CSEE Department
Marshall University
EE 420 Capstone II – Phase II

Date: 04/26/2021

Title of the Project: Heavy Lift Delivery Drone with OpenCV Object Avoidance

Team Members:
Name(s): ______Caleb Johnson________________901# __84-1801_____________________
Name(s): ______Peter Mills___________________901# __47-5389_____________________
Name(s): ______Bailey Protzman ______________901# __89-8446_____________________
Name(s): ______Thomas Williams______________901# _ 79-0608_____________________

Advisor: __Dr. Pingping Zhu Pingping Zhu

Table of Contents

List of Equations

List of Tables

List of Figures

Summary

Over the past several years, the use of Unmanned Aerial Vehicles (UAV) for delivery of packages has grown in favor of several large companies. The idea of autonomous delivery has emerged for trials in the consumer market and as recently as September 2020, both Walmart and Amazon offered same day delivery of packages by drones. The concept of drone construction has been conducted by numerous private companies, hobbyists, and militaries around the world for several years, but it has only recently gained attention from industrial, logistical, shipping, and medical companies for uses other than recreation or national defense. It has been determined that the use of drones for delivery on a broad scale has been deemed feasible [1].

It was determined by the team to be a visionary project, not only for the design, build, and test aspects, but also for the device's potential applicability. Not only could this device be used to deliver consumers their packages with relative convenience and ease, but also offers the ability to easily and quickly send and deliver medical supplies from neighboring hospitals or replenish resources in disaster situations where larger ground or air-based vehicles may be unable to reach. Furthermore, it is more prominent in today's world as a logistical company's response to the world-wide Covid-19 outbreak, where social distancing is encouraged and the need for less face-to-face interaction is required for worker and customer safety. Additionally, the team determined that the challenge to design, construct, and test such vehicle at as low cost as possible would be applicable to our future endeavors in the work force and in academia.

The team has developed skills and gained knowledge from courses that will be helpful in completing this project. These courses include control systems, physics, project management, circuits, and communication systems.

1. Project Goal

To design and construct a drone that can be easily loaded with a payload of at least 2.3kg, safely deliver a package to a determined delivery area, and return and land at the initial takeoff location. This will be achieved by improving on previous heavy-lift-drone (HLD) designs and implementing object avoidance during flight.

2. Project Objectives

- Improve on designs of payload delivery system (PDS) such that payload is easy to load and unload with no human interaction at a delivery site with a payload of at least 2.3kg and up to 16.8kg
- Implement autonomous flight that can takeoff, fly to a delivery location, deliver payload, return to the home location, and land with minimal human interaction
- Implement computer vision for object detection and avoidance during flight
- Select parts for the drone, such as motors, flight controller, and battery, that are cost effective, while still being able to reliably lift the payload.

3. Background and Rationale

Drones have been around for decades, and their use has rapidly grown in recent years. Understanding the basics of flight of a multirotor vehicle and its theoretical equations is a necessity when a UAV is scaled up in size [2]. While the concept may remain the same, the components would need to be upgraded to handle larger amounts of power to compensate for the size of the vehicle. Additionally, a UAV would become inherently safer if it were to be able to avoid objects while in takeoff, landing, and during flight. This would reduce the number of mid-air collisions that could potentially take place with other aerial vehicles, wildlife, and other potential obstructions below 400 ft above ground level [3]. It is also more sustainable for a drone not to land at a delivery location due to potential issues in suitable landing locations and complications during liftoff, so a method to deliver a package from air to ground would be necessary.

The flight of a multi-rotor UAV is based on physics concepts of kinematic and potential energy conservation, Newton’s laws of motion, momentum, and basic power equations [4]. To understand how a drone hovers, consider Newton’s 3rd law, the force of the air pushed down by the rotors would equal and opposite to the force applied to the vehicle by gravity. To rise, the drone would need to exert an upward force greater than what is enacted upon it by gravity. To apply a directional force, the drone would tilt, which would be dependent on motor speeds that would cause the drone to move in the tilted direction, following Newtons 1st law of motion [5].

Apart from theoretical modeling, thrust can be represented in terms of propeller diameter, power absorbed, and air density. [6]

When the above formula for power was compared with data sheets provided by Advanced Precision Composites Propellers®, a propeller manufacturer, it was found that the equation overestimated power, which caused a higher thrust and current to be derived [7]. It was also limited because it does not explain how the propeller constant related to pitch or diameter. There were only constants listed for a limited number of propellers, so without the relation to propeller size, this power formula is not useful for calculating the thrust for any other propellers.

This formula for power from the article "Understanding Electric Power Systems” underestimates the power used. The first formula is from 2016 and the second is from

1998 [8]. Over the last 22 years, technology has improved, and the material and shape of propellers have been refined. The newer formula uses an rpm exponent of 3.2. This is more accurate. The propeller constant, K_p, is outdated as well. Even though the constants are outdated, the formula is still useful because it shows the relationship between pitch, diameter, and power. The formula can be set equal to known values of power and rpm in order to update the old constant to fit the current propellers. Any propeller can be modeled using this method. Actual values to input were provided by a company that manufactures propellers [9].

As pitch increases, power, and therefore thrust, increases. However, stability decreases. The best way to increase thrust without losing stability is to use a propeller with a larger diameter and a smaller pitch. Pitch is directly related to speed. Smaller pitch means the drone will move slower [9]. Ideal parameters are about twice the realistic parameter. The calculated rpm is found from multiplying the motors kv rating by the voltage supplied by the battery [10]. This is the ideal rpm. The ratings given for the motor are without a load so the actual rpm will be about half. The static thrust needs to be divided by two as well in order to find the amount of mass that can lifted.

The mass of the drone was determined by deciding on components and adding the masses in a spreadsheet. It was found that the necessary components for an autonomous drone are the frame, electronic-speed-controllers (ESC), power distribution board (PDB), GPS module, flight controller (FC), lithium-ion polymer battery (LiPo), motors, and propellers [11]. According to FAA regulations, the maximum amount of weight a drone can be is 55 pounds [12]. While there are few sources with information on heavy lift drones, there are many about smaller drones. The same theoretical equations used for smaller drones can be used on a much larger scale [13].

There have been several different iterations of payload delivery systems including winches, rotational motion, and parachute drops [1][2][14]. A separate project focused solely on the PDS used a weight-based hook pick point, and winch [2]. When applied with a downward force inside the mouth of the mechanical hook, the hook clamped a pick point, and would open when a downward force is applied from the top of the hook. The project also presented a prebuilt which that was able to lift weights of 5 pounds for a price of $100 [2].

For implementation of object avoidance, the process of creating a basic collision avoidance system on a raspberry pi 2 utilizing OpenCV [15]. This method is able to accurately measure depth and be implemented with object avoidance. This is obtained through the generation of a disparity map, that is set with a threshold, and properly implemented contour detection to the response of the drone. The collision avoidance is accomplished through a "see and avoid" style algorithm which only allows the drone to proceed forward or backward to avoid an obstacle. This process was accomplished on a less powerful system, but the fundamentals are similar to other projects and the compatibility of the systems discussed applies directly to the capstone [16].

While object avoidance can be implemented for lower power systems, it can still have a problem of random depth miscalculations that occur in stereo camera-based collision avoidance systems. Miscalculations can unnecessarily trigger the collision avoidance system which makes the drone maneuver around non-existing objects negatively impacting its ability to follow its flight path. However, the algorithm for further processing can be further improved by techniques used in a previous study where the system was tested in an outdoor environment. Techniques that could be used included changing the camera's exposure level and downsizing the image in post-processing to smooth out camera noise. [17].

Similar autonomous, HLD projects have been completed between a range of a 2kg lift to 25kg (~55lbs) for a cost of $300-$13,921 [12][18]. The most similar project was found to be a heavy lift drone that was able to provide enough lift for 55lbs, at a cost of $13,921 [12]. The proposed cost of this projects drone is set to $1,377.33. To reduce costs, several components will need to be designed and constructed instead of purchased out right. To further reduces costs, the availability of 3D printing will be utilized since it has proven acceptable to construct drones from 3D printed parts [19]. These printed parts will be utilized on the drone body and payload delivery system. Furthermore, there is a lack of academic literature on how to implement a flight planning software with object avoidance through OpenCV. Overall, the proposed design will rely on alterations to previous algorithms for object avoidance, implementation of a winch system with alteration to reduce cost and complexity and include 3D printed parts to secure components to the drone. All of this will be done while reducing the cost of the vehicle to $1,377.33

4. Design Approach and Potential Challenges

The design of the Autonomous Delivery Heavy Lift Drone with OpenCV Object Avoidance was portioned into 4 different areas deemed necessary for the design. These areas were to determine the calculations necessary for flight, understand the implementation of OpenCV, determine necessary components, and the development of a PDS. The main task of a UAV is to fly, so the task of understanding the equations of flight were considered first. Figure 1.

4.1. Flight Calculations

The purpose of finding and designing proper equations for lift is to develop a mathematical model that will make tuning and adjustments for getting the drone into the air easier. With a mathematical model following principles of physics, such as Newton’s Laws and Bernoulli’s Equations, it can be determined how much lift is required to cause the drone to ascend and then hover and descend.

These actions were deemed necessary due to their ability to determine the amount of force needed to be applied by the rotors to create lift. Thrust can be represented in terms of propeller diameter, power absorbed, and air density, as can be seen in the below equation, where T is thrust, D is diameter of the propeller, ρ is the fluid constant of air and P is power in watts [6].

$$T = \sqrt[3]{(0.5\pi\, D^2\, P^2\, \rho)} \qquad (1)$$

Additionally, Power can be modeled as [6]

$$\mathrm{P} = p * \mathrm{rpm}^{pf} \qquad (2)$$

Where p is prop constant, pf is power factor, and rpm is revolutions per minute of the rotor. The power was then modeled by [8].

$$Power = p\, K_p D^4 * rpm^3 \qquad (3)$$

This formula uses the value K_p = 1.11 for APC propellers. It suggests using 1.25 as an estimate for all other propellers. The diameter, D, and pitch, p, are in feet and the rpm is in thousands. This would also allow the team to plan for how wind may affect the drone.

While the data obtained by the equations would be solely theoretical, it was expected that the real word application of the vehicle would differ due to drag, friction and efficiencies of different components. Additionally, the equations were initially meant to represent static bodies, which would cause the dynamic nature of the drone to differ from theory. However, the resulted issues will be resolved by component selection due to the availability of real-world test data and adjustments made during testing.

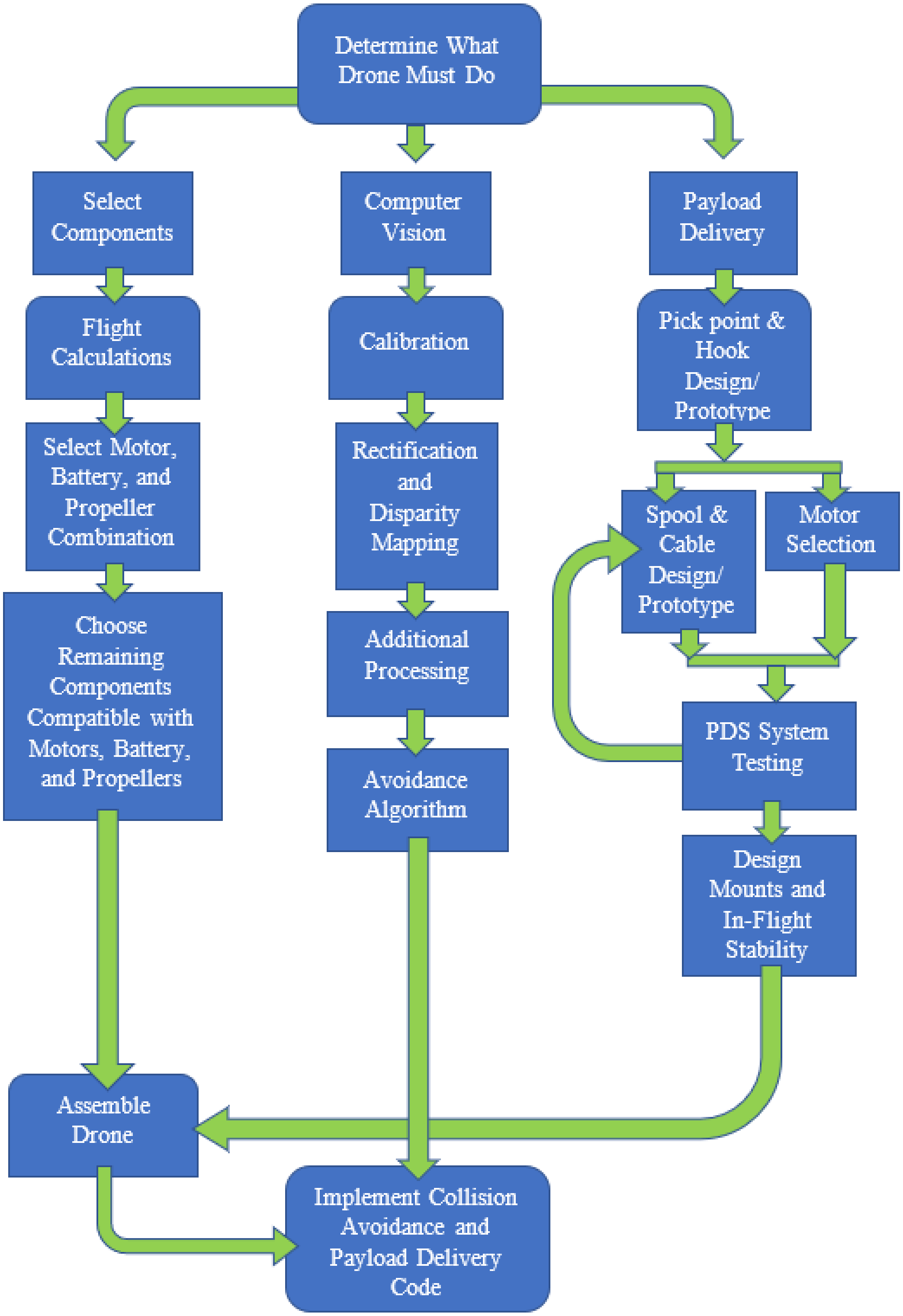

Determine What Drone Must Do
Select Components
Computer Vision
Payload Delivery
Flight Calculations
Calibration
Pick point & Hook Design/ Prototype
Select Motor, Battery, and Propeller Combination
Rectification and Disparity Mapping
Spool & Cable Design/ Prototype
Motor Selection
Choose Remaining Components Compatible with Motors, Battery, and Propellers
Additional Processing
PDS System Testing
Avoidance Algorithm
Design Mounts and In-Flight Stability
Assemble Drone
Implement Collision Avoidance and Payload Delivery Code

Figure 1: Design Process Flowchart

4.2. Component Selection

A multitude of parts are needed to construct an autopilot drone. There were many considerations when choosing parts, such as part compatibility, lift generation, and cost. Since the drone must be able to deliver a package, it requires a lot of thrust to lift the payload. Not all drone parts work together. Certain motors and frames only fit a small range of propeller sizes. It is important to make sure each part is compatible with the others. Cost is a very important aspect of the design. The team aims to design a heavy lift drone as cost effective as possible.

A hexacopter design was chosen over a traditional quadcopter design because it increases lift and flight stability. The drawback to this design decision is that a hexacopter cost more than a quadcopter. It was decided that a quadcopter could not produce the lift required to lift and deliver a package.

The selected parts were:

- 5010 750KV High Torque Motors – This particular motor was selected because it produces significantly more thrust than other motors at a comparable price point. The motor is able to produce more thrust because it can be used with a 6S battery, and larger propellers. The equations for maximum RPMs, power, and thrust were utilized to evaluate motor-propeller combinations. Equation 4 below shows the maximum rotations per minute. Most of the motors with a higher speed were not compatible with large propellers. This motor was perfect for the application because it is a high torque motor for large drone propellers. This was also the only high torque motor for large propellers under $100 per motor. This motor cost about $22 per motor. Fig. 2 below shows a single arm of the drone with double stacked motors and propellers.

$$rpm_{max} = kv * V = 750 * 22.2 = 16,650\, rpm \qquad (4)$$

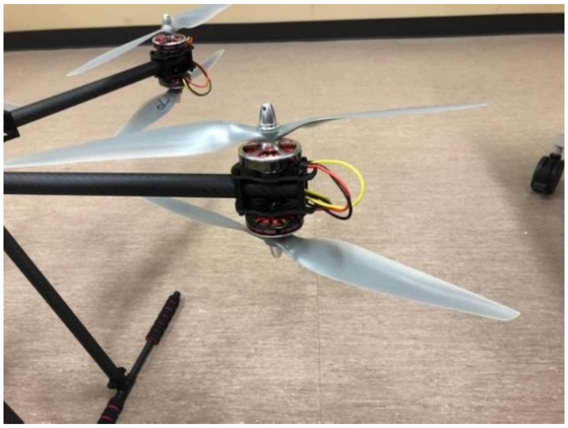


Figure 2: Motors and Propellers

- Electronic Speed Controllers – Each motor is connected to an electronic speed controller or ESC. An ESC is necessary to regulate the speed of a motor. This component can be seen in Fig. 3.

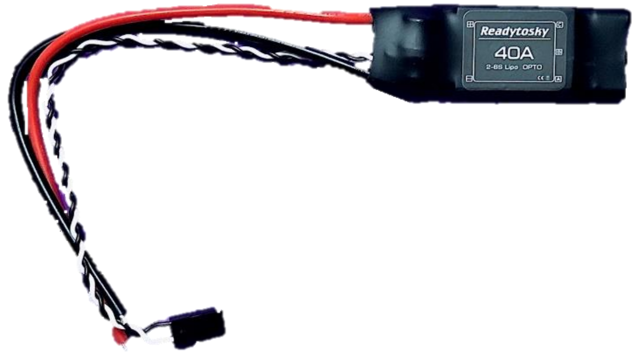


Figure 3:Original ESC

- 15x5.5 Propellers – The selected propellers have a diameter of 15 inches and a pitch of 5.5 inches. The selected motor is compatible with propellers between 12 and 16 inches in diameter. Larger propellers draw exponentially more power, which drains the battery quicker and reduces flight time. A 15-inch propeller was chosen because it creates enough lift without draining too much of the battery. Equations 1 and 2 were used to analyze various motor and propeller combinations to decide which combination would optimize lift and power consumption.

- ZD-850 Carbon Fiber Frame – The ZD850 frame was chosen because of its compatibility with the selected motors and propellers. The carbon fiber material is very lightweight, which is important for maximizing the payload mass. The motor mounts on the frame fit a 5010 motor, and the frame is designed for 14-16-inch propellers. Fig. 4 below shows the ZD-850 frame with the 5010 750 kv motors and APC 15x5.5-inch propellers. Fig. 5 shows the drone in a folded arm orientation. With arms out and propellers on, the drone is 4 feet in diameter.

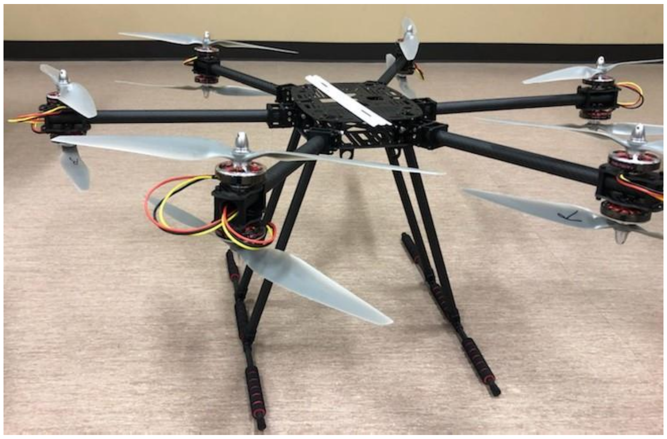


Figure 4: Partial Drone Build

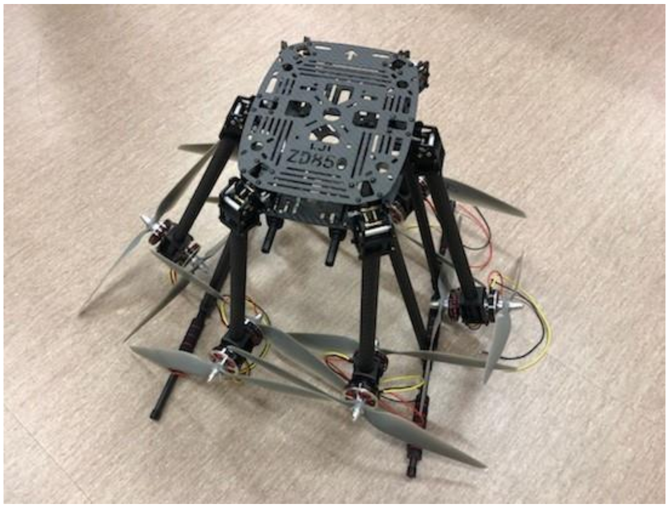


Figure 5: Folded Arm Orientation

- 6S Lipo Battery 27000mAh – Due to the large current draw from the motors, a large capacity battery is needed. The current draw for each component was calculated. Equation 5 was used to calculate the flight time that a battery could sustain. This battery was selected because it can support at least 15 minutes of flight time. 6S corresponds to 22.2V. The power needed to generate a certain amount of lift for a certain propeller will be the same regardless of voltage. Our high torque motors can operate at 11.1V to 22.2V (3-6S). Choosing the highest voltage of this spectrum lessens the current draw, providing longer battery life.

$$Flight\ Time\ (minutes) = 60\ *\ Capacity\ (mAh)\ \ /\ I \qquad (5)$$

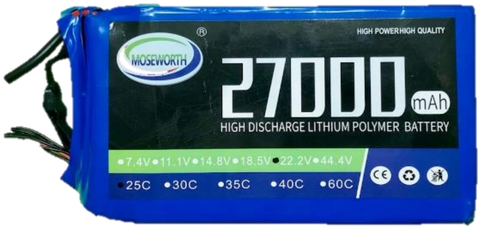

MOSEWORTH
HIGH POWER HIGH QUALITY
27000 mAh
HIGH DISCHARGE LITHIUM POLYMER BATTERY
•7.4V •11.1V •14.8V •18.5V •22.2V •44.4V
•25C •30C •35C •40C •60C

Figure 6: 6s 22.2V 27,000mAh Battery

- Battery Charger – It is necessary for the battery to be charged in order to get multiple flight tests completed.

Figure 7: 6s Battery Charger

- Pixhawk Autopilot Kit with GPS and 915MHz Telemetry – This flight controller was chosen because it supports autopilot flight and is compatible with the open-source software Mission Planner. The kit chosen comes with GPS which has the capability for autonomous flight. It also comes with 915MHz telemetry. This frequency was chosen because it is the recommended frequency for drone telemetry in the US.

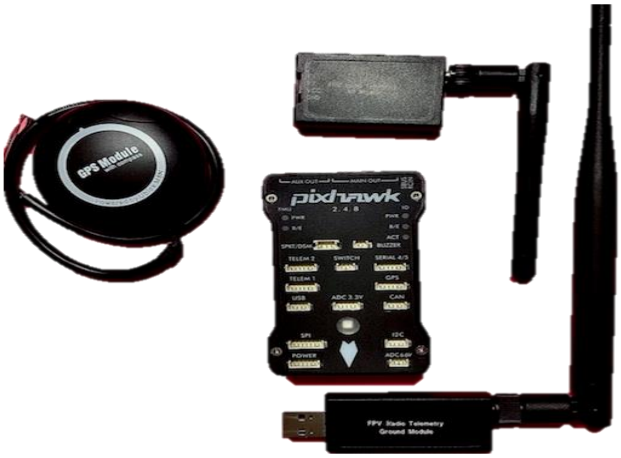


Figure 8: Pixhawk Kit

- Power Distribution Board – A power distribution board, or PDB, is necessary for distributing the battery’s power across the different components. It has ports that step down the voltage to power components like the flight controller or raspberry pit that run on less than 22.2V.

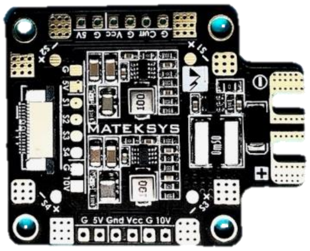


Figure 9: Power Distribution Board

- Payload Delivery System – A winch made using a servo motor and 3D printed parts will lower and release the payload/package.
  - 60kg Servo motor – the large torque of the motor was necessary to allow for lift at further distances than 1cm from the rotor which would be based on the spooled cable.
  - 110lb test 3/8” polypropylene rope- used as cable for winch.
  - 100mm M3 all thread- used as shaft to secure spool to servo and to assemble PDS.
  - M3 nuts- used to assemble PDS
  - 3D printed parts: winch box, hook, pickpoint.
  - 3/8”ID 1/4” groove, 1/2” panel hole, Rubber grommet- used for mounting all equipment to drone frame rails.
- Cameras – Two Logitech C615 cameras will be used for computer vision.
  - 1/4-20 allthread- cut to length to support camera mount.
  - 3D printed parts- bottom plate, top plate, rail mount.
- Raspberry Pi4 – A raspberry pi will be used to control the cameras and payload delivery system.
- Remote Controller – A remote controller is necessary for flight.

Figure 10: Fs-I6b Remote Controller

- FAA Registration – In order to adhere to federal law, the hexacopter will be registered as a recreational unmanned air system prior to flying.

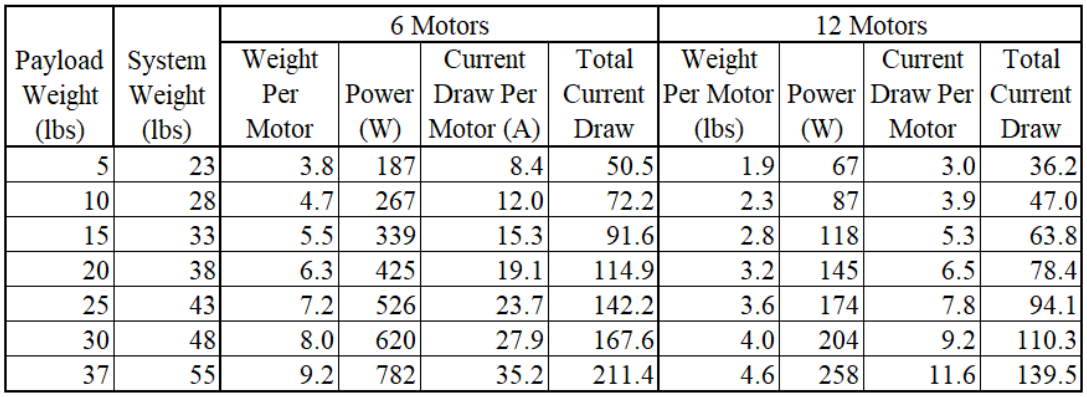
The weight of the drone is equally distributed among the propellers and motors. Lift data obtained from APC Propellers is shown below. It related generated lift to power.

Table 1: Lift Vs Power for 15" Propellers [20]

Lift (lbs)	Power (hp)
0.613	0.017
1.380	0.055
2.461	0.126
3.881	0.259
5.653	0.473
7.790	0.793
10.321	1.252
13.190	1.746
16.486	2.339
20.258	3.046
24.662	3.892

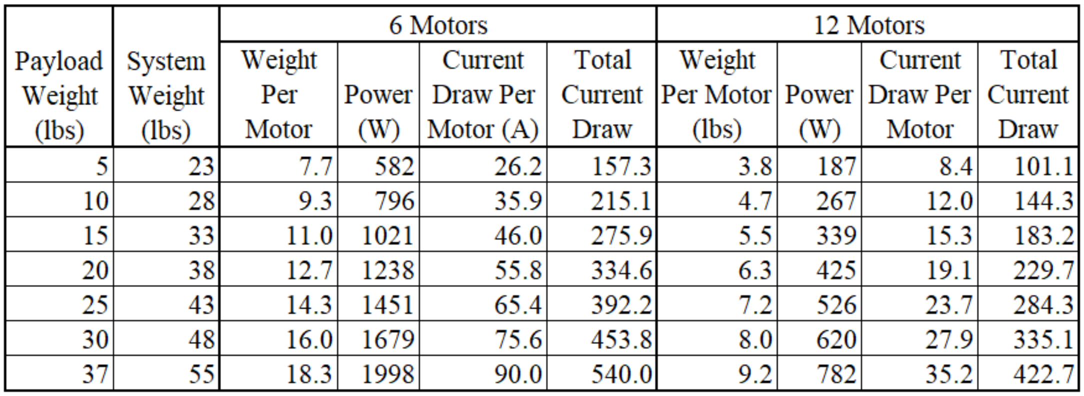
The data in table 1 was used to find a curve of best fit. The weight of the drone, not including the payload was calculated to be about 18 pounds. The weight per motor was calculated for payload weight of 5, 10, 15, 20, 25, 30, and 37 pounds. A 37-pound payload would bring the drone to the maximum 55-pound weight allowed by the FAA. Each weight per motor was plugged into the curve of best fit to find the required power for that lift. That power was then converted to watts. Given a battery voltage of 22.2 V, the current for each motor was calculated and then multiplied by the number of motors to get the total current draw. This was completed for a typical 6-motor hexacopter orientation and a 12-motor double stack orientation. The maximum current draw, which will occur when the drone is increasing in altitude, was calculated by doubling the pound force. The typical current draw, which will occur when the drone remains at a constant altitude, was calculated by distributing the weight of the drone evenly to all motors. Tables 2 and 3 below show the typical and maximum current draw for a 6 and 12 motor configuration. This process of calculating current draw was also used to evaluate different motor, propeller, and battery combinations.

Table 2: Typical Current Draw



Payload Weight (lbs)	System Weight (lbs)	6 Motors				12 Motors			
		Weight Per Motor	Power (W)	Current Draw Per Motor (A)	Total Current Draw	Weight Per Motor (lbs)	Power (W)	Current Draw Per Motor	Total Current Draw
5	23	3.8	187	8.4	50.5	1.9	67	3.0	36.2
10	28	4.7	267	12.0	72.2	2.3	87	3.9	47.0
15	33	5.5	339	15.3	91.6	2.8	118	5.3	63.8
20	38	6.3	425	19.1	114.9	3.2	145	6.5	78.4
25	43	7.2	526	23.7	142.2	3.6	174	7.8	94.1
30	48	8.0	620	27.9	167.6	4.0	204	9.2	110.3
37	55	9.2	782	35.2	211.4	4.6	258	11.6	139.5

Table 3: Maximum Current Draw



Payload Weight (lbs)	System Weight (lbs)	6 Motors				12 Motors			
		Weight Per Motor	Power (W)	Current Draw Per Motor (A)	Total Current Draw	Weight Per Motor (lbs)	Power (W)	Current Draw Per Motor	Total Current Draw
5	23	7.7	582	26.2	157.3	3.8	187	8.4	101.1
10	28	9.3	796	35.9	215.1	4.7	267	12.0	144.3
15	33	11.0	1021	46.0	275.9	5.5	339	15.3	183.2
20	38	12.7	1238	55.8	334.6	6.3	425	19.1	229.7
25	43	14.3	1451	65.4	392.2	7.2	526	23.7	284.3
30	48	16.0	1679	75.6	453.8	8.0	620	27.9	335.1
37	55	18.3	1998	90.0	540.0	9.2	782	35.2	422.7

The total current is significantly higher for a 6-motor configuration. Because of this a double layered 12-motor hexacopter design will be implemented. A larger current will drain the battery quicker. It is important to get enough flight time to deliver the package. Because of a large potential current draw 8-gauge wire will be used.

Fig. 11 below shows an CAD design prototype. The double layered hexacopter design can be seen.

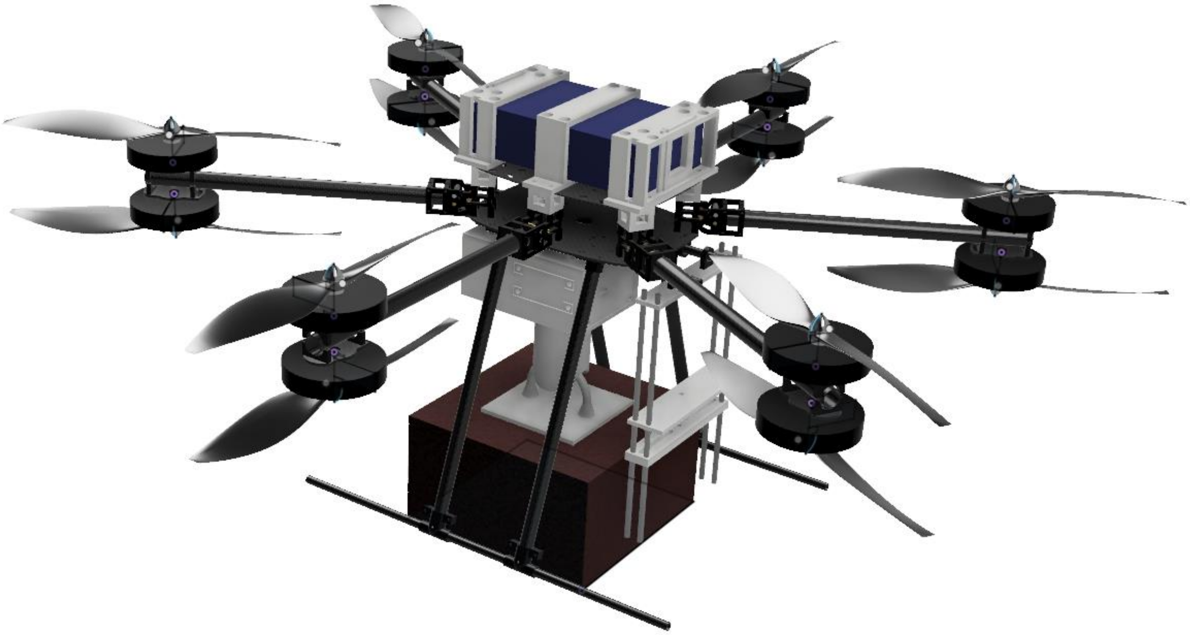


Figure 11: CAD Prototype

Some problems have arisen during the design implementation. One of those is quality control of parts ordered. During our initial test flight, we had a capacitor blow. This capacitor was part of one of the electronic speed controllers. When the capacitor blew, it caused the associated motor to stop, and the drone flipped over. The initial test was a small hover, so there was no major damage. A few propellers broke and the electronic speed controller was replaced. Drone propellers are the most common part of a drone to break. The team anticipated that propellers would break during testing and ordered a pack of 20 propellers instead of purchasing 12 individual propellers. The propellers also cut a couple wires, so shorter wires were soldered to the drone to prevent this problem from reoccurring. The team also ran into problems with the telemetry unit. While trying to connect Mission Planner to the drone via 915MHz telemetry, the firmware on the ground unit was accidentally updated. The ground unit was no longer able to connect to the air unit. An FTDI cable was used to update the air unit, but it was an unfruitful attempt. Unfortunately, a new telemetry unit had to be ordered. The replacement telemetry works with no problems. The largest challenge has been the cost. Heavy lift drones are costly. To mitigate this, the team applied for additional grants, and per Dr. Yoo, the remaining cost will be covered by the CSEE Department.

4.3. Flight Model

Before physical tests were started, the team developed the control model of the Pixhawk flight controller. The drone was proposed to be controlled by reading GPS position, gyroscope, and accelerometer values to estimate the vehicle’s location in space. The control method used would be a feedback control loop involving the drone, its visible position, and the observer (pilot) which would adjust position of the drone. The system was modeled to verify the design of the drone and that the proposed payload weight would be theoretically achievable. The flight controller model can be easily found for a quadcopter, but not for a hexacopter. We developed the control model using the equations for hexacopter flight [21] and an existing model for a quadcopter [22]

The Pixhawk was determined to have 11 PID controllers used to control the position and orientation of the drone. The vehicle had to be modeled in a 6-dimensional space consisting of X, Y, Z position and the roll, pitch, and yaw of the vehicle. This was due to the nature in which a drone is controlled and exists in a relative 3-D space as seen by the observer and controlled by adjustments of the vehicles tilt along 3-axes. This is better visualized through Fig. 12.

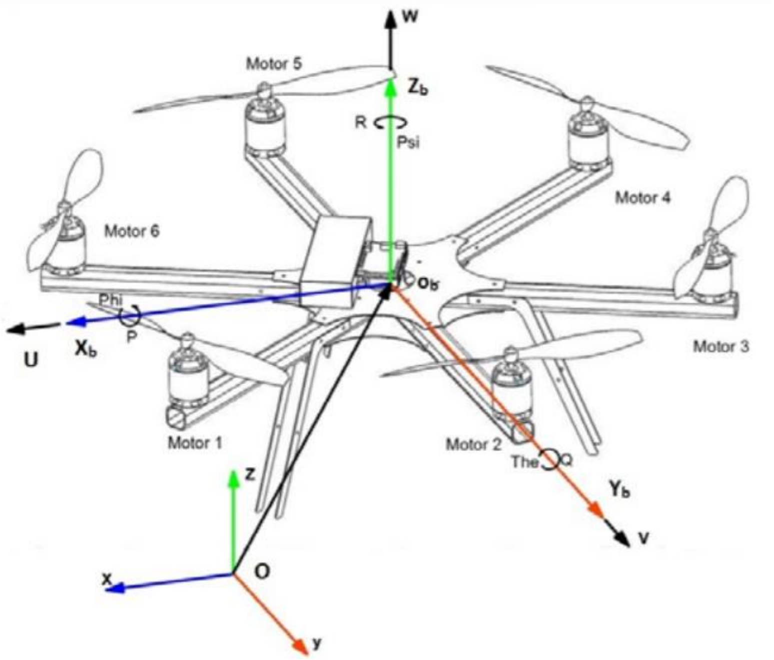


Figure 12: Hexacopter Axes of Motion

Based on the above assumptions, the system was mathematically modeled by use of quaternions. The mathematical model was represented by the variable outlined in Table 4.

Table 4:System Model Symbols & Variables

C	Description	Unit
m	Mass of the system	Kg
g	Gravitational constant	m/s/s
T	Thrust	N
x	Position along x axis	M
y	Position along y axis	M
z	Position along z axis	M
ϕ	Roll angle	Rad
θ	Pitch angle	Rad
ψ	Yaw angle	Rad
$\dot{\phi}$	Roll angle rate of change	rad/s
$\dot{\theta}$	Pitch angle rate of change	rad/s
$\dot{\psi}$	Yaw angle rate of change	rad/s
p	Angular velocity of roll	rad/s
q	Angular velocity of pitch	rad/s
r	Angular velocity of yaw	rad/s
U_1	Altitude control	
U_2	Roll control	
U_3	Pitch control	
U_4	Yaw control	
u_x	Linear motion control – y axis	
u_y	Linear motion control – x axis	
ω	Motor speed	rad/s
d	Drag torque coefficient	kg-m^2
b	Thrust force coefficient	kg-m
l	Motor to center of mass	m
k	Lift constant	kg/s^{2}mPa
I_M	Inertia moment of rotor	kg-m^2
J_{xx}	Moment of inertia – x axis	kg-m^2
J_{yy}	Moment of inertia – y axis	kg-m^2
J_{zz}	Moment of inertia – z axis	kg-m^2
τ	Torque	Nm

The relationship between motor speed and the controls for altitude, roll, pitch, and yaw can be seen in Equation 6.

$$
\begin{gathered}
{\omega_1}^2 = \frac{U_1}{6b} + \frac{U_2}{3bl} - \frac{U_4}{6d} \\
{\omega_2}^2 = \frac{U_1}{6b} + \frac{U_2}{bl} - \frac{\sqrt{3}U_3}{6bl} + \frac{U_4}{6d} \\
{\omega_3}^2 = \frac{U_1}{6b} - \frac{U_2}{bl} - \frac{\sqrt{3}U_3}{6bl} - \frac{U_4}{6d} \\
{\omega_4}^2 = \frac{U_1}{6b} - \frac{U_2}{3bl} + \frac{U_4}{6d} \\
{\omega_5}^2 = \frac{U_1}{6b} - \frac{U_2}{bl} + \frac{\sqrt{3}U_3}{6bl} - \frac{U_4}{6d} \\
{\omega_6}^2 = \frac{U_1}{6b} + \frac{U_2}{bl} + \frac{\sqrt{3}U_3}{6bl} + \frac{U_4}{6d}
\end{gathered}
\tag{6}
$$

The inverse of these equations was then converted into matrix form to make the controls the output, shown in Equation 7.

$$
\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b & b & b \\ -\frac{bl}{2} & -bl & -\frac{bl}{2} & \frac{bl}{2} & bl & \frac{bl}{2} \\ -\frac{bl\sqrt{3}}{2} & 0 & \frac{bl\sqrt{3}}{2} & \frac{bl\sqrt{3}}{2} & 0 & -\frac{bl\sqrt{3}}{2} \\ -d & d & -d & d & -d & d \end{bmatrix} \begin{bmatrix} {\omega_1}^2 \\ {\omega_2}^2 \\ {\omega_3}^2 \\ {\omega_4}^2 \\ {\omega_5}^2 \\ {\omega_6}^2 \end{bmatrix}
\tag{7}
$$

The motor speed equations for hexacopter flight can be used to model the motor speed block as seen below in Fig. 13.

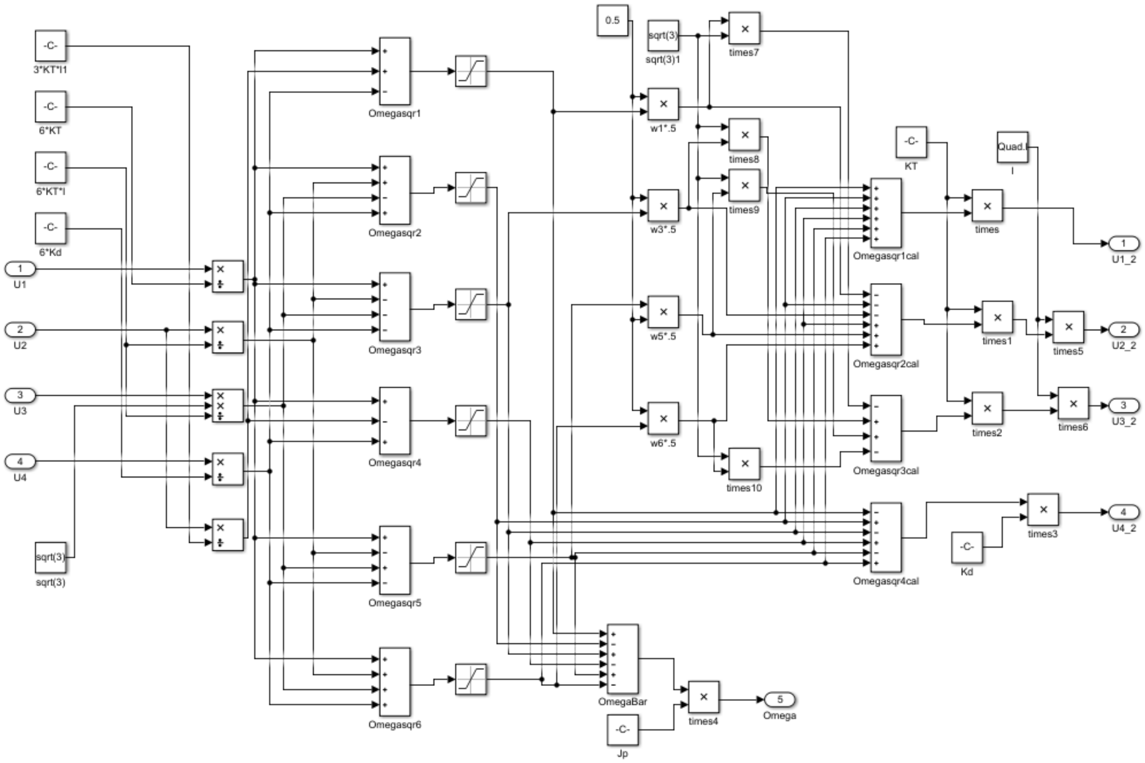


Figure 13: Motor Speed Block

The torque matrix represents the relationship between the motor speed and the torque of roll, pitch, and yaw as described by Equation 8.

$$\begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} \frac{3}{4}kl({\omega_2}^2 + {\omega_3}^2 - {\omega_5}^2 - {\omega_6}^2) \\ kl(-{\omega_1}^2 - \frac{{\omega_2}^2}{4} + \frac{{\omega_3}^2}{4} + {\omega_4}^2 + \frac{{\omega_5}^2}{4} - \frac{{\omega_6}^2}{4}) \\ b(-{\omega_1}^2 + {\omega_2}^2 - {\omega_3}^2 + {\omega_4}^2 - {\omega_5}^2 + {\omega_6}^2) + \\ I_M\left({\dot{\omega}_1}^2 + {\dot{\omega}_2}^2 + {\dot{\omega}_3}^2 + {\dot{\omega}_4}^2 + {\dot{\omega}_5}^2 + {\dot{\omega}_6}^2\right) \end{bmatrix} \tag{8}$$

The $[p\ q\ r]^T$ matrix, shown in equation 9, can be used to relate the angular velocity to the angular rate of change for roll, pitch, and yaw.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -sin\theta \\ 0 & cos\phi & sin\phi cos\theta \\ 0 & -sin\phi & cos\phi cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{9}$$

The $[\dot{p}\ \dot{q}\ \dot{r}]^T$ matrix, Equation 10, can be used to relate the angular acceleration to moment of inertia, angular velocity, and torque. ω_Γ is the sum of all the motor speeds, where motors 1, 3, and 5 were assumed to spin clockwise and motors 2, 4, and 6 were counterclockwise.

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} (J_{yy} - J_{zz})qr/J_{xx} \\ (J_{zz} - J_{xx})pr/J_{yy} \\ (J_{xx} - J_{yy})pq/J_{zz} \end{bmatrix} + J_r \begin{bmatrix} q/J_{xx} \\ -p/J_{yy} \\ 0 \end{bmatrix} \omega_\Gamma + \begin{bmatrix} \tau_\phi/J_{xx} \\ \tau_\theta/J_{yy} \\ \tau_\psi/J_{zz} \end{bmatrix} \tag{10}$$

Linear motion control was modeled using u_x and u_y as shown in Equation 11.

$$\begin{aligned} u_x &= cos\phi cos\psi sin\theta \\ u_y &= cos\phi sin\psi sin\theta \end{aligned} \tag{11}$$

Acceleration along the xyz space can be represented by Equation 12.

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \frac{T}{m} \begin{bmatrix} cos\phi sin\theta cos\psi + sin\phi sin\psi \\ cos\phi sin\theta sin\psi - sin\phi cos\psi \\ cos\phi cos\psi \end{bmatrix} \tag{12}$$

These equations were then used for the plant model of the control system. These systems were modeled using Matlab and Simulink. The resulting block diagram for the drone's control system was determined to be represented by Fig. 14.

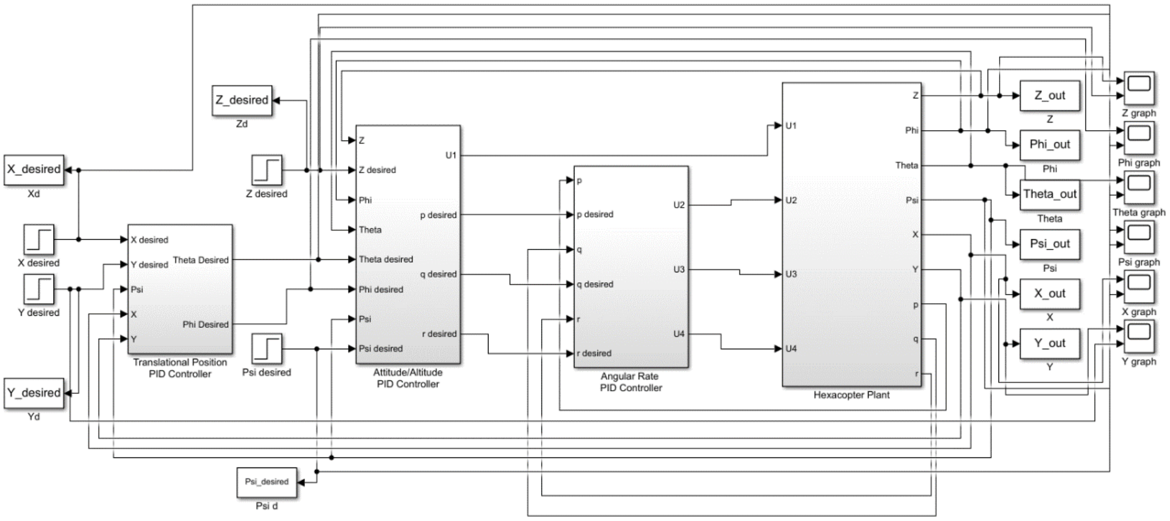


Figure 14: Drone Control Block Diagram

Fig. 15 shows the hexacopter plant in Simulink. Within the plant there were blocks for motor speed, angular velocity translation, rotational dynamics, and translational dynamics.

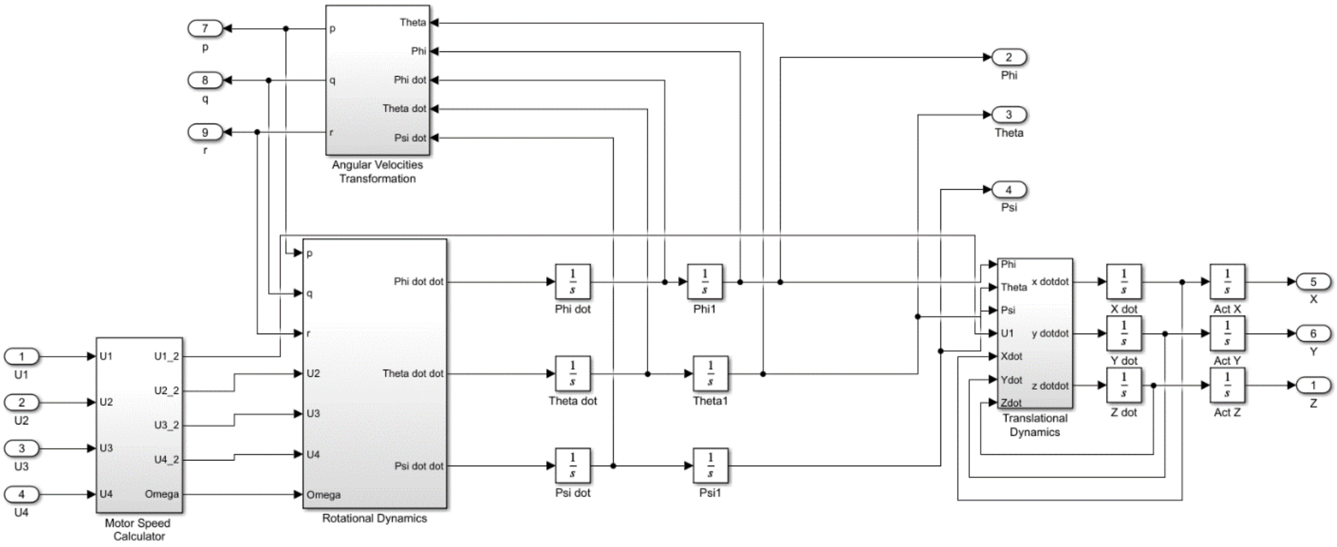


Figure 15: Hexacopter Plant Block Diagram

The proportional, integral, and derivative gains for all PID controllers were found using MATLAB's auto tune function in this case. The simulation time was set to 10 seconds. Using Simulink scopes, the desired and output values were plotted for x, y, z, phi, theta, and psi, shown in Fig. 16-21 respectively. The blue lines represent input/desired values, and the yellow lines represent the output.

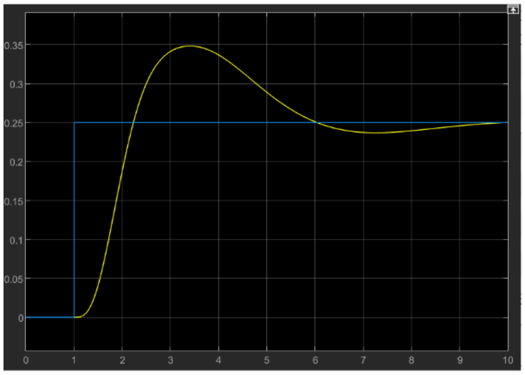


Figure 16: X Input VS Output

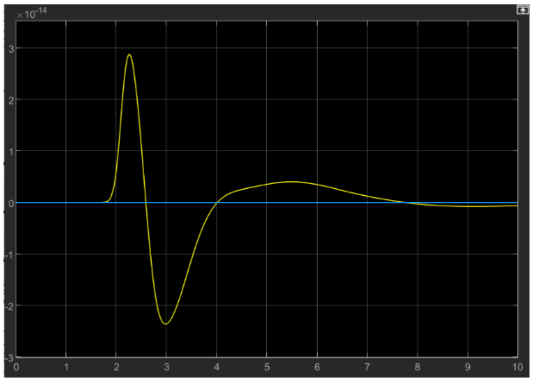


Figure 17: Y Input VS Output

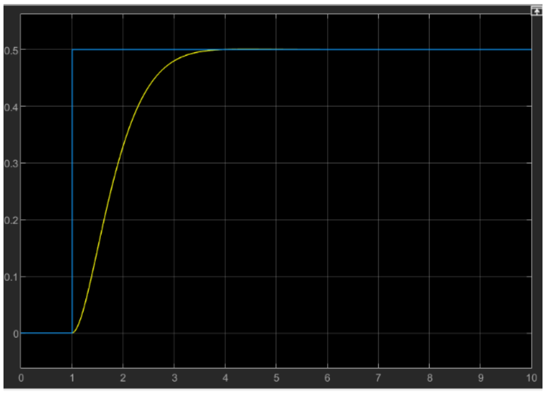


Figure 18: Z Input VS Output

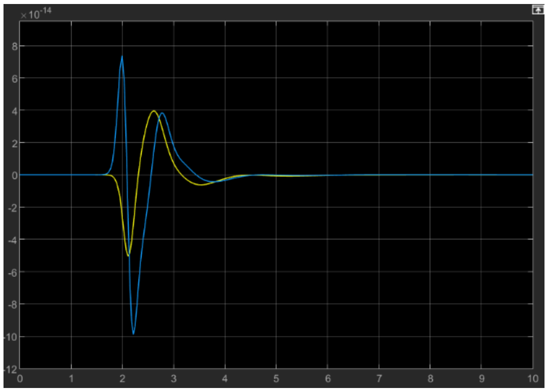


Figure 19: Roll (Phi) Input VS Output

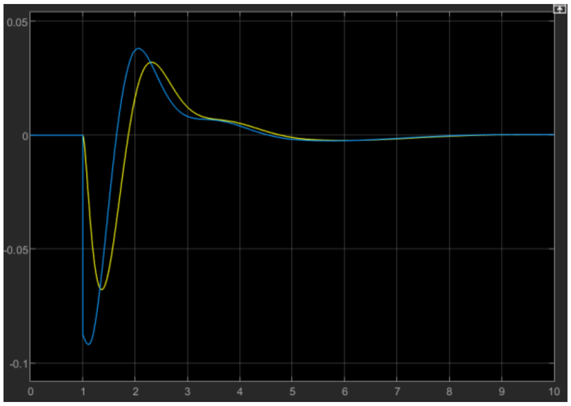


Figure 20: Pitch (Theta) Input VS Output

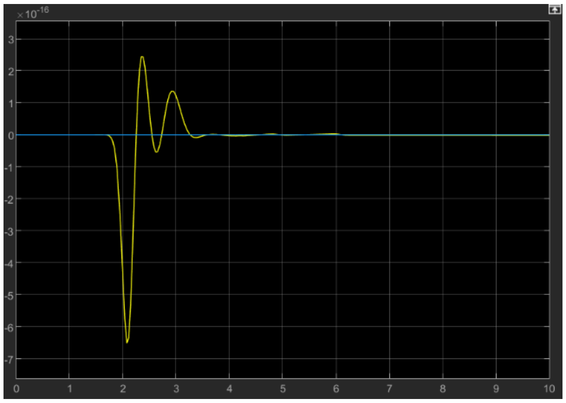


Figure 21: Yaw (Psi) Input VS Output

The model was then discretized based on the speed of 50Hz and the PIDs were tuned. Overall, the model acted as proposed. The settling times for all the components was below 10 seconds, which would be acceptable for a vehicle of this size hovering in the air. This would allow the heavy lift drone to slowly reach a stable steady state position without sharp movements and quick jerks, which would be beneficial when the payload is heavy or fragile.

4.4. Computer Vision

Computer vision is one of several commonly used methods for collision avoidance. With proper calibration computer vision can create an accurate avoidance system that can perform well in an environment with several obstacles. It is a very useful method for this project which will involve a drone that will operate both in open areas and within residential areas that may have several objects that are either stationary or moving inside of the flight path of the drone.

The Raspberry Pi (Rpi) will utilize an open computer vision library, OpenCV, as well as a stereo set of cameras to develop a reliable depth map for collision avoidance. The stereo camera set up was selected because of its well documented ability to accurately measure depth as long it is calibrated appropriately. The below equation describes how a stereo camera system estimates depth.[16]

$$Z = \frac{Bf}{x - x'} \tag{13}$$

Where x and x' represent the projection of a real-world point on the left and right images, B is the distance between the two cameras, f is the focal length, and Z is the depth [16]. An aluminum heatsink was chosen for passive cooling of the Rpi 4. Two Logitech C615 webcams were chosen for the stereo pair because of their video quality and compatibility.

This type of collision avoidance system has been used before, but the Rpi 4 is more powerful than previous versions which should allow for higher frames per second and a more accurate depth map. The collective field of view of the cameras will prevent any gaps in front of the drone that may occur with other sensor arrangements.

Collision avoidance will be achieved by following a multi-step process:

- Calibration – Proper calibration of each camera individually and as a pair is an extremely important step in computer vision. Calibration will be achieved by first taking several pictures of a chessboard pattern with each camera. The pictures must vary in distance and angle to ensure proper calibration. The calibration script will evaluate images taken by both cameras and will discard any images where the chessboard corners cannot be found. These pictures will then be used by three OpenCV functions findChessboardCorners, cornerSubPix, and drawChessboardCorners. The three functions will sort through each picture and determine the internal corners of chessboard, refine the location of the corners down to sub pixel values, and connect the points with a pattern. Any picture that doesn’t pass the initial check of findChessboardCorners will be deleted from the folder via os.remove(). This can be seen in Table 5.
- Stereo Camera Calibration – Once the image points and object points are obtained through the previous functions they can be used as inputs into the calibrateCamera and stereoCalibrate functions. calibrateCamera will return the distortion coefficients and camera matrices of each camera and stereoCalibrate will output the rotation matrix and translation vector using both camera’s parameters. These two outputs effectively bring the points given in the first camera’s coordinate system to the second camera’s coordinate system. Both calibration functions will also return a reprojection error value. This value must be less than 1 in order to have acceptable calibration. The returned values will be stored via numpy.savez() which will allow multiple arrays to be stored in one location. The camera calibration code can be seen in table 6.
- Rectification – Rectification is the process of projecting two images onto a common image plane. This will make the epipolar lines horizontal which will simplify the correspondence problem that disparity mapping must solve. The functions that we will use for this process are stereoRectify and initUndistortRectifyMap followed by remap. These functions will compute rectification transforms and undistortion and rectification maps. The code for this section can be found in table 7.

Table 5: Identify Chessboard Corners

```
import cv2
import numpy as np
import glob
import os

criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 30, 0.001)
objp = np.zeros((6*8,3), np.float32)
objp[:,:2] = np.mgrid[0:6,0:8].T.reshape(-1,2)
objpoints = [] # 3D points in real world space
imgpoints = [] # 2D points in image plane
imgpoints2 = []

images = sorted(glob.glob('/home/pi/Pictures/Cam1Rough/*.jpg'))
images2 = sorted(glob.glob('/home/pi/Pictures/Cam2Rough/*.jpg'))

# Counter for image points. Image points from each camera must be equal
for cv2.stereoCalibrate.
counter1 = 0
counter2 = 0

for fname in images:
    img = cv2.imread(fname)

    # Convert from color to grayscale
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

    # Find the corners of the chessboard pattern
    ret, corners = cv2.findChessboardCorners(gray,(6,8),None)
    if ret == True:
        objpoints.append(objp)

        # Narrow image points down to subpixel level for more accuracy
        corners1 = cv2.cornerSubPix(gray,corners,(11,11),(-1,-1),criteria)
        imgpoints.append(corners1)

        # Connect corners with lines to provide a visual aid for
calibration
        img = cv2.drawChessboardCorners(img,(6,8),corners1,ret)

cv2.imwrite('/home/pi/Pictures/Cam1Corners/cam1c{:1}.jpg'.format(counter1)
, img)
        counter1 += 1
        cv2.imshow('img',img)
        cv2.waitKey(500)
    else:

        # Remove any image where chessboard corners cannot be located
        os.remove(fname)
```

Table 6: Calibration Code

```
# Utilize cv2.calibrateCamera to find camera's intrinsic values
ret,mtx,dist,rvecs,tvecs = cv2.calibrateCamera(objpoints, imgpoints,
gray.shape[::-1],None,None)
ret2,mtx2,dist2,rvecs2,tvecs2 = cv2.calibrateCamera(objpoints, imgpoints2,
gray2.shape[::-1],None,None)

# Save each camera's returned values
np.savez('cam1.npz', mtx=mtx, dist=dist, imgpoints=imgpoints, gray=gray)
np.savez('cam2.npz', mtx2=mtx2, dist2=dist2, imgpoints2=imgpoints2)

# Fix camera's intrinsic values that were previously returned and fix
focal length. Utilize cv2.stereoCalibrate to find extrinsic values for the
camera pair
criteria2 = (cv2.TERM_CRITERIA_MAX_ITER + cv2.TERM_CRITERIA_EPS, 100, 1e-
5)

flag = cv2.CALIB_FIX_INTRINSIC,cv2.CALIB_SAME_FOCAL_LENGTH,
cv2.CALIB_ZERO_TANGENT_DIST
retval, cameraMatrix1, distCoeffs1, cameraMatrix2, distCoeffs2, R, T, E, F
= cv2.stereoCalibrate(objpoints, imgpoints, imgpoints2, mtx, dist, mtx2,
dist2, gray.shape[::-1], criteria2, flag)

# Save extrinsic values
np.savez('stereo_cal.npz', R=R, T=T)
```

Table 7: Rectification Code

```
import cv2
import numpy as np

# Load intrinsic and extrensic camera values
data_cam1 = np.load('cam1.npz')
mtx = data_cam1['mtx.npy']
dist = data_cam1['dist.npy']
imgpoints = data_cam1['imgpoints.npy']
gray = data_cam1['gray.npy']
data_cam2 = np.load('cam2.npz')
mtx2 = data_cam2['mtx2.npy']
dist2 = data_cam2['dist2.npy']
imgpoints2 = data_cam2['imgpoints2.npy']
data_stereo_cal = np.load('stereo_cal.npz')
R = data_stereo_cal['R.npy']
T = data_stereo_cal['T.npy']


# Computes rectification transforms
flag = cv2.CALIB_ZERO_DISPARITY
R1, R2, P1, P2, Q, validPixROI1, validPixROI2 = cv2.stereoRectify(mtx, dist,
mtx2, dist2, gray.shape[::-1], R, T, alpha = -0.5)
```

```
# Computes the undistortion and rectification transformation map to be used by
the remap function
mapX1, mapY1 = cv2.initUndistortRectifyMap(mtx, dist, R1, P1, gray.shape[::-1],
cv2.CV_32FC1)
mapX2, mapY2 = cv2.initUndistortRectifyMap(mtx2, dist2, R2, P2, gray.shape[::-1],
cv2.CV_32FC1)

cap1 = cv2.VideoCapture(0)
cap1.set(4,360)
cap1.set(3,480)
cap2 = cv2.VideoCapture(2)
cap2.set(4,360)
cap2.set(3,480)
MinContourArea = 4000
counter = 0

# Write live video to file
#w = cap1.get(cv2.CAP_PROP_FRAME_WIDTH);
#h = cap1.get(cv2.CAP_PROP_FRAME_HEIGHT);
#writer = cv2.VideoWriter('test.mp4',cv2.VideoWriter_fourcc(*'mp4v'), 15,
(int(w),int(h)))

while True:

    _, frame1 = cap1.read()
    _, frame2 = cap2.read()

    gray_frame1 = cv2.cvtColor(frame1, cv2.COLOR_BGR2GRAY)
    gray_frame2 = cv2.cvtColor(frame2, cv2.COLOR_BGR2GRAY)

    # Find frames per second and dislay it on screen
    timer = cv2.getTickCount()
    fps = ((cv2.getTickFrequency()/(cv2.getTickCount() - timer))*0.0001)
    cv2.putText(frame1, 'fps = ' + str(int(fps)),(0,20),cv2.FONT_HERSHEY_SIMPLEX,
0.75, (0,255,0), 2)


    # Finds interpolated pixel value from gray_frame1 that relates to new
cap1_rectified
    cap1_rectified = cv2.remap(gray_frame1, mapX1, mapY1, cv2.INTER_LINEAR)
    cap2_rectified = cv2.remap(gray_frame2, mapX2, mapY2, cv2.INTER_LINEAR)
```

- Disparity Mapping – OpenCV uses a disparity mapping algorithm called StereoBM. This algorithm uses the sum of absolute differences (SAD) approach to match regions of pixels in one image to another and calculate the difference between them. The size of the region and number of disparities can be adjusted to make it more accurate. This can be seen in Table 8.
- Additional Techniques – Techniques like thresholding, contour detection, and morphology can all be used to increase the accuracy of the disparity map. This can be seen in Table 8.

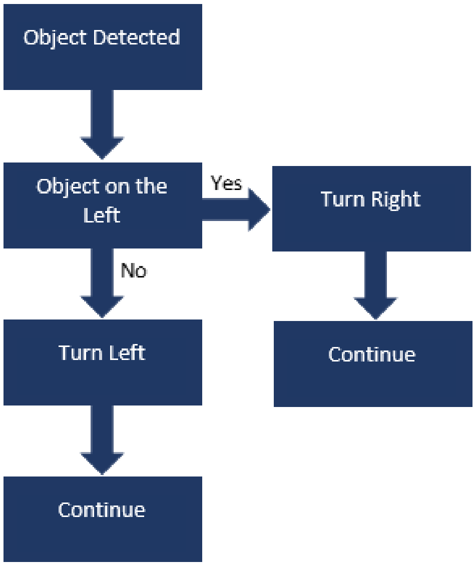
Table 8: Disparity Mapping Code

```
    # Initializes disparity map
    retval1 = cv2.StereoBM_create(numDisparities=96, blockSize=19)

    # Computes raw disparity map
    disparity_map = retval1.compute(cap1_rectified, cap2_rectified)
    rotate_disparity = cv2.rotate(disparity_map, cv2.ROTATE_180)

    # Adjusts pixel intensity to create a clearer image
    convert = cv2.normalize(rotate_disparity, None, 0, 255, cv2.NORM_MINMAX,
dtype=cv2.CV_8U)

    # Converts values to binary
    ret, thresh = cv2.threshold(convert,127,255,cv2.THRESH_BINARY)

     # Dilate, blur, and find contours
    blur = cv2.medianBlur(convert, 1)
    kernel = np.ones((3,3),np.uint8)
    dilate = cv2.dilate(blur, kernel, iterations=1)
    contours, hierarchy = cv2.findContours(dilate, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
```

- Object Detection and Avoidance Algorithm – Due to the size and purpose of the drone the avoidance algorithm will focus primarily on turning left or right. It will rely on accurate depth perception and object detection. If an object is considered close the algorithm will determine if that object’s centroid is on the left of the screen. If the object’s centroid is on the left the system will alert the drone to turn right and if not, the system will alert the drone to turn left. The code for these processes can be seen in Table 9. The algorithm can be viewed as a flowchart in Figure 22 and a test example is shown in Figure 23. Object detection is sensitive to lighting conditions, object size, and angle of view. A large object can be detected at ranges of up to 2 meters in a situation with good lighting. The angle of view is more difficult to predict as the block matching algorithm used for the disparity map can vary significantly frame by frame. Object detection and avoidance were able to operate at an average of 15 frames per second.

Table 9: Object Detection and Avoidance

```
    for c in contours:

        if cv2.contourArea(c) < MinContourArea:
            continue

        # Generates a bounding rectangle for a contour area bigger than
MinContourArea
        (x, y, w, h) = cv2.boundingRect(c)
        cv2.rectangle(frame1, (x, y), (x + w, y + h), (0, 255, 0), 2)

        # Generates the object's centroid
        CoordXCentroid = (x+x+w)//2
        CoordYCentroid = (y+y+h)//2
        ObjectCentroid = (CoordXCentroid,CoordYCentroid)
        cv2.circle(frame1, ObjectCentroid, 1, (0, 0, 0), 5)

        # Display concept behind avoidance algorithm. Screen is split into right
and left sides with message displayed depending on location of centroid.
        if CoordXCentroid < 320:
            cv2.putText(frame1, 'TURN RIGHT',(150,200),cv2.FONT_HERSHEY_SIMPLEX,
1, (0,255,0), 3)
        elif CoordXCentroid >= 320:
            cv2.putText(frame1, 'TURN LEFT',(0,200),cv2.FONT_HERSHEY_SIMPLEX, 1,
(0,255,0), 3)

    cv2.imshow('Left', frame1)
    cv2.imshow('Right', dilate)
    #writer.write(frame1)
    k = cv2.waitKey(32)

    # Save images from video by pressing 't' and break with spacebar.
    if k == ord('t'):


cv2.imwrite('/home/pi/Pictures/test_images/object_detection{:1}.jpg'.format(count
er), frame1)

cv2.imwrite('/home/pi/Pictures/test_images/disp{:1}.jpg'.format(counter),
convert)
         counter += 1

    elif k == 32:
        break

cap1.release()
cap2.release()
#writer.release()
cv2.destroyAllWindows()
```



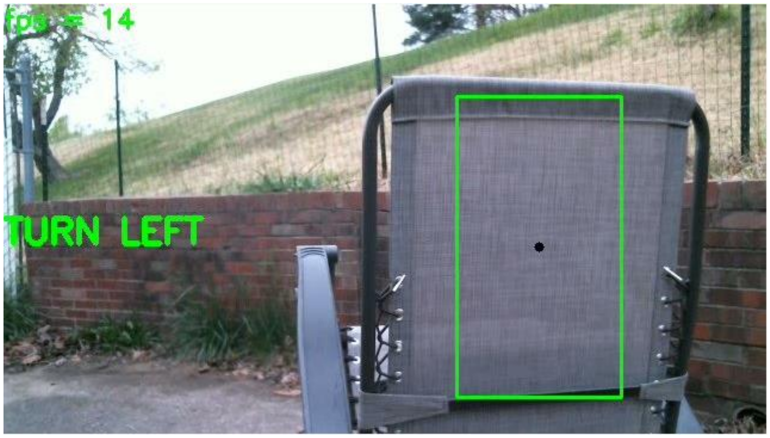


Figure 22: Avoidance Algorithm *Figure 23: Detection and Avoidance*

4.5. Camera Mount System

To implement the computer vision properly, there was need for a system to mount the web cameras in stereo onto the drone. The requirements for this structure were that it needed to be light weight, sturdy, able to mount the cameras horizontally, and provide a way to give stability to the cameras. This was achieved by taking measurements of the assembled drone frame to design a connection. The team decided that the best way to mount the cameras would be by using the rails that were underneath the frame which were intended to mount auxiliary parts to. These measurements were taken, and a 3D design was drawn.

The mounting bracket was then designed by knowing the required space to mount 2 cameras, 3 inches each, and providing enough room so that they would fit in between the all-thread. Additionally, there was a slit added in the middle of the plate so that the camera mount would fit and tighten down to, the bottom plate was made 20mm thick, which would allow for enough space for the camera mount to grip the plate firmly.

Next, a plate was made to produce pressure on the top of the cameras so they would not move during flight. This plate was made half the thickness of the bottom plate and extended by 25mm towards the front so that pressure could be achieved on the top of the camera when it was mounted. The plates were then held together by ¼-20 all-thread cut into 12” sections. This setup can be visualized in a CAD representation in figure 22. All files were then 3D printed at 15% infill and mounted on the drone.

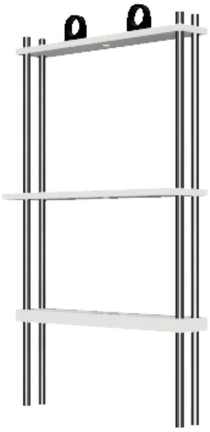


Figure 22: Camera Mount System CAD

4.6. Payload Delivery System

The PDS is integral to the construction of a delivery drone because it will safely deliver the attached payload (package). This system will include a pick point, hooking device, and winch system to raise and lower the payload when it is loaded and delivered [2]. The system allows the drone to decrease in altitude to a height between 10-15m above ground level, and slowly and safely lower the payload at the delivery location. The winch system will require a motor, spool, cable, hook, pick point, a winch box, and a mounting system to attach to the drone's undercarriage.

The hook and pick point, shown below, were designed simultaneously. The pick point design was taken directly from Aero Drone Package Delivery System [2] and altered in size to suite the project's needs. This part was printed with white IC3D high temperature PLA at 20% infill. The hook was modeled after a simple eye grab hook, the hook used with metal chain. The mouth of the hook was made 10% larger than the diameter of the pick point's peak. The hook was then made into a rounded shape to stabilize and reduce the effects of wind on the object. The hook was then printed at 50% infill to account for the bottleneck section's weigh distribution. The cross-sectional area was found to be 0.34" based on the CAD file, and the rated tensile strength of the PLA was determined to be 60MPA (8702.26 lbs/in^2). The infill was 50% and the proposed weight was doubled for safety measures. The proposed pounds per square inch for the smallest cross-sectional area for the hook was calculated according to equation 14.

$$(2 * 10lbs)/(50\% * 0.34in^2\ ) = 117.65\ lbs/in^2 \tag{14}$$

It was then determined that the designed hook was well within the range of safety, with only 117.65 lbs/in^2. The 3D printed pick point and hook can be seen below in Fig. 23 and 24.

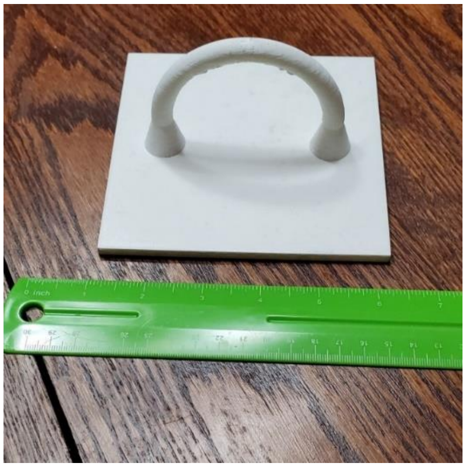


Figure 23: Pick Point

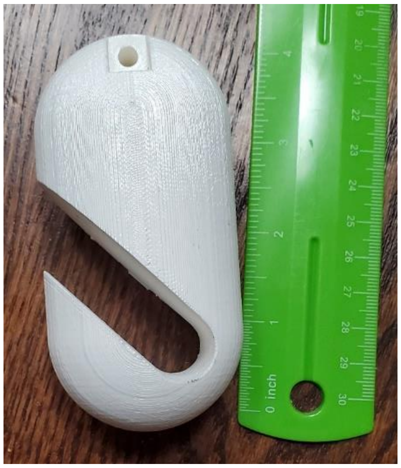


Figure 24: Hook

After a first iteration of a hook and pick point, the motor was to be chosen. This was designed by using the proposed lifting weight of 10lbs (2.3kg) and applying a lift speed of 0.25m/s. A spool radius was changed to determine the best achievable lift. This was achieved by changing the value in newton's laws of motion for lifting an object. Is outlined below

The process to determine the motor was [15]*:*

1. Determine the force of gravity acting on the mass that is being lifted. A safety factor of twice the mass is used to account for safety.

$$F = 2 * 10kg * 9.8\, m/s^2 \tag{15}$$

2. Use the calculated force in the above equation and multiply it by the radius. This value was changed to suit the needs of the project. It was determined that distance from the center of the spool wall would be used instead of the inner dimension of the spool to allow for an average applied torque.

$$\tau = F * r \tag{16}$$

3. Next convert from Newton-meters to kilogram-centimeters

$$(\tau\, Nm)/(0.0980665\, Nm) * 1\, kgcm = x\, kgcm \tag{17}$$

4. The obtained value would be the minimum acceptable torque of the selected motor if the weight of the spool and cable were neglected.

An optimal radius was initially found to be 40mm with a spool wall of 5mm to contain the proposed metal cable. However, in initial testing, the proposed spool was unable to lift a 10lbs package and would unspool whenever there was no load applied to the attached hook. Because of this, the spool was redesigned with 110lbs test rope. The inner spool diameter was also decreased to 10mm and a wall height of 25mm and the width of the spool was increased to contain the rope. The spool was printed with an infill of 50%. The finished 3D print of the spool is shown in Fig. 25. This reduction in radius allowed more torque to be achieved by the motor.

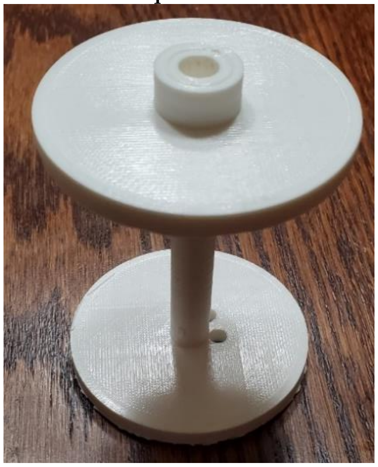


Figure 25: Spool

However, in testing, this design had repeated high failure and the part was reprinted so that the inner radius was solid. This adjustment eliminated this type of failure in testing. The failure is pictured below in Fig. 26.

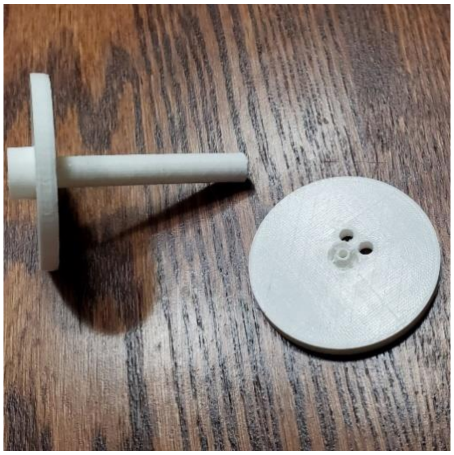


Figure 26: Spool Failure Point

From the above equation, a motor was selected based on its torque output at distances. These values are used to show how much torque a motor produced at a certain radius. For the original spool, it was found that the necessary motor to suit the needs of the project was a 60kg motor. A 360 servo was chosen due to its ability to be used without adding an ESC to the project and its ability to be interfaced with the flight controller. When the radius of the spool was reduced, to eliminate additional costs, the same spool was used even though it was overrated for the required task.

After testing, it was realized that the package would need a way to be stabilized during flight, and a system would need to be designed to attach the motor to the under carriage of the drone. To reduce the amount of weight of the system, these systems were integrated together and referred to as the winch box. It was also determined that these parts would need to be 3D printed to rapidly prototype and to reduce weight. Originally, a single box was designed that would be able to connect to the drone, securely attach the servo motor, and provide stability in flight. This was based on measurements of the components and of the drone's undercarriage. Two rails were available underneath for mounting equipment on the drone, so these rails were utilized. An additional part was designed for stability to firmly secure the pick point to the underside of winch box. This part was designed by making a hollow cylinder with the inner diameter of 110% that of the hook's outer diameter. This was then made so that the hook could be reeled into a portion of the winch box and would allow the pick point to be seated in a cut angle of the hollow cylinder, which will now be referred to as the can. The original design of the winch box can be seen below, in Fig. 27.

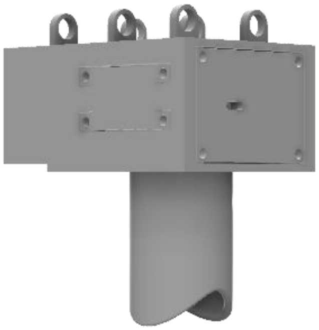


Figure 27: Original Winch Box

Once designed in CAD, the separate parts were placed in a slicing software to prepare for 3D printing. It was determined that the proposed design was not optimal for printing and would need to be reevaluated to reduce part failure during prints. Due to this, the part had to be re designed into 8 separate parts. The mounting system was deemed to need to be printed separately and needed holes placed along the edges for assembly. The constructability, placing the motor and spool inside box, was found to be suboptimal. The removal of one of the walls was also deemed necessary to increase the printability of the part. To help with this, one of the walls was to be printed as a separate part and assembled later and two plate were to be constructed separately to assist with motor placement. The two plates were proposed to be inserted into the opposite side of the missing wall and would more firmly hold the motor in place. A spool guide was also seen as a necessary addition to the system since it would be able to help the spool reel properly during use. The can was also redesigned to allow for the new mounting and assembly holes. The updated design's front and back views as well as a blown-out picture can be seen in Fig. 28, 29, and 30 respectively.

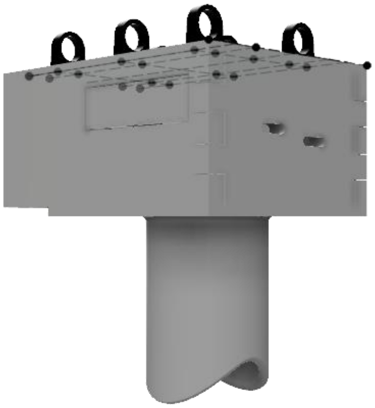


Figure 28: Redesigned Winch Box - Front

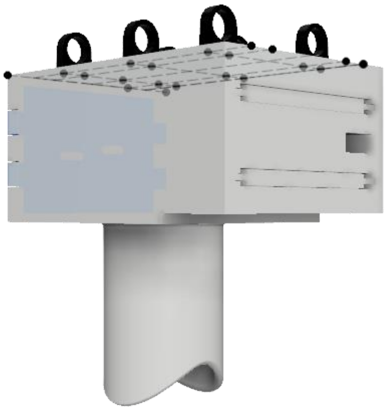


Figure 29: Redesigned Winch Box - Back

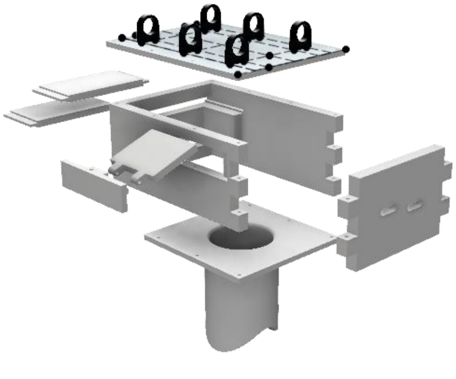


Figure 30: Redesigned Winch Box - Blown Out

The device pictured above is currently in production and is expected to be finished within 1 week. This system was attached to the runner rails beneath the drone and the motor was connected to a battery eliminator circuit which will step down the 22.2V from battery to the 7.4V working voltage of the motor. The ground wire was run in parallel to the ground of the battery eliminator circuit and to the auxiliary port 1 ground on the Pixhawk. The signal wire was attached to the auxiliary port 8 of the flight controller and an up and down button was programmed onto the remote’s channel 6 so that it could be manually controlled. During flight tests, the 3D printed spool was found to strip on the rotor of servo motor when reeled tightly. To counteract this, an M3 nut was inserted into the spool on each side of the spool. This was determined to resolve the issue with the spool.

4.7. Assembly

The construction of the drone took place over several days with different members of the team taking on different roles, Tom oversaw most of the soldering, Bailey and Caleb oversaw the construction of the physical drone, and Pete oversaw the design of the camera mounts for the addition of Computer Vision. The assembly itself had no large issues only a minor issue of space inside of the drone, which was fixed by 3-D printed spacers that allowed an additional inch of room for wiring and electrical components.

The construction of the drone began with the frame. This was done by assembling multiple carbon fiber rods and screwing the top and bottom plates all the electrical equipment was attached using zipties, electrical tape, or velcro. The twelve motor mounts were then assembled and positioned onto the arms.

Next, the motors were mounted, and the electrical connections were soldered. The motors were attached through the use of the motor mounts that were included in the frame kit. The electrical components were then soldered. These components include the power distribution board, ESC's, power distribution cable, and the signal wires from the ESC's. XT-90s connectors were used for all main connections to PDB's Pixhawk power, and winch power. These were used based on their nominal current ration of 90A. Each PDB had an individual connection to the main power cable wired in parallel. The flight controller and BEC for the winch were wired in the parallel configuration as well. The ESC's were connected to each port on the PDB. They were connected so that the top motors on arms 3, 4, 5, and 6 were connected to the top PDB. The bottom motors of arms 3, 4, 5, and 6 were connected to the middle PDB. The motors on arms 1 and 2 were ten connected to the bottom PDB. This was done so that the top X configuration of motors were on a single board, the bottom X configuration were connected to a single PDB, and the middle arms were connected to a single PDB. This was proposed to evenly distribute the power distribution of each PDB. This was due tot the fundamental knowledge that the middle motors were the average between the front and back of the X configuration, so all motors with the average current draw were placed together. Additionally, while moving forward, the here would be more power needed for the back motors on the X configuration and less power for the front, thus the power needed would be averaged between that of the front and back motors of the X configuration. Additionally, if a PDB were to fail, there was potential for reduced control that would allow the pilot to take control and land the drone. Due to the flight controller's specifications of having only 8 ports, the ESC's on each arm had their signal wires connected such that only 6 ports were used on the flight controller as was necessary for the hexacopter configuration. Once the components were soldered together, they were placed on the bottom plate of the frame and arranged so space was conserved. To reduce the potential of wiring pushing on the flight controller, the wires were run through the frame. The motor wires were run through the center of the carbon rods of each arm and connected to the ESC's.

Once the electrical components were connected, the Pixhawk flight controller was attached to the center most point of the drone. The controller was to be the the central control system for the drone including flight control, position data recording, and the controller for the PDS. Once the ESC's and the power distribution boards were connected to the power cable and the Pixhawk, along with the Telemetry for wireless control and the GPS. The top plate of the drone and the battery mounts were then installed. The drone was then ready to begin initial tests.

5. Design Evaluation, Demonstration, & Recommendations

5.1. Computer Vision Test

Stereo vision depends on having two cameras that are mounted at a fixed baseline apart with their lenses in parallel. Multiple designs were considered before deciding on a dual camera bracket where the cameras can screw onto the base and a plate can be screwed on top of the cameras to hold them still.

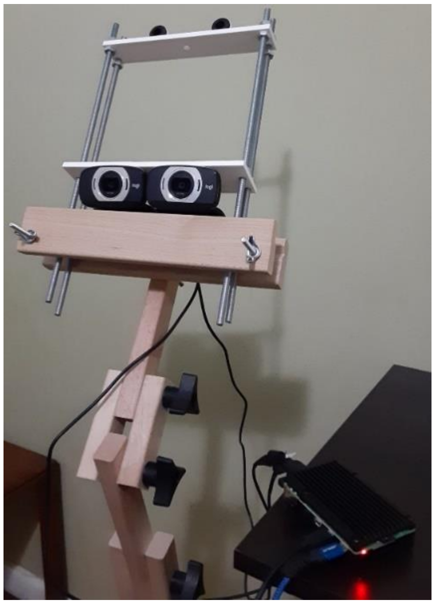


Figure 31: Cameras, Mount System, & Raspberry Pi 4

The calibration pattern used for the cameras is a 6 by 8 chessboard pattern with 40mm squares. Good calibration was achieved with this setup as individual camera calibration and calibration of the pair achieved a less than 1-pixel reprojection error. The quality of the final disparity map was determined through fine tuning of numerous OpenCV functions until the object detection could effectively locate objects over a minimum contour size. The map is tested indoors and outdoors, but for outdoor use the cameras are connected to a laptop for portability.

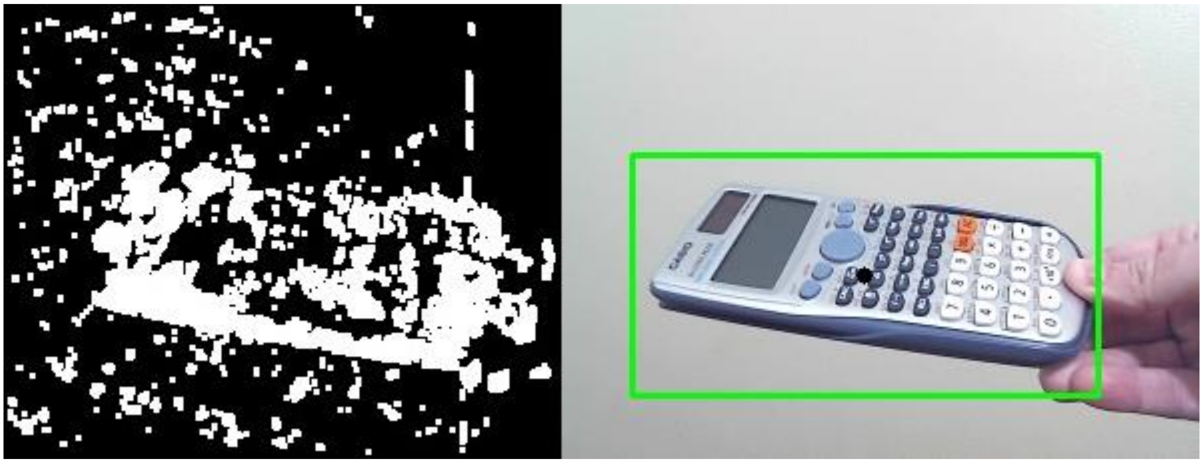


Figure 32: Disparity Map w/ Additional Processing (left) & Object Detection (right)

The system performs as expected, but further adjustments must be done to remove a high degree of noise contamination that threatens the accuracy of the collision avoidance system. Noise can be caused by the sensor or the environment and further testing outdoors will be necessary to adequately replicate the environment that the drone will be operating in.

5.2. Payload Delivery System Testing

The PDS required preliminary testing to verify the design would work before mounting it to the drone. The system was needed to be tested for torque produced by the motor, and if the setup would be able to pick up and release the payload.

To achieve this, the parts were tested for the torque produced by the motor with the rope wound around the spool. It was determined that the produced torque was 10.61kg (23.3lbs) which satisfied the requirement of double the potential weight. The test rig needed to also be constructed for this. It was made from materials that were readily available and did not require additional parts to be purchased. The test used an Arduino and an external 7.4V power source to power the motor. A circuit was constructed with 2 push buttons which controlled the rotation of the spool, clockwise and counterclockwise. The counterclockwise motion was used to determine the force produced by the motor through the rope and spool. An example of this test is pictured in Fig. 33.

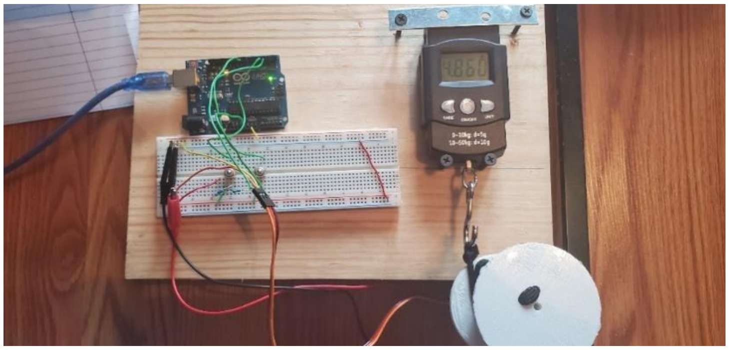


Figure 33: Torque Test

Additionally, the hook and pick point set up needed to be tested. A separate rig 5 ft tall was constructed to model the drone hovering. A package was attached using 30lbs tape attached to the pick point. The package was filled to 10lbs and was lifted and lowered. During this test, it was found that the motor configuration was able to lift and hold the 10lbs package. The hook and pick point setup were also tested for their success rate. It was found that out of 50 trials, the setup failed to release the package only twice. One time was just a system failure, and the second time, the package did not land directly on the ground, but rater on of the legs of the rig. It was determined that the package did not release due to it not being level with the ground, causing the system to fail. Based on the test, it was found that out of 50 trials, the setup worked properly 96% of the time, which was an increase from the mechanical hook in the Aero Drone Package Delivery System [2].

The system was then attached to the drone and tested to ensure proper integration. During testing, it was found that the motor tended to rotate in the clockwise direction when initially connected to the Pixhawk's AUX 1 port. To counter act this the trim, or middle operating value, was lowered on the remote control to ensure that the spool would not move upon powering the drone and, also when the remote-control switch assigned to control the motor was placed in the middle position. Once this was achieved, a test of the PDS's ability to lift a 12lbs payload was attempted. This can be seen below in Fig. 34.

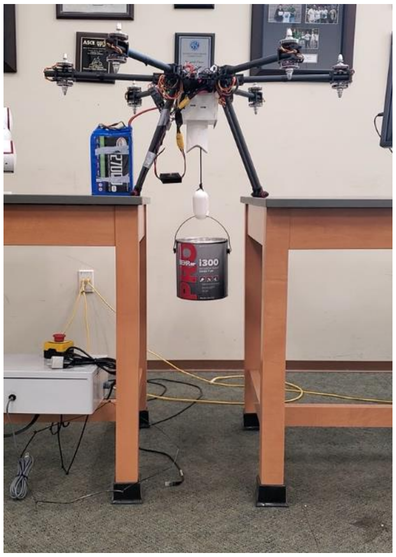


Figure 34: PDS Integration with 12lbs Payload

The test results achieved by the PDS are outlined in Table 10.

Table 10: PDS Results

Test Type	Pass/Fail	Measured Value
Torque Test	Pass	2.6 Nm
10 lbs Lift Test	Pass	100%
Release Test	Pass	96%

5.3. Flight Test

The team first tested the drone motors individually. The default thrust for a motor test was 5%. The motors would not spin, so the throttle was increased in 1% increments until the motors were all at a steady speed. This occurred at 16% throttle. A piece of tape was put onto all the motors to ensure that each one spun in the correct direction. Fig. 35 shows the initial motor test setup. For this test, propellers were off, and a wired connection was used.

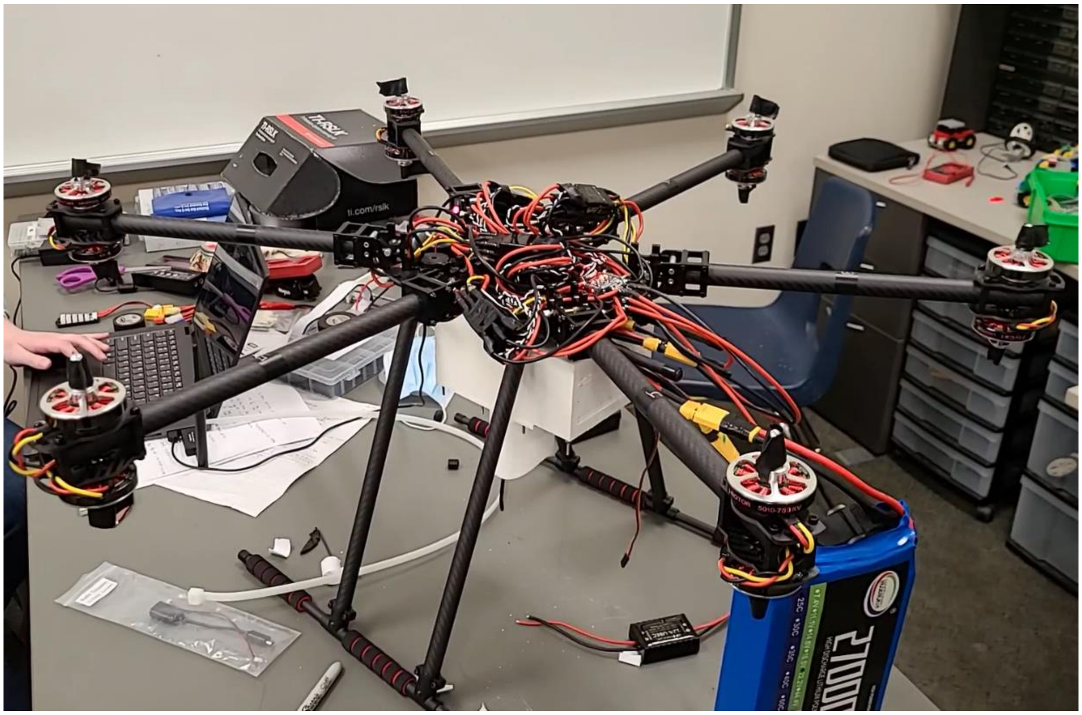


Figure 35: Initial Motor Test

Next the propellers were screwed onto the motors and the drone legs were put into cinderblock slots so that it could not lift into the air. The drone was wirelessly connected this time. As the throttle was increased, a couple of the propellers flew off. The team took the propellers off and added adapter rings and screwed the propellers on tighter. The grounded test was then redone. The grounded test can be seen in Fig 36. During this test, the drone communicated correctly with mission planner and no propellers came off.

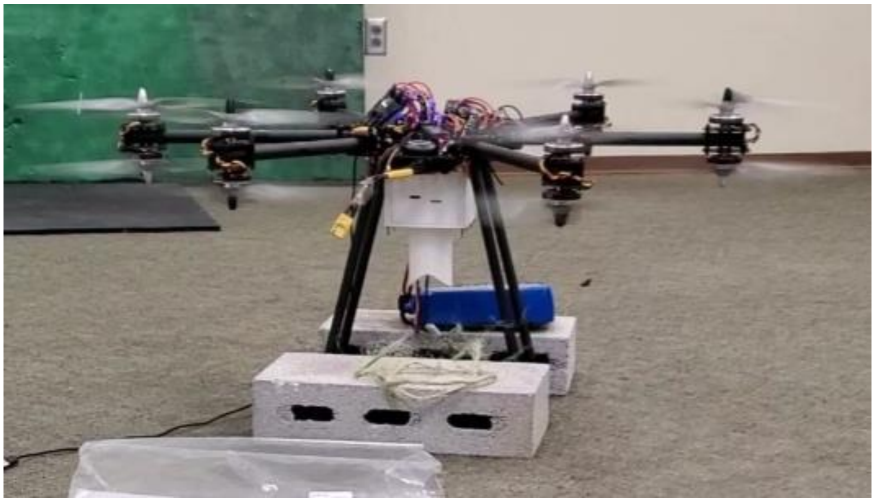
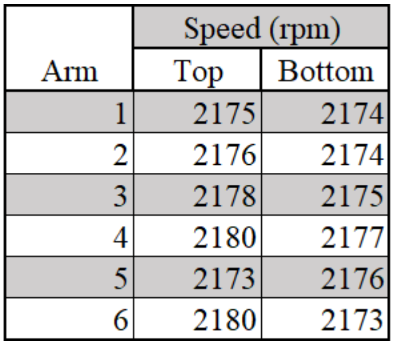


Figure 36: Grounded System Test

After ensuring that the drone responded properly with the computer, the team attempted a first hover. During this first hover attempt, a capacitor blew. This capacitor was part of an electronic speed controller. When the capacitor blew it made an arm of the drone stop spinning and caused an isolated fire near the faulty component's location. This in turn caused the drone to flip, melted the coating of nearby wires, and rendered the original GPS inoperable. As a result, 3 propellers were broken, and the main power cable was severely damaged. A replacement ESC was ordered. The component was then replaced. After several tests, it was discovered that the original ESC's continued to stop working. They were replaced with higher quality ESC's. There are no problems with the new ESCs. During the next hover attempt, the drone would start to tip. It was secured in cinderblocks to prevent damage. Using slow motion video analysis, it was discovered that 5 of the 12 motors were spinning in the wrong direction. Either the motors or ESC's had the polarity incorrectly labeled. Because the motors are DC, the polarity can easily be switched by swapping the ground and power wires. After swapping the polarity of 5 of the motors, the drone successfully achieved flight.

The motor speeds were tested using a tachometer to ensure that each of the 12 motors are spinning at roughly the same speed. The speed measured was the idle speed. Table 11 contains the speeds of the motor. This speed is at 16% throttle.

Table 11: Measured Motor Speeds at 16% Throttle



Arm	Speed (rpm)	
	Top	Bottom
1	2175	2174
2	2176	2174
3	2178	2175
4	2180	2177
5	2173	2176
6	2180	2173

In order to protect the drone during outside flight, we attached pex tubing to the legs so that we could grab it if it started to flip. Fig. 37 shows the drone being tested with the pex tubing attached.

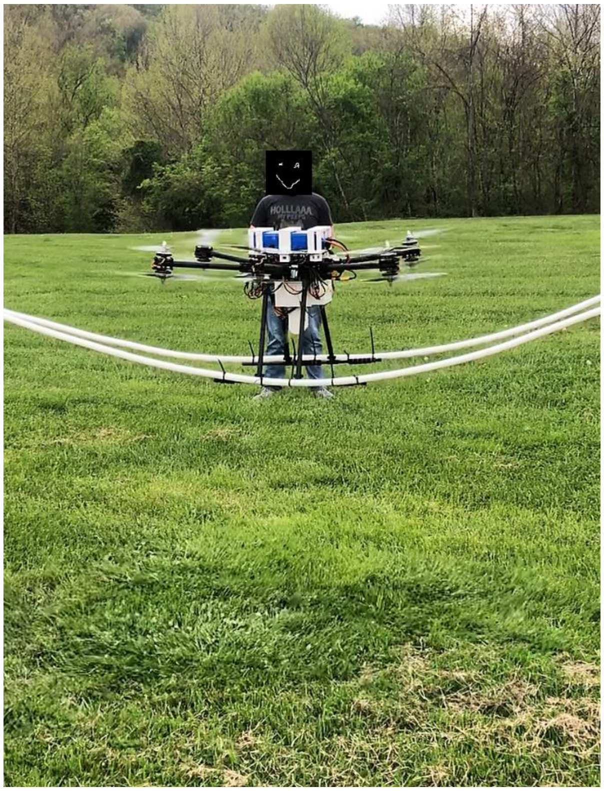


Figure 37: Flight Test with Pex Protection

Fig. 38 displays the drone in mid-air.

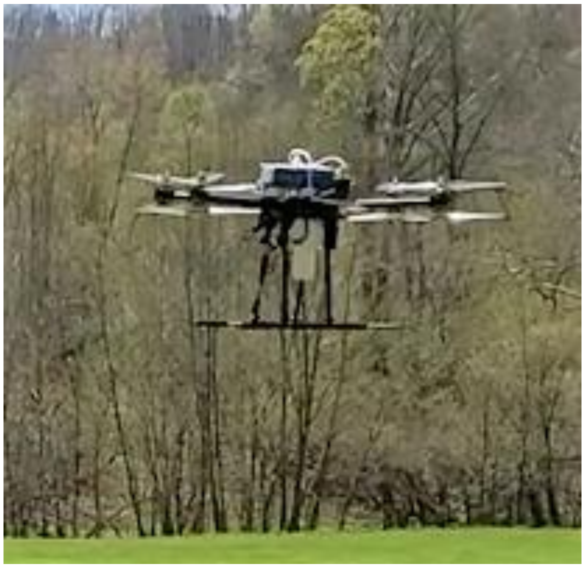


Figure 38: Flight Test Mid-Air

Table 12 below shows the results of the flight test.

Table 12: Flight Results

Test	Pass/Fail
Telemetry Connection	Pass
GPS Lock	Pass
Motors	Pass
Payload Delivery	Pass
10 lb Payload Minimum	Pass

5.4. Results

The results the team achieved were as follows:

1. The object avoidance system was well calibrated and able to determine which objects were closer in its field of view and then identify them.
2. The PDS was capable of providing enough torque to lift a 23lbs package.
3. The PDS was easily loaded through use of the pickpoint and hook system.
4. Once attached, the PDS provided in-flight stability to the attached package.
5. The package was able to be unloaded at the destination with no human interaction as designed.
6. The drone connected to the computer via telemetry without losing connection.
7. The drone achieved a GPS lock.
8. The drone achieved a stable takeoff and landing.

This was all completed while achieving a total cost of $1,552.37, 11.7% of the cost of Heavy Lift Drone [12] project’s cost of $13,291.

5.5. Recommendations

- Propeller Guards –

  This drone could be further improved by adding propeller guards. This would improve the safety of the drone and reduce the chance of a propeller cutting wires. It would also prevent propellers for breaking. Propeller guards are expensive to buy. Instead, they could be designed using CAD software and 3-D printed.

- Drone Frame –

  One of the issues the team faced was that the package was too tall to fit the clearance underneath the drone and PDS without being raised up on cinderblocks. Construction of an additional rig or the addition of longer and sturdier legs to assist with package attachment to accommodate a wider variety of package dimensions. Carbon fiber tubing can be purchased and cut to a better length. The frame also lacks durability. Some adjustments should be made to improve its design.

- Flight Stability –

  Because of the limited space underneath the drone, the battery had to be placed on the top plate. While this alone will not cause the drone to flip, it can make it more likely to flip if a gust of wind hits the drone during flight or the flight controller over corrects. This could be fixed by moving the battery in between the bottom plate of the frame and the PDS.

- Autonomous Flight –

  In order to integrate the obstacle avoidance, autonomous flight must be achieved. The biggest piece of further work would be to add autonomous flight. The team attempted autonomous flight and found that the current flight controller and GPS do not work accurately enough to achieve a stable autonomous flight. The flight controller and GPS should be replaced with a higher quality name brand product.

- Telemetry –

  Currently the telemetry can communicate at a distance of 500 meters away from the ground station. The antennas could be upgraded to accommodate a long distance delivery.

6. Project Timeframe

The project was outlined as seen in Table 13.

Table 13: General Team Tasks

General Tasks	Start Date	End Date
Research	9/6/2020	10/24/2020
Determine Components	9/27/2020	10/31/2020
Design/Test Payload System	10/4/2020	12/12/2020
Test Payload System	12/13/2020	1/9/2021
Design Object Avoidance system	10/11/2020	12/12/2021
Calibrate object avoidance	12/13/2020	1/9/2021
Construct Drone	1/10/2021	2/20/2021
Test and Refine Drone	2/21/2021	4/15/2021

It was planned that the team will construct the drone in January, once parts are received. Upon completion, the vehicle would be refined until April 15th, when it expected to be completed and the team would be able to focus solely on writing final papers for the project. The timeline can be visualized by correlating Table 13 with Tables 14 and 15. Additionally, specific milestones have been planned out for the spring semester and can be seen in Tables 16 and 17.

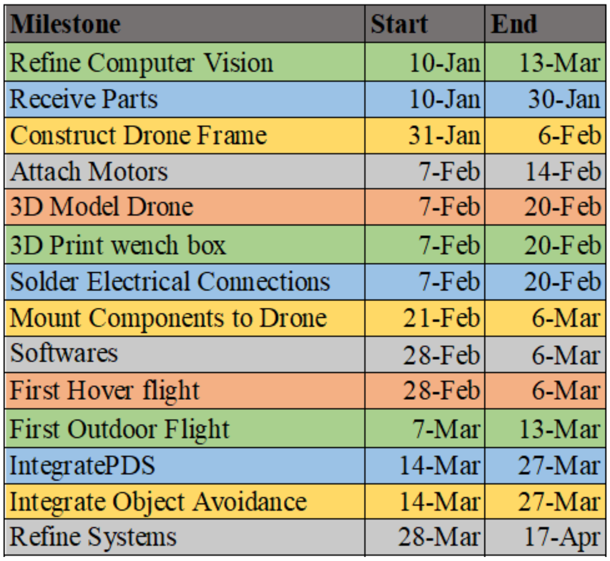
Table 14: Fall 2020 Schedule

6-Sep	13-Sep	20-Sep	27-Sep	4-Oct	11-Oct	18-Oct	25-Oct	1-Nov	8-Nov	15-Nov	22-Nov	29-Nov	6-Dec	13-Dec	20-Dec
🟩	🟩	🟩	🟩	🟩	🟩	🟩									
			🟦	🟦	🟦	🟦	🟦								
				🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨		
					▨	▨	▨	▨	▨	▨	▨	▨	▨		
														🟧	🟧
														🟩	🟩

Table 15: Spring 2021 Schedule

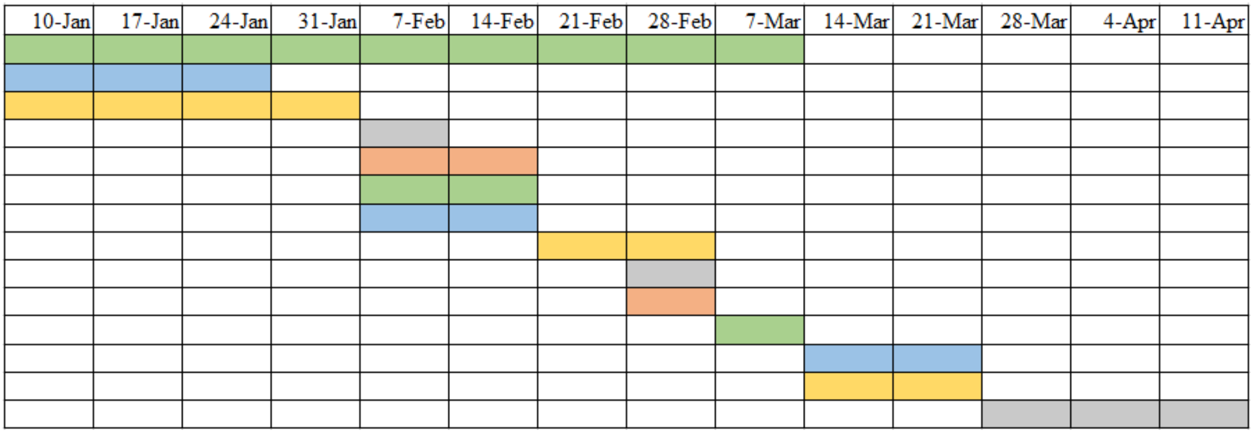
27-Dec	3-Jan	10-Jan	17-Jan	24-Jan	31-Jan	7-Feb	14-Feb	21-Feb	28-Feb	7-Mar	14-Mar	21-Mar	28-Mar	4-Apr	11-Apr
🟧	🟧														
🟩	🟩														
		🟦	🟦	🟦	🟦	🟦	🟦								
								🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨

Table 16: Spring Milestones



Milestone	Start	End
Refine Computer Vision	10-Jan	13-Mar
Receive Parts	10-Jan	30-Jan
Construct Drone Frame	31-Jan	6-Feb
Attach Motors	7-Feb	14-Feb
3D Model Drone	7-Feb	20-Feb
3D Print wench box	7-Feb	20-Feb
Solder Electrical Connections	7-Feb	20-Feb
Mount Components to Drone	21-Feb	6-Mar
Softwares	28-Feb	6-Mar
First Hover flight	28-Feb	6-Mar
First Outdoor Flight	7-Mar	13-Mar
IntegratePDS	14-Mar	27-Mar
Integrate Object Avoidance	14-Mar	27-Mar
Refine Systems	28-Mar	17-Apr

Table 17: Spring Milestone Schedule

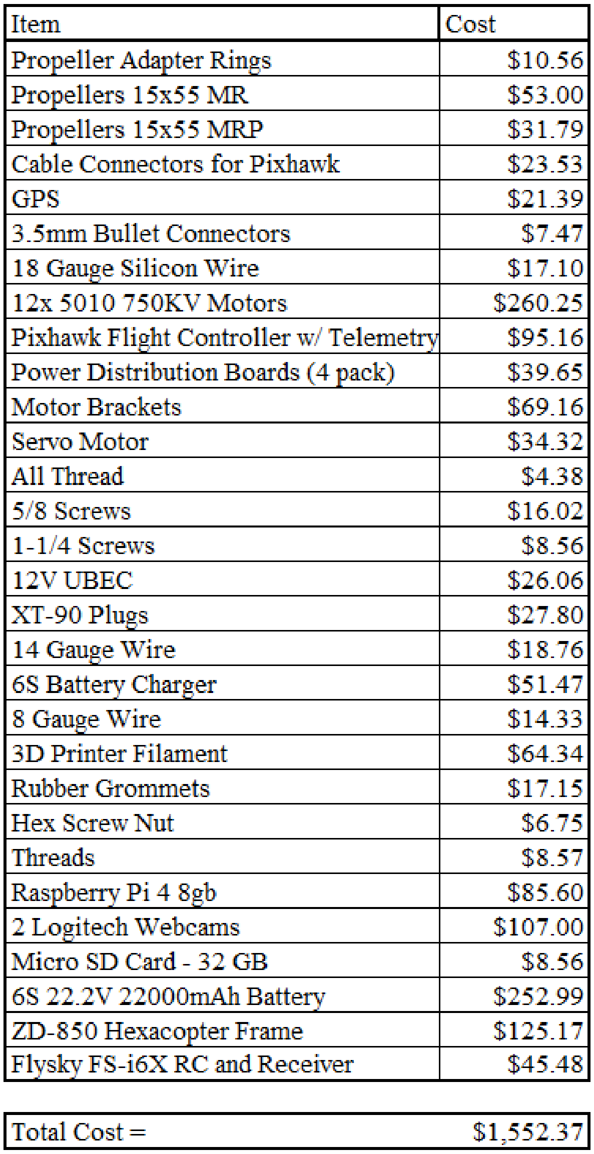


10-Jan	17-Jan	24-Jan	31-Jan	7-Feb	14-Feb	21-Feb	28-Feb	7-Mar	14-Mar	21-Mar	28-Mar	4-Apr	11-Apr
🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩					
🟦	🟦	🟦											
🟨	🟨	🟨	🟨										
				▨									
				🟧	🟧								
				🟩	🟩								
				🟦	🟦								
						🟨	🟨						
							▨						
							🟧						
								🟩					
									🟦	🟦			
									🟨	🟨			
											▨	▨	▨

7. Cost Estimate

Table 18 shows the projected overall cost of the project. The cost will be potentially subsidized by grants through the College of Engineering and Computer Sciences, and Marshall University which the students applied for. The team has received a $250 grant from Marshall University and the remainder of the funds were secured by Marshall University’s Computer Science and Electrical Engineering department with the agreement that the project would be retained by the department upon completion to be used as a recruiting tool for future students.

Table 18: Cost Estimate



Item	Cost
Propeller Adapter Rings	$10.56
Propellers 15x55 MR	$53.00
Propellers 15x55 MRP	$31.79
Cable Connectors for Pixhawk	$23.53
GPS	$21.39
3.5mm Bullet Connectors	$7.47
18 Gauge Silicon Wire	$17.10
12x 5010 750KV Motors	$260.25
Pixhawk Flight Controller w/ Telemetry	$95.16
Power Distribution Boards (4 pack)	$39.65
Motor Brackets	$69.16
Servo Motor	$34.32
All Thread	$4.38
5/8 Screws	$16.02
1-1/4 Screws	$8.56
12V UBEC	$26.06
XT-90 Plugs	$27.80
14 Gauge Wire	$18.76
6S Battery Charger	$51.47
8 Gauge Wire	$14.33
3D Printer Filament	$64.34
Rubber Grommets	$17.15
Hex Screw Nut	$6.75
Threads	$8.57
Raspberry Pi 4 8gb	$85.60
2 Logitech Webcams	$107.00
Micro SD Card - 32 GB	$8.56
6S 22.2V 22000mAh Battery	$252.99
ZD-850 Hexacopter Frame	$125.17
Flysky FS-i6X RC and Receiver	$45.48

Total Cost =	$1,552.37

8. Environmental and Safety Aspects

Safety should be considered in any situation but is paramount when it can affect the health and safety of the public. That is why safety was considered as one of the key aspects of this project. Not only the safety of the public was considered, but federal regulations were needed to be followed for the testing of the drone. Additionally, the team followed all lab procedures and developed precautions for hands on testing. The lab safety procedures were as follows:

1. Safety glasses are to be worn during soldering.
2. There will be no eating or drinking in the lab.
3. Shirts, long pants, and closed toed shoes are always to be worn during lab.
4. Appropriate PPE will be worn for the task at hand. i.e. wearing leather gloves while cutting material.

The additional precautions specifically for this project were:

1. Rubber coated gloves will be worn while connecting/disconnecting the battery.
2. Treat all electrical circuits as they are energized, even when they are not.
3. The battery will be disconnected when working on electrical components.
4. The battery will be disconnected once the drone is disarmed
5. Leather gloves will be worn when working with the drone while the propellers are attached.
6. All team members will remain at least 10ft away from the drone once it is armed (loaded and ready to fly).
7. The team will place the remote controller on a flat surface ensuring that no toggles are push while arming/disarming drone, there will be no team member holding the remote to reduce the risk of injury during the arm/disarm sequence.

The Federal regulations required to be followed are as stated:

The team will be operating the drone under F.A.A. Part 107, exception 49 U.S.C. § 44809, which allows unmanned aerial vehicles weighing less than 55lbs to be flown for recreational flyers and community-based organizations. These rules are:

1) Register your drone, mark (PDF) it on the outside with the registration number and carry proof of registration with you.
2) Fly only for recreational purposes.
3) Fly your drone at or below 400 feet above the ground when in uncontrolled (Class G) airspace.
4) Obtain authorization before flying in controlled airspace (Class B, C, D, & E). You can obtain authorization in three ways:
   a) LAANC
   b) DroneZone
   c) A written agreement with the FAA for fixed flying sites.
5) Keep your drone within your visual line of sight, or within the visual line-of-sight of a visual observer who is co-located (physically next to) and in direct communication with you.
6) Do not fly at night unless your drone has lighting that allows you to know its location and orientation at all times.
7) Give way to and do not interfere with manned aircraft.
8) Never fly over any person or moving vehicle.
9) Never interfere with emergency response activities such as disaster relief, any type of accident response, law enforcement activities, firefighting, or hurricane recovery efforts.
10) Never fly under the influence of drugs or alcohol. Many over-the-counter medications have side effects that could impact your ability to safely operate your drone.
11) Do not operate your drone in a careless or reckless manner.

In addition to safety aspects, the effect of the project on the environment was also considered. The drone was expected to create excessive amounts of noise while used. To reduce and eliminate harm from this, the user shall not fly the vehicle near people or animals. Additionally, the user shall not fly the vehicle over any people, vehicles, or animals, to reduce the potential of falling objects that could potentially cause damage or harm. Also, the packing materials for the project came in cardboard boxes and plastic bags. To reduce the amount of waste generated through packaging, the team will recycle materials such as plastic or cardboard. All 3D printed parts were made from PLA, which is produced from a corn-based plastic and is biodegradable and recyclable.

9. Ethical Consideration

The design of the vehicle was portioned into 4 different areas deemed necessary for the design. These areas were dispersed among the team members. The team performed enough research on various topics regarding the project to gain adequate knowledge of the work to be performed and are cited in the references section. The general categories of research were divided and assigned as shown in table 19**.**

Table 19: Team Member Assigned Design Challenges

Team Member	Design Challenge
Caleb Johnson	Determine and Perform Calculations for Flight
Peter Mills	Understand Implementation of Computer Vision
Bailey Protzman	Determine All Necessary Drone Components
Thomas Williams	Development of a Payload Delivery System

Each team member oversaw all additional tasks that pertained to the area of focus. For example, if a team member were assigned the OpenCV implementation, they would need to research, construct, calibrate, test, and write about the smaller tasks involved. Each team member also provided a weekly report of the updates on their portion of the project to be presented during weekly group meetings. An expectation of team members can be seen in table 20.

Table 20: Team Member Tasks

General Tasks	Start Date	End Date	Owner
Research Physics of Flight	9/6/2020	10/24/2020	Caleb
Research Drone Components	9/6/2020	10/24/2020	Bailey
Research Computer Vision	9/6/2020	10/24/2020	Peter
Research Payload Delivery System	9/6/2020	10/24/2020	Thomas
Determine Components for the Drone and Order	9/27/2020	12/12/2020	Bailey
Design Payload System	10/4/2020	12/12/2020	Thomas
Test Payload System	12/13/2020	1/9/2021	Thomas
Calculate Current, Lift, and Flight Time	12/13/2020	2/1/2021	Bailey
Design Object Avoidance System	10/11/2020	12/12/2020	Peter
Calibrate object avoidance	12/13/2020	3/1/2021	Peter
Construct Drone	1/10/2021	2/20/2021	Team
Test and Refine Drone	2/21/2021	4/27/2021	Team

10. References

[1] B. Butcher and K. W. Lim, “Assessing Feasibility of the Delivery Drone,” thesis, MIT, Cambridge, 2019.

[2] A. Joyce, A. Pederson, and M. Rodriguez, “Aero Drone Package Delivery System,” thesis, 2019.

[3] J. Feist, “Delivery drones: Understanding the challenges for drone delivery,” *Drone Rush*, 23-Apr-2020. [Online]. Available: https://dronerush.com/google-ups-delivery-drones-19058. [Accessed: 27-Sep-2020].

[4] J. S. Reid, "Drone Flight - what does basic physics say?," 2020.

[5] P. Sarin, "Quadcopter Dynamics," Indian Institue of Technology, Bombay.

[6] J. Dickey, “Static Thrust Calculation,” *quadcopterproject*, 24-Feb-2016. [Online].Available: https://quadcopterproject.wordpress.com/static-thrust-calculation. [Accessed:05-Oct-2020].

[7] P. Connelly and W. Geck, “Hyperion Emeter Datasheet,” *HYPERION Prop Constants*. [Online]. Available: https://www.empirerc.com/hp-propconstants.htm. [Accessed: 05-Oct-2020].

[8] J. Bourke, “Understanding Electric Power Systems - Part 3,” *RC Groups*, 01-Nov-1998. [Online]. Available: https://www.rcgroups.com/forums/showthread.php?393251-Control-Tower-November-1998. [Accessed: 05-Oct-2020].

[9] P. Connelly, “HYPERION Prop Talk,” *HYPERION*. [Online]. Available: https://www.empirerc.com/hp-proptalk.htm. [Accessed: 05-Oct-2020].

[10] J. Reid, “Multirotor Motor Guide,” *RotorDrone*, 11-Apr-2019. [Online]. Available: https://www.rotordronepro.com/guide-multirotor-motors/. [Accessed: 20-Sep-2020].

[11] C. Bergquist, “Learn How to Build Your Own Drone from Scratch: A 2020 DIY Guide,” *Learn How to Build Your Own Drone from Scratch | A 2020 DIY Guide*, 22-Sep 2020[Online]. Available: https://dojofordrones.com/build-a-drone/. [Accessed: 20-Sep-2020].

[12] S. Carhart, D. Cooper, L. Gaitan, M. Kaliterna, S. Lama, P. Rogel-Herrera, and Y. Yabe, “Heavy Lift Drone,” thesis, Scholar Commons, Colombia, SC, 2020.

[13] B. Y. Suprapto, M. A. Heryanto, H. Suprijono, J. Muliadi and B. Kusumoputro, "Design and development of heavy-lift hexacopter for heavy payload," *2017 International Seminar on Application for Technology of Information and Communication (iSemantic),* Semarang, 2017, pp. 242-247, doi: 10.1109/ISEMANTIC.2017.8251877

[14] D. Mardiansyah and A. Budi, “UAV Vision System for Rescue Payload Delivery,” *IOPConference Series: Materials Science and Engineering*, vol. 384, Nov. 2017.

[15] “Torque Calculations for Gearmotor Applications,” *Precision Microdrives*, 2020. [Online]. Available: https://www.precisionmicrodrives.com/content/torque-calculations-for-gearmotor-applications/. [Accessed: 29-Oct-2020].

[16] Freeman, D. Suen, R. Chapman, and S. Biaz, “UAV Collision Avoidance with Stereo Vision on a Low-Power Embedded System”, Department of Computer Science and Software Engineering at Auburn University, Auburn, AL, USA, Tech. Rep. #CSSE17-7, Jul.2017.

[17] D. Pohl, S. Dorodnicov, and M. Achtelik, “Depth Map Improvements for Stereo-based Depth Cameras on Drones,” *Intel® RealSense™ Developer Documentation*, 2020.[Online]. Available: https://dev.intelrealsense.com/docs/depth-map-improvements-for-stereo-based-depth-cameras-on-drones. [Accessed: 29-Oct-2020].

[18] E. Ramsay, “Super cheap autonomous cargo ‘drone’.,” *diydrones*, 12-Mar-2016. [Online]. Available: https://diydrones.com/profiles/blogs/super-cheap-autonomous-cargo-drone.[Accessed: 28-Sep-2020].

[19] R. Radharamanan, A. Rosa, B. Neto, and V. Santos, “Use of 3-D Printers to Design, Build, and Test a Quadcopter Drone,” *Journal of Management & Engineering Integration*, vol.9, no. 1, pp. 24–33, 2016.

[20] J. B. Brandt, R. W. Deters, G. K. Ananda, O. D. Dantsker, and M. S. Selig, “UIUC propeller data site,” 25-Aug-2020. [Online]. Available: https://m-selig.ae.illinois.edu/props/propDB.html. [Accessed: 15-Nov-2020].

[21] V. Artale, C. Milazzo and A. Ricciadello, "Mathematical Modeling of Hexacopter," Kore University of Enna, Enna, Italy.

[22] W. Selby, "Simulation Environment," May 2015. [Online]. Available: https://www.wilselby.com/research/arducopter/simulation-environment/.