

2015

University of North Carolina Wilmington
Master of Science in
Computer Science and Information Systems
Proceedings

<https://csbapp.uncw.edu/mscsis>

INVESTIGATION OF USING AN UNMANNED AERIEAL VEHICLE
TO ASSIST IN HELPING DISTRESSED SWIMMER

Jason Gillikin

A Capstone Project Submitted to the
University of North Carolina Wilmington in Partial Fulfillment
of the Requirements for the Degree of
Master of Science

Department of Computer Science
Department of Information Systems and Operations Management

University of North Carolina Wilmington

2016

Approved by
Advisory Committee

Dr. Gene Tagliarini

Dr. Jeffrey Cummings

Dr. Thomas Janicki, Chair

Accepted By

Dean, Graduate School

Table of Contents

	Page
Chapter 1: Introduction	5
Chapter 2: Background	7
A. Examples of Modern Robots	8
B. Characteristics of Modern Robots	11
Chapter 3: Technologies Employed	13
A. Software	13
B. Hardware	13
Chapter 4: Drone Operations and Lessons Learned	16
A. Safety and things not to do	16
B. Tuning	19
C. RC controls	20
D. Flight Modes	21
E. Arming and Disarming	22
F. Lessons Learned	23
Chapter 5: Using Computer Vision to Distinguish Between Water and People	24
A. Detection Techniques	24
B. Future Research Related to Computer Vision	28
Chapter 6: Investigate Video Feed to Laptop	29
Chapter 7: Mapping GPS Coordinates to Pixels	30
Chapter 8: MavLink Protocol	34
Chapter 9: Investigate Dropping a Life Jacket	37
Chapter 10: Summary of Work Completed	39
Chapter 11: Future Research	40
References	41
Figures	
1. Movements of the Hyper Redundant Robot	9
2. A Typical QuadraCopter	10
3. Amoeba Qualities of a Metamorphic Robot	11
4. Gaussian Blur Matrix	24
5. People in the Water	26
6. People Identified in the Water (ideal)	27
7. People Identified in the Water (non Ideal)	27
8. Morphological Gradient Transformation	28
9. Mapping Pixels into Displacement and Bearing	30
10. Lifeguard Application (proof of concept)	32
11. Lifeguard Application (simulated screen)	33
12. MavLink Packet	34
13. Base Station GUI	35
14. MavLink Command	36
15. 555 Timer as PWM Control Circuit	37
16. Transistor for Hot Knife Circuit	38

ABSTRACT

Investigation of Using an Unmanned Aerial Vehicle for Helping Distressed Swimmers. Gillikin, Jason, 2016. Capstone Paper, University of North Carolina Wilmington.

From 2005-2009, there were an average of 3,533 fatal unintentional drowning (non-boating related) annually in the United States. The study includes Utilizing Computer Vision, GPS, and Optical Calculations. This paper will discuss the study of the implementation of unmanned aerial vehicles to assist lifeguards in potentially identified distressed swimmer. Topics include, identifying distressed swimmers, programming an Unmanned Aerial Vehicle, problems encountered, and potential future enhancements.

LIST OF FIGURES

Figure	Page
1. Movements of the Hyper Redundant Robot	9
2. A Typical QuadraCopter	10
3. Amoeba qualities of a metamorphic robot	11
4. Gaussian Blur Matrix	24
5. People in the Water	26
6. People Identified in the Water (ideal)	27
7. People Identified in the Water (non Ideal)	27
8. Morphological Gradient Transformation	28
9. Mapping Pixels into Displacement and Bearing	30
10. Lifeguard application (proof of concept)	32
11. Lifeguard application (simulated screen)	33
12. MavLink Packet	34
13. Base Station GUI	35
14. MavLink Command	36
15. 555 timer as PWM control circuit	37
16. Transistor for hot knife circuit	38

CHAPTER 1: INTRODUCTION

This capstone project involved the study of the potential use of an Unmanned Aerial Vehicle (UAV) or drone that can assist in the rescue of distressed swimmers before their situation worsens. The capstone involved the combination of the following hardware components: hexacopter platform, cameras, GPS, computer vision and a laptop. It also involved writing the specific software necessary to operate the UAV.

The initial goal of the system was to aid lifeguards in rescuing swimmers by dropping a floatation device and then returning to the platform. The system operates by the base station's camera monitoring the surf for swimmers. Once the lifeguard identifies a distressed swimmer the drone flies to the GPS coordinates and drops a floatation device. This allows assistance to reach the person faster and possibly save lives. The importance of the project comes from the Center for Disease Control(CDC) statistics:

“Drowning ranks fifth among the leading causes of unintentional injury death in the United States...From 2005-2009, there were an average of 3,533 fatal unintentional drowning (non-boating related) annually in the United States, about ten deaths per day. An additional 347 people died each year from drowning in boating-related incidents...About one in five people who die from drowning are children 14 and younger. For every child who dies from drowning, another five receive emergency department care for nonfatal submersion injuries... More than 50% of drowning victims treated in emergency departments (EDs) require hospitalization or transfer for further care (compared with a hospitalization rate of about 6% for all unintentional injuries). These nonfatal drowning injuries can

cause severe brain damage that may result in long-term disabilities such as memory problems, learning disabilities, and permanent loss of basic functioning (e.g., permanent vegetative state). ”[12]

CHAPTER 2: BACKGROUND

Unmanned Aerial Vehicles (UAV) or drones as they are more commonly known are a growing topic in our daily lives. Commonly the news media portray UAVs as negative in content in personal use or related to military applications. However, drones and the autonomous revolution that is just on the event horizon will change our everyday lives in ways we wouldn't expect. The immediate future will probably not be full of large humanoid robots depicted in the Jetsons and Lost in Space. Rather, they will be small specifically designed automatons that work in a multi-agent system or swarm to accomplish a task. Currently there are several household robots that handle mundane chores we do. There are robots for cleaning floors, windows, gutters, and mowing lawns[1].

The word robot is derived from the Czech word "robota" which means "slave". From the root of the word, it can be inferred that any mechanical invention that relieves a person from work can be referred to as a robot. The latter half of the 20th century produced the programmable microprocessor that bridged the gap between machines and what we today refer to as robots. The first industrial robot was Devol's Unimate robotic arm[3]. This robotic arm was used to transfer hot pieces of metal and spot weld in a General Motors plant. The Unimate robot was soon adapted in all major automobile manufactures. After the widespread use of a robotic arm, the next step in the evolution of the modern robot was to attempt to create a humanoid robot. This is an ongoing project that has made lots of headway since 1973 with the invention of WABOT-1 and 2 by Ichiro Kato at Tokyo University[4]. These robots had full limb control, were able to walk, grip objects, talk, play piano, and were estimated to have the intelligence of an 18 month old child.

A. Examples of Modern Robots

Soon after the WABOT project ended, Honda started creating their line of Humanoids which has culminated in the ASIMO series[5]. The ASIMO has become the benchmark for humanoid robots. Capable of running at 9km/s (forwards and backwards) navigating over uneven surfaces, and even hopping on one leg. The sensors allow it to pick up things as fragile as paper cups and pour liquid without spilling the contents [5]. The Human form allows for many advantages over other animals. The primary advantage is the opposable thumb that allows for gripping objects. However, nature allows other animals to excel in vastly different areas. For example, birds have the power of flight, octopi have multiple limbs, and serpents have the ability to writhe and contort their bodies to get into tight spaces. Since nature saw fit to provide these creatures with specialized adaptations to thrive in nature, why should robots be created solely on the model of man? In fact many of these biologically inspired advantageous features in nature have been adapted to robots. These specialized features become invaluable because most applications that robots are designed for are inherently difficult for the human form such as: search and rescue in collapsed buildings, space exploration, and delicate surgeries.

Another example is the Hyper Redundant robot which is modeled after a snake. It consists of a large number of joints that have multiple degrees of freedom. These features give the robot the ability to writhe and move freely[6]. A Hyper Redundant robot has the ability to rotationally manipulate objects by wrapping around them and either pulling or pushing (see figure 1a). This robot has the ability to squeeze into crevasses by contorting its frame around obstacles (see figure 1b)[6]. The many degrees of freedom also allow the robot to efficiently swim using the peristaltic abilities of its tail. The Hyper Redundant style of robot does have some challenges associated with it. The wiring of all the nodes is an issue. With so many

modules it is necessary to multiplex the wiring so each actuator doesn't require its own dedicated connection to the controller. Another problem associated with the writhing motion is the non-linear movement caused by the motors. This can be reduced by dampening.

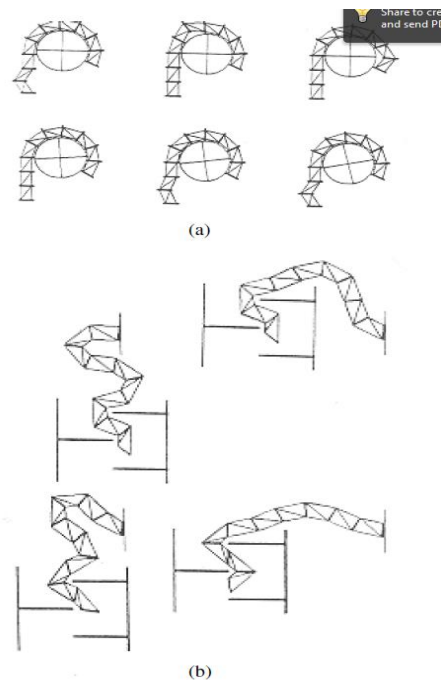


Figure 1. Movements of the Hyper Redundant Robot. (a) rotational manipulation. (b) writhing around objects.[6]

Micro Air Vehicles (MAVs) are small autonomous aerial vehicles with an enormous potential in multiple applications like mapping, reconnaissance, and surveillance. These Robots are often less than 30 cm in size and some are even 10 cm in wingspan. The size and weight of these vehicles gives them an enormous advantage in stealth over typical aerial vehicles. A typical design for the MAV is the quadcopter (see figure 2). Having 4 rotors, the quadcopter has superior stability and is agile in all 6 degrees of freedom. These MAVs usually have on board visual systems and can either be autonomous or piloted remotely. Since the quadcopter has the ability to hover as compared to a typical fixed wing UAV, it can effectively maneuver around obstacles and tactically hold its position.



Figure 2. A typical Quadcopter [7]

Other MAVs such as the DelFly is 10cm in wingspan and weighs only 3 grams. The DelFly can fly at 5km/hour however due to the size of its battery it can only be operated for 3 minutes. [7] Since the DelFly is not a rotary craft, it can fly faster and more efficiently. Also, it is modeled after the humming bird and can fly either forward or backward. [7]

The Metamorphic robotic system is a series of independent robots that have the ability to connect to one another to form a superstructure. [6] The connection can either be through a physical docking mechanism or through electromagnets. While connected, these robots have the ability to recharge one another and also transfer data in a secure manner. The robots are best viewed as one large collection or swarm rather than each individual robot. There are properties that differentiate metamorphic robots from other modular robots. A list of these properties for metamorphic robots follows.

- 1. All modules have the same physical structure, including the same hardware for computation and communication.*
- 2. The mechanical structure of the modules possesses symmetries that ensure that they fill planar and spatial regions with minimal gaps and form a lattice.*

3. The modules must each be kinematically sufficient with respect to the task of locomotion; that is, they must have sufficient degrees of freedom to be able to “walk” over adjacent modules, so that they can reconfigure without outside help.

4. Modules must adhere to adjacent modules, for example, there must be electromechanical or electromagnetic connectors between modules, which can carry load. This causes the collection of modules to act as a single physical object. [6]

Since the individual robots form a lattice and are autonomous, the size of the swarm can theoretically be infinite. As the size grows, the mechanical organism can take on amoeba like qualities. Including the ability to change into any form it wants to and engulf foreign objects [6] (see figure 3).

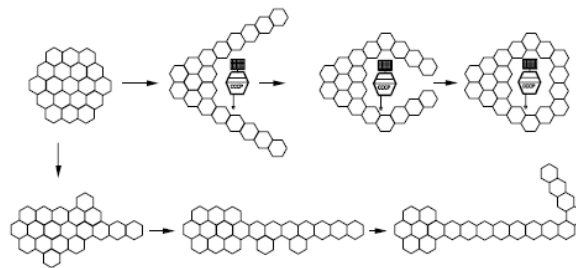


Figure 3. Amoeba qualities of a metamorphic robot[6].

B. Characteristics of Modern Robots

Robotic sensors can vary depending on the application and environment.[9] Most robots have: bump sensors which let them know when physical contact has been made, infrared sensors which can sense heat and also distance from an obstacle, sometimes the IR sensors are replaced by laser ranging sensors to provide more accurate readings.[10] In between the two is the ultrasonic sensor that is capable of accurate readings at a distance and has very little interference.[10] Radio transmitters/receivers are used to get information from other robots,

accelerometers are used to measure movement on all 6 axes allowing for full control of motion.[8] GPS is the preferred method of navigation overall, because it covers most of the planet, however. However, in conditions where GPS cannot be utilized (i.e. tunnels, enclosed environments, and canyons) an optical system is required for navigation. [8]

Object identification is an integral part of this project and comes with special challenges that need to be. The problem of computer vision is compounded because the background is not constant. Since the objects of interest are people in the water, it is difficult to eliminate the water because in most cases there are waves that create a moving background. Other issues include “sun glint” caused by sun reflected off the surface of the water and by white caps caused when the wave breaks. UAV’s have been used to track marine mammals in the open ocean. The algorithm used to identify the whales converts the target image to HSV and creates an HS histogram and compares that to the HS Histogram of the ocean.[14]

CHAPTER 3: TECHNOLOGIES USED

In order to make this project possible, several pieces of technology were implemented on both the software side (vision, UAV control, and Interface) and the hardware side (drone and laptop). Following is an outline of the different technologies implemented and their challenges.

A. Software

1. Vision

2. The computer vision aspect of the problem was the most challenging. The program implements several methods to identify people versus water. The solution to this software problem will be discussed later.

3. UAV Control

The UAV control uses an open source protocol called MavLink which has methods `WAYPOINT_REQUEST_LIST()`, `WAYPOINT_COUNT()`, and `WAYPOINT_CLEAR_ALL()`. These methods are used to plot a course to the distressed swimmer.

4. Interface

- i. The Software interface will be written in C# using Visual Studio and can be deployed on a laptop.

B. Hardware

i) Drone

The hexacopter UAV studied consists of a light weight frame, 6 motors and propellers, a 2200mAh battery. The UAV is controlled by an Atmel 328 Microcontroller which is the heart of the Arduino board. A 915 MHz telemetry kit is used for communicating with the UAV while in the air. The on board camera is was

going to be relayed with a 1.3 GHz transmitter back to the base station where image processing would have taken place, but due to scope it is not used. Other peripherals include a 3 axis accelerometer (IMU), GPS, sonar, and barometer. Because of its accuracy sonar can be utilized for measuring altitude close to the ground, whereas high altitude measurements use the barometer for its efficiency. This specific UAV comes compiled with the ArduCopter software that allows for both autonomous navigation and remote control (RC). There are many safety measures in this software that help prevent fly away, crashes, and help improve flight control. The base station consists of a video camera, 1.3GHz receiver and a laptop.

(a) GPS

Will utilize the standard GPS that comes with the drone.

(b) Camera

A CMOS and CCD Camera are both implemented in this project. CCD or Charge-Coupled Detector cameras are more expensive and use up to 100 times more battery power than the CMOS or Complementary Metal Oxide Semiconductor. However, the CCD offers better light sensitivity and noise reduction than the CMOS[13]. For purposes of time frame for this scope, the camera is not used

(c) Sonar

Used for determining the altitude when close to the ground (<10ft) because the barometer is not very accurate. For purposes of time frame for this scope, the Sonar is not used.

(d) Telemetry

1. Remote

Used for Human control of drone

2. 915MHz transmitter

Used for sending new GPS coordinates and communicating with
MavLink protocol

- ii) Tablet

Since the program is written in C# any windows tablet or laptop maybe used as long as it has a USB port

- iii) Video Transmitter

The 1.3GHz transmitter and receiver was recommended by the vendor to have the range needed for this project. However, for purposes of time frame for this scope, the camera is not used because no video from the drone is analyzed.

CHAPTER 4: DRONE OPERATIONS AND LEASSONS LEARNED

This section will detail the basic operation of the drone and how I have learned to safely fly and calibrate the drone, both by reading community forum [18] and from the mistakes I have made.

A. Safety and things NOT to do:

In order to properly operate the drone, I researched the blogs on the community website [18] The site detailed mistakes that enabled me to get up to speed on safety issues quicker.

Following are a list of safety factors I tried to employ:

- 1) Multicopters are essentially flying lawnmowers, and ArduCopter is an experimental aircraft running community-created code.
- 2) Crashes can happen, because of pilot error or hardware or software malfunction.
- 3) If you are flying anywhere near other people, you are putting them at risk!
- 4) Maintain a safe distance away from people.
- 5) Safe distance - You have to make your own determination of what is a "safe distance" from people and property.
- 6) At full power, an average sized multi-copter can exceed 20 mph (32 km/h), can ascend to hundreds of feet and travel easily more than a mile in distance before running out of battery.
- 7) Remove the battery cable to the power distribution board until you're ready to fly, and removing props while you're testing motors.
- 8) When testing or flying any of the navigation modes (using GPS) check that your home position on the Mission Planner is in fact correct, sometimes GPS's do not report the

home position accurately, reboot if not accurate and wait for 8 or more satellites (not just 3D lock) and check again.

- 9) Keep it close to you, i.e. no further than 30ft (10m) from you. If the copter gets further than about 100ft (30m) it starts getting difficult to be able to maintain orientation and can easily crash.
- 10) It is much better to simply descend and land rather than have an orientation-induced crash or worse still a fly away. This kind of mishap often happens when the copter is commanded to tilt back towards the pilot but has rotated in the meantime and is so far away that orientation is lost. Result: the copter flies further away and crashes or is lost.
- 11) Always have Stabilize mode as one of your 3 options. If your copter gets caught up in the wind for some reason it can be difficult to fly. High winds can spin the copter around to the point you can't make forward progress. The higher you are, the more likely high winds are a problem. Stabilize mode saves your copter in this situation.
- 12) Avoid flying at high speed or high altitude until you have built confidence in both manual and automatic modes. It is not uncommon to lose control by ascending above tree level you may catch the wind once clear of surrounding obstacles, may lose radio signal, may be unable to discern the "front" and become disoriented, or may lose sight of the copter all together.
- 13) Keep low (7-10 ft) and slow (walking pace) in a very large open area, until you have tested and can confidently and consistently use SIMPLE, RTL, geo-fencing and other failsafe features.
- 14) Avoid sudden or extreme control deflections (don't yank the sticks). If the copter is properly calibrated and balanced it should require only small stick inputs to change

direction and should be more or less stable on the horizontal plane without any control inputs.

- 15) If you are "fighting" the copter, land and calibrate/test - something is not right.
- 16) Avoid doing a soft reboot of your APM with the flight battery attached.
- 17) In some situations, the timer outputs can get corrupted causing the motors to spin up.
- 18) If you can, avoid having the flight battery attached at all with the USB cable attached. It is best if possible to, avoid having the flight battery attached at all unless you're flying.
- 19) Assume the motors are armed. Check with a short throttle pulse. Don't pick up the model and the radio at the same time, you may bump the throttle.
- 20) Always check that the copter is stabilizing correctly by holding it securely from the top and giving it some power. You should not feel a circular wobble. This means the + or X are set incorrectly. Taking off in this condition only ends in a crash.
- 21) Avoid flying high with a low battery. You may end up crashing down in a violent spin. Make sure your battery can't fall out. Use a piece of thin foam rubber under it to make it sticky, or use Velcro in addition to the Velcro strap.
- 22) If you suddenly lose yaw orientation in stabilize, try only flying forward and using yaw to steer like a car.
- 23) The drone is not a toy, there have been several injuries where people have been hit by the drone, or the drone motors have engaged because the drone was not in "disarm" mode.
- 24) Do not fly in close quarters, the drone can be damaged by walls or worse, someone could get hurt.

25) Use Loctite on the propeller nuts, the nuts that hold the prop to the motor have a tendency to spin off if not secured with Loctite. One incident reports that the copter lost the prop at around 50 feet and crashed into the water completely destroying the Copter.

26) FAA limits the Altitude of drone operation to 400 feet in a line of sight orientation. Also, the new regulations will require a drone registration and inspection.

In order to gain experience in a safe manner for myself and others, I plan to start by operating the drone at low altitudes (under 20 feet) and taking advantage of the preprogrammed flight modes mentioned latter.

B. Tuning

There are 3 different rates (P,D,I) that effect the handling and operation of the drone. These variables have to be calibrated for each drone in each flight condition. The ideal settings differ with temperature, humidity, and pressure. It is also suggested that a tuning jig be constructed.

An explanation of P,D, and I are as follows:

RATE-P

- Too much rate P will oscillate quickly, and cause to copter to sound angry under stick input, bouncing rather than smoothly following your inputs. It also shakes more at full throttle and under hard turning.
- Not enough you won't feel like you have full control. It feels sloppy and be very easy to over correct with your inputs. It feels delayed.
- Perfect is where it feels locked in, stiff in the air, but not shaky. (although if I'm sport flying I turn it up a bit for maximum 'wang' and just tolerate the slight oscillations)

RATE_I

- Too much rate_I oscillates if you get high enough (a much slower oscillation than a rate_P shake). But quite a long while before it oscillates it will have other detrimental effects on flight performance, like a sluggish feeling or a tendency to flip over on take-off. This is why I suggest tuning this in flight rather than in your hand/jig.
- Not enough causes the copter to get pushed by a constant wind, then it fights back using just P. It doesn't hold a very firm angle during forward flight and needs more correction. This will not be as smooth as it could be in either case.
- The perfect amount will cause the copter to lean gracefully into a constant wind, but also allow you to set a lean angle and stay there as you fly about. As you modify rate_I pay close attention to the feel of the copter. You are not looking to create oscillations here at all, and you should notice a strange 'feel' long before this point.

RATE_D

- Too much rate_D will oscillate very fast, you will see a twitch forming then a fast buzzing oscillation
- Not enough rate_D simply means you can't dial enough rate_P and so you will suffer the effects of having rate_P too low.
- A perfect rate_D helps fight the wind and follow your sticks as its fast to react, but also allows you to reach a maximum rate_P level for you frame, thus giving better control.

C. RC Controls:

At first glance there are a lot of buttons and sticks on the remote. However, after reading the blogs, very few settings on the remote need to be changed. The remote has two basic

functions which are controlling the UAV and setting the flight mode. Control is accomplished by the sticks the left stick controls the throttle and the orientation, while the right stick controls the movement left, right, forward, and backward. The toggle switches are used to select the flight modes

D. Flight Modes

Acro: The mode where the pilot has the most control over the copter. The stick movements are directly translated into motor commands with no buffering or limits. This setting is for experienced users and not for beginners as it is easy to crash.

Stabilize: This mode is recommended for beginners because the UAV automatically stabilizes itself when not actively being piloted. If you get in trouble, all you have to do is release the sticks and the UAV stabilizes itself.

Alt Hold: This mode maintains a constant altitude using data from the barometer and from the sonar at <26 feet. To adjust the altitude, simply throttle up or down. Since this mode keeps an altitude, it is impossible to land while using. In order to land and disarm the motors, you must switch out of altitude mode.

Simple: This mode sets the orientation of the controls to your orientation. It is the easiest to fly as long as you stay facing the same way. For example, if you want the UAV to move forward from your perspective, push up on the right stick. This movement is completely independent of the orientation of the drone, because it uses the on board compass to determine the correct motor controls. This mode is often compiled with other modes such as Stabilize with Simple mode.

Auto: This mode makes the drone follow the preloaded mission plan. This is the radio mode that is used most often for the present application.

RTL: Return to Launch mode makes the UAV fly back to wherever it was most recently armed.

Guided: Same as RTL, the landing point is pre set

Loiter: Allows the UAV to fly to maintain its current position in all three dimensions.

Position: Same as Loiter except you have control over the throttle

Circle: The UAV circles around a coordinate at a given radius

Follow Me: with a GPS dongle attached to your computer, the UAV follows your position at a given altitude.

GeoFence: An area can be specified (by polygon or circle) and if the drone flies out of that area, it automatically returns to the center coordinate. This mode is used as a safety feature to prevent fly away.

E. Arming and Disarming

In order to safely operate the drone the motors can be armed and disarmed. The drone should be disarmed anytime it is not flying. An analogous situation would be disconnecting the power before working on electrical equipment. Accidents do happen, and this drone can cause serious injury. In order to arm the UAV, hold the left stick down and left. In order to disarm, hold it down and right.

F. Lessons learned

After flying the hexacopter for several hours I have realized things don't always go as planned. It is very hard to judge distance when the hexacopter is in the air. Several times I have crashed into trees, buildings, and have had several close calls. Also, even when flying with a flight plan it is possible for the hexacopter to fly away and be hard to find when it crashes. The cause of the crash was undetermined, but I have two hypotheses. The first is that the hexacopter lost GPS communication and simply was unable to find where it was supposed to go. The second is that due to mechanical failure, the hexacopter simply couldn't fly in that direction, this is due to a failing motor.

CHAPTER 5: USING COMPUTER VISION TO DISTINGUISH BETWEEN WATER AND PEOPLE

As this project involves flying over the coastal shore, some special concerns about image processing caused by waves and reflection on the ocean arise. The most important is that the ocean is not a static entity. Some days it can be rough and the others calm. These rapidly changing conditions cause problems with computer vision when trying to discern a person from the ocean. Being able to detect swimmers in the water can be a valuable asset. Since people are not all the same size or color and the weather conditions are variable, this problem can be tough to solve.

A. Detection Techniques

I decided to try three techniques to solve these problems: Canny Edge detection filter, Morphological Transformations, and the Hough Circle algorithm. Canny Edge detection is used to identify well defined edges in an image. There are three main steps to the Canny algorithm (noise detection, finding the intensity gradient, and thresholding). During the Noise Reduction phase the image is convolved with a 5X5 matrix (fig. 4) to create a Gaussian blur. This process allows noise to be eliminated.

$$\mathbf{B} = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} * \mathbf{A}.$$

Figure 4. Gaussian Blur Matrix

Four filters are used to find intensity gradients for the image. The filters get gradients in the horizontal, vertical, and both diagonal directions. The thresholding used in the Canny algorithm is referred to as “Thresholding by Hysteresis”. This requires both upper and lower threshold parameters. The upper value marks all the edges that are definitely edges. The lower value is used along with the gradient information to mark faint lines.

Hough transformations are used to detect shapes and lines in images. For the purpose of this research I have concentrated on the Hough Circle algorithm. In order to reduce computational complexity, it’s recommended that the diameters of the desired circles be known. If the diameter is not known the run time is $O(n^3)$. The algorithm begins by creating bins for each pixel (i, j) and setting them to zero. For each edge pixel use the equation $(i^2-a)^2+(j-b)^2=r^2$ where “a” is the center of the circle. Then find all values of “b” that satisfy the equation. Once this is completed, a search for local maximums is executed and these are the circles.

Morphological transformations are a handy tool when trying to increase the contrast between features. The two most common transformations are Erosion and Dilation. The transformation matrix can be any shape. Circles, crosses, and squares are the most common shapes. The matrix also contains one cell that is designated as the anchor point. The anchor point is where the final value is stored in the new image. Erosion puts the minimum intensity value for the matrix in the anchor point, making the darker areas larger. Dilation puts the maximum intensity value for the matrix in the anchor point, making the brighter areas larger. A gradient transformation can also be performed by subtracting the dilation from the erosion.

A picture of people in the ocean taken from an elevated point of view was required for testing. After research, a good candidate was chosen (fig.5). An application was created that

implemented the Canny Edge detection and Hough Circles algorithm in succession. The threshold values can be manipulated to find the highest resolution between people in the water without eliminating people from identification. A second method of identification was also attempted using a gradient morphological transformation.



Figure 5. People in the Water (Notice People's heads look like circles)

After trial and error, the ideal threshold values were 364 and 222 for the upper and lower thresholds respectively. The resulting image can be seen in figure 6, the red circles are used to identify people in water. The same threshold values were applied to different images and the results (fig. 7) were less than satisfactory.



Figure 6. People Identified in the Water (ideal)

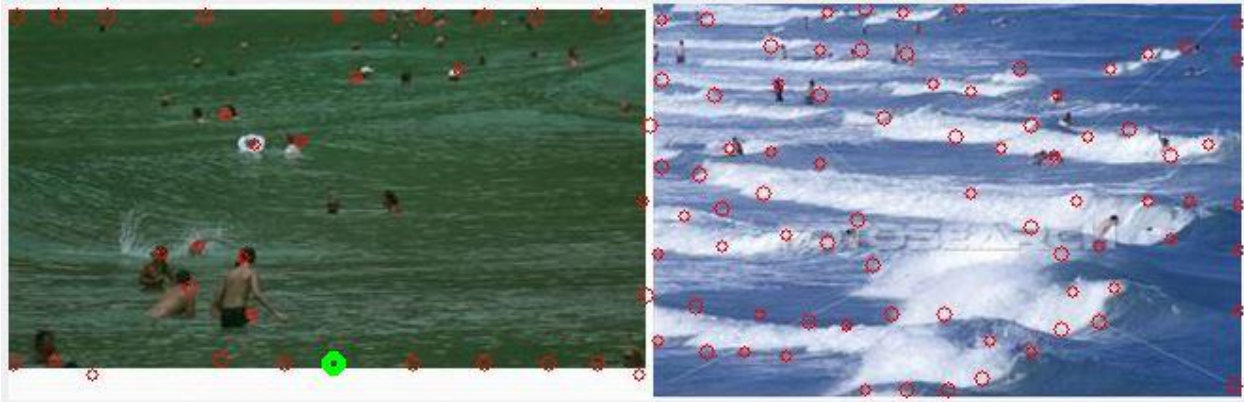


Figure 7. People Identified in the Water (non ideal)

The Morphological Gradient transformation was performed 5 times with a threshold window of 120-140. The results (figure 8) show a good contrast between people and water.

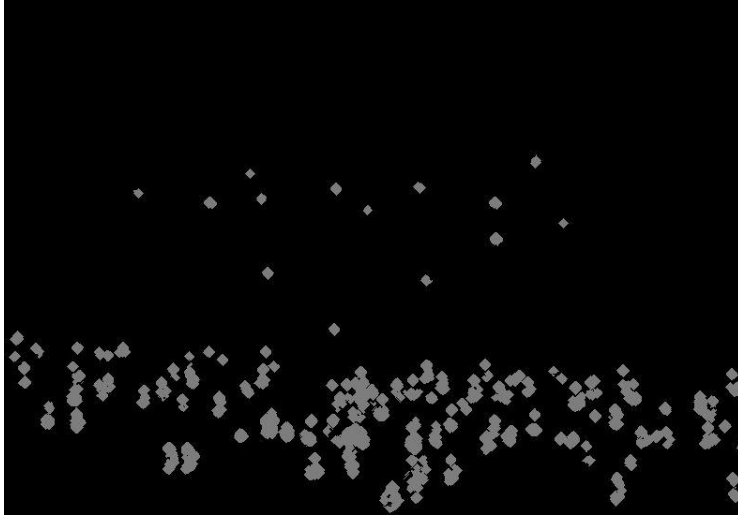


Figure 8. Morphological Gradient Transformation

B. Future Research Related to Computer Vision

Since the weather and lighting conditions cause such a vast change in the parameters needed to detect people. It is not possible to have one fixed number for identification purposes. It is recommended that a more intelligent system be designed to aid in further identification. This will also require more images than are currently available at this time. Using the Adaptive Resonance Theorem (ART) or a Bayesian Classifier to detect people in the water with varying weather conditions would be a route that could prove fruitful.

CHAPTER 6: INVESTIGATED VIDEO FEED TO LAPTOP

The basic architecture consists of a stationary webcam whose image is updated as often as possible. The standard video feed is 30fps; however it takes about 1 second for computer vision function to work, so the feed should be around 1fps. This image is be used to possibly identify the people in the water, but definitely allow the user to deploy the drone to the specified GPS coordinate. This is accomplished by mapping a coordinate grid to the pixels. The real life implementation of this setup would require a separate server and a longer range video transmitter (UHV). For the purposes of this project, the actual video feed system was not be implemented due to the time required to implement it. Instead, still images have be analyzed.

CHAPTER 7: MAPPING GPS COORDINATES TO PIXELS

In order to map GPS coordinates to pixels, I first had to map pixels to a bearing and a displacement. In order to accomplish this, I used some known information from the camera and the environment (focal length, lenses, height of the camera, distance to the bottom of the picture). With these knows, we can use simple geometric formulas to calculate the x and y displacement in feet, and also the bearing.

- Pythagorean theorem: $A^2+B^2=C^2$
- Law of Sines: $\sin(A)/a=\sin(B)/b=\sin(c)/c$
- Field of View Calculation: $2\text{ArcTan}((\text{height or Width})/(2*\text{focal Length}))$

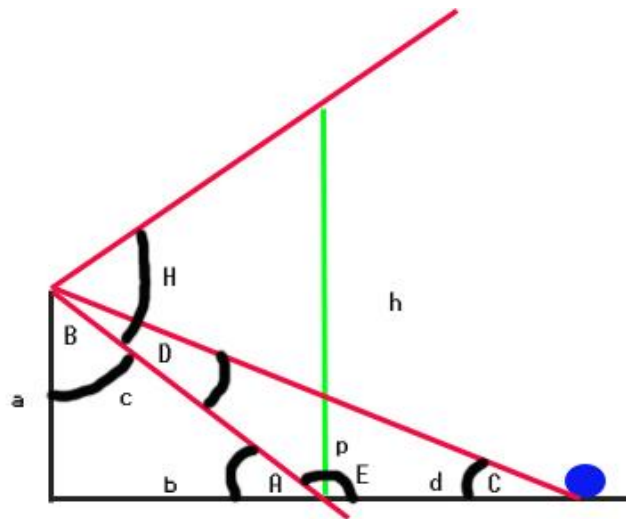


Figure 9. Mapping Pixels into Displacement and Bearing

The diagram in figure 9 explains how these formulas were implemented to map pixel distances to feet. The slope of the beach and tides were not taken into consideration. This two dimensional image represents a view where the camera vertex is on the left at the vertex of the

pink lines, and the drowning person is represented by the blue ball on the right. In figure 9, all of the capital variables are angles and all lower case variables are lengths. The redlines represent useful camera angles (angle to the top of picture, angle to a person in the water person, and angle to the bottom of the picture) the green line represents the photographic plane the image is projected on to. The field of view (h) is determined by $2 * \text{ArcTan}(\text{Height}) / (2 * \text{Focal Length})$. Using the Pythagorean theorem to find 'c'. since we know 'a' and 'b' (height of camera, and distance to the bottom of the image). Next we use the ratios of angles and lengths to determine 'D' ($D/H = p/h \rightarrow D = p * H/h$). To determine 'A' we can use simple trigonometry ($A = \sin(a/c)$). 'E' is found by $180 - A$. Since we know 'E' and 'D' we can now determine 'C' ($C = 180 - E - D$). Finally we can now determine the distance to the drowning person by rearranging the Law of Sines ($d = \sin(D) * c / \sin(C)$). The same Algorithm can be used to find the horizontal displacement and substituting widths for heights. After we know the x, y displacement in feet, the bearing is determined by the $\text{Tan}(x/y)$.

Now that we have an origin, bearing and displacement we can use the large sphere formula.

$$|\varphi_2| = \arcsin(\sin(\varphi_1) * \cos(d/R) + \cos(\varphi_1) * \sin(d/R) * \cos(\theta))$$

$$|\lambda_2| = \lambda_1 + \arctan_2(\sin(\theta) * \sin(d/R) * \cos(\varphi_1), \cos(d/R) - \sin(\varphi_1) * \sin(\varphi_2))$$

Where φ is latitude, λ is longitude, $|\theta|$ is the bearing (in radians, clockwise from north), $|d|$ is the distance traveled, $|R|$ is the earth's radius (d/R is the angular distance, in radians).

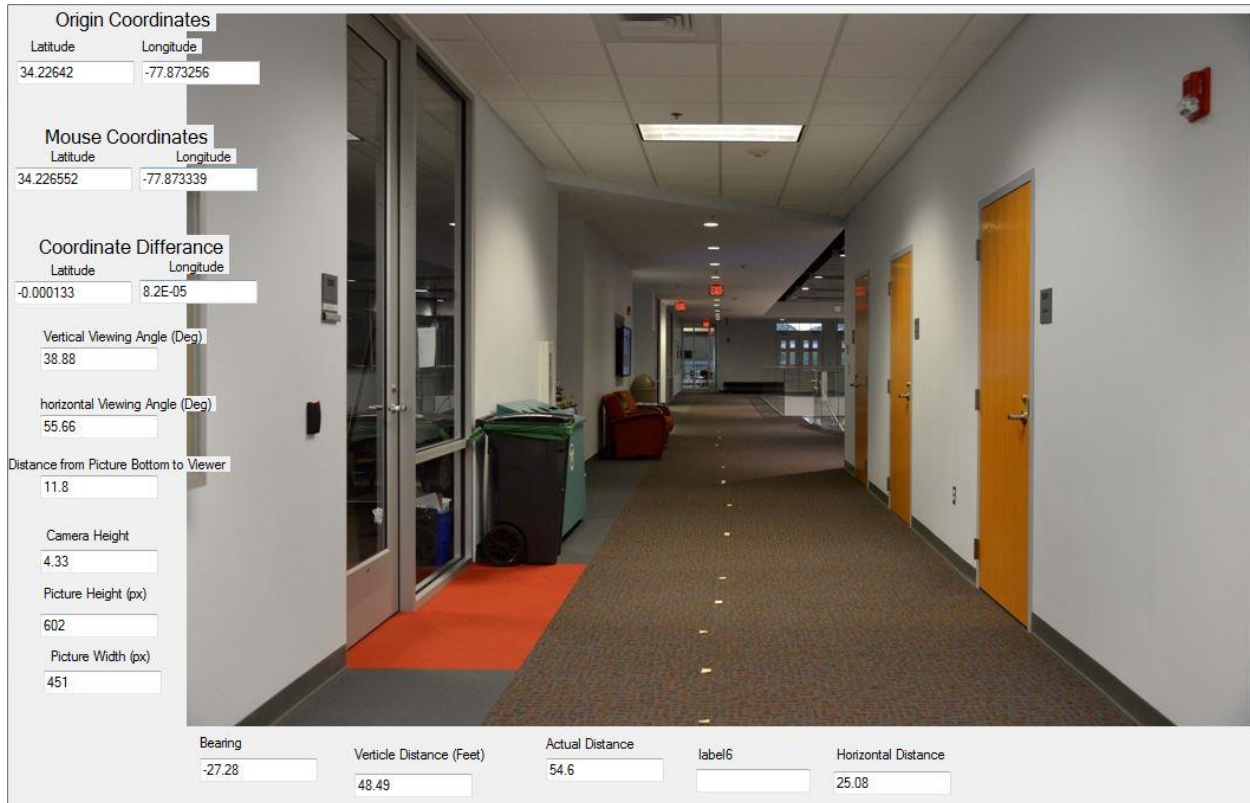


Figure 10. Life Guard Application (proof of concept)

In order to provide a proof of concept of the algorithm, a 35mm camera was used with a vertical viewing angle of 38.88 degrees and a horizontal viewing angle of 55.66 degrees. The camera was placed at a height of 4.33 feet in the middle of the 10 feet wide hallway. The bottom view found in figure 10 is 11.8 feet from the camera. In order to mark distances, markers were placed every 2 feet up to 20 feet. At 20 feet the distance between markers was raised to 5 feet. The proof of concept tests indicated that all marker 'y' distances can be verified within .2 feet (i.e. the range would be 49.8 to 50.2 for a 50 foot distance) and the hallway width can be verified to the end of the hallway. When a click event occurs, the mouse coordinates are mapped to GPS coordinates and sent to the drone via the MavLink protocol.

Figure 11 shows how the tablet view would look to a lifeguard when the hallway is replaced by a beach view and the Hough Circle algorithm is implemented. Note the red circles are people identified in the water.

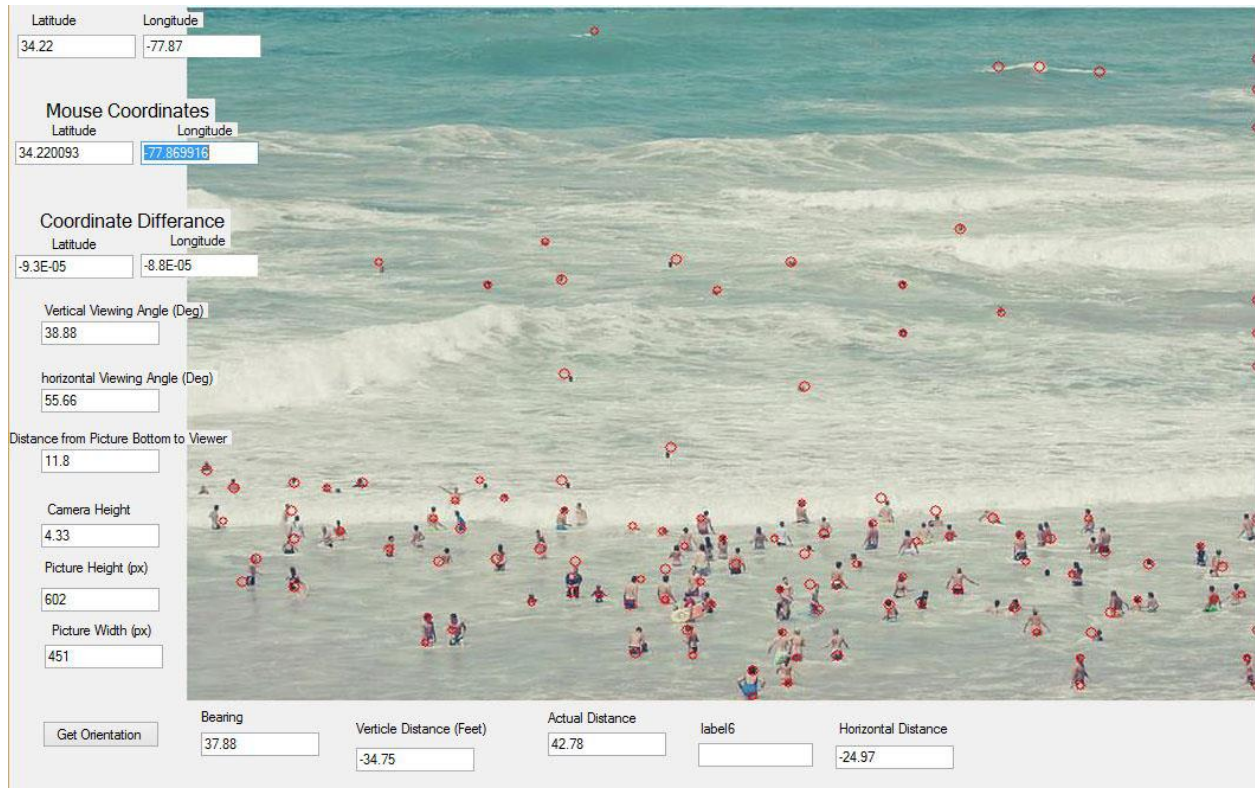
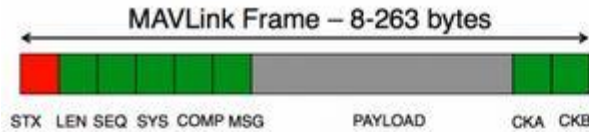


Figure 11. Life Guard Application (simulated screen)

CHAPTER 8: THE MAVLINK PROTOCOL

The hexacopter communicates with the base station over a serial interface using the MavLink Protocol. An example of the MavLink header can be seen in figure 12.



Byte Index	Content	Value	Explanation
0	Packet start sign	v1.0: 0xFE (v0.9: 0x55)	Indicates the start of a new packet.
1	Payload length	0 - 255	Indicates length of the following payload.
2	Packet sequence	0 - 255	Each component counts up his send sequence. Allows to detect packet loss
3	System ID	1 - 255	ID of the SENDING system. Allows to differentiate different MAVs on the same network.
4	Component ID	0 - 255	ID of the SENDING component. Allows to differentiate different components of the same system, e.g. the IMU and the autopilot.
5	Message ID	0 - 255	ID of the message - the id defines what the payload "means" and how it should be correctly decoded.
6 to (n+6)	Data	(0 - 255) bytes	Data of the message, depends on the message id.
(n+7) to (n+8)	Checksum (low byte, high byte)	ITU X.25/SAE AS-4 hash, excluding packet start sign, so bytes 1..(n+6) Note: The checksum also includes MAVLINK_CRC_EXTRA (Number computed from message fields. Protects the packet from decoding a different version of the same packet but with different variables).	

Figure 12.MavLink Packet

Each packet starts with (0xFE) to indicate the start of a transmission. The next byte indicates the length of the payload. The sequence number is used in case a packet is dropped or

mis-transmitted. The System ID is the ID of the device on the network that is sending the command. It can either be a base station or another hexacopter. However, the hexacopter only parses packets from the specified ID of their base station (0xFF by default). The component ID can be used to differentiate what part of the drones hardware the message is originating from. The message ID is used to correctly parse the data in the payload. There is a 16bit check sum to verify message content. MavLink Protocol also uses reverse byte order on anything greater than 8 bits. For example, if my 32 bit integer was equal to 12 (0x0000000C) it would need to be transposed to (0x0C000000) upon transmission.

In order to easily send and parse received messages, a base station GUI was built. A screen shot can be seen in figure 13.

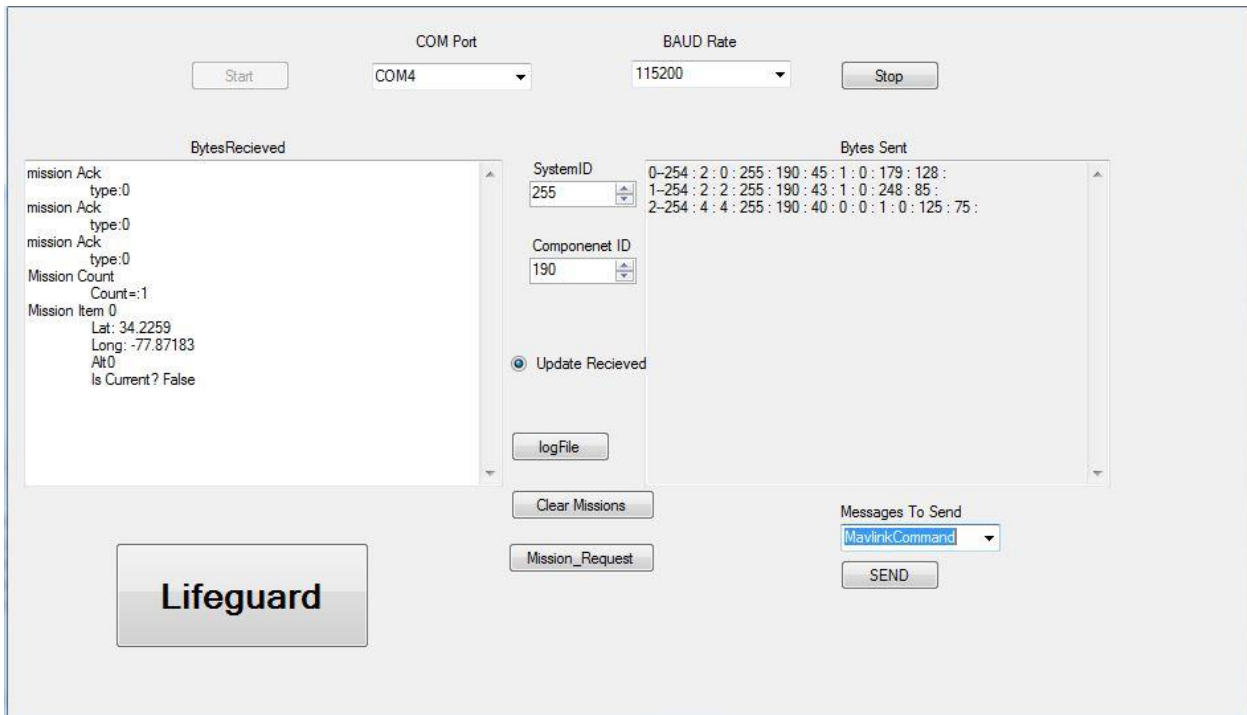


Figure 13. Base Station GUI

This platform allows me to become a base station and communicate with the drone. The text area on the left is a parsing of messages received from the hexacopter. The text area on the right is the bytes sent. Messages can be sent using the “MavLink Command” message that uses a subset of commands. The MavLink command always has a payload of 7 floats. The interface for sending a MavLink command can be seen in figure 14.

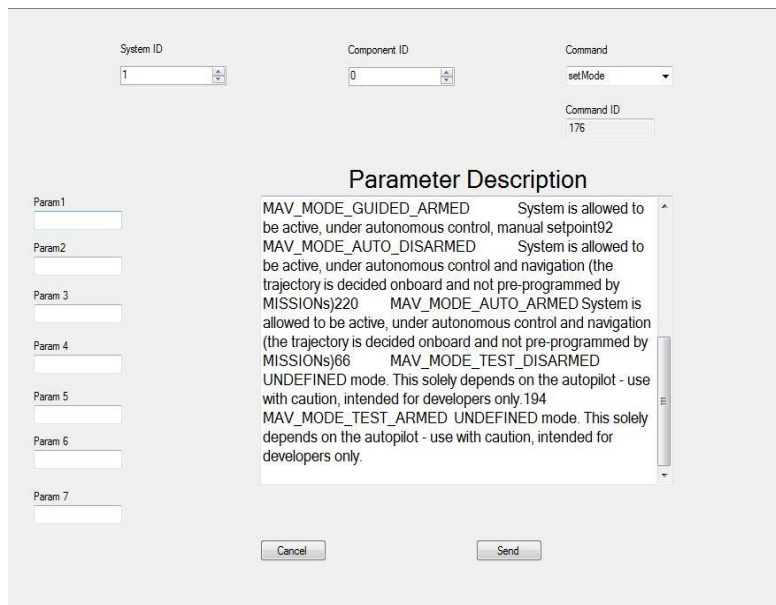


Figure 14. MavLink Command

Messages were sent to the hexacopter setting up a data stream that updates the current GPS position of the unit. These positions are then logged and using the Google maps API we can track the hexacopter in real time. This was implemented after a flyaway occurred so I could easily track the location of the hexacopter. While I have implemented most of the messages in the MavLink protocol, this particular project only requires a fairly simple flight plan. The first step is to fly to the coordinates of the victim. Once this position is achieved, the next step is to drop the life jacket. Finally, the hexacopter uses the RTL(return to landing) command to return back to its station and land.

CHAPTER 9: INVESTIGATE DROPPING A LIFE JACKET

Many dropping mechanisms are available for hexacopters. Most rely on a gripping mechanism that secures the object until it needs to be released. Another approach is an electromagnetic coupling; however, both these solutions require either cumbersome attachments or excessive power drain. My solution to this is to attach the life jacket to the hexacopter with monofilament line and use a “hot knife” resistor to melt the line releasing the payload. The resistor is hooked to transistor because the current draw is too much for the microcontroller. PWM to the gate of the transistor can be sent either from the microcontroller, or a 555 timer. As this was outside of the project scope the dropping mechanism will remain in the investigation phase. Shown in figures 15 and 16 are two possible circuit diagrams that can be used for the PWM (figure 15) and the transistor circuit (figure 16).

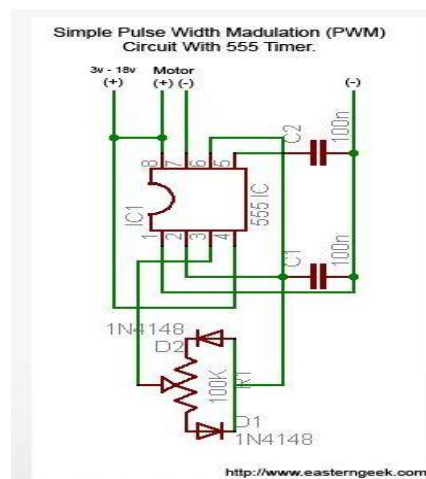


Figure 15. 555 Timer as PWM Control Circuit

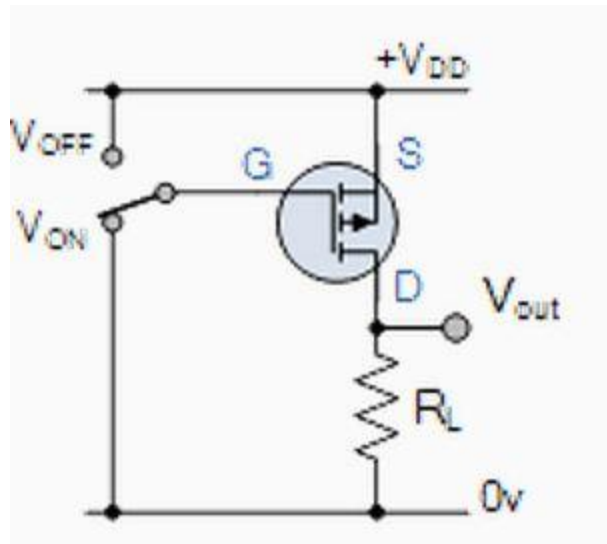


Figure 16. Transistor as hot knife circuit

In the future, it might be ideal to have the drone land in the water and turn into a floatation device itself. This would require a lot of advancement on the mechanical design of the hexacopter.

CHAPTER 10: SUMMARY OF WORK COMPLETED

The following items were completed as part of this project:

1. Build a hexacopter
2. Learned to fly the hexacopter
3. Set up C# software application to communicate with hexacopter using MavLink Protocol
4. Investigated various approaches for identifying swimmers, and implemented both the Hough Circle Algorithm and the erosion and dilation transformation. Using C# and EMGU libraries
5. Mapped pixel coordinates to GPS coordinates

The following items were investigated and found not to be feasible for this study

1. Live video feed loop
2. Dropping a life preserver to a potential victim

CHAPTER 11: FUTURE RESEARCH

There are three major areas for additional research. They are:

1. Improving the identification of ‘victims’ in the water -The Computer Vision segment of this project has the most room for improvement. The use of a Bayesian Classifier, FFT, or a Neural Network could offer possibilities.
2. Implementing a live video feed from the lifeguard stand so the system can work in real time is another area for improvement. Due to the victim drifting with the current and in accuracies in GPS data, a live video feed from the hexacopter to the water would be ideal for the tracking of the person in the water and adding pinpoint life jacket dropping or landing of the hexacopter.
3. The hexacopter itself has design flaws, mainly it is extremely susceptible to water and salt spray. Finally, the program to identify people in the water and the program that sends commands to the hexacopter need to be merged into a single application.

REFERENCES

- [1] <http://mashable.com/2012/09/27/household-robots/> viewed on 2/13/2013
- [2] http://www.cbsnews.com/8301-3445_162-57568571/drones-eyes-in-the-sky/ view on 2/13/2013
- [3] Pearce, Jeremy. 2011. *George C. Devol, Inventor of Robot Arm, Dies at 99*. The New York Times
- [4] http://www.humanoid.waseda.ac.jp/booklet/kato_2.html
- [5] <http://world.honda.com/ASIMO/history/>
- [6] Chirikjian, G. 2001 *Design and Analysis of Some Nonanthropomorphic, Biologically Inspired Robots: an Overview*. Journal of Robotic Systems.
- [7] Delft, T.U. 2008 *Micro Air Vehicle: Three Gram 'Dragonfly' Takes Flight*. ScienceDaily.
- [8] Sun, Ron. 2004. *Simulating Organizational Decision-Making Using a Cognitively Realistic Agent Model*. Journal of Artificial Societies and Social Simulation.
- [9] Weiss, Scaramuzza, Siegwar. 2011. *Monocular-SLAM-Based Navigation for Autonomous Micro Helicopters in GPS-Denied Environments*. Journal of Field Robotics
- [10] Santana, Correia, Mendoca. 2010. *Tracking Natural Trails with Swarm-based Visual Saliency*. Journal of Field Robotics
- [11] <http://www.robotshop.com/sensors.html> viewed on 2/18/2013
- [12] <http://www.cdc.gov/homeandrecreationalafety/water-safety/waterinjuries-factsheet.html> viewed on 2/13/2013
- [13] <http://electronics.howstuffworks.com/cameras-photography/digital/question362.htm> viewed on 3/5/2013
- [14] Selby, Corke, Rus. 2009. *Autonomous Aerial Navigation and Tracking of Marine Animals*.
- [15] Wikipedia.org http://en.wikipedia.org/wiki/Hough_transform
- [16] Wikipedia.org http://en.wikipedia.org/wiki/Canny_edge_detector
- [17] Badski, Gary. Learning OpenCV. 2008
- [18] <http://diydrone.com/forum/> viewed on 2/13/2013