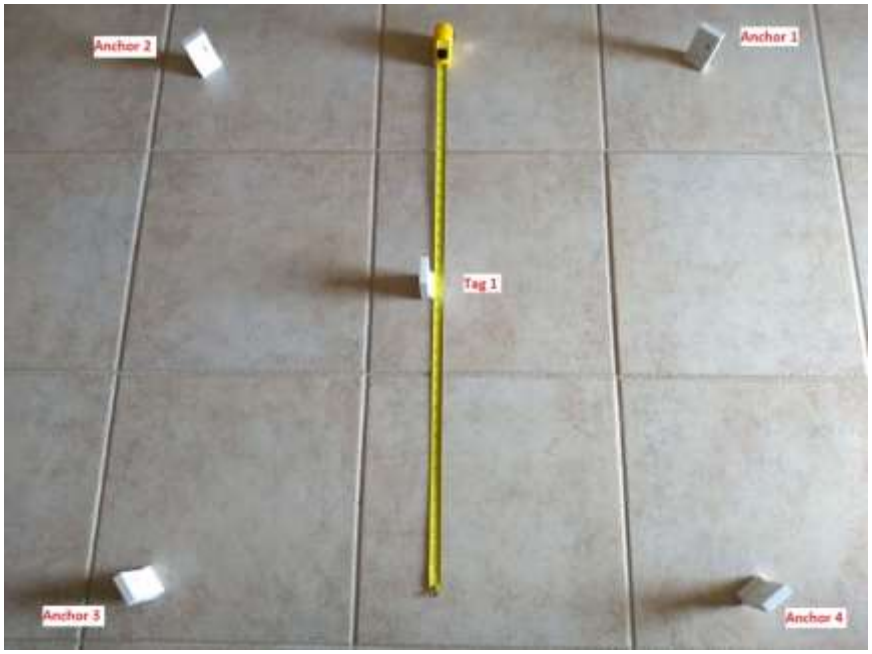

Figure 22: Initial test Orientation of the Anchors and Tag with Labels.

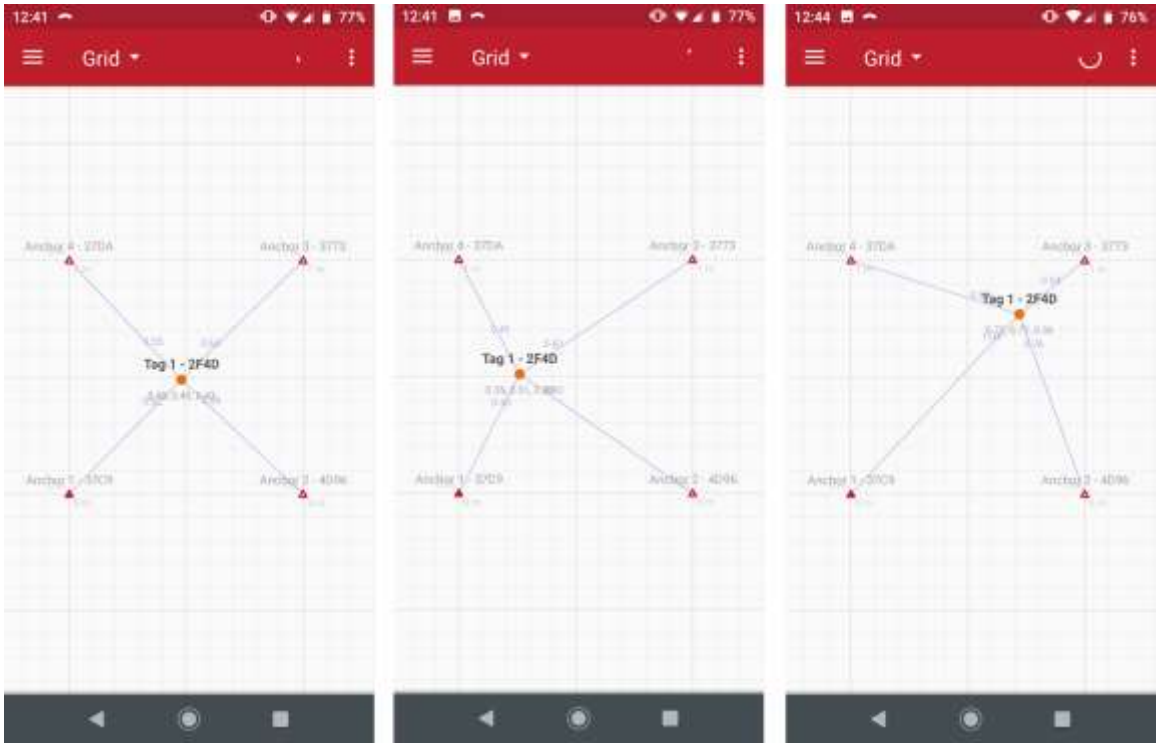

Figure 23: App Display of the System with the Tag at Three Different Locations.

According to the manual, the Android device must be within the range of the Bluetooth communications to record the information of the tracker's location. To test this range, the app was installed on three different available android phones, a Moto G6 Play,

a Moto G4 Plus and a Moto X. The maximum distances the phones were able to pick up data from the system are shown in Table 7. The variations are most likely due to varying manufacture years and different qualities in the phones' Bluetooth modules.

Table 7
Phones and their Maximum Communication Range

Phone	Maximum Range Between Devices
Moto G6 Play	32 m
Moto G4 Plus	47.5 m
Moto X	17.5 m

The tracking system also could be set up using internet of things (IoT) where the system is equipped with a device that connects to the internet to transfer the data to a network that can be reached at any location with an internet connect. In this case, one of the trackers was disassembled and its circuit board was attached to a Raspberry Pi 3 model B+. The tracker circuit board is used as a listener in the locating system to relay the information to a network page via the Raspberry Pi as shown in Figure 24. A box was 3D printed to protect the circuit board and a hole was left of the side for a power source connection. The device can be powered from a micro USB wall charger or power pack.

Figure 24: DMW1001 Module Circuit Board Connected to the Raspberry Pi 3 Model B+.

This setup is useful because when using an android device with an app, the device must be within Bluetooth range. With the listener, the listener device just needs to be within range of the system and have an internet connection. The Raspberry Pi can transfer the information over to the network page that can be accessed anywhere if its address is known.

The initial test of the listener-based system involved placing the anchors in a square at a distance of 1 m from each other. This test was conducted indoors with no obstructions in the way of the anchors or tag as demonstrated in Figure 25. The system did connect to the Raspberry Pi and transfer the data to the network page as shown in Figure 26.

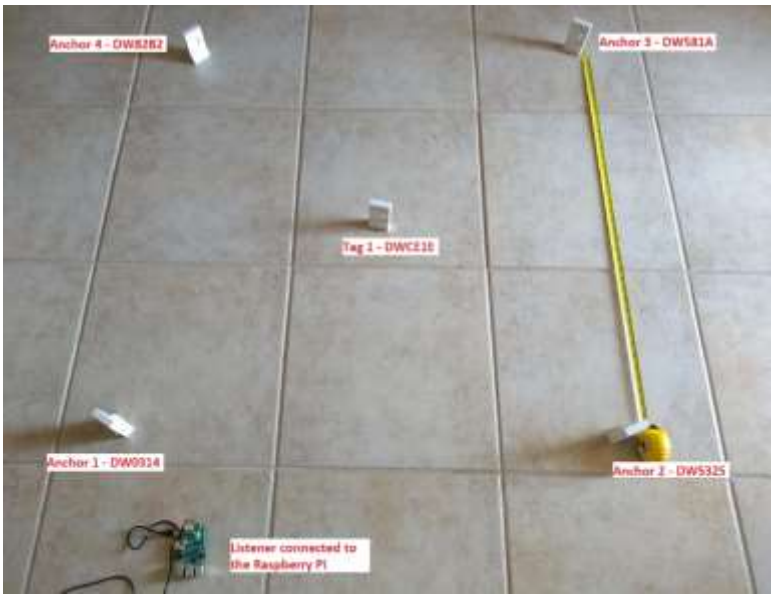

Figure 25: Initial Test of the System Using the Raspberry Pi Indoors with Labels.

Figure 26: Network Page Displaying the Locations of the Anchors and Tag in the Initial Test.

The system was again tested with the anchors set 5 m way from each other. A graphic of the locations of the anchors and tag is shown in Figure 27 and a screenshot of the network page is shown in Figure 28.

Figure 27: Second Test of the System Using the Raspberry Pi Outdoors with Anchors Labeled at 5m Away.

Figure 28: Screenshot of the Network Page Displaying the Locations of the Anchors and Tag in the Second Test.

The last test conducted tested the system with the anchors 13.5 m away from each other. A picture of the locations of the anchors and tag is shown in Figure 29 and a screenshot of the network page is shown in Figure 30.

Figure 29: Third Test of the System Using the Raspberry Pi Outdoors with Anchors Labeled at 13.5 m Away.

Figure 30: Screenshot of the Network Page Displaying the Locations of the Anchors and Tag in the Third Test.

During all the tests, it was observed that the tag can be located by the anchors inside and outside of the square created by the anchors. However, when the tag came within approximately 25 cm of the location of an anchor the tag's location projected on the location grid would shift around the location of the anchor signifying there are contradicting communications within close ranges between modules. It also was observed that there is some lag present between the actual location of the tag and what the software portrays. The nominal and stationary update rates of the system were set up to be 100 ms/10 Hz as it is the fastest update rate that the system can provide. It should be noted

that some of the lag was due to the individual walking around with the tag so the software could be observed as they blocked some of the communications between the tags and anchor.

Another important observation was that the listener needs to be placed at a location higher than the rest of the system. For the tests ran, this height was 50 cm. This ensures that the anchors will be able to communicate with the listening device so their locations can be plotted on the network page.

CHAPTER 5

CONCLUSION

The suction stabilized floats studied in this thesis have been determined to be able to establish a floating platform that remain stable for a range of environments. The inverse slack tank effect created by the fluid within the float creates a moment that restores the float to its equilibrium position at a faster rate than a normal float.

The real time location system has proven itself to be able to set up a localized coordinate frame that can work in a GPS denied environment. The anchors and tags will connect to a network page using a listener attached to a Raspberry Pi that is able to be reached anywhere if the address of the Raspberry Pi is known. The floats are stable in most environments as demonstrated in the stability plot. They also can be used and connected locally using a device with Bluetooth capabilities.

One way that this system can be set up on a large scale is to set up the anchors in a grid of any specified length between 5 m and 50 m. To expand the range of the system, more anchors can be added to increase the size of coordinate grid. The listener for the system can be positioned in between two of the anchors at raised height of at least 50 cm to be able to listen to the system and transport the data over the internet to its network page. The antenna used by the Raspberry Pi should remain in a vertical direction to prevent obstructions that can cause disruptions with the communication from the rest of the system. The anchors can be placed in an environment ahead of time or they can be set up when they are needed.

The system has been reconfigured to provide battery boxes on the trackers so they can easily be turned on and off with a switch. During most of the testing, the trackers had

to be disassembled and the battery had to be removed. This is especially useful for the tag so it can be switched on when needed. It is also possible for the anchors to be connected to a battery that can be charged by the sun so they can remain in place and have a lower chance of running out of battery.

It is important to note that the floats used in this situation can be increased in size to produce a larger platform for the anchors or for other applications. This type of float was originally design for use for offshore wind turbine farms which require a much larger base since the wind turbine extends far above the surface of the platform. They also were design to help prevent large cargo ships remain stable in rough waters. Other applications for the floats could be to create inflatable rafts for people to float on as demonstrated in Chapter 3, helicopter platforms that can be place in a large body of water and stationary platforms for research equipment to float on. The applications are endless for any floating platform that is needed.

REFERENCES

Argos. (n.d.). International Cooperation. Retrieved November 13, 2019.

Argos. (2008). GPS/satellite Telemetry Process. Photograph.

Biran, A. (2003). Ship Hydrostatic and Stability. Oxford: Butterworth-Heinemann (Elsevier).

Fallon Naval Air Station (2001). Combat Survival Evader Locator (CSEL) System. Photograph, Fallon Naval Air Station.

Fryar, C. D., Kruszon-Moran, D., Gu, Q., & Ogden, C. L. (2018). Mean Body Weight, Height, Waist Circumference, and Body Mass Index Among Adults: United States, 1999–2000 Through 2015–2016. *National Health Statistics Reports*, (122), 1–15.

Gibson, Staff Sgt. A. (2011). *Search and Rescue Exercise*. photograph, Langley Air Force Base.

Gode, T. (2015, February 20). Long Baseline Ranging Acoustic Positioning System (Master's Thesis). Virginia Polytechnic Institute and State University.

Holden, C. (2011). Modeling and Control of Parametric Roll Resonance. Norwegian University of Science and Technology.

Nagao, N., Tamaki, K., Kuchiki, T., & Nagao, M. (1995, June). A New Gas Dilution Method for Measuring Body Volume. Retrieved November 25, 2019.

Nayfeh, A. H. (2011). *The Method of Normal Forms* (2nd ed.). Weinheim: Wiley-VCH.
Olive-Drab (n.d.) CSEL System Architecture and Components. Photograph.

R.A. Ibrahim, I. G. (2009, June). Modeling of Ship Roll Dynamics and Its Coupling with Heave and Pitch. Mathematical Problems in Engineering. Detroit, MI, USA: Hindawi Publishing Corporation.

Redkar, S. (2020). *Normal Forms* [Handwritten Notes].

Redkar, S. (2020). *Stability Criteria by Eigenvalues* [Handwritten Notes].

Subramanian, S. C. (2014). Hydrodynamic Study of a Suction Stabilized Float (SSF) (Master's Thesis). Arizona State University.

Symmetry Electronics. (2019). Development Kit for DWM1001 module. Retrieved November 13, 2019.

US Department of Commerce, & National Oceanic and Atmospheric Administration. (2013, March 5). How Does the Temperature of Ocean Water Vary?

APPENDIX A

SSF EQUATION DERIVATION

The SSF equations are based upon the system in figure 6 that was presented in Chapter 2. The system includes a rotational and a vertical component of motion with accompanying damping and stiffness components (Subramanian, 2014, pp. 93). In order to begin the calculations of the model, the following variables must be defined.

The position vector for mass 1:

$$\underline{z} = \begin{bmatrix} z + \alpha cos(\omega t) \\ 0 \end{bmatrix} \quad \text{(A.1)}$$

The velocity vector of mass 1:

$$\underline{\dot{z}} = \begin{bmatrix} \dot{z} - \alpha\omega sin(\omega t) \\ 0 \end{bmatrix} \quad \text{(A.2)}$$

The acceleration vector of mass 1:

$$\underline{\ddot{z}} = \begin{bmatrix} \ddot{z} - \alpha\omega^2 cos(\omega t) \\ 0 \end{bmatrix} \quad \text{(A.3)}$$

The position vector for mass 2:

$$\underline{\varphi} = \begin{bmatrix} \underline{z} - lcos(\varphi) \\ 0 \end{bmatrix} \quad \text{(A.4)}$$

The velocity vector of mass 2:

$$\underline{\dot{\varphi}} = \begin{bmatrix} \underline{\dot{z}} + l\dot{\varphi}sin(\varphi) \\ 0 \end{bmatrix} \quad \text{(A.5)}$$

The potential energy of the system:

$$V = m_2 gl(1 - cos\varphi) + \frac{1}{2}k\underline{z}^2 + \frac{1}{2}k_t\varphi^2 \quad \text{(A.6)}$$

The kinetic energy of the equation:

$$T = \frac{1}{2}m_1\underline{\dot{z}}^2 + \frac{1}{2}m_2(\underline{\dot{z}} + l\dot{\varphi}sin(\varphi))^2 + \frac{1}{2}m_2(l\dot{\varphi}cos(\varphi))^2 \quad \text{(A.7)}$$

To determine the Lagrange equations of the system, the potential energy of the system needed to be subtracted from the kinetic energy of the system, resulting in equation A.8.

$$L = \frac{1}{2}(m_1 + m_2)\underline{\dot{z}}^2 + m_2\underline{\dot{z}}l\dot{\varphi}sin(\varphi) + \frac{1}{2}m_2 l^2\dot{\varphi}^2 - m_2 gl + m_2 glcos(\varphi) - \frac{1}{2}k\underline{z}^2 - \frac{1}{2}k_t\varphi^2 \quad \text{(A.8)}$$

The resulting Lagrange equation for the z direction is derived using equation A.9. The derivative steps are shown in equations A.10 through A.12 for system.

$$\frac{d}{dt}\left[\frac{\partial L}{\partial \dot{z}}\right] - \left[\frac{\partial L}{\partial z}\right] = 0 \quad \text{(A.9)}$$

$$\frac{dL}{\partial \dot{z}} = (m_1 + m_2)\underline{\dot{z}} + m_2 l\dot{\varphi}sin(\varphi) \quad \text{(A.10)}$$

$$\frac{\partial L}{\partial z} = -kz \quad \text{(A.11)}$$

$$\frac{d}{dt}((m_1 + m_2)\underline{\dot{z}} + m_2 l\dot{\varphi}sin(\varphi)) + kz = -b\dot{z} \quad \text{(A.12)}$$

The resulting simplified Lagrange equation for the z direction of the system is presented in equation A.13.

$$(m_1 + m_2)\underline{\ddot{z}} + m_2 l\ddot{\varphi}sin(\varphi) + m_2 l\dot{\varphi}^2 cos(\varphi) + kz + b\dot{z} = 0 \quad \text{(A.13)}$$

The resulting Lagrange equation for the ϕ direction is derived using equation A.14. The derivative steps are shown in equations A.15 through A.17 for system.

$$\frac{d}{dt}\left[\frac{\partial L}{\partial \dot{\varphi}}\right] - \left[\frac{\partial L}{\partial \varphi}\right] = 0 \quad \text{(A.14)}$$

$$\frac{\partial L}{\partial \dot{\varphi}} = m_2 \underline{\dot{z}} l sin(\varphi) + m_2 l^2 \dot{\varphi} \quad \text{(A.15)}$$

$$\frac{\partial L}{\partial \varphi} = m_2 \underline{\dot{z}} l \dot{\varphi} cos(\varphi) - m_2 g l sin(\varphi) - k\varphi \quad \text{(A.16)}$$

$$\frac{d}{dt}(m_2 \underline{\dot{z}} l sin(\varphi) + m_2 l^2 \dot{\varphi}) - m_2 \underline{\dot{z}} l \dot{\varphi} cos(\varphi) + m_2 g l sin(\varphi) + k\varphi = -c\dot{\varphi} \quad \text{(A.17)}$$

The resulting simplified Lagrange equation for the z direction of the system is presented in equation A.18.

$$m_2 l^2 \ddot{\varphi} + c\dot{\varphi} + m_2 g l sin(\varphi) + m_2 \underline{\ddot{z}} l sin(\varphi) = 0 \quad \text{(A.18)}$$

APPENDIX B

MASS DENSITY VALUES FOR SOLIDWORKS

Table 8 shows all the mass densities used in Solidworks to derive masses and the densities used in calculations for the metacentric heights and stability of the floats.

Table 8

Mass Densities for Solidworks

Material	Mass density
PVC Sheet/Vinyl	1300 kg/m^3
Air	1.1 kg/m^3
Freshwater	1000 kg/m^3
Saltwater	1025 kg/m^3
PLA Filament	1290 kg/m^3

APPENDIX C

IN-DEPTH METACENTRIC HEIGHT CALCULATIONS

The calculations for finding the metacentric height of each float involved several intermediate calculations. For all the calculations, the dimensions, volume, mass and center of gravity of the float and the density of the fluid are recorded in Table 9. The numerical calculations in this appendix will relate to the final design of the prototype in freshwater.

Table 9

Dimensions and Details about the SSFs

Measurement	Symbol	Value
Height of top PLA (cm)	h_t	9.5
Length of top PLA (cm)	l_t	16
Height of bottom PLA (cm)	h_b	1
Length of bottom PLA (cm)	l_b	9
Height of top water (cm)	h_{tw}	2
Length of top water (cm)	l_{tw}	15
Height of bottom water (cm)	h_{bw}	8
Length of bottom water (cm)	l_{bw}	8
Height of tracker holder (cm)	h_h	1.5
Length of tracker holder (cm)	l_h	5.8
Width of tracker holder (cm)	w_h	3.4
Mass of float in freshwater (kg)	m_t	1.8818
Center of gravity in freshwater (cm)	cog	6.21
Volume of tracker (cm^3)	v_{tr}	4.381
Mass of tracker (kg)	m_{tr}	0.0568
Volume of top PLA (cm^3)	V_t	2432
Volume of top water (cm^3)	V_{tw}	450
Volume of bottom PLA (cm^3)	V_b	81
Volume of bottom water (cm^3)	V_{bw}	512
Volume of tracker holder (cm^3)	V_h	16.58
Total volume (cm^3)	V_t	2529.6
Freshwater density (kg/m^3)	ρ_{fresh}	1000

The first calculation is determining the density ρ of the float. This is derived using the total mass of the float m and the volume of the float V in Equation C.1. This value was calculated to be 743.92 kg/m^3.

$$\rho_{float} = \frac{m_t}{V_t} \tag{C.1}$$

The specific gravity of the float was determined using the density of the float and the density of freshwater in Equation C.2. The equation produced a result of 0.744.

$$sg = \frac{\rho_{float}}{\rho_{fresh}} \tag{C.2}$$

The immersed volume was then calculated using Equation C.3 by multiplying the specific gravity by the total volume of the float. The immersed volume was determined to be 1881.8 cm^3.

$$V_i = sg * V_t \tag{C.3}$$

Using the immersed volume, the depth of immersion could be calculated. This value is measured from the bottom of the float to the water line. Due to the float's irregular shape, the volume of the smaller bottom section had to be removed and its height added to the depth of immersion value as demonstrated in Equation C.4. The depth of immersion was calculated to be 8.034 cm.

$$h_i = \frac{V_i - V_b}{{l_t}^2} + h_b \quad \text{(C.4)}$$

In order to find the metacentric height for the float, the moment of inertia, I_{yy} must be calculated. This is done by taking a vertical cross section of the float and finding the area sum of the total float in Equation C.5, the height center of the top area in Equation C.6, and the height center of the bottom area in Equation C.7. The values derived from these equations are 161 cm^2, 5.75 cm and 0.5 cm, respectively.

$$A_{sys} = A_t + A_b = h_t * l_t + h_b * l_b \quad \text{(C.5)}$$

$$\overline{y}_t = \frac{h_t}{2} + h_b \quad \text{(C.6)}$$

$$\overline{y}_b = \frac{h_b}{2} \quad \text{(C.7)}$$

After these values were derived, the sum of the areas of the two areas times their height centers, were calculated in Equation C.8 and was calculated to be 878.5 cm^3.

$$A * \overline{y} = \overline{y}_b * A_t + \overline{y}_b * A_b \quad \text{(C.8)}$$

The height of the center of the entire system was then derived as 5.457cm in Equation C.9.

$$\overline{y}_{sys} = \frac{A * \overline{y}}{A_{sys}} \quad \text{(C.9)}$$

Then the moment of inertia was calculated for both sections. The value for the top section was derived in Equation C.10 as 1156.3 cm^4. The value for the bottom section derived in Equation C.11 was determined to be 221.85 cm^4.

$$I_{yy_t} = \frac{1}{12} l_t * {h_t}^3 + A_t * (\overline{y}_t - \overline{y}_{sys})^2 \quad \text{(C.10)}$$

$$I_{yy_b} = \frac{1}{12} l_b * {h_b}^3 + A_b * (\overline{y}_{sys} - \overline{y}_b)^2 \quad \text{(C.11)}$$

These two values can be added together to get the moment of the inertia of the float which was calculated to be 1378.1 cm^4 as shown in Equation C.12.

$$I_{yy} = I_{yy_t} + I_{yy_b} \quad \text{(C.12)}$$

After this, the metacentric height of the float can be calculated using the moment of inertia, immersed volume, the center of gravity of the float and the immersion depth as demonstrated in Equation C.13. The metacentric height was calculated as -1.460 cm. This value was measured from the center of gravity of the float.

$$\overline{GM} = \frac{I_{yy}}{V_i} - (\overline{cog} - \frac{h_i}{2}) \quad \text{(C.13)}$$

Up to this point, the water chamber inside the float has been disregarded. In order to factor it in, the moment of inertia must be calculated for the water inside the float. Equations C.5 through C.12 can be used and the variables changed to deal with the top water and bottom water chamber values instead of the overall float dimensions. The moment of inertia for the water, I_{yy_w} was calculated to be 1654.6 cm^4.

Next, the displaced volume of water needed to be calculated from the inside of the float below the waterline. This was done by taking the volume of the bottom water and

adding the volume of the upper float below the waterline in Equation C.14. The displaced volume was 519.73 cm^3.

$$V_{disp} = V_{bw} + {l_{bw}}^2 * (h_i - h_{bw}) \quad \text{(C.14)}$$

The suction stabilized float metacentric height was calculated using the previously calculated metacentric height, the moment of inertia of the water and the volume of water displaced inside the float. This is demonstrated in Equation C.15 and the resultant metacentric height was calculated to be 1.723 cm.

$$\overline{GM}_{SSF} = \overline{GM} + \frac{I_{yyw}}{V_{disp}} \quad \text{(C.15)}$$

APPENDIX D

MATHEMATICA AND MATLAB CODE

The following section are copies of the codes used in Mathematica and Matlab to create a plot of the stability of the system.

Mathematica Code

```
(*Calculation is for FRESHWATER*)
ClearAll["Global`*"]
<< NormalForm`;
```

```
x = {x1, x2, x3, x4, x5, x6};
u = {u1, u2, u3, u4, u5, u6};
k = 197.63; (*Translational stiffness coefficient (N/m)*)
(*Rotational stiffness coefficient is 1.361 (N-m/rad)*)
m1 = 1.74398; (*Mass 1 - total mass of float in freshwater (kg)*)
R = 0.075; (*Radius of float - half of the water wedge's length (m)*)
[Rho] = 1000; (*Density of freshwater (kg/m^3)*)
g = 9.81; (*Gravity (m/s^2)*)
L = 0.00556; (*Mass moment of inertia of float (kg*m^2) - derived in Solidworks file*)
b = 0.201078764; (*Translational damping coefficient (N-s/m) - derived in Excel file*)
c = 0.18246; (*Rotational damping coefficient (N-s-m/rad) - derived in Excel file*)
```

```
table = Do[{m = {{0, 1, 0, 0, 0, 0}, {-k/m1, -b/m1, (-2/(3 m1)) R^3 [Rho] g, 0, a
        [Omega] - a k/m1, 0}, {0, 0, 0, 1, 0, 0}, {(-2/(3 L)) R^3 [Rho] g, 0, (-14/(15 L))
        R^3 [Rho] g R, - c/L, (-2/3 L) R^3 \[Rho] g a, 0}, {0, 0, 0, 0, 0, 1}, {0, 0, 0, 0, -
        [Omega], 0}};
        MatrixForm[m];
        {s, j} = JordanDecomposition[m]; MatrixForm[s];
        MatrixForm[Chop[N[Inverse[s]], 0.06]];
        N[MatrixForm[j]];
        {y1, y2, y3, y4, y5, y6} = s.{x1, x2, x3, x4, x5, x6};
        rhs = Flatten[j.{x1, x2, x3, x4, x5, x6} + Inverse[s].{0, (2/(3 m1)) a [Omega]^2
        R^3 [Rho] y3 y5 - (2/(3 m1)) a R^3 \[Rho] y4 y6, 0, (2/(3 L)) a R^3 [Rho] y2 y6
        + (1/(3 L)) a^2 R^3 [Rho] y6^2 + (7/(15 L)) a [Omega]^2 R^3 [Rho] R y3 y5, 0,
        0}];
        rhs1 = Simplify[rhs];
        MatrixForm[rhs1];
        $Assumptions =  [Lambda]1 [Lambda]2 [Lambda]3 > 0;
        noiseFree = N[rhs1 /. Thread[[Sigma] -> 0]];
        MatrixForm[noiseFree];
        {newrhs, trans} = NormalFormTransformation[noiseFree, x, u, 2];
        {s1, s2, s3, s4, s5, s6} = {x1 /. trans, x2 /. trans, x3 /. trans, x4 /. trans, x5 /. trans,
        x6 /. trans};
                A1 = newrhs[[1, 1]] /. {u1 -> 2*Pi};
                B1 = newrhs[[2, 1]] /. {u2 -> 2*Pi};
```

```
            C1 = newrhs[[3, 1]] /. {u3 -> 2*Pi};
            D1 = newrhs[[4, 1]] /. {u4 -> 2*Pi};
        h1 = FullSimplify[{s1 /. {u1 -> Exp[A1], u2 -> 0, u3 -> 0, u4 -> 0}}];
        h2 = FullSimplify[{s2 /. {u1 -> 0, u2 -> Exp[B1], u3 -> 0, u4 -> 0}}];
        h3 = FullSimplify[{s3 /. {u1 -> 0, u2 -> 0, u3 -> Exp[C1], u4 -> 0}}];
        h4 = FullSimplify[{s4 /. {u1 -> 0, u2 -> 0, u3 -> 0, u4 -> Exp[D1]}}];
        H14 = {{h1 + h2 + h3 + h4} /. {u5 -> 1, u6 -> 0}};
        Print[TableForm[{{a}, {\[Omega]}, {H14}}]]
, {a, 0, 100, 1}, {\[Omega], 0, 100, 1}];
```

Matlab Code

```
clear, clc      %This code is to plot the data derived from Mathematica.
FID2=fopen('0to100freshwater.txt','r');
AB_data_back=textscan(FID2,'%s');
fclose=(FID2);
AB_data_char=char(AB_data_back{:,:});
Ab_data_complex=str2num(AB_data_char);
B=reshape(Ab_data_complex,[3,11308]);
C=B'
for i=1:11308
if(real(C(i,3))<=0)
        plot(real(C(i,1)),real(C(i,2)),'g.')
        axis([0 100 0 100])
        xlabel('Amplitude, a')
        ylabel('Frequency, \omega')
end
hold on
end
```