WORCESTER POLYTECHNIC INSTITUTE

Surveillance UAV

A Major Qualifying Report
submitted to the faculty of the
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the degree of Bachelor of Science

Submitted by:

Andrew Gallagher
Mechanical Engineering

Steven Guayaquil
Robotics Engineering / Electrical and Computer Engineering

Walter McIntyre
Robotics Engineering

Arianna Niro
Robotics Engineering

Antonio Puzzi
Mechanical Engineering

Arman Uygur
Mechanical Engineering

Advisor:

Professor Taskin Padir
Robotics Engineering / Electrical and Computer Engineering

Submitted on:

1 May 2014

Acknowledgements

The Intelligent Surveillance MQP team would like to give a special thanks to a few people, without whom this project would not have been possible. First, we would like to thank Professor Padir, the advisor of the team. He gave the team timely and helpful feedback on the project on several occasions. Joe St. Germain, the robotics lab manager of Atwater Kent, spent countless hours assisting the team in many ways such as 3D printing materials for us and ensuring we had our parts in time to meet our deadlines. For that, we are extremely grateful. Tracey Coetzee, the robotics department administrative assistant, made several orders for our team and helped us out in several other ways. Natalie Guardino assisted us with orders during Tracey's absence.

We would also like to thank a few professors who spent time assisting us on the project. Professor Stafford, of the RBE department, gave the team excellent advice about different manufacturing methods. Professor Onal, of the ME department, assisted the team with materials selection and analysis.

Abstract

Surveillance is critical for military, law enforcement, and search and rescue operations. In the past, stealth aircraft and helicopters were used for these types of missions. Recently however, unmanned aerial vehicles (UAVs) have grown in popularity and are an excellent resource that can be utilized for surveillance missions. Since this is a common capability of drones, this project sought to create a surveillance UAV that was autonomous, inexpensive, lightweight, and easy to manufacture. The drone was designed as a quadrotor that houses two cameras with a wireless transmission system that provides live feed from the cameras to the ground station. It was also intended to be able to carry a payload for future developments. Though not all of the goals were fully realized by the project's conclusion due to stability and networking complications, the drone met size and cost standards, and could successfully localize its position with GPS and IMU sensors. Additionally, its controls were understood through simulation and testing.

Table of Contents

Acknowledgements 1
Abstract 2
Table of Contents 3
Table of Figures 6
Table of Tables 8
1. Background Research 9
1.1 History of UAVs 9
1.2 Basic Flight Physics 11
1.3 Physical Limitations 12
1.4 System Limitations 13
1.4.1 System Power 13
1.4.2 System Sensors 14
1.5 System Components 14
1.5.1 Materials 15
1.5.2 Motors 18
1.5.3 Propellers 18
1.5.4 Batteries 19
1.5.5 Sensors 22
1.5.6 Microcontrollers 23
1.6 Future of UAVs 24
2. Initial System Development 26
2.1 Project Specifications 26
2.2 Materials 26
2.2.1 Derivations 27
2.2.2 Thickness Calculations for Various Materials 28
2.2.3 Decision Matrices 29
2.2.4 Results 30
2.3 Router 30
2.4 Motors and Propellers 31
2.5 Battery 32

2.6 Motor Speed Controller 33
2.7 DC to DC Converter 33
2.8 Previous MQP Components 34
3. Design 35
3.1 Initial Concept Designs 35
3.2 Graphic User Interface (GUI) Design 36
3.3 Controls and Simulation 37
4. Prototypes 44
4.1 Prototype 1 44
4.1.1 CAD Modeling and Manufacturing 45
4.1.2 Results and Conclusions 46
4.2 Prototype 2 47
4.2.1 CAD Modeling and Manufacturing 47
4.3 Final prototype 52
5. Final System Components Selection 54
5.1 Motors and Propellers 54
5.2 Battery 55
5.3 Motor Speed Controller 55
5.4 Landing Gear 56
5.4.1 Impact Force Calculations 56
5.4.2 Damping Coefficient Calculation 58
5.4.3 Spring Coefficient Calculations 60
5.4.4 Change in Springs 61
6. System Tests 62
6.1 Drop Tests 62
6.1.1 Initial Drop Test 62
6.1.2 Initial Drop Test Analysis 64
6.1.3 Final Drop Test 64
6.1.4 Final Drop Test Analysis 65
6.2 Flight Tests 66
6.2.1 Introduction 66

6.2.2 Flight Tests and Analysis 66
6.2.3 Flight Test Conclusions 71
6.3 Flight Time Test 71
7. Results, Conclusions, and Recommendations 73
7.1 Results 73
7.2 Conclusions 74
7.3 Recommendations 74
Appendices 76
Appendix A: Authorship 76
Appendix B: Rod Calculations and Decision Matrices 78
Appendix C: Hull Calculations and Decision Matrices 81
Appendix D: Motor Decision Matrix, Parameters, and Weight Distribution of Factors for Decision Matrix 82
Appendix E: Battery Decision Matrix, Parameters, and Weight Distribution of Factors for Decision Matrix 83
Appendix F: Material Properties 84
Appendix G: Landing Gear Spring Suggestions 85
Appendix H: Landing Gear Calculations MATLAB Code 87
Appendix I: Future Work and Alternative Recommendations 89
Appendix J: Multirotor Design Models 91
Appendix K: Quadrotor Simulation MATLAB Code 94
Appendix L: Basic Arduino to Raspberry Pi Communication MATLAB Code 101
Appendix M: Arduino Stability Control MATLAB Code 102

Table of Figures

Figure 1: Aerial Steam Carriage 10
Figure 2: Titan Aerospace Solara 50 14
Figure 3: Generic Carbon Fiber Comparison Table 16
Figure 4: Deflection properties of Honeycomb Core 17
Figure 5: Battery Weight vs. Power Example Comparison 21
Figure 6: DJI Phantom UAV 23
Figure 7: Area Moments of Inertia 27
Figure 8: Decision Matrix for the Material Selection of Rods 29
Figure 9: Linksys WRT54GL Router 31
Figure 10: C2020 Micro Brushless Outrunner 3500Kv 32
Figure 11: 6x3 Standard Propellers 32
Figure 12: Mystery 12A Brushless Speed Controller 33
Figure 13: OKI-78SR-5/1.5-W36-C DC to DC Converter 34
Figure 14: Preliminary Design Concept 35
Figure 15: Initial GUI Design 36
Figure 16: Final GUI Design 37
Figure 17: Quadrotor Inputs Model 37
Figure 18: Quadrotor Dynamics 38
Figure 19: Quadrotor Physical Model 38
Figure 20: X-position Block Diagram 39
Figure 21: Basic X-Y-Z Position Control 39
Figure 22: Linear Trajectories 40
Figure 23: Sinusoidal Trajectory 40
Figure 24: Higher Frequency Sinusoidal Trajectory 41
Figure 25: 0.4 Second Quintic Trajectory 41
Figure 26: 0.4 Second Quintic Velocity Trajectory 42
Figure 27: 0.4 Second Quintic Acceleration Trajectory 42
Figure 28: 0.2 Second Quintic Trajectory 43
Figure 29: Prototype 1 Isometric View 45
Figure 30: Prototype 1 Left View 45

Figure 31: Modified Prototype 1 Isometric View 46
Figure 32: Modified Prototype 1 Rod 46
Figure 33: Prototype 2 Isometric View 48
Figure 34: Prototype 2 Left View 48
Figure 35: Landing Gear Sub-Assembly 49
Figure 36: Battery Casing 49
Figure 37: Hull Top 50
Figure 38: Hull Base 50
Figure 39: Middle Hull's Wall 51
Figure 40: Final Hull Base 52
Figure 41: Final Motor Rod 52
Figure 42: Final Leg Assembly 53
Figure 43: Final Design 53
Figure 44: EMAX GT Series 1180Kv Outrunner Brushless Motor 54
Figure 45: APC 10 x 4.7 propeller 55
Figure 46: Hobbywing FlyFun 30 A Brushless ESC 56
Figure 47: Vector Components on Impact Force 58
Figure 48: Damping Calculation Characteristics 59
Figure 49: Assembled Prototype 62
Figure 50: Additional Weight Added 63
Figure 51: Measured Height (4 feet) 63
Figure 52: Final Drop Test Location (17 feet) 65
Figure 53: Flight Test #3 set up 67
Figure 54: Flight Test #3 close up 68
Figure 55: Flight Test #7 70
Figure 56: Flight Test #8 71
Figure 57: Project Specification Results Summary 74

Table of Tables

Table 1: Material Selection Formulas[12] 17
Table 2: Potential and Kinetic Energy Equations Used 56
Table 3: Calculated Impact Velocities Based on Initial Heights 57
Table 4: Calculated Impact Forces (IF) Based on Estimated Bounce Back Distances 57
Table 5: Calculated Net Impact Forces (NIF) Based on Estimated Bounce Back Distances 58
Table 6: Calculated Total Energy (TE) Based on Initial Velocities 59
Table 7: Calculated Stroke Energies (SE) Based on Various Damper Diameters 60
Table 8: Calculated Damping Forces (DF) Based on Various Damper Diameters 60

1. Background Research

The development of unmanned aerial vehicles (UAVs) has been growing significantly over the last decade. UAVs have been expanding from military applications into civilian purposes like aerial photography, field surveillance, and disaster relief. However, most are often found to be expensive and difficult to deploy. To address these issues, this project sought to implement a lightweight drone capable of performing surveillance while communicating in real time to the user. Before the team could establish project specifications, they conducted extensive background research to gain a deeper understanding of the current technological advancements within the drone industry.

1.1 History of UAVs

Though modern-day technology is quickly advancing and improving UAVs and drones, developments in this field began decades ago, even before the first manned airplane flight occurred in 1903. The first and most primitive designs centered on balloons. The first attempts began in France in 1782 by the Montgolfier brothers[1]. These attempts continued through the years, one of which was developed by Charles Perley in February 1863, two years after the Civil War began. Perley attempted to design an aerial bomber, a hot-air balloon that carried explosives in its basket. The explosives were attached to a timing mechanism, and upon the timer going off, the explosives dropped out and a fuse was ignited[2]. However, due to the unpredictability of air currents and weather patterns then, Perley's aerial bomber was never successfully deployed and experimentation into other designs was expanded[3]. Another model reliant upon wind and weather was a surveillance kite. A novelty put together by Douglas Archibald in 1883, the concept was successfully applied during the Spanish-American War in 1898. A corporal captured hundreds of images through a kite with a camera attached, with a long shutter release attached to the string[4].

---

[1] Terault, Cam. "A Short History of Unmanned Aerial Vehicles (UAVs)." *Draganflycom UAV News RSS*. N.p., n.d. Web. 19 Sept. 2013. <http://www.draganfly.com/news/2009/03/04/a-short-history-of-unmanned-aerial-vehicles-uavs/>.

[2] Krock, Lexi. "Timeline of UAVs." *PBS*. PBS, n.d. Web. 20 Sept. 2013. <http://www.pbs.org/wgbh/nova/spiesfly/uavs.html>.

[3] Scheve, Tom. "How the MQ-9 Reaper Works." *HowStuffWorks*. N.p., n.d. Web. 20 Sept. 2013. <http://science.howstuffworks.com/reaper1.htm>.

[4] Krock, Lexi. "Timeline of UAVs." *PBS*. PBS, n.d. Web. 20 Sept. 2013. <http://www.pbs.org/wgbh/nova/spiesfly/uavs.html>.

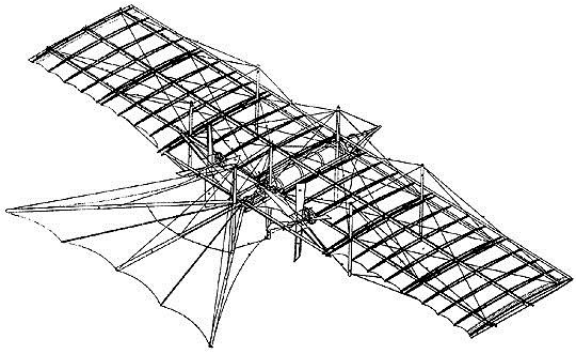


Figure 1: Aerial Steam Carriage

Beginning in mid-1800s, winged designs were also taking shape. One such example is the Aerial Steam Carriage, shown in Figure 1, which was built in 1848 by John Stringfellow and William Henson in England. Their initial model flew successfully for roughly 60 yards. In 1868, the pair also developed a model based on a tri-winged structure attached to a wire guide that could effectively generate lift and only used the wire guide to steer around obstacles[1]. 1896 saw the development of a model created by Samuel Langley. Steam-powered, his "Aerodrome Number 5" flew for almost a mile. In the years leading up to and during World War I, development began to expand even more, especially with war demands for fighter aircraft. One such model was the Kettering Aerial Torpedo, a 300-pound plane with the capacity of carrying an equal load that could be launched at a pre-programmed target. Though ordered in great numbers in the final few months of World War I, the close of the war led to the cancellation of the orders[2]. 1917 saw the advent of the first radio-controlled UAV, invented by Dr. Peter Cooper and Elmer A. Sperry after they received the first military contract for an unmanned flight system[5]. It also employed the use of their automatic gyroscopic stabilizer to help it fly straight and level. Though it could successfully travel for 50 miles with a 300-pound bomb loaded, it was never employed in combat after the Navy cancelled the project in 1922[2].

After World War I, the development in this field abruptly declined until almost 1940 when an impactful design called the "Queen Bee" was developed in Great Britain. Though initially designed as a practice target during training, it expanded as its capabilities were capitalized on; it could reach a height of 17,000 feet and fly for 300 miles at a speed over 100 mph. Radio-controlled, the 380-strong fleet was used until its retirement in 1947, although its primary use was still for training[3]. A key aspect of this UAV was its ability to land and be recovered for reuse. This drone was the most prominent one until World War II when the Interstate BQ-4 was effectively employed in combat in 1942. In 1944, the United States reached 18 hits on targeted Japanese units with this drone[1].

Another advancement was the V-1 in Germany, which Adolf Hitler claimed would be used for non-military targets. Flying at 470 mph, it had a 2,000 pound load capacity and could fly 150 miles. With such a dangerous UAV in Germany's control, programs within the United States

[5] Blom, John D. "Unmanned Aerial Vehicle Systems: A Historical Perspective." *Combat Studies Institute Press*. N.p., Sept. 2010. Web. 20 Sept. 2013. <http://usacac.army.mil/cac2/cgsc/carl/download/csipubs/OP37.pdf>.

significantly took off in an effort to counteract and destroy them. In the years following World War II, there were many improvements, not only to the propulsion and guidance systems, but also to overall capabilities of the UAVs[1]. The United States Naval and Air Force programs began to convert surplus aircraft into practice target drones and the first jet-propelled UAVs, Ryan Firebees, came onto the scene in the early 1950's. These drones were responsible for over 34,000 surveillance flights in the 1960's and 1970's[3].

Developments in Israel in the late 1970's and 1980's had a significant impact on the programs within the United States. The Israeli "Scout" and "Pioneer" UAVs began the trend towards lighter drones with more of a glider-based design. These drones also began the transition to less expensive, smaller models, which increased their stealth capabilities[3].

In the last two decades, UAV research and development has continued to focus on military surveillance and attack applications. However, recently UAVs have become more popular in the civilian sector as well. The applications of these drones have been broad. These include disaster relief, crop dusting, and mapping new geographic areas. These UAVs have also functioned as toys controlled by smartphones. The surge in the popularity of drones in the civilian sector demonstrates that they still have the potential for growth and further developments.

1.2 Basic Flight Physics

Similar to other flying objects, a quadrotor has a group of forces and torques acting on it while it flies. There are four main forces acting on the drone: drag, lift, weight, and thrust. In order for the drone to fly, these different forces need to be balanced. This can be seen by utilizing Newton's Second Law.

Newton's Second Law:

$$F = ma \qquad (1)$$

For constant velocity acceleration is zero (a=0). Thus the sum of the forces is equal to zero. So for steady, constant velocity flight, completing a force balance in the horizontal direction on the diagram obtains:

$$F_{thrust} - F_{drag} = 0 \qquad (2)$$

$$F_{thrust} = F_{drag} \qquad (3)$$

Since this is for a constant velocity, the aircraft is either moving or at rest. An analysis in the vertical direction will produce similar results.

$$F_{lift} - F_{weight} = 0 \qquad (4)$$

$$F_{lift} = F_{weight} \qquad (5)$$

Thus, for steady level flight the thrust must equal the drag and the lift must equal the weight. In order to gain altitude, the force of lift must be greater than the force due to gravity. Similarly, in order to accelerate the vehicle the force of thrust must be greater than the force of drag.[6]

For the quadrotor to perform different types of movements, the sum of the forces and torques have to follow a certain pattern. It is important to note that the quadrotor's motors are divided into pairs in order to cancel moments that would make the aircraft spin in an uncontrollable way. The interaction of these motors and the movement they produce are explained below.

Dividing the motors into pairs allows the aircraft to perform movements, such as going forward or backwards, and rotating either clockwise or counter clockwise. The pairs are grouped by the diagonals of the quadrotor (or if seen as a cross, the two located in the vertical line would form the first pair, and the two located in the horizontal line would be part of the second pair).

For forward movement, one pair of motors would keep working with the same power/force output, while the motor in the back of the other pair would operate at a higher power than the one in front, making the robot tilt and move forward. For a backwards movement, the back and front motors would change roles, and the same interaction would occur with the other motor pair if the quadrotor would like to move left or right.

To understand the physics involved in the rotation of the robot, it is important to understand the rotation produced by the motors produce a torque in one direction, which the robot tries to compensate or balance by rotating in the other direction. In other words, if one pair of motors rotate in a clockwise direction, the robot would balance the torques by rotating in a counterclockwise direction and vice versa. If the four motors are operating at the same time with the same force, the torques are balanced in the middle of the robot. This is because one pair of motors rotates clockwise and the other counterclockwise, canceling out the torques. In order to turn the UAV in a certain direction, more power is added to the motors associated with the desired direction.

1.3 Physical Limitations

There are several physical limitations that effect the performance of the UAV. One of the biggest components of these limitations is weather conditions. There are some specific weather conditions that affect the ability of any vehicle to fly; these conditions are rain, snow, and wind. The first two conditions are the hardest ones to overcome since the UAVs often have delicate electronics that are outside of the main cabin that end up being unprotected. Rain and water, in addition to wind and lightning that often accompany these weather conditions, can easily damage all these electronics and affect the stability of the drone since the design analysis is made using some assumptions for ideal flying. For other natural limitations, environmental conditions of the area where the UAV has to perform can be included. This is because different terrain and climates, such

[6] "The Physics of Flight (Newton and Bernoulli)." The Physics of Flight (Newton and Bernoulli). N.p., n.d. Web. 17 Sept. 2013.

as a desert or a city, can have extremely different temperatures, flora and fauna, man-made and natural structures, etc. that could potentially affect the UAV.

In terms of stability, the size and design of the UAV greatly influence its performance. In the current market there exists a wide variety of options, the predominant the ones being at a size around 0.5m x 0.5m x 0.15m and with a weight below the 2 kg. These characteristics are determined by the needs and the specific applications of the UAV.

Another important consideration that needs to be made on the physical limitations is the altitude that the UAVs are flying at. The different altitudes can bring different issues, like an increase in wind velocity. The typical altitude where most of the existing UAVs perform is at less than 15 m, because of the simplicity of operation and control at that altitude. Thus, exploring higher altitudes requires more stability.

1.4 System Limitations

There have been many system developments for UAVs recently. Despite all these developments, however, there are still limitations that must be considered.

1.4.1 System Power

System power has been one of the biggest limitations for all UAV systems. There have been different approaches and ideas about how to power the system, but the most common one is the use of conventional rechargeable batteries. The main problem related to the powering of the system has to do with the actual flight time, which is directly affected by the amount of power required by all the different sensors and processors on board. Considering the amount of sensors—including altimeters, cameras, gyroscopes, and many others—and their importance to the proper function of the system, the power needed for these is not something that can be changed to a great extent.

The main alternative to the conventional batteries is the use of solar energy. The main issue with solar energy is that it requires a large number of panels or cells to produce enough energy to power the system. This implies that there needs to be a significant increase in size of the UAV and possibly other adaptations with new designs.

Figure 2: Titan Aerospace Solara 50

Figure 2 depicts an example of a solar energy powered UAV. It has a 50-meter wingspan and a length of 15.5 meters. This shows that the size for this particular UAV is a lot greater than the common UAVs. Another particularity about this UAV is its need to be at higher atmospheric altitudes to receiver better solar radiation.

1.4.2 System Sensors

The sensors used in a system are a critical and important decision in the design process. The selection of these sensors will depend on the purpose of the system. Some of the purposes of these sensors are to perform a successful flight and also to stabilize the system. The most commonly used sensors for these purposes are gyroscopes, accelerometers and altimeters. The performance of the UAVs depends significantly on the sensors chosen and their correct integration, which is the main reason why sensors are a crucial aspect of the system. The sensors and their application to this project are explained in further detail in Section 1.5.5.

1.5 System Components

When developing a UAV, there are several parts that are combined in the system in order for it to achieve its objectives. Mechanical and electrical components, such as motors, propellers, sensors, and microcontrollers, must be integrated to create a fully functional drone. The correct implementation and selection of all these different parts will determine the success of the project, which is the reason why having a deep knowledge of all of these concepts is key to be able to correctly design the UAV.

1.5.1 Materials

General Assumptions

Material selection is one of the key factors that has a direct effect on mass and other physical properties of the vehicle. A good selection will lead to a weather resistant, lightweight, and strong product. Five primary factors were considered when choosing the materials: deflection, strength, weight, water resistance, and cost. Based on these criteria several materials were considered and compared. The materials considered were 3K, Plain Weave Carbon Fiber, Generic Carbon Fiber, Nomex Honeycomb Core, Airex Foam Core, Aluminum 6061, G-AIMg5 fiberglass, PVC (hard), Styrene (plastic), and Balsa Wood Class IV. Of all of these materials, carbon fiber, aluminum, and wood are the most common materials used in quadrotor designs[7]. However, most of the articles, papers, and blogs researched show that material selection is mostly based on commonly known facts rather than calculations. Below are three such facts.

- "Carbon fibers are the most rigid, light and vibration absorbent but they are expensive." [7]
- "Aluminum is relatively light, affordable and has average rigidness but may have problems with motor vibrations since they have worse damping effects comparing to carbon fiber. In severe vibration problems, sensor readings can be affected seriously."[7]
- "Wood is easy to shape and has a better absorption ratio of vibrations than aluminum but it is not very rigid and can be broken easily"[7]

These facts led designers to choose their materials without modeling and analyzing the effects the materials have on the system. However, the team decided to have a broad analysis to ensure that the optimum material was selected for the system.

Since carbon fibers are popular materials for aerospace applications, they were added in the comparison list for further calculations. However, there are many different carbon fiber types commercially available. Therefore it was decided to compare 4 different types of carbon fiber along with other materials. Since the stiffness to weight ratio was the top priority, Nomex Honeycomb and Airex Foam Cores were picked for further analysis from the specific carbon fibers below.

[7] Liang, Oscar. "Build A Quadrotor From Scratch." Web log post. *OscarLiang.net*. N.p., 25 June 2013. Web. 17 Oct. 2013.

PRODUCTS	COMPARISON CRITERIA				
	Stiffness to Weight	Toughness	Crushability	Moisture Resistance	Sound Absorbency
Solid Carbon Fiber	GOOD	GOOD	BEST	BEST	POOR
Birch Core	BETTER	BEST	BEST	GOOD	POOR
Balsa Core	BETTER	GOOD	BETTER	POOR	GOOD
Polypropylene Honeycomb Core	BEST	GOOD	GOOD	BEST	BEST
Nomex Honeycomb Core	BEST	BETTER	BETTER	BETTER	BEST
Depron Foam Core	BETTER	POOR	POOR	BETTER	BETTER
Airex Foam Core	BEST	GOOD	GOOD	BETTER	BETTER
Divinycell Foam Core	BETTER	BEST	BETTER	BETTER	GOOD

Figure 3: Generic Carbon Fiber Comparison Table[8]

There are certain tradeoffs of carbon fibers even though they have great electrical, thermal and physical proprieties. Under loads, carbon fibers bend but they do not remain permanently deformed. Instead, once ultimate strength of the material is exceeded, it will fail suddenly. Thus, if one of carbon fibers is chosen as the optimum material among those compared, the factor of safety will need to be raised.[8]

Material Calculations and Assumptions

For comparing parameters like deflection and strength, several calculations must be made, including but not limited to bending stress, factor of safety (FS), and deflection. Bending stress and factor of safety calculations will be used for determining effects of loads and moments on various parts of the quadrotor. Deflection calculations will be used for determining the displacement of several parts such as the rods. Since forces caused by the thrust and torque directly impact deflection, displacement calculations play a significant role for the control of the vehicle. Ideally, the material will have a negligible deflection under the estimated load.[9] Furthermore, deflection may vary greatly when thickness of the same material is changed, as demonstrated in the graphs below. Therefore, thickness calculations will also be considered in the design process.

[8] DragonPlate Company. *What Is Carbon Fiber?* N.p.: DragonPlate, n.d. *DragonPlate Company*. Web. 17 Oct. 2013. <http://www.dragonplate.com/sections/technology.asp>.

[9] Carlos, Nate, Ben Cole, John Cook, Jonathan Forest, Sansen Johnson, Ed Massie, and Chris Rogers. *2008-2009 IARC Team Quadrotor*. *IARC Team Quadrotor Final Report*. Virginia Tech, n.d. Web. 17 Oct. 2013. <http://www.dept.aoe.vt.edu/~mason/Mason_f/IARC-FinalReport-v6.0.pdf>.

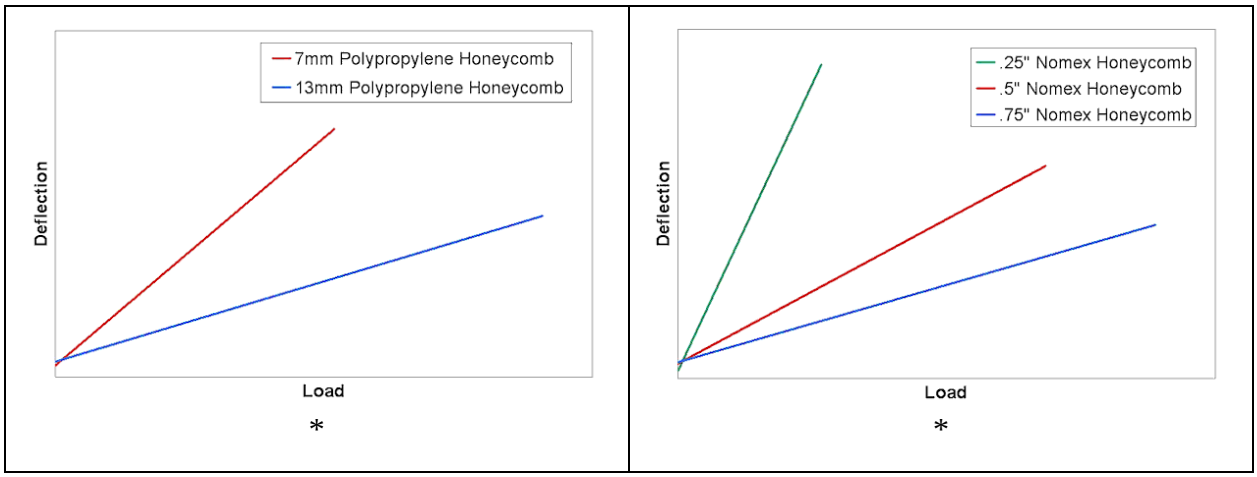


Figure 4: Deflection properties of Honeycomb Core[10]

The factor of safety is a vital parameter for the protection of the UAV and will be chosen as 2 since most aircraft applications use a FS of 1.2 to 3 depending on the material. Ductile and metallic materials use lower values, while brittle materials use higher values. A FS of 2 was chosen to lower the cost of production. In aerospace industry, the FS is chosen as small as possible because of the costs associated with structural weight are high and having a higher FS increases the structural weight. The more complex structures use local FSs for each different part. For example, buildings use an FS of 2 while their pressure vessels use an FS of 4. However, the quadrotor for this project will keep a uniform FS of 2.[11]

Knowing that factor of safety is 2 and the tensile strength of each material, an approximate bending stress can be calculated. From the physical dimensions of the design and estimation for weight, I_x can be determined. Once I_x, the assumed load (force), and Young's (tensile) Modulus are known, deflection can be estimated. Formulas that were used for the calculations are seen below.

Bending Stress	***Factor of Safety (FS)***	***Deflection***
$\sigma_{bending} = \frac{M_y}{I_x}$	$FS = \frac{\sigma_{tensile}}{\sigma_{bending}}$	$\upsilon_{deflection} = \frac{F\ x\ L^3}{3\ x\ E\ x\ I_x}$

Table 1: Material Selection Formulas[12]

[10] DragonPlate Company. *Carbon Fiber PP Honeycomb Core*. N.p.: DragonPlate, n.d. *DragonPlate Company*. Web. 17 Oct. 2013. <http://dragonplate.com/docs/DPSpecPP-Honeycomb.pdf>

[11] Burr, A and Cheatham, J: Mechanical Design and Analysis, 2nd edition, section 5.2. Prentice-Hall, 1995

1.5.2 Motors

Since the drone is a quadrotor, it necessitates four motors and four propeller. The motor is the device that converts electrical power output from the battery into mechanical power. The mechanical power then turns the propellers and generates the force needed by the robot to fly. This motor also creates a torque that rotates the robot either in a clockwise or counter clockwise direction. As mentioned before, this torque is cancelled by the torque generated by one of the other motors which produces a torque of the same magnitude but opposite direction.[12]

There are many types of motors, such as DC brush motors and AC motors. For this quadrotor, 4 DC brushless motors were implemented into the design. This type of motor has 2 constants which are important. These are the Km and the Kv values. Km is known as the motor constant and it is a ratio of the motors torque and the square root of the resistive power loss. Kv is known as the motor's velocity constant and it measured in RPM (revolutions per minute) per volt. This last constant is the ratio of the unloaded motor's RPM to the peak voltage output. In other words, if a motor has a Kv value of 1500 RPM/volt and a supply of 9 volts, the nominal speed would be 13,500 RPM.[12]

1.5.3 Propellers

The propellers are the components of the propulsion system that provide the thrust and lift needed by the quadrotor lo leave the ground, hover, turn, and rotate. These forces are produced as the propellers rotate. The propeller hubs are attached to the motor and then a nose cap is screwed on top of the propeller to secure it to the motor.

Propellers can be set up into two different ways. These are the pushing configuration and the pulling configuration. The difference between these two configurations is the direction the propeller is facing and the position it is with respect to the aircraft. For the pushing configuration, the propeller is pointing to the back of the aircraft and is usually located behind the wings, which makes the propeller push the aircraft through the air. On the other hand, the pulling configuration has the propeller pointing to the front of the aircraft and it is usually at the front of the wings, which makes the propeller pull the aircraft through the air.

In a quadrotor, the propellers are usually on top of the vehicle, i.e. in front of it, and usually pulling the aircraft up while they push the air down. They are also separated into clockwise and counter clockwise configurations (CW and CCW respectively)[13] since quadrotors need to have a balance

---

[12] Du Plessis, Francois. "Brushless DC Motor Characterisation and Selection for a Fixed Wing UAV." *Academia.edu*. N.p., n.d. Web. 29 Sept. 2013. <http://www.academia.edu/1360191/Brushless_DC_Motor_Characterisation_and_Selection_for_a_Fixed_Wing_UAV>.

[13] "AIRCRAFT PROPELLER INTRODUCTION." *AIRCRAFT PROPELLER INTRODUCTION*. N.p., n.d. Web. 29 Sept. 2013. <http://www.thaitechnics.com/propeller/prop_intro.html>

between the number of motors turning in a clockwise direction and those turning in a counter clockwise direction.

Motor-Propeller Configuration and Analysis

When choosing the type of combinations of motors and propellers, there are some factors that have to be taken into consideration. First of all, the goal and purpose of the quadrotor must be determined before choosing any configuration. For example, if the user wants a more acrobatic quadrotor, it will require a configuration that outputs a huge amount of speed and allows for quick speed variations. On the other hand, if the user is looking for a quadrotor that will hover most of the time without making quick and violent changes of speed and directions, the configuration requirement will be completely different.

It is important to understand that both the speed of rotation of the motor and the area covered by the propeller are values that affect the output force and speed in a significant way. The complexity of choosing the right configuration comes when other limitations of the design affect the decision. Motors that output high amounts of power are usually bigger and heavier than those that output lower levels of power. Motors require a lot of the battery's power, and if these get bigger, the battery will last a shorter amount of time. This is where the propellers come into play. A bigger propeller will output a larger amount of force than a smaller one while rotating at the same angular speed. The most obvious problem with a bigger propeller is that it covers more space and it can mess up the design's dimensions. The other issue with bigger propellers is that they take more time to vary their speed, while smaller propellers can change their speed faster. Clearly there are many decisions that must be made in order to determine the optimum motor-propeller setup for this project.

1.5.4 Batteries

When looking for a battery to use for a quadrotor, there are a number of specifications to consider in order to establish a balance between the weight of the battery and its capacity. A battery cannot be chosen solely on its own specifications, but the other components of the drone must also be taken into consideration. Each part has an impact on each other and therefore there are multiple combinations of components that could be used to create the ideal UAV.

An initial consideration must be the weight of the vehicle. Generally speaking, a good guideline is to have the total weight at around half of what the motors' thrust value is equal to[14]. By doing this the motors will not overheat and fail, and they will be able to lift the drone off the ground and hover at approximately 50% throttle. For instance, if there are four motors that each produce 654g of thrust each, the system's has a total thrust of 2616g[14]. Dividing that total in half gives its 50%

[14] "Quadrotor - LiPo Batteries." *Quadrotor - LiPo Batteries*. N.p., n.d. Web. 2 Oct. 2013. <http://quadrotor.wikispaces.com/LiPo Batteries>.

value, 1308g in this case. Subtracting the total weight of the system from that number will give an ideal range to work with to determine the weight of the battery[14].

Beyond just looking at the weight, the capacity of the battery must also be considered. Looking for batteries can be confusing to those who are not familiar with how they are named. For instance, "FreeFly LiPo 4S 14.8V 9000mAh 25C" and "G6 Pro Lite 6600mAh 25C (5S)" are names of different batteries. A consumer needs to know how they are categorized so that the proper battery for their needs is purchased. The S-number is the number of battery cells in series, so a 3S is three cells, a 4S is four cells, and so on. Each cell is equivalent to 1.7V, so the voltage of a 3S battery is 11.1V[15]. The more battery cells there are in series, the higher the voltage, which makes it more efficient while lessening current flow[16]. Batteries can also be listed in a form such as 3S1P, where the 3S is the same as before, but the added 1P indicates one parallel block. Adding more cells in parallel increases the capacity of the battery as it adds more amps[16]. The mAh is the milliamps per hour discharge rate, which is essentially the battery capacity. Using the two batteries cited above, it is seen that the FreeFly LiPo (lithium polymer) discharges at a rate of 9 amps per hour, while the G6 discharges at 6.6 amps per hour.

Finally, the C rating gives an indication of how long the battery will be able to run on full power, so the two mentioned above have the same capability, but a battery with a 30C rating will be able to run longer[15]. Though this is a good indication for the capabilities of the battery, the battery should never be run at its max for an extended period of time in order to avoid damaging the unit. In general, batteries with higher capabilities (4S versus 3S, 30C vs. 25C) add more weight to the battery, so it is important to balance capabilities with weight. Unfortunately, there is no graph to give an easy indication of what would be ideal for a system, so it needs to be calculated for every system according to its needs and components. However, the chart below gives a good example of the impact the weight of a system has on batteries with varied capacity[17].

---

[15] "Quadrotor LiPo Battery Weight/capacity Trade off." *Batteries*. N.p., n.d. Web. 2 Oct. 2013. <http://robotics.stackexchange.com/questions/554/quadrotor-lipo-battery-weight-capacity-trade-off>.

[16] "Choosing a Battery for a Quadrotor." *Everything about Quadrotor Flying Machine*. N.p., 4 Oct. 2010. Web. 2 Oct. 2013. <http://quadrotorflyingmachine.blogspot.com/>.

[17] "Recommended Components for Your AeroQuad." *AeroQuad*. N.p., n.d. Web. 2 Oct. 2013. <http://aeroquad.com/showwiki.php?title=Recommended%20components%20for%20your%20AeroQuad&redirect=no>.

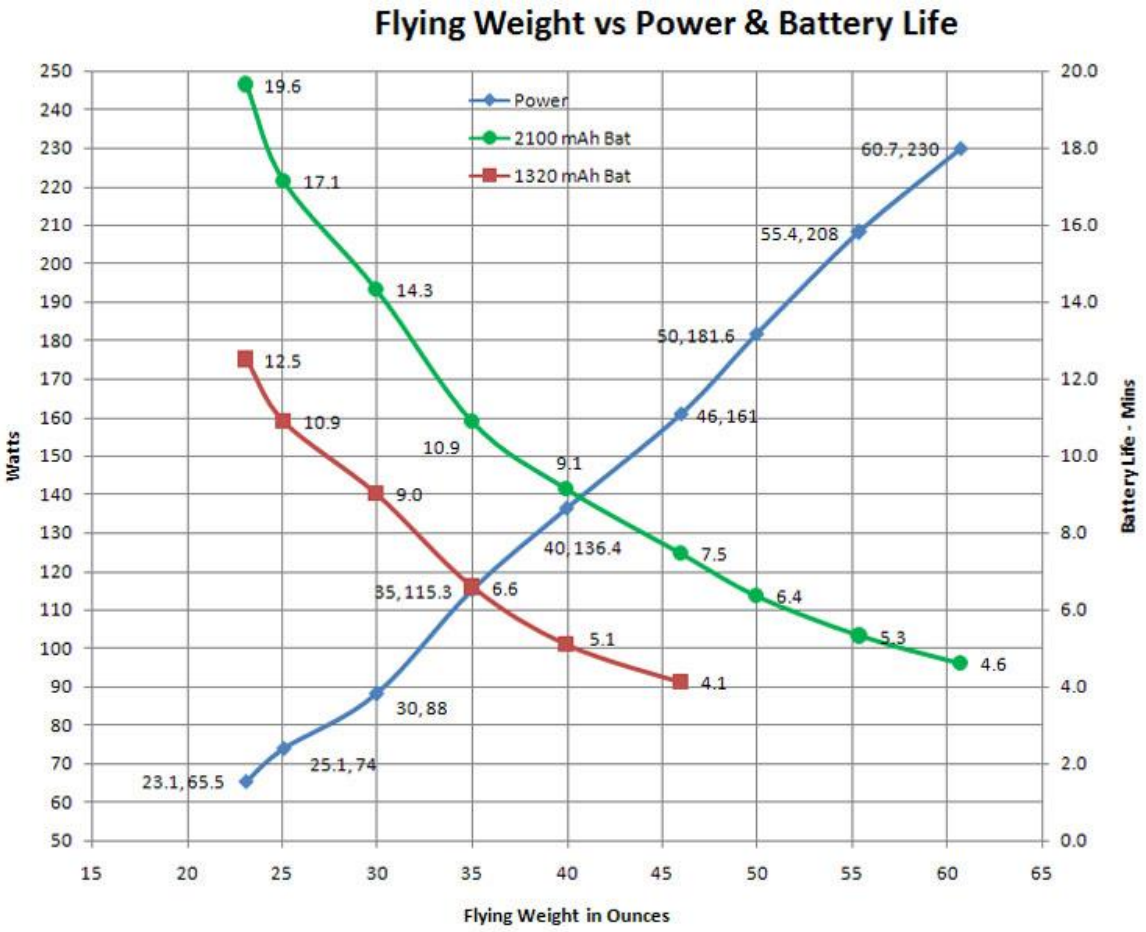


Figure 5: Battery Weight vs. Power Example Comparison [18]

The motors play an important role in choosing a final battery. For example, if a motor recommends a 3S1P battery, that is an indication of a recommended 11.1V. A 2200 mAh 30C battery for this system can output 66 A of current safely. If the maximum current for each motor is 14 A, then having four of them will require four times this amount, so 56 A[16]. Therefore, this battery would be a good fit for this system. Now, a 3S battery pack with these specifications would weigh about a quarter less than a 4S since it has one less cell, but it requires more amps to accomplish the same amount of work. A 4S require fewer amps, but due to the heavier weight it would require more work to be done due to its added weight[16]. Again, the graph above can be utilized to see the relationship between the weight, capacity, and power output of different batteries. There are tradeoffs for each scenario, so an ideal battery must be chosen based on the factors discussed above.

[18] *Quadcoptor LiPo Battery weight/capacity trade-off.* (2012, November 28). Retrieved from Robotics: http://robotics.stackexchange.com/questions/554/quadcopter-lipo-battery-weight-capacity-trade-off

1.5.5 Sensors

Sensors are the link between the physical world and the system. They are crucially important in the design process of the UAV, since they provide all the information needed to perform all the different tasks. These sensors are used for different purposes and they can provide different data. For this project, the UAV needs to know about its positioning, location, stabilization, and other data. As it can be seen, there are several important concepts that need to be provided to the system. All this data is later used and processed according to the needs of the vehicle. This section provides a deeper overview of the different sensors.

IMU

An Inertial Measurement Unit (IMU) is extremely important for any autonomous UAV. The IMU can be used for stability control, velocity estimation, and position estimation. An IMU normally consists of an accelerometer, gyroscope, and sometimes a magnetometer. An accelerometer is a sensor that measures acceleration in the x, y, and z axis. Depending on how the IMU is mounted, these directions will change. The gyroscope is a sensor that measures the roll, pitch, and yaw of the UAV. This sensor is also dependent on how it is placed on the UAV. The magnetometer is a sensor to detect the strength and direction of magnetic fields. They can be used to help calibrate the IMU to ensure that the accelerometer and gyroscope measurements are valid.

The IMU is one of the prime sensors that allow the UAV to fly without crashing. An IMU normally doesn't take up much power but must be processed fairly fast to ensure that no data is lost. If the IMU is not processed fast enough, a UAV could easily flip upside down without correcting itself. The cost of an IMU can still be fairly expensive depending on the brand. The average price of an IMU used for hobby UAVs is around $100, while commercial and military systems require much more highly developed IMUs at a higher price point.

GPS

A Global Positioning System (GPS) device is a helpful and commonly used sensor for a UAV. As most people know, a GPS device can be used to help determine its own altitude, longitude, and latitude positions. A GPS device typically receives a signal from a satellite to calculate these positions. Depending on the GPS device chosen, some devices give the internal clock and standard deviations of its positions.

Cameras

One of the most important sensors that is often used in UAVs is a camera. There are many different types of cameras and they can be configured on the UAV in several different ways. There are some systems that have space for external cameras like the GoPro, but for most of the UAVs the cameras

are built-in. Most of the built-in cameras are printed circuit board (PCB) cameras that connect directly to a microcontroller. The PCB board is fairly small because it doesn't need all of the extra functions of a normal digital camera. For most of the commercial UAVs, the cameras are HD quality, which provides high quality video footage for the user. The use of high quality cameras also affects the system overall because it has additional requirements like higher transmission rate, a better processing memory, and more memory space.

Figure 6: DJI Phantom UAV

In the image above there is an example of a UAV with a GoPro mount, so the camera can be put there if desired. It has to be clarified that in this type of systems the camera is totally separated from the UAV, which means that the camera works independently from the system.

1.5.6 Microcontrollers

Microcontrollers are practically the motherboards of most UAVs. A microcontroller is essentially a small computer with a processor, memory, and programmable inputs and outputs. Microcontrollers are usually where batteries are plugged into and all of the sensors and actuators are split off of. For current UAVs, some microcontrollers are built from scratch just to reduce the cost of the product. Famous microcontrollers include Arduino, Raspberry Pi, and Vex. Depending on the brand and version, a microcontroller can consist of different peripherals and battery voltages. As stated above, the system power and system sensors are important and both are limited by which microcontroller is chosen.

Arduino

Arduinos are inexpensive microcontrollers that focus on flexibility by allowing a large amount of the pins to have different functionality. Their pins include SPI, I2C, PWM, analog inputs/outputs, and digital inputs/outputs. Arduinos also have large amounts of software support that works for any version of the board purchased.

Vex

Vex microcontrollers gained their population from FIRST robotics. These microcontrollers are fairly similar to Arduinos except they are for less experienced users. Their pins include motor/servo control, analog inputs, and digital inputs/outputs. Vex also has a lot of support for their software but doesn't have anywhere near the same amount of flexibility as Arduinos do.

Raspberry Pi

Raspberry Pi is a cheap microcontroller that focuses on graphic interfaces and cameras. The general purpose input/output (GPIO) is not completely supported by the company but community support helps for using the necessary pins. The Raspberry Pi contains two USB ports, an ethernet port, a video input/output, microphone/headphone jack, HDMI port, and camera port. This microcontroller is coded through its own operating system from an SD card that is Linux based.

1.6 Future of UAVs

Recently the military started developing multi-capable, attack drones called MQ-1 Predators. Initially, they were designed as medium altitude, long endurance unmanned aircraft for spying, however developers decided to add a weaponry functionality. After a couple of good test results, they modified the system to increase its attacking capabilities and since then, it is used solely for long distance military operations in Afghanistan and the Pakistani tribal areas and elsewhere.[19]

According to the current trend in the latest Special Operations Forces Exhibition (SOFEX), which was held in Jordan, the military drone market is heading towards an autonomous, smaller size attack quadrotors. They should be able to be self-aware and to suggest actions to its operator.[20]

UAV technology is also being used and implemented in the public sector and other commercial areas. In near future, it is expected that users will be able to transmit camera views to goggles or glasses. This will enable people such as rock climbers or campers to see their location more thoroughly. Using processing software, image stitching, and 3D modeling techniques, people will be able to guide ships in harbors with hard winter conditions by mapping out sea ice thickness so they can navigate through it. This technology can also serve for nature photography, film making, research, and many others applications.[20]

Another promising future of UAV technology, "swarm" technology, is being developed by the GRASP Laboratory of the University of Pennsylvania. Inspired by nature, they developed a system of drones which manifest complex behavior without having a main drone. Each drone has its own simple rules such as steering in the same direction or moving towards the center, etc. Moving as a

---

[19] *General Atomics MQ-1 Predator*. N.p.: n.p., n.d. *Wikipedia*. Wikimedia Foundation. Web. 17 Oct. 2013. <http://en.wikipedia.org/wiki/MQ-1_Predator>.

[20] *Drone On: The Future of UAV Over the US*. *YouTube*. MotherboardTV, 05 Dec. 2012. Web. 17 Oct. 2013. <http://www.youtube.com/watch?v=kwkxx84wXNo>.

group, these drones can be as powerful as a large aircraft capable of hauling a heavy object. Other applications for "swarm" technology are search and rescue missions that would include high risk buildings and sites like Fukushima, crop pollination, traffic monitoring, assisting emergency workers in disaster areas, and overwhelming standard missile-defense systems.[21]

As more technology is being developed, the desire towards drones with longer endurance is growing. The drone company AeroVironment recently developed a 13 pound high altitude UAV called "PUMA AE" with 4 times longer air endurance than its current counterparts. The main idea is to keep UAVs at a high altitude with very efficient solar cells (instead of low efficient silicon cells, gallium arsenide solar cells were used), which could keep drones flying for months, even years at a time. Since it is possible to achieve longer endurance in near future, advancement in camera technology would greatly improve these high altitude UAVs. They could serve as atmospheric satellites.[22] Clearly there is a lot of room for UAV research and development in the future.

[21] Hambling, David. "The Future Of Flight: Swarms Will Dominate The Sky." *The Future Of Flight: Swarm Technology*. Popular Science, 02 July 2013. Web. 17 Oct. 2013. <http://www.popsci.com/technology/article/2013-06/future-flight-swarms-will-dominate-sky>.

[22] Schechter, Erik. "Solar-Powered Drones' Bright Future." Popular Mechanics, 13 Aug. 2013. Web. 17 Oct. 2013. <http://www.popularmechanics.com/technology/aviation/solar-powered-drones-bright-future-15803525>.

2. Initial System Development

2.1 Project Specifications

Once the team determined the problem they wanted to tackle while completing this MQP, project specifications were developed to serve as the driving forces behind various design decisions and component selection. These specifications are an indicator of why this drone should be chosen over others. For instance, being a cost effective system would hypothetically appeal to a larger market. With its desired capabilities, the quadrotor will also be able to be adapted for a variety of purposes, such as aerial support for police patrols or military applications, such as for the collection of photographic intelligence. It will also be convenient for these applications as it is desired to be lightweight and sturdy, with a smaller frame that will allow it to be easily carried. Below are all of the project specifications for this MQP.

1. The drone will be able to fly and hover at a maximum altitude of 100 ft.
2. The entire system must not exceed 2 kg with a payload of 0.5 kg for a total weight of 2.5 kg.
3. The drone will have 4 motors and 4 propellers for its propulsion system.
4. The drone will have “legs” for landing.
5. The drone will be able to survive a fall from 10 ft.
6. The drone will have a minimum of 1 camera to provide imagery during flight.
7. The drone will have a battery life of at least 15 minutes.
8. The drone will be able to sense its location and its altitude.
9. The drone will be able to transmit data wirelessly before, during, and after the flight.
10. The total cost of the entire system will not exceed $1000.
11. The drone will be easy to carry and to deploy. It will be user friendly and intuitive.
12. The drone will be waterproof.

2.2 Materials

The analysis and research on different materials was of great interest for this project because there are several components, specifically all the housing parts, that needed to be manufactured by the team. This made the selection of the materials used very important. It is also important to remark that the selection of the material had a close relationship with the manufacturing tools available, since these machines have certain materials requirements.

2.2.1 Derivations

After obtaining the tensile strength from the material's property, the maximum bending stress that each material should have can be calculated, resulting as a Factor of Safety of 2. All the theoretical calculations are based on the main assumption that the system behaves as a cantilever model which is mostly true for the hovering phase. For the following calculations, the value considered for bending stress is the maximum and assumed to be a constant.

$$\sigma_{bending} = \frac{\sigma_{Tensile}}{2} \tag{6}$$

The deflection formula can be derived and written in terms of bending stress, as shown in the following equations:

$$\sigma_{bending} = \frac{M_y * h/2}{I_x} = \frac{F * L}{I_x} * \frac{h}{2} \tag{7}$$

$$\upsilon_{deflection} = \frac{F * L^3}{3 * E * I_x} = \frac{F * L}{I_x} * \frac{h}{2} * \frac{1}{\frac{h}{2}} * \frac{L^2}{3 * E} \tag{8}$$

In Equations 7 and 8, F is applied force, L is the length of object, M_y is moment with respect to y axis, h is the diameter of the cross-section area, and E is Young's Modulus.

$$\upsilon_{deflection} = \frac{2 * \sigma_{bending} * L^2}{3 * E * h} = \frac{2}{3 * E} * \sigma_{bending} * \frac{L^2}{h} = \gamma * \frac{L^2}{h} \tag{9}$$

The term $\frac{2}{3*E}$ is a constant value, which comes from the material's property, as well as bending stress. γ is defined as a product of bending stress and $\frac{2}{3*E}$. Based on this formula, deflection can be calculated and easily improved with altering the length of the rod.

From the equations derived, it can be seen that deflection does not get affected by the cross-section area. The team created different scenarios for the length of the rods and compared all the materials accordingly. The minimum length of the rod turned out to be 10 cm with which materials like ABS, carbon fiber, etc. can also pass the stress test for a given thickness. To define the minimum thicknesses for each material, the inertia formula, shown in Figure 7 below, is used.

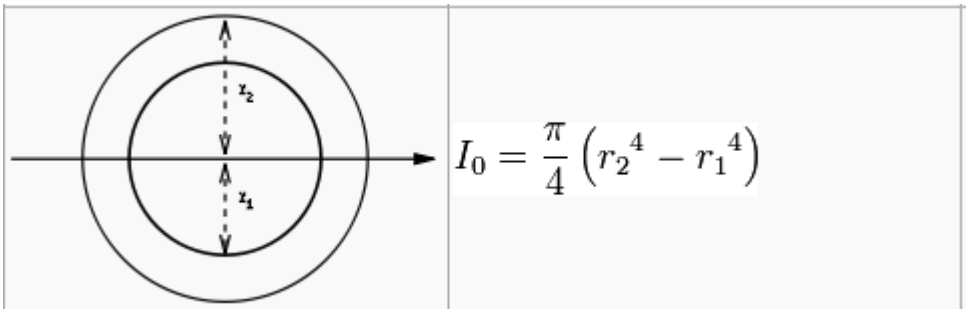


Figure 7: Area Moments of Inertia

$$I_x = I_{outer} - I_{inner} \quad (10)$$

From the material's property, the ultimate bending stress that the material can resist is obtained (Equation 7). When I_x gets equated with the bending stress equation, the following is found:

$$I_x = I_{outer} - I_{inner} = \frac{\mathrm{M_y} * \frac{h}{2}}{\sigma_{\mathrm{bending}}} = \frac{F_{\max Thrust} * L * \frac{h}{2}}{\sigma_{\mathrm{bending}}} \quad (11)$$

The area moment of inertia can be calculated with the maximum theoretical thrust of the motors used and the ideal length of the rod (found to be 10 cm earlier). The inner circle radius can be found from the area moment of inertia, and with this value the minimum thickness can be calculated. The equations used are shown below, where $\frac{h}{2}$ is the outer radius of the rod.

$$I_x = \frac{\pi}{4} * (r_{outer}^4 - r_{inner}^4) = \frac{F_{\max Thrust} * L * \frac{h}{2}}{\sigma_{\mathrm{bending}}} \quad (12)$$

$$r_{inner} = \sqrt[4]{r_{outer}^4 - \frac{4 * F_{\max Thrust} * L * r_{outer}}{\pi * \sigma_{\mathrm{bending}}}} \quad (13)$$

2.2.2 Thickness Calculations for Various Materials

Taking into consideration the materials chosen to further analyze based on background research, the team compared them using the formulas derived in Section 2.2.1. The inner radius of the material is a function of the maximum theoretical thrust applied to the rod, the rod's length and outer radius, as well as the material's bending stress. Maximum theoretical thrust was calculated to be 25.8 N based on the selected motor's properties and propeller's size.

The team initially assumed a length of 20cm and an outer radius of 0.7 cm for the rods and all the calculations in Appendix B are done accordingly. As a result, for a Factor of Safety of 2, Nomex Honeycomb Core, Airex Foam Core, ABS and TangoBlack materials failed the stress requirements that the team set.

Among all the materials, the thickness varies greatly from 0.0015cm to 0.3825cm. The thinnest material was calculated to be 3K, Plain Weave Carbon fiber however it was not selected due to its high cost, manufacturability issues and its high deflection. Each material was compared with its deflection behavior versus its weight. E-glass fiber, was found to have a strong strength; this material spread as thin as 0.00195 cm was enough to pass the stress test. However calculations showed that this material can cause high deflection and weight problems because it is super dense.

2.2.3 Decision Matrices

Decision matrices were created to oversee the effect of different factors for material selection. These can be found in Appendices B-E; the decision matrix for the rods is demonstrated below as an example. The team defined five important factors for selecting the right material for the rods: deflection, weight, impact strength (which is the resistivity of material against collision), cost, and water resistivity. Since rods are the support elements of the quadrotor and receive the highest stress and moment affects them, the highest weight was given to the deflection property. From the calculations, the best material for the lowest deflection behavior was found to be Aluminum 6061, but its high weight made it not feasible. Overall Balsa Wood Class IV and Styrene (plastic) were some of the better options; however, they were eliminated due to wood's poor water resistivity and styrene's poor impact strength. PLA was chosen as the main material for the rods.

Decision Matrix -Rods

Material Name	Stress Analysis	Deflection	Weight	Impact Strength*	Cost	Water Resistivity	Total
Nomex Honeycomb Core	Fail						
3K, Plain Weave Carbon Fiber	Pass	3	2	2	5	2	2.85
Generic Carbon Fiber	Pass	3	4	2	4	2	3.2
Airex Foam Core	Fail						
Aluminum 6061	Pass	1	5	3	3	3	2.7
E-glass Fiber	Pass	5	5	3	4	2	4.35
PVC (hard)	Pass	2	4	4	2	2	2.7
Styrene (plastic)	Pass	2	3	4	2	2	2.45
Balsa Wood Class IV	Pass	2	2	5	1	5	2.45
PLA	Pass	2	3	3	2	2	2.35
ABS	Fail						
TangoBlack	Fail						
VeroClear	Pass	3	3	4	2	2	2.85

*This property is the resistivity of material against a collision

Weight Distribution

Deflection	Weight	Impact Strength*	Cost	Water	Total
0.4	0.25	0.1	0.15	0.1	1

Figure 8: Decision Matrix for the Material Selection of Rods

For the middle hull's decision matrix, the team removed the stress and deflection parameters, because they were found to have a very small effect on the system after consulting with mechanical engineering professors and conducting research. However, the rods were secured by fillets to the middle hull connections to avoid stress concentrations. The team defined four important factors for selecting the right material for the middle hull: weight, impact strength, cost, and water resistivity. The highest weight was given to the weight parameter since the middle hull is large and will be the heaviest part of quadrotor. From the calculations, the best materials for the lowest weight were found to be the Nomex Honeycomb Core and the Airex Foam Core, and their overall material properties had the best score in the decision matrix. However due to manufacturability issues, the team decided to go for the second best material options. Balsa Wood, PLA, and ABS were overall in the second best material category. To ease the manufacturing by using a 3D printer, PLA was chosen as the main material for the middle hull.

2.2.4 Results

The team was inherently aware of the importance of selecting the right material for the system. After conducting careful calculations on the rod's weight and deflection, the spherical/cubic design for the middle hull, and thickness-stress testing, the team came up with the most feasible material among all these options for the given dimensional properties of the prototype. Carbon fiber has been traditionally the most desired option for drones, however due to its high deflection properties, cost and manufacturing issues led the team to consider other materials in the decision making process. At the end of this process, PLA was found to be the most feasible option given its extensive use for rapid prototyping, good weight to deflection ratio and cost.

The variations of a spherical versus a cubic design were compared with different scenarios for each material, and it was found that a spherical design with 0.2 cm thickness has better properties overall than a cubic design for the middle hull. Based on the chosen material, PLA, the thickness was calculated to be 0.23 cm for the rods. The team also learned that the best material based on the decision matrix evaluations cannot always be selected due to manufacturability capabilities of facilities available. Even though PLA was selected to be the best for this project, the middle hull could be made with Nomex HoneyComb Core if any future team has the capability to manufacture and shape carbon fiber cores. Overall, a carbon fiber-PLA hybrid mix would provide better results.

A more detailed description and overview of the calculations and the decision matrices is given in the Appendices.

2.3 Router

There were a few parameters considered for the selection of the router. The most important ones were: range, speed, and cost. It needed to reach an altitude of 100 feet and be able to communicate with a base station. It is necessary to have reliable communication all the time, which is the reason why having an efficient router is so important. The desired range was at least 100 feet outdoors with a transmission speed over a few megabytes per second. Multiple images are sent every second and each image is only a few hundred kilobytes. From the math, a 54 Mbps router is plenty to gather all of the pictures and send information to the quadrotor at the same time. It was even considered that with this router, multiple quadrotors could be attached and controlled by the ground station. A more likely problem for limiting the processing power is that the Raspberry Pi can't gather the image data fast enough. Another objective for this project was to make the UAV low-cost, so the router chosen had to meet all three of these parameters.

The Linksys WRT54GL, seen in Figure 9 below, was the router selected for this project and is shown below. It has a range of 300 feet outdoors and a maximum speed of 54 megabytes per second, which clearly meets all the design parameters defined previously. The 300 feet range also allows for an estimated 280 foot movement in longitude and latitude. The price for this router was $45, which is also between the desired price ranges.

Figure 9: Linksys WRT54GL Router

2.4 Motors and Propellers

In order to pick the appropriate motors for the quadrotor, many factors were taken into consideration. The selected motors had to be capable of lifting the whole quadrotor and an extra payload of 500 grams. Added to that, the motors themselves needed to be light and work without using too much current in order to make the battery last longer, thus resulting in a longer flight time.

As mentioned in the Section 1.5, the motors had to be selected at the same time as the propellers in order to get the best and most effective combination. The propellers come in different sizes and with different number of blades. The bigger the propeller, the more thrust it generates, but it also gets heavier and makes it harder for the motor to reach its max performance.

For this project many motors were reviewed, all of them with different values for weight, size, cost, and the Kv constant, which is the number of rotations per minute (rpm) per volt. A group of 5 was then picked for deeper analysis. The decision matrix and motor parameters, as well as the weight placed on each factor, can be seen in the Appendices. After all calculations were completed, the C2020 Micro brushless Outrunner 3500Kv was the motor chosen to operate the quadrotor. This motor turned out to be the best of all the options considered since the amount of rpm it could output was really high while using less current. At a max voltage of 11.1 this motor is theoretically capable of producing 38850 rpm. The chosen motor can be seen in the Figure 10.

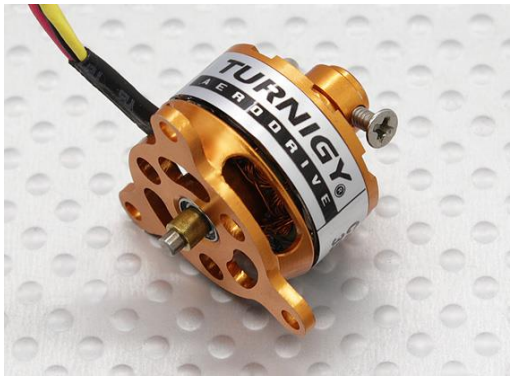


Figure 10: C2020 Micro Brushless Outrunner 3500Kv

At the same time the group had to select the size of the propellers to be used. After browsing for the available options and calculating the static thrust created while being used with the chosen motor, it was decided to use 6x3 standard propellers. Theoretically, these propellers will be capable of generating 2.59 Kg of thrust while running at max speed. At the same time, these propellers are not too heavy, allowing the motor to run at the desired speed without getting too hot or breaking. The chosen propellers are seen in Figure 11.

Figure 11: 6x3 Standard Propellers

2.5 Battery

Based on the initial research into various batteries, the ideal choice seemed to be one with a higher capacity, which would therefore increase the flight time of the quadrotor. A significant factor was balancing capacity and weight. All of the batteries were fairly similar in a weight to capacity ratio, so a 4900mAh battery was chosen as it was projected to give a flight time of about 20 to 30 minutes without adding too much weight to the drone. This seemed like the ideal choice, and the battery was purchased. Initial projections when considering smaller capacity batteries indicated that the flight time would only reach about 10 minutes, which falls short of the project goals. Due to this reason, smaller batteries were not chosen. However, further research demonstrated that the purchased battery may not be the best one to use. For starters, the one purchased is longer than ideal as its length impacted the size of the hull, increasing the overall size of the quadrotor. After

creating the decision matrix, it showed that the smaller capacity battery had a better weight to capacity ratio and shorter length. Therefore, a new consideration is to buy two 2500mAh batteries and wire them in parallel. This will give the same capacity, but with slightly less weight. The shorter length will also allow more flexibility in the hull design, although a thicker width will need to be accounted for when fitting all of the components together. This idea will be furthered more once testing is completed and transition into a redesign for prototype 2 is made.

2.6 Motor Speed Controller

In order to control the amount of current being fed into the motor, it is imperative to use a motor controller. This motor controller would prevent the motor from breaking or burning, and it would also prevent a short circuit from happening. Since the motors chosen for the project were small and did not need huge amounts of current to operate, the speed controller could actually have a low amp value. For the project the Mystery 12A Brushless Speed Controller was chosen. As its name suggests, it allows up to 12 amps of current to flow. This speed controller has some safety functions like reducing power if the temperature goes above 120°C and reducing power or shutting off as a low current protection. This speed controller will be connected to the motors on one of their ends, and to the battery and the Arduino UNO on the other. The selected controller is shown below in Figure 12.

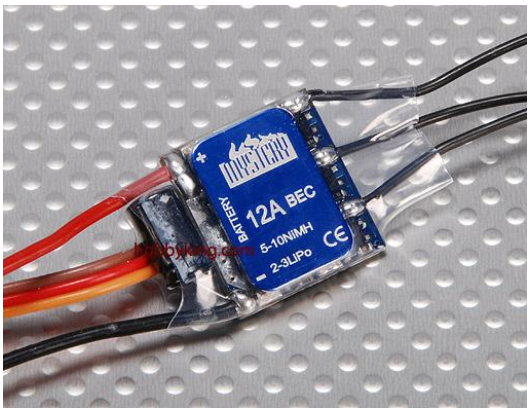


Figure 12: Mystery 12A Brushless Speed Controller

2.7 DC to DC Converter

One of the most important components of the quadrotor is the Raspberry Pi, which controls the cameras and the image interface. This element can only operate at a voltage of 5V. If the voltage goes higher than 5V, the Raspberry Pi could suffer damage and break, altering the performance of the quadrotor. Since the voltage that is being output into the Raspberry Pi needs to stay at 5V, a DC to DC converter needs to be placed and connected between the source of power and the Pi. While several converters were researched, the OKI-78SR-5/1.5-W36-C converter was ultimately

chosen for this project. This converter, shown in Figure 13, converts voltages from 7V up to 36V into 5V, allowing the Raspberry Pi to work properly without risking breakage.

Figure 13: OKI-78SR-5/1.5-W36-C DC to DC Converter

2.8 Previous MQP Components

Prior to working on building a quadrotor for surveillance purposes, a group of students was given an MQP for creating another device for surveillance. From their work, some parts that they had purchased were deemed useful for this project as well. The microprocessors for the project were chosen to be the Raspberry Pi and Arduino Uno. Although there are many other microprocessors, these microprocessors were believed to have enough support for a fairly cheap price. An Arduino Uno was used in the previous project to control motor speeds and work an IMU. The Raspberry Pi was considered for their project but didn't work with the cameras that they had used so another microcontroller was bought. However, the Raspberry Pi was chosen for this surveillance project due to the fact that it could work with other cameras and also could have a GPS and Wireless adapter plugged into it. As of now, all of the devices attached to the Raspberry Pi work as intended.

The cameras that are used for the project were also researched by a previous group. The cameras are USB compatible and give a fairly clear image from a height of 100 ft. These cameras were not tested in the Raspberry Pi and instead were used by another microcontroller. After testing, it was seen that these cameras work perfectly well with the Raspberry Pi through USB. One drawback to the Raspberry Pi was that it could only supply 500 amps, which is not enough to power all of the devices attached to it. Due to the need of more amperage, a USB hub was also included into the design so that the Raspberry Pi didn't get overpowered.

The Inertial Measurement Unit (IMU) that was chosen for the project was also the same IMU tested by the previous group of students. This IMU was already programmed for the Arduino Uno, but needed some adjustments to get the Gyroscope working properly due to the drift. These controls are discussed in the next chapter. The cost of the IMU is about $90, which seems overpriced, but not many other IMUs are cheaper. The IMU being used for this project has six degrees of freedom, which allows us to control the trajectory of the quadrotor reliably, while also accounting for uncontrolled elements, such as the wind.

3. Design

Early on in the project, the team began working on the project specifications as well as the preliminary designs. Although the team knew the basics of what was desired for the design--i.e. it would be an unmanned aerial vehicle with more than one rotor--specifics about the design had to be not yet been chosen. Thus the processes of creating the preliminary designs aided with specificity and brevity of the developing project specifications during the early phases of the project.

3.1 Initial Concept Designs

The members of the project group came up with several initial concepts for the quadrotor. For brainstorming purposes, each member came up with a few preliminary designs. Most were simple sketches, but some were 3D CAD models, as shown in Figure 14.

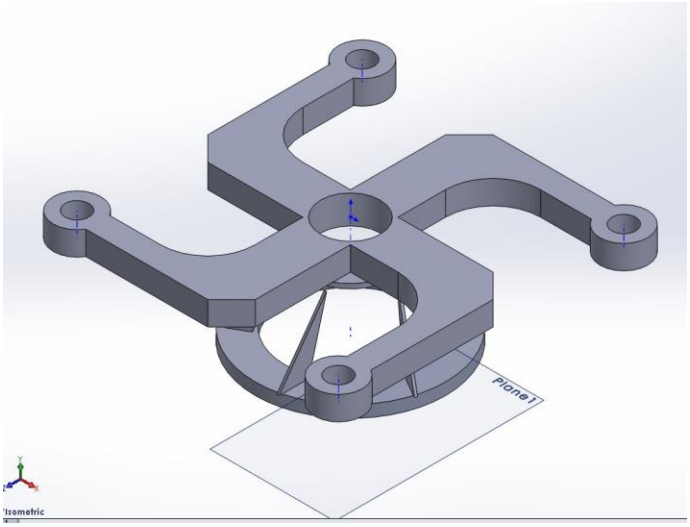


Figure 14: Preliminary Design Concept

Although the majority of the initial ideas were quadrotors, several of the designs had different qualities. For example, some of the initial designs included room for solar panels, night vision devices, or even parachutes. The design above was created with the concept of a vortex or tornado and actually was the only design to have 5 rotors (the middle would include a larger rotor).

After initial brainstorming to determine the design basics, the team was able to make several decisions about the design. For instance, it was decided to create a quadrotor, as opposed to having more or less rotors. The reason for this was to create an unmanned aerial vehicle that would be easy to reproduce as well as easy to fly, whereas utilizing five rotors would considerably complicate the flight dynamics. Using fewer rotors would sacrifice stability and the hover power of the drone. Also, several of the more extravagant ideas for the design, such as solar panels and night vision were no longer considered. The team decided that realistically, ideas such as these could potentially be added to the prototype later on in the project, but were left aside from the

project specifications. The team decided that the main purpose of the drone would be for surveillance, and thus all further designs would include space for the cameras.

3.2 Graphic User Interface (GUI) Design

The design of the Graphic User Interface (GUI) had to be useful for the user and also user-friendly. In order to make it user-friendly, it was necessary to make the GUI as simple to use as possible, as well as clear and concise.

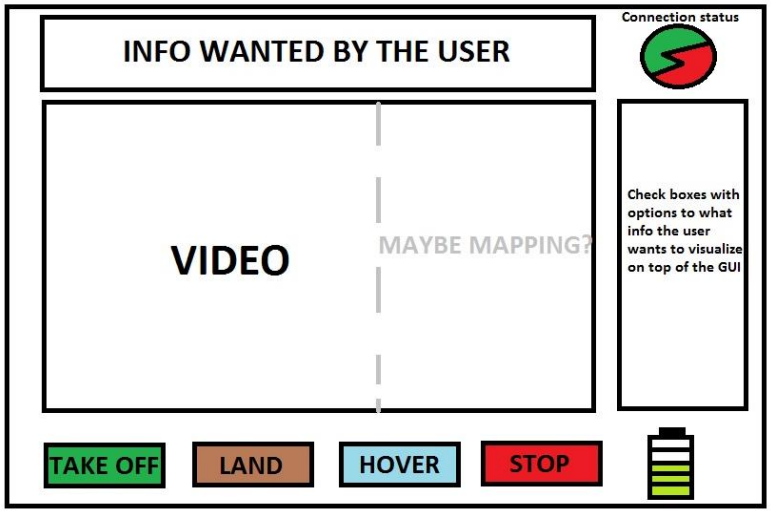


Figure 15: Initial GUI Design

The initial design of the GUI can be observed in Figure 15. There are control buttons, a UAV information panel, a video panel with a possible extension for mapping, and battery and connection statuses. In the control buttons panel there are four options: take off, land, hover and stop. These are the only four options that should be available in autonomous mode for the UAV, since the system is performing the rest of the tasks. On the top part of the GUI, the information about the UAV should be displayed. The idea was to have some important information as a fixed part of the information panel, but also provide the user some options to be selected from the panel on the right. The main reason to give the user the ability to see or not see some information was because as a user there is some information that is not necessary or useful for the current purpose. The fixed information is the X, Y, and Z positions. The video panel in the center of the GUI is probably the most important part of it, since this is where the user's center of attention is going to be. There is also a consideration for a possible extension of this panel to include mapping that should also be visible for the user. The last piece in the GUI is the connection and battery statuses; these two components have to be displayed because of the importance of maintaining both and the need to troubleshoot them.

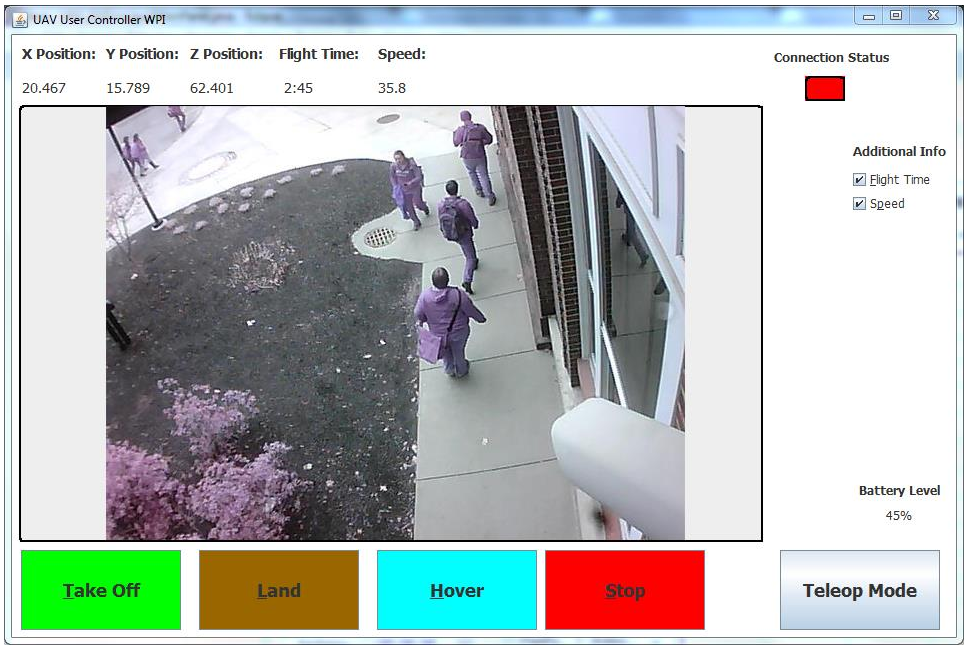


Figure 16: Final GUI Design

Figure 16 shows the latest update on the GUI design, and what the user should expect to see on the base station. Comparing this design with the original one, it can be seen that the main shape and content of the GUI was kept. All the information that needed to be displayed is very visible for the user. The most important thing is that this interface is very simple to use and self-explanatory, which means that the operator does not need significant training in order to use it.

3.3 Controls and Simulation

For creating the controls and simulation, Matlab was used to simulate a quadrotor by coding the dynamics and having a quadrotor figure move around on a 3D plot. To understand how to control a quadrotor, a model was used based on articles written by users that had experience with quadrotors. The models inputs are designated in Figure 17.

$$\begin{bmatrix} 0 \\ 0 \\ U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ b\left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right) \\ b\,l\left(\Omega_4^2 - \Omega_2^2\right) \\ b\,l\left(\Omega_3^2 - \Omega_1^1\right) \\ d\left(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2\right) \end{bmatrix}$$

Figure 17: Quadrotor Inputs Model

In this model, the Ω variables represent the angular velocities of the four motors. The constants *b, l,* and *d* represent the motor thrust coefficient, distance from the motor to the center of the quadrotor, and drag coefficient respectively. The values of *U* are the forces and torques that are applied to the quadrotor. The given dynamics of the quadrotor were a foundation for developing the controls. The dynamic equations can be seen in Figure 18.

$$\begin{cases} \ddot{X} = (\sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi)\dfrac{U_1}{m} \\ \ddot{Y} = (-\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi)\dfrac{U_1}{m} \\ \ddot{Z} = -g + (\cos\theta\cos\phi)\dfrac{U_1}{m} \\ \dot{p} = \dfrac{I_{YY} - I_{ZZ}}{I_{XX}}\, q\, r - \dfrac{J_{TP}}{I_{XX}}\, q\, \Omega + \dfrac{U_2}{I_{XX}} \\ \dot{q} = \dfrac{I_{ZZ} - I_{XX}}{I_{YY}}\, p\, r + \dfrac{J_{TP}}{I_{YY}}\, p\, \Omega + \dfrac{U_3}{I_{YY}} \\ \dot{r} = \dfrac{I_{XX} - I_{YY}}{I_{ZZ}}\, p\, q + \dfrac{U_4}{I_{ZZ}} \end{cases}$$

Figure 18: Quadrotor Dynamics

In these equations, θ, φ, and ψ represent the angles of roll, pitch, and yaw respectively. The accelerations of the X, Y, and Z positions are determined by the current orientations of the quadrotor and the force represented by U. The values of *p, q,* and *r* represent the angular velocities of the quadrotor. The angular accelerations are solved from the current angular velocities, torques, and necessary constants. The values of *I* represent the inertia of the quadrotor about the x, y, and z axes. The value of Ω in this equation is the total angular velocity from the motors that cause a gyroscopic effect on the system.

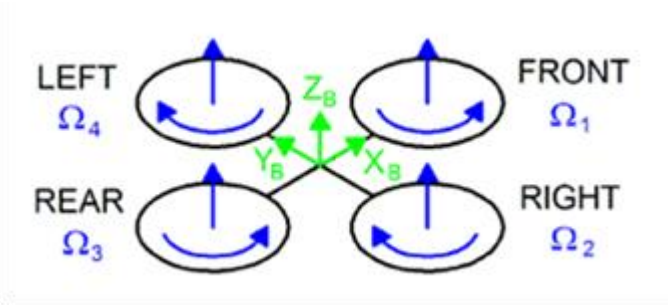


Figure 19: Quadrotor Physical Model

After understanding these equations, PID and PD controllers were created to control the Z position and the three angles of rotation as seen in Figure 19 above. Creating these simple controllers made a very simple but effective way of hovering. The next thing that needed to be created was control over the X and Y position so that if wind had pushed the quadrotor aside, it could re-center itself over the desired X, Y, and Z positions. The team was unable to develop the controls nonlinearly, so a linear approach was applied instead. A PD controller was created for the X and Y positions

by controlling the angles that corresponded with their movement. Figure 20 describes in a block diagram how these position controllers would work.

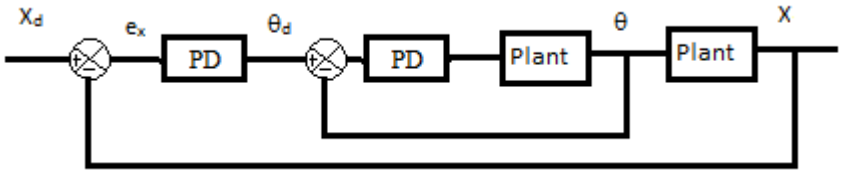


Figure 20: X-position Block Diagram

In Figure 20, e_x represents the error in the x direction. This error is used to control the desired θ position through a PD controller. After applying another PD controller based on the error of θ, the output is used to calculate the motor velocities which will make the robot change orientation. This control loop will iterate continuously until the error of x and θ are 0. After simulating this approach, it was seen that it would converge if the controls of the Z position were not changed frequently. Figure 21 shows that without a trajectory and by simply setting a desired X-Y-Z position, the quadrotor will move appropriately.

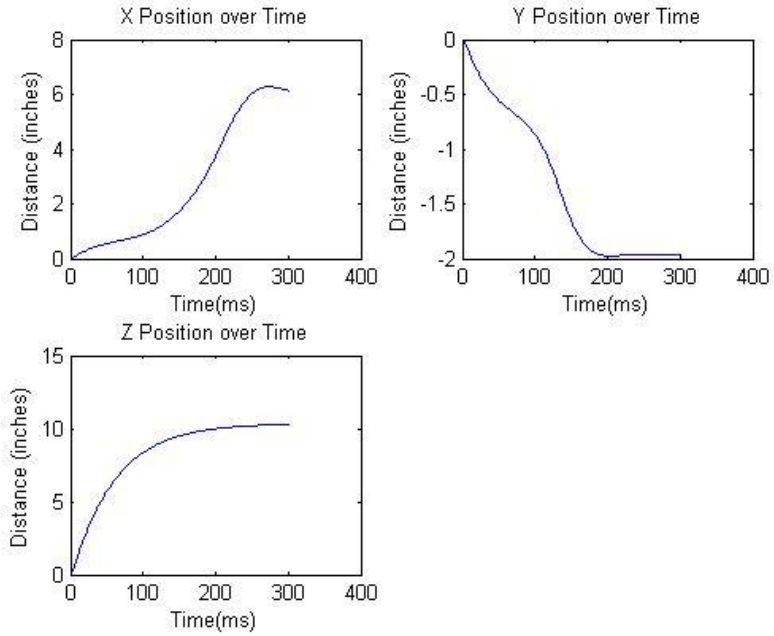


Figure 21: Basic X-Y-Z Position Control

Different trajectories were attempted to see how the control algorithms would act. Linear, sinusoidal, and quintic trajectories were all attempted to see how well the quadrotor would move. The trajectories can be seen in Figures 22-25.

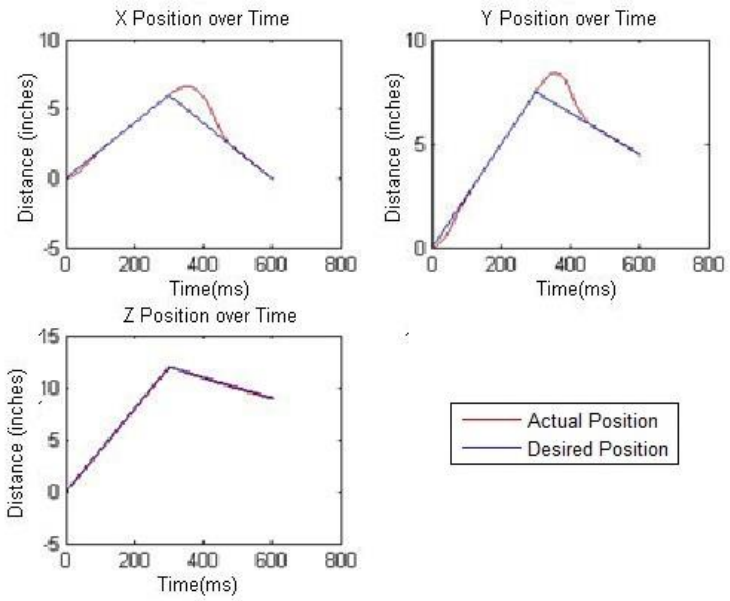


Figure 22: Linear Trajectories

As it was observed in Figure 22, the linear trajectory worked perfectly for the Z position but there was a delay in the X and Y position controls because of the nonlinearity of their motion. The positions will always converge on these linear trajectories but the delay is not preferred. Figure 23 shows sinusoidal trajectories for the X, Y, and Z positions.

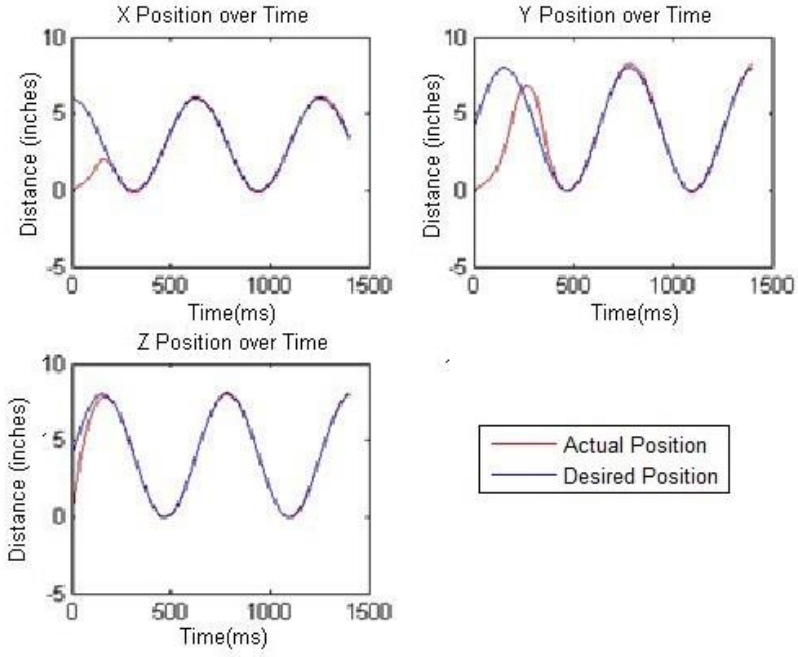


Figure 23: Sinusoidal Trajectory

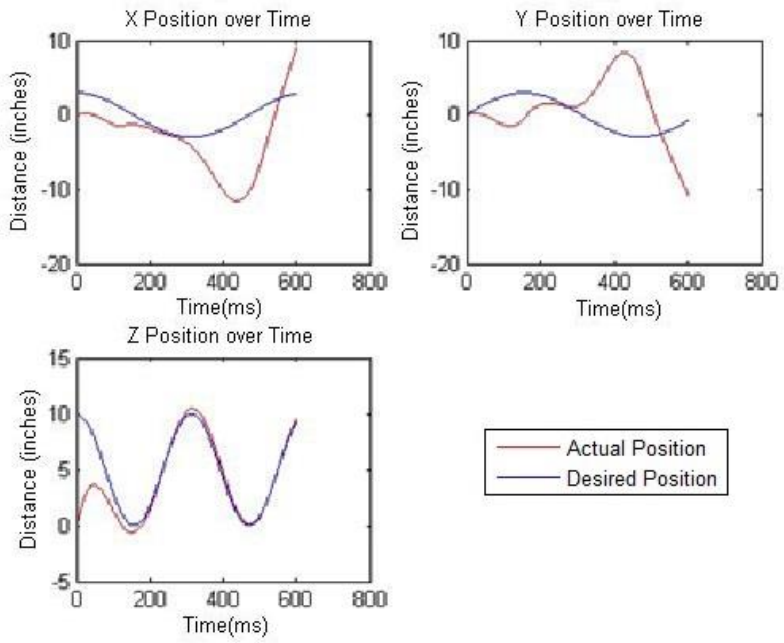


Figure 24: Higher Frequency Sinusoidal Trajectory

The sinusoidal trajectories address some interesting problems with the controls of the X and Y positions. The higher the rate of change for the Z position controls makes the X and Y controls have the possibility of becoming divergent. Although sinusoidal trajectories will not be directly used in the project, it is an important observation for controlling the positions. A quintic trajectory was attempted and results are shown Figures 25-28.

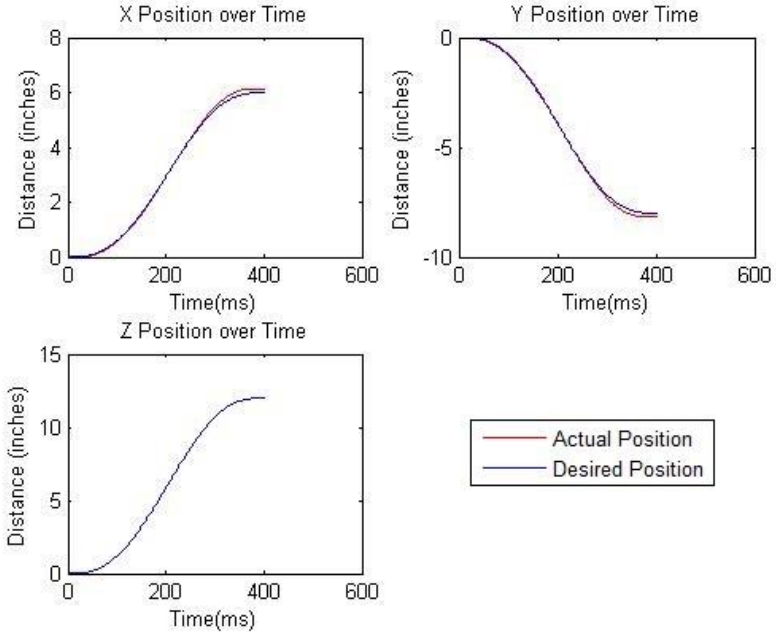


Figure 25: 0.4 Second Quintic Trajectory

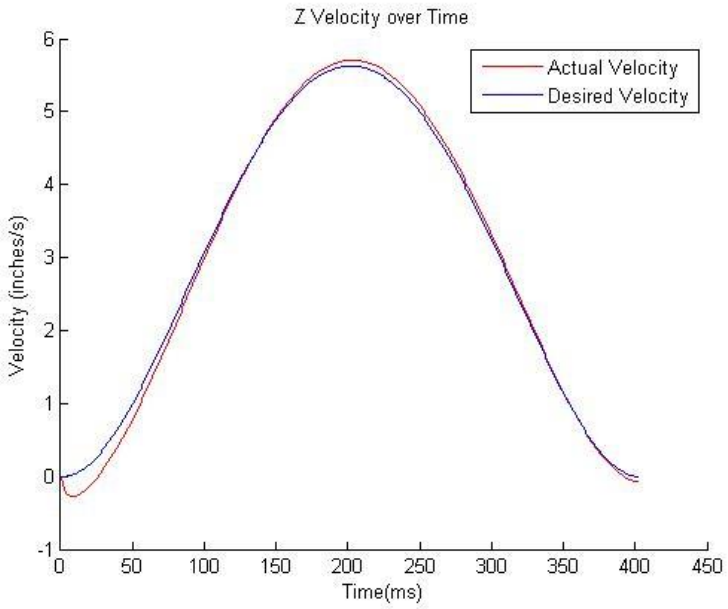


Figure 26: 0.4 Second Quintic Velocity Trajectory

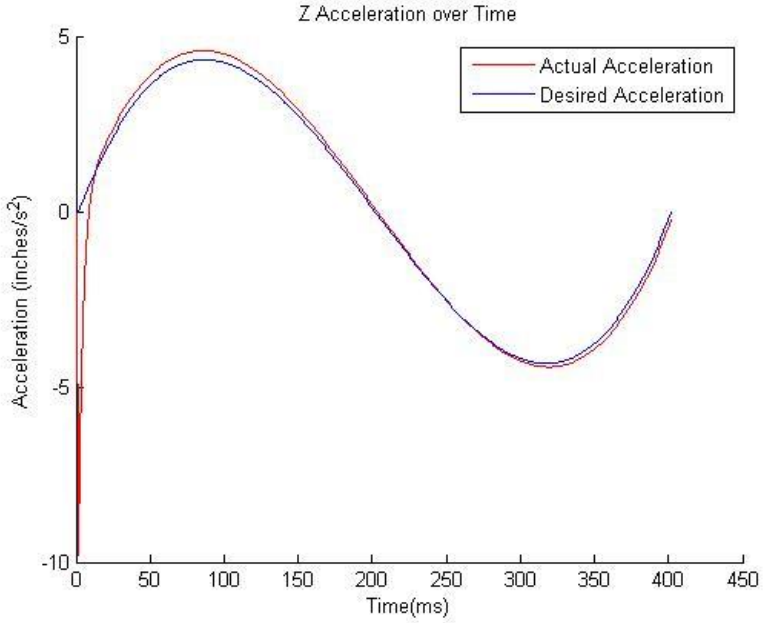


Figure 27: 0.4 Second Quintic Acceleration Trajectory

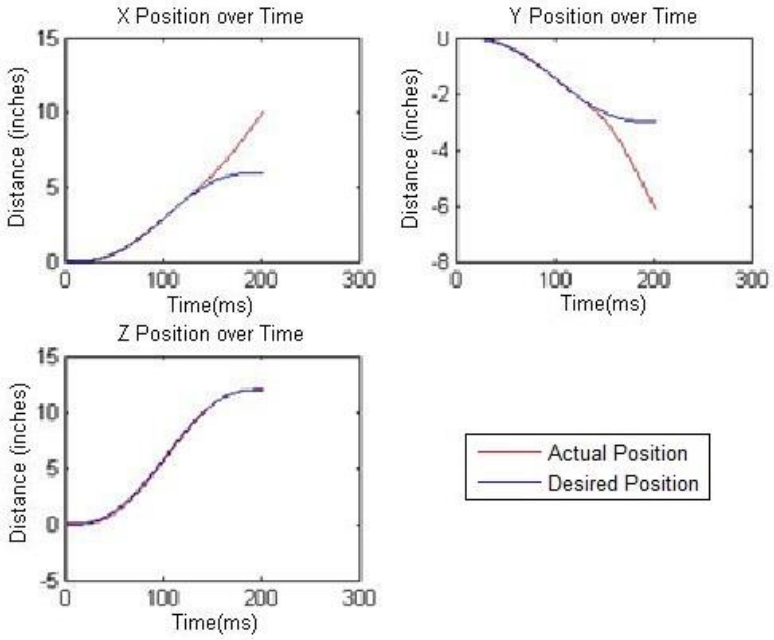


Figure 28: 0.2 Second Quintic Trajectory

As seen in Figures 25-28, the quintic trajectory works perfectly if the time duration is large enough. A shorter time causes the Z position control equation value to increase drastically, affecting the control over the X and Y positions. The quintic trajectory gives a great ability of ensuring that there are near negligible amounts of excess velocity and acceleration when the quadrotor reaches its desired position. This is useful for the internal components and can later be transformed into a more complex path planning technique.

As of simulation completion, it was believed that a basic control system had been created and that prototype 1 could be able to hover effectively, but the controls for moving in the X and Y positions were highly tested to ensure that it did not diverge like the sinusoidal case. A nonlinear control theorem could be applied to the X and Y positions to ensure that the entire system is completely stable.

After conducting tests on Prototype 1 it was determined that the motor-propeller configurations had to be changed because the produced thrust was not enough to lift the drone. Therefore, the stability controls were not tested until Prototype 2 was assembled. These results are described in Section 7 of the report.

4. Prototypes

Prototyping is one of the most important parts during the course of a project. The fact that the prototypes are a real version of the designs, usually made using software tools, can provide a better perspective of that design. This is really important since, as it is known, simulations are a good approximation of the reality but are not identical, and can also carry some issues that are not necessarily observed until after the model is built.

The idea for this project was to have at least two fully assembled prototypes by the end of the project. The fact that this consideration was made since the beginning means that the team always had in mind the idea of improvement of the designs. Another important consideration that needs to be made is that, even though the team considered on having at least two prototypes, there were several small changes made on each of the prototypes. This because there is no need to redo an entire prototype when the changes that need to be done are small. As it was mentioned before, the idea was to have a finalized model of the project fully assembled by the end of the project.

There were several procedures that took place to manufacture the different prototypes. The first step was to design the prototype on a software based tool. This was probably the most important step in the whole process, since here the prototype was basically made. The team decided that this design process had to be reviewed several times by different team members to ensure that everything was done correctly. The following step was the manufacturing of the different parts, which was done either by 3D printing or laser cutting. This step was very simple, since it was only necessary to put the CAD models in the respective machine with the right configurations. The final step was the assembly of the model with all its components. It was after this last step and some testing that some conclusions were derived for each model, and the small changes mentioned before took place.

Sections 4.1 through 4.3 provide a deeper description of each of the prototypes, including the models and important results and conclusions made.

4.1 Prototype 1

After determining initial project specifications and brainstorming several possible designs, the next step was to select one of the preliminary designs, add more detail, and finish the first design. The team decided to create the hull in the shape of a sphere to improve aerodynamics of the quadrotor as well as to look good aesthetically. The sphere was chosen over a cubic shape since it had a drag coefficient of 0.5 which was smaller compared to the 0.8 of the cubic structure[23]. Also, instead of having a center plate in the middle of the hull for all the rods to connect at or only having two long rods, the team decided to attach the rods to the exterior of the hull in order to have more free space

[23]The Engineering ToolBox (n.d.). *Drag Coefficient*. Retrieved from http://www.engineeringtoolbox.com/drag-coefficient-d_627.html

in the hull for the sensor module, as well as to make it easier to replace the rods if they break. Holes were created for the cameras, antennas, and wires for the rods.

4.1.1 CAD Modeling and Manufacturing

Solidworks 2013 was used for the 3D modeling portion of the project. Below is the completely assembled model with the antenna and the camera assembled as well as the motors attached to their rods.

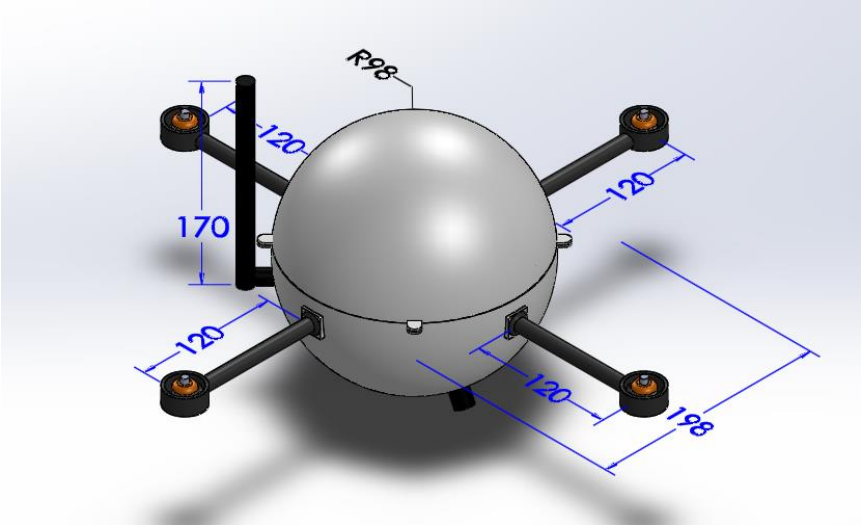


Figure 29: Prototype 1 Isometric View

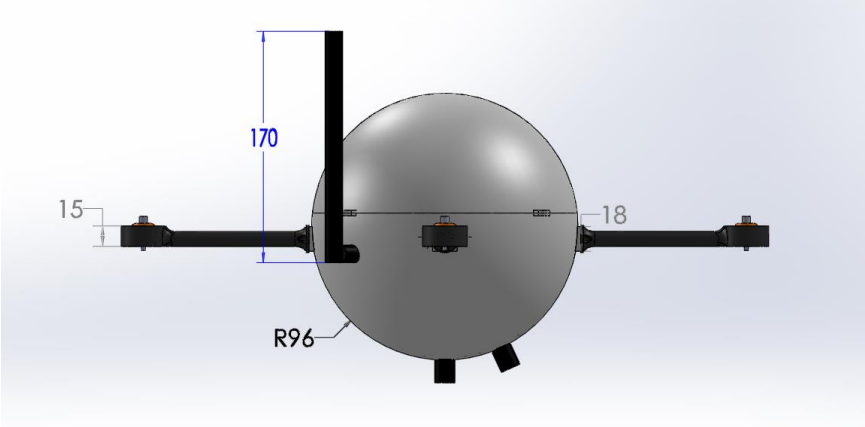


Figure 30: Prototype 1 Left View

The above model in Figures 29 and 30 was rejected for 3D printing due to the hull top and bottom shells being too large for the 3D printer. Thus, the hull was divided into four separate and smaller. Also, due to the additional hull separations, flanges were created on the edge to use fasteners to keep the hull together. Additionally, screw holes were added where they were missing on the flanges as well as on the arms, as seen in Figures 31 and 32.

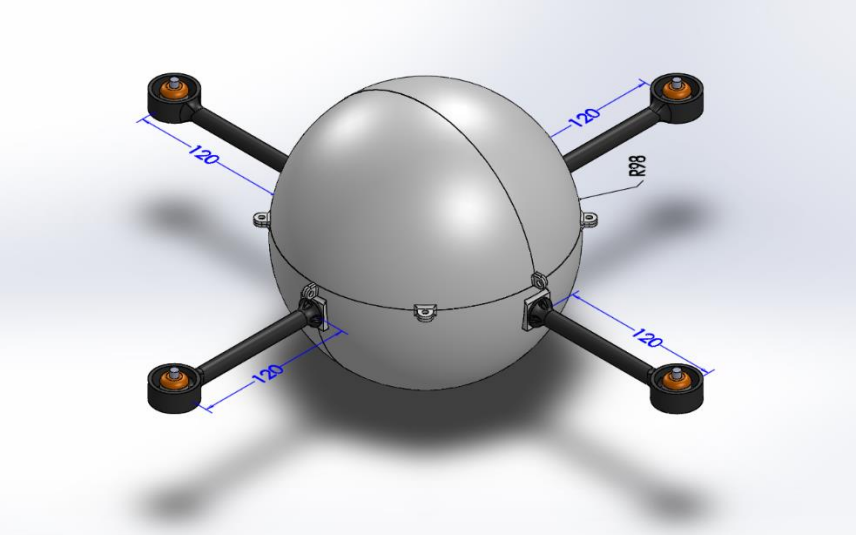


Figure 31: Modified Prototype 1 Isometric View

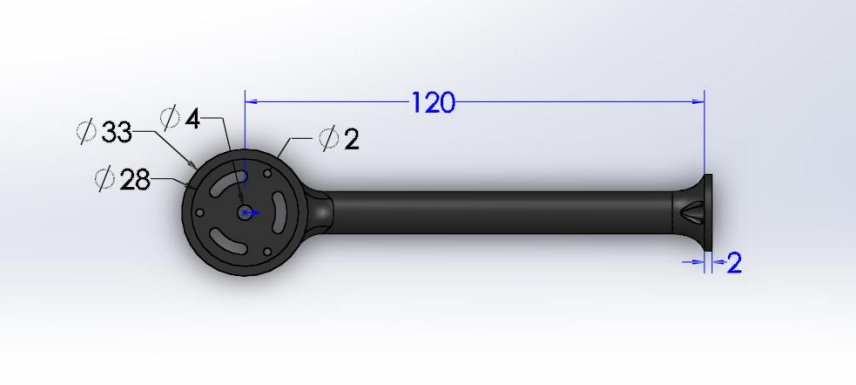


Figure 32: Modified Prototype 1 Rod

The UAV is moderately sized in comparison to average quadrotors. The hull is slightly larger than normal due to the battery needing to fit within the hull. The hull's outer diameter, and thus its height, was 196 mm with a thickness of 2 mm. The flanges for the arms were each 1 mm at their widest point. The rod's length was 139 mm. The total UAV length and width was 476 mm, which is slightly larger than a foot and half in length. This moderate size enables the UAV to navigate through a variety of spaces without issues. For instance, it can fit through door frames, hallways, and rooms with no issues and is small enough to fit under tables and most chairs if desired.

4.1.2 Results and Conclusions

The team learned several lessons during the creation and testing of prototype 1. First of all, when 3D modeling in CAD, more detail is needed in order to have a simpler assembly process. For example, some screw holes were missing as well as a mounting structure for the electronics. Thus,

the team decided for prototype 2 to work from the inside-out for the design portion; namely, this means to start with the mounting positions for all of the electronics and then building around it.

Another issue that the team faced with prototype 1 was the tolerance of the 3D printer. The 3D printer was not as accurate as expected, and thus the team had issues assembling the four parts of the hull since they did not all line up perfectly. For this reason, as well as strength issues with the material, the team considered different manufacturing methods for creating the next prototype.

The team also learned some other basic lessons. For instance, ordering parts well in advance is imperative; several parts such as the battery charger arrived late, leading to further setbacks in the project's development. The team also learned more about each of the teammate's working styles and how to be more productive with meetings by prioritizing and delegating tasks.

4.2 Prototype 2

The team worked on a prototype 2 right after completing analysis on prototype 1. The outcomes were carefully considered, such as the mounting and assembling issues raised in the previous prototype. These were found to be the crucial factors for designing the next prototype. Due to the large size of spherical middle hull in the prototype 1, the 3D printer had to manufacture it in 4 quarter parts and the team had to assemble these parts with a hot glue gun and other small size screws. This caused instability and mechanical weakness at the flanges holding the screws as well as at the reinforced/glued parts. These did not have a high tolerance in terms of stress and strain due to the vibrations caused by motors and also due to impact forces such as those occurring during a drop test. Another issue was the difficulty of mounting the boards, batteries and other electronic parts on non-flat surfaces. These problems led the team to redesign the hull. The new UAV hull was divided in 3 parts: a bottom base, middle cubic hull and the hull top. The hull top was added to compensate the loss of aerodynamics when the model was changed from a sphere to a cube. The hull bottom was added to give a strong base and support for the drone, having additional mountings for the landing gear subassembly and the battery housing, which are two additional features of prototype 2. Additionally, the hull base had mounting for the downward facing camera and holes for the hull walls to screw into. The hull walls had holes for the motor housing arms, which were essentially unchanged from the first design except for a few size changes.

4.2.1 CAD Modeling and Manufacturing

Solidworks 2013 was used for updating 3D model. Figures 33 and 34 show the completely assembled model with motors, propellers, and landing gear.

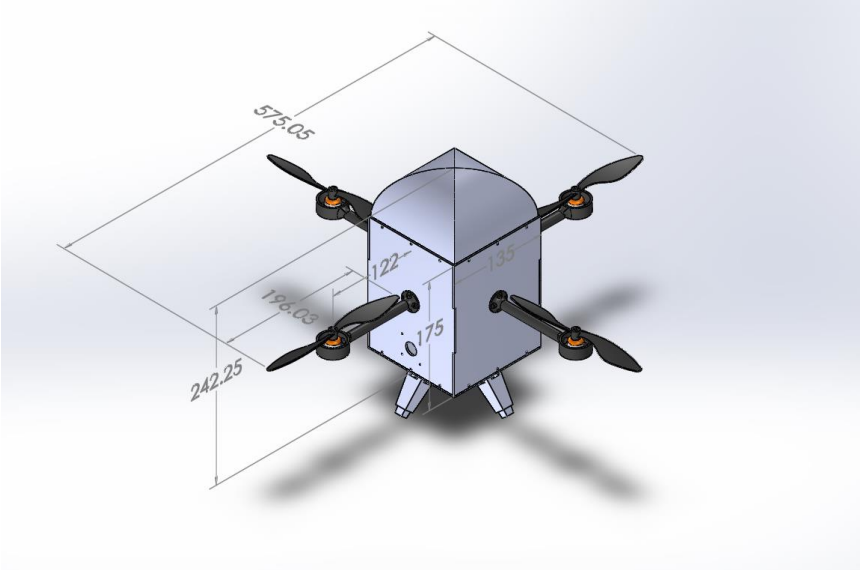


Figure 33: Prototype 2 Isometric View

As seen in Figure 33 above, this prototype is also moderately sized, again allowing for the common maneuverability desired from most quadrotors. The drone is also narrow enough to get through doorways and hallways.

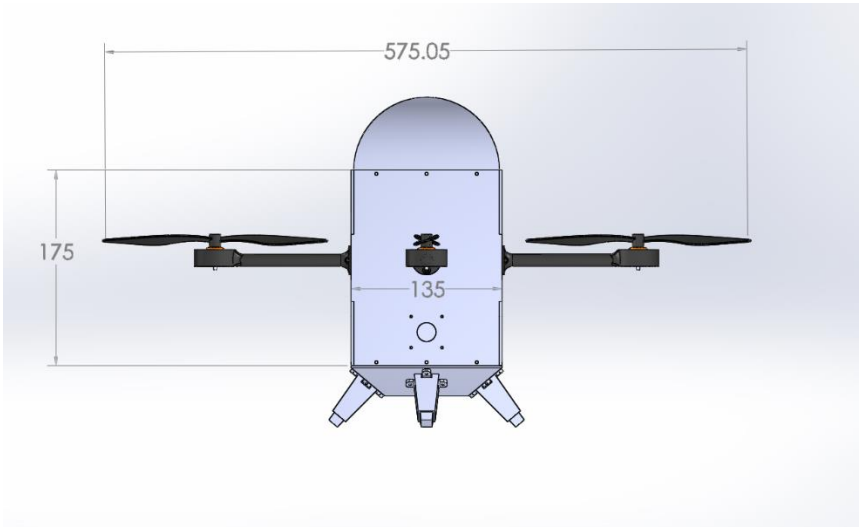


Figure 34: Prototype 2 Left View

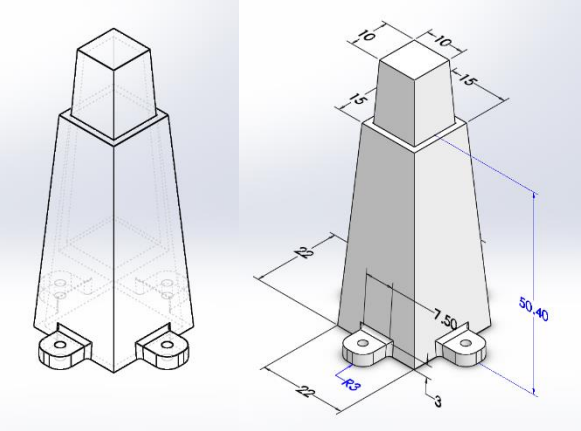


Figure 35: Landing Gear Sub-Assembly

The landing gear subassembly, seen in Figure 35, consists of 3 parts: the inner leg, the outer leg casing, and the spring. The inner leg and outer leg casing were designed based on spring and impact force calculations for the given estimated weight of the prototype, which can be found in Appendices G and H. Different types of landing gears were researched as well as dampers and other spring assemblies. The team determined that a landing gear with a spring would absorb impact force from a rough landing or a low fall so as to protect the sensor module within the drone.

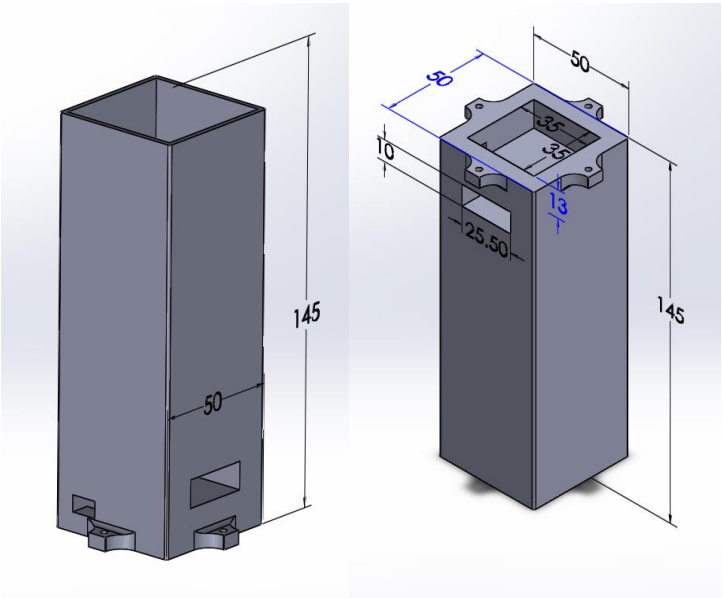


Figure 36: Battery Casing

As shown in Figure 36, the battery casing was primarily designed to maintain the two 2500 mAh batteries safely in the middle. These batteries are extremely delicate and should not be damaged during the actual flight, as these can cause fires and in extreme cases, explosions. Thus, the team decided to create a housing that will prevent the batteries from moving freely inside the drone. The

dimensions of the battery casing were chosen so both batteries could fit tightly, while having the top part clear to pull their wires down to feed power into the rest of the electronics. The casing also had other design purposes. The first one was to keep the batteries, the heaviest elements in the whole system, in the middle of the drone in order to keep the center of gravity in the middle of the horizontal axis along the base floor of the drone. The second one was to provide a place that would hold the IMU and prevent it from shifting. The IMU should be positioned as close as possible to the center of gravity for it to work accurately, so it was positioned in the bottom middle of the entire system where it would be closest to the center of gravity. Finally, the bottom part of the casing was designed to provide enough space for the camera placed at the bottom of the system. The casing would protect it from anything that could move on the inside and it would still allow its wires to come out through a small hole that was left on one of the sides.

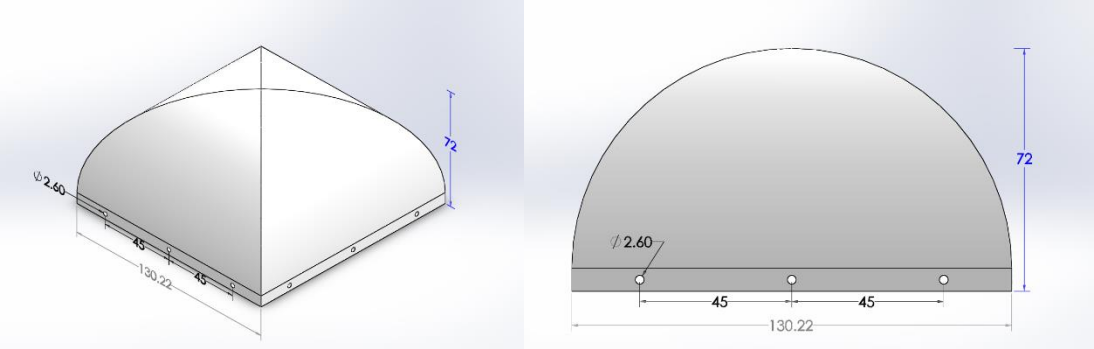


Figure 37: Hull Top

As seen in Figure 37, the hull top is rounded in order to be more aerodynamic. Since this quadrotor will not be reaching supersonic velocities, a sharp and point edge at the top is not necessary to break any shock waves. Thus, for steady, level, subsonic flight, the hull top created should be sufficient to prevent any extra drag as well as minimize turbulence.

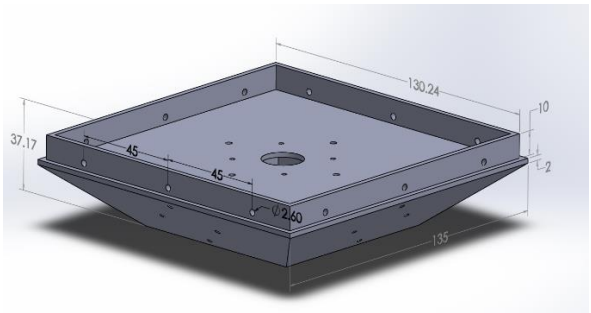


Figure 38: Hull Base

The base of the model is demonstrated in Figure 38; it was designed taking various aspects into consideration. The first one was to give the bottom camera enough clearance from the ground, while still giving it room for visibility. The second one was to create an angled surface to attach

the landing gear assemblies. This angle was expected to be smaller, but due to 3D printing restrictions and performance, it had to be changed to a 45 degree angle. With an angled bottom the landing gear would be able to compensate for non-straight landings. Finally, the last aspect of this base design was to keep the added weight to a minimum. In order to do this, the inside of the base was left empty, thus reducing both waste and material cost.

At the top of the base, a 10x2 mm lip was raised around the entire design to give space for the bolts which attach the Delrin walls to this section of the hull. Three holes were created on each side to fit the desired bolts. Finally, eight holes for the bolts and one bigger hole for the camera were designed to hold the camera and battery casing, while four more holes were added to each side of the base to support and hold the landing gear assemblies.

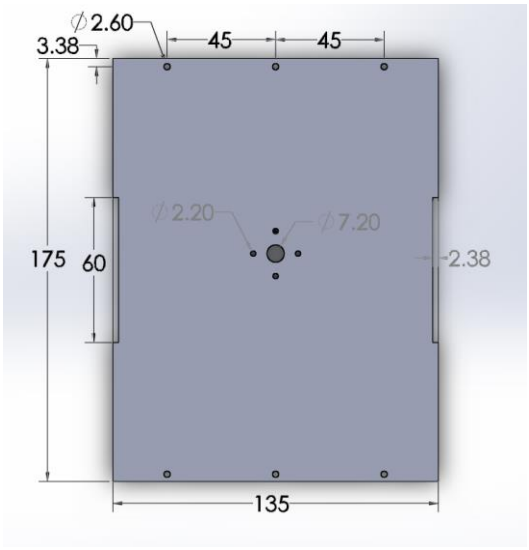


Figure 39: Middle Hull's Wall

Whereas the entirety of the previous prototype components were 3D printed, and all the other parts for this prototype were 3D printed as well, the hull walls for prototype 2 were laser cut. Since prototype 2 is a cuboid, or a rectangular cube, it did not make sense to 3D print a 2 dimensional extrusion. Furthermore, since the middle part of the hull contains the sensor module, the team wanted a stronger material to house the components. Also, as stated before, the flat surfaces of the hull wall were desired for mounting the electronics. Thus, these walls were laser cut. The material selected was Delrin, due to its high strength to weight ratio and ease of manufacturability. The hull walls were made like puzzle pieces so that they would easily fit with one another; one such wall is depicted in Figure 39. These walls were tightened using two L-brackets per each pair of walls. Additionally, holes were cut in them for mounting the hull top and bottom, as well as the motor housing arms and camera.

4.3 Final prototype

After completing and conducting several tests with prototype 2 the team made some slight changes to come up with the final prototype. These changes included changing the hull base into another laser cut piece of Delrin, adding a fillet feature to the motor rods to make them more stable, and designing a new set of legs that were going to work as the landing gear. This decisions were taken in order to increase the overall strength of the whole structure.

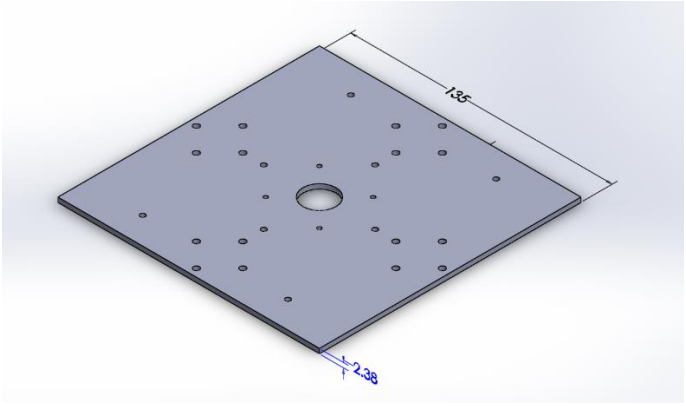


Figure 40: Final Hull Base

Shown above in Figure 40, the new base of the outer hull was designed using SolidWorks just as the rest of the walls. This time a square was created with all the holes to bolt both the camera and the battery casing to it. This base also included four extra holes to attach the L-brackets to the rest of the hull. It was chosen to have the base laser cut from the Delrin sheets in order to make it more resistant compared to the previously used one which was 3D printed with PLA. This part also included four extra holes on the corners to insert the tubes used for the thrust test.

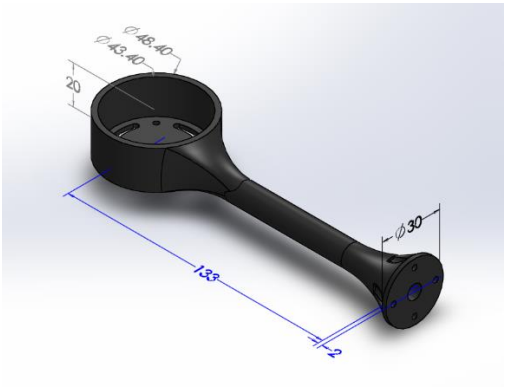


Figure 41: Final Motor Rod

The motor rods shown in Figure 41 were modified and a fillet feature was added to both ends of the tube in order to make them more resistant. The previous rods were always cracking at the same spot which was close to the end that was bolted to each hull wall. By adding the fillet, the extra material would make it harder for the rod to crack at that spot, while reducing the amount of vibrations by making the arm stiffer.

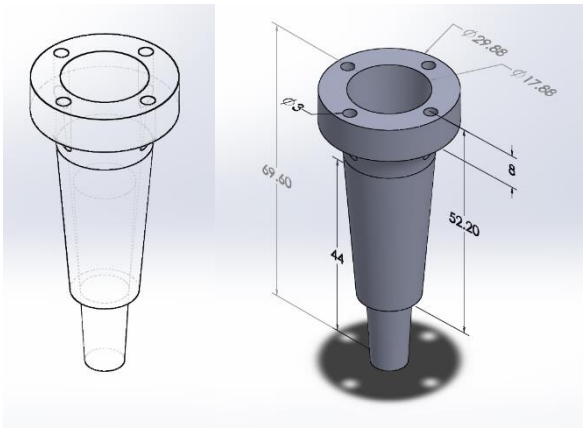


Figure 42: Final Leg Assembly

Finally, the legs used in the landing gear system were updated and changed from a cubical to a cylindrical form, as demonstrated in Figure 42 above. The previous legs were not moving up and down freely because of the method in which they were 3D printed, and they were continually breaking at the spot where they were being attached to the hull base. The fillet feature was also added to the cylindrical legs to prevent them from breaking as easily as they were.

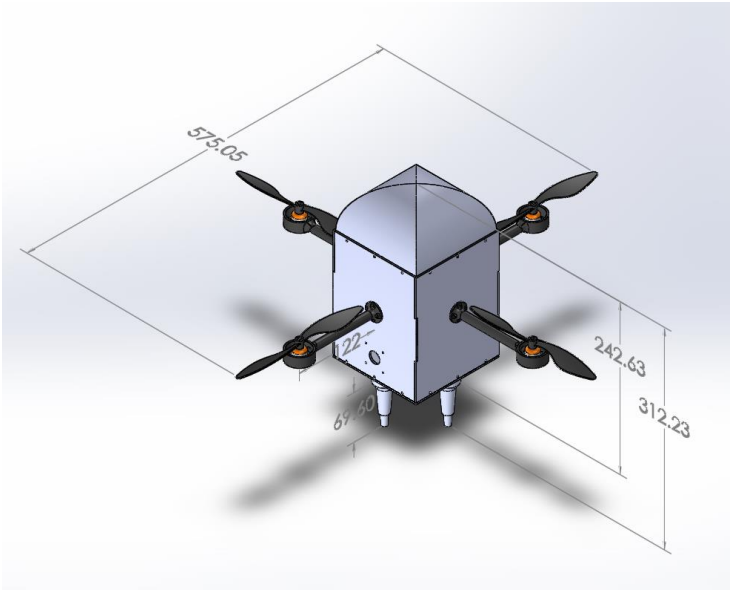


Figure 43: Final Design

The final prototype, shown in Figure 43, incorporated all these new parts and proved to be a much stronger design.

5. Final System Components Selection

5.1 Motors and Propellers

After going through a first stage of selection of components, as described in Section 2, it was necessary to change some of the parts initially selected due to the change of the overall design of the drone. With that consideration in mind, there were also some additional calculations made which are shown next. A decisive factor when choosing a motor is the total mass of the quadrotor. The mass determines the thrust needed and that can be calculated by the equation below.

$$T^3 = \frac{\pi}{2} * D^2 * p * P^2 \qquad (14)$$

In the equation above, T is thrust, D is diameter of the propeller, p is air density, and P is power of the motor. This calculation combined with the following equation was used to help determine the components for the motors and propellers.

$$Power = Kp * D^4 * P * (RPM^3) \qquad (15)$$

In this equation, Kp is the propeller constant, D is diameter, P is pitch of the propeller, and RPM is rotations per minute. The rotations per minute can be calculated by the Kv constant and voltage used on the motor. Through these equations, it can be seen that diameter is the most dominant variable followed by rpm. Another variable to be considered is the battery life which can simply be calculated by solving for the amperage in the basic power equation of power equals voltage times current. This showed that although rpm is important, a lower power motor with a larger diameter propeller would have a longer flight time. This would mean that there would be less maneuverability in flight because larger propellers are harder to change direction instantaneously. Based on the worst case scenario of the quadrotor weighing 2.5kg, calculations were done to simulate different combinations of diameter, pitch, and rpm. After understanding that the quadrotor still needs some maneuverability and would be more user friendly if the propellers were smaller, a combination of motors and propellers were decided. A combination of EMAX GT Series 1180Kv brushless motors with 10x4.7 propellers would generate 19.8 N of thrust or be able to each lift approximately 2kg at max power. This is much higher than is necessary but it is best to assume that the motors may not be as efficient as the equations show.

Figure 44: EMAX GT Series 1180Kv Outrunner Brushless Motor

Figure 45: APC 10 x 4.7 propeller

5.2 Battery

Due to the higher weight of the drone, a larger battery is necessary for a long flight time. Based on the initial research into various batteries, the ideal choice seemed to be one with a higher capacity, which would therefore increase the flight time of the quadrotor. A significant factor was balancing capacity and weight. All of the batteries were fairly similar in a weight to capacity ratio, so a 10000mAh battery was chosen as it was projected to give a flight time of about 13 minutes. Initial projections when considering smaller capacity batteries indicated that the flight time would only reach about 8 minutes, which falls short of the project goals. Due to this reason, smaller batteries were not chosen.

5.3 Motor Speed Controller

The motor speed controller, similarly to the motors had to be changed after the design changed. Because of the new requirements set by the new motors, there was a new motor speed controller that need to be acquired. For this new prototype the Hobbywing FlyFun 30A Brushless ESC was chosen. It allows up to 30 amps of current to flow. The motors only show that they can go to 26A and this will stop the ESC from overheating. Similarly to the previous one, this speed controller has safety functions like reducing power if the temperature goes above 120°C and reducing power or shutting off as a low current protection. In the same way as the previous controller, this controller will be connected to the motors on one of their ends, and to the battery and the Arduino UNO on the other. The selected controller is shown below in Figure 46.

Figure 46: Hobbywing FlyFun 30 A Brushless ESC

5.4 Landing Gear

5.4.1 Impact Force Calculations

Impact force calculations play a crucial role for landing gear design process of UAVs. Calculations use a basic energy balance, as seen below, which is commonly used in classical mechanics problems. The impact force cannot be calculated without knowing how far the object will travel after the impact. This distance depends on the toughness of the ground or impact area, the landing gear's stiffness, and air viscosity if the object is significantly light. Using the first law of thermodynamics, namely the conservation of energy, the impact velocity and kinetic energy of the quadrotor can be predicted for ideal conditions. Table2 below shows the Potential Energy (PE) and Kinetic Energy (KE) equations used for impact force calculations.

Before Impact, at a height of h	Just before the Impact, at height ~ 0
$PE = mgh$	$PE = 0$
$KE = 0$	$KE = \frac{1}{2} * m * \vartheta^2$

Table 2: Potential and Kinetic Energy Equations Used

In order to design the damper/spring for the landing gear, different scenarios were considered and a factor of safety needed to be added accordingly. In the first scenario, the kinetic energy (KE) of the object is assumed to be zero, which is a falling object situation. Equating both sides above allows to define impact velocity in terms of initial height.

$$\vartheta_{impact} = \sqrt{2 * g * h} \qquad (16)$$

Using the 5 different height scenarios with the known parameters of mass = 1500 g and gravity = 9.81 m/s^2, the calculations for impact velocities are found below:

Height (m)	Impact Velocity (m/s)
1	4.429447
3	7.672027
5	9.904544
8	12.52837
10	14.00714

Table 3: Calculated Impact Velocities Based on Initial Heights

Based on different heights of the distance traveled after impact, average impact forces (IF) can be calculated. Using the Work-Energy Principle, IF can be written in terms of distance travelled during the impact and the net work done (change in kinetic energy). This is a convenient method and a different application of conservation of energy, from which it is derived.

$$W_{net} = \frac{1}{2} * m * \vartheta_{final}^2 - \frac{1}{2} * m * \vartheta_{initial}^2 = \frac{1}{2} * m * \left[\vartheta_{final}^2 - \vartheta_{initial}^2\right] \quad (17)$$

$$F_{Avg.Impact} = \frac{W_{net}}{d_{AfterImpact}} \quad (18)$$

The table below shows the calculated results for different distances traveled after impact.

d (m)	IF (N) for h=1	IF (N) for h=3	IF (N) for h=5	IF (N) for h=8	IF (N) for h=10
0.02	735.75	2207.25	3678.75	5886	7357.5
0.05	294.3	882.9	1471.5	2354.4	2943
0.1	147.15	441.45	735.75	1177.2	1471.5
0.15	98.1	294.3	490.5	784.8	981
0.2	73.575	220.725	367.875	588.6	735.75

Table 4: Calculated Impact Forces (IF) Based on Estimated Bounce Back Distances

From the calculations above, it is found that the IF ranges between 73.58 N and 7357.5 N for a given initial velocity of 0. Next, the same calculations are run for initial velocities of 1 m/s, 2 m/s, 3 m/s and 5 m/s.

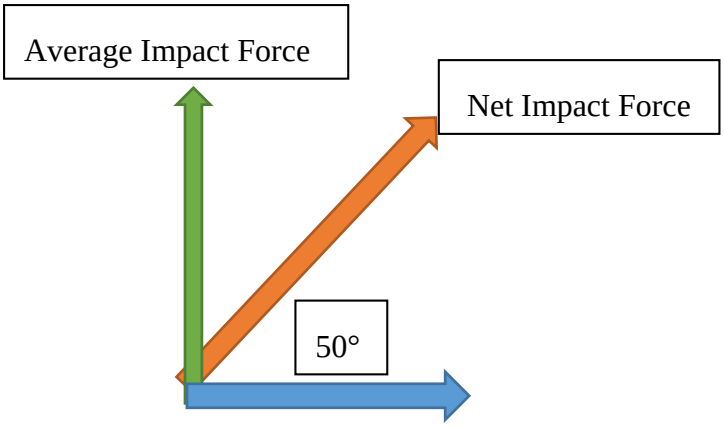


Figure 47: Vector Components on Impact Force

The landing gear designed for prototype 2 is at an angle of 50° with reference to the normal of the vehicle. The average impact forces calculated above has only partial effect on the drone. Thus, the range for net impact force (NIF) actuated on the landing gears can be seen below

d (m)	NIF (N) for h=1	NIF (N) for h=3	NIF (N) for h=5	NIF (N) for h=8	NIF (N) for h=10
0.02	472.931	1418.793	2364.655	3783.448	4729.31
0.05	189.1724	567.5172	945.862	1513.379	1891.724
0.1	94.5862	283.7586	472.931	756.6896	945.862
0.15	63.05746	189.1724	315.2873	504.4597	630.5746
0.2	47.2931	141.8793	236.4655	378.3448	472.931

Table 5: Calculated Net Impact Forces (NIF) Based on Estimated Bounce Back Distances

5.4.2 Damping Coefficient Calculation

Damper calculations require various inputs, including the correction factor of the possible damper of choice. Due to uncertainty of what size of damper is needed for the problem, the solution should be obtained via a recursive process. To solve the problem, total energy (TE), impact mass, acceleration due to gravity, height of fall, damping/stroke distance (initially assumed), driving force (NIF in this case) and an estimated correction factor are needed. Figure 48 below shows the characteristics and parameters used in the damping force calculation.

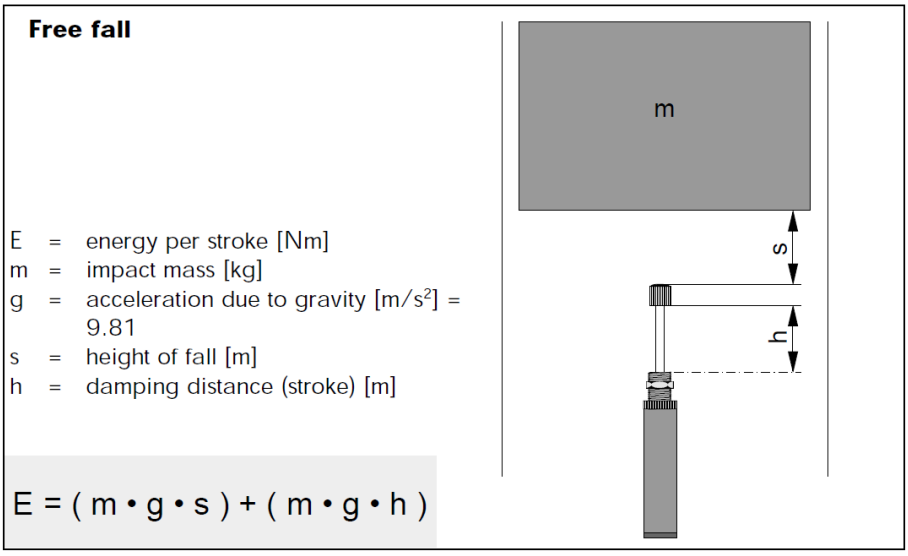


Figure 48: Damping Calculation Characteristics[24]

From the calculations made in Section 5.4.1, the total energy just before the impact is calculated, as shown below in Table 6.

Initial velocity (m/s)	TE at h=0.5(J)	TE at h=1(J)	TE at h=1.5(J)	TE at h=2(J)	TE at h=3 (J)
0	14.715	44.145	73.575	117.72	147.15
1	15.465	44.895	74.325	118.47	147.9
2	17.715	47.145	76.575	120.72	150.15
3	21.465	50.895	80.325	124.47	153.9
5	33.465	62.895	92.325	136.47	165.9

Table 6: Calculated Total Energy (TE) Based on Initial Velocities

$$F_{Damping} = \frac{E_{stroke} * correction\ factor * 1000}{stroke[mm]} \quad (19)$$

In order to calculate the damping force, energy per stroke needs to be calculated as

$$E_{stroke} = TE + F_{NetImpact} * d_{stroke} \quad (20)$$

Equation 20 is useful for 1 compact object. However, as the drone has 4 landing gears, the formula is modified to be:

[24] Damping Calculation Examples. Dictator. Dictator Co., n.d. Web. 12 Mar. 2014. <http://www.dictator.de/uploads/tx_rfdownloadsdivers/Calculation_Examples.pdf>.

$$E_{stroke} = \frac{1}{4} * \left[TE + F_{NetImpact} * d_{stroke}\right] \quad (21)$$

For the worst case scenario parameters, NIF is 4729.31 N with different stroke diameter sizes of 12 mm, 15 mm, 18 mm, 22 mm and 25 mm and max total energy of 165.9 J. the stroke energy (SE) is found below.

SE, d=15 mm (J)	SE, d=18 mm (J)	SE, d=22 mm (J)	SE, d=25 mm (J)	SE, d=28 mm (J)
59.20991	62.75689	67.4862	71.03319	74.58017

Table 7: Calculated Stroke Energies (SE) Based on Various Damper Diameters

Using the Equation 19, the damping forces (DF) are found for a given correction factor of 2:

DF, d=15 mm (N)	DF, d=18 mm (N)	DF, d=22mm (N)	DF, d=25 mm (N)	DF, d=28 mm (N)
7894.655	6972.988	6135.109	5682.655	5327.155

Table 8: Calculated Damping Forces (DF) Based on Various Damper Diameters

The current design has space for a maximum diameter of 15.5 mm for a potential damper, and based on the calculations, it was planned to change the landing gear sub-assembly in order to create space for up to 28 mm. Due to physical constraints in the design, it is not possible to have a space for more than the calculated diameters. Because the required damping force is considerably above the industry standards, which is an average of 2500N for a 30 mm diameter of damper, it was decided to use a spring system instead.

5.4.3 Spring Coefficient Calculations

Hooke's Law is used to calculate the desired type of spring.

$$F = k * X = k * \left(L_{free} - L_{deformed}\right) \quad (22)$$

The particular type of spring appropriate for free fall and shock absorbency would be the compression spring. The maximum net impact force on the object is calculated to be 4729.31 N, making it 1182.33 N on each spring, while the minimum net impact force required is found to be 354.7 N for each spring (based on a 10 ft. drop). The current design allows up to 20 mm of free length for the springs with changeable solid heights and a maximum of 15.5 mm diameter in the landing gear sub-assembly. With a rough assumption of 10 mm for solid height after compression for the given net impact force requirements, the spring constant is found as:

$$1182.33\ N = k * (20\ [mm] - 10\ [mm]) \rightarrow k_{max} = 118.2\ N/mm\ \ [32.8\ ft\ drop]$$

$$354.7\ N = k * (20\ [mm] - 10\ [mm]) \rightarrow k_{min} = 35.4\ N/mm [10\ ft\ drop]$$

Based on the calculated ranges for the stiffness constant *k*, net impact forces, spring diameters and free lengths, the appropriate springs were compared and the team concluded that the Gardner Spring with Part Number MC078-0205 is the best option. The spring suggestions can all be found in Appendix G.

5.4.4 Change in Springs

After receiving the selected springs from the dealer, the team found that the chosen springs were significantly stiff and impractical to use for the prototype. In theory they can resist and absorb up to the maximum calculated force, however initial testing results show that springs will act as a rigid object compared to the flexible and light PLA material used for landing gears. Thus a possible crash will likely cause the springs to break the main body rather than absorb the impact. Therefore, the team decided to use weaker springs with lower stiffness constant. The aim was to alleviate the initial impact force rather than completely absorbing it. The legs were also supported with SUGRU, a flexible silicone rubber that will absorb the force partially and create an additional damping effect. The 17 feet drop test showed that landing gear subassembly could absorb enough initial shock force to protect the Delrin middle hull.

6. System Tests

6.1 Drop Tests

In order to demonstrate the strength of the structure of the UAV as well as determine any potential weaknesses in the second prototype, a drop test was conducted. According to the project specifications, the quadrotor must be able to survive a 10 foot drop. The drop test was first conducted at 4 feet to ensure that the procedure was conducted properly. Assuming all would go well with the preliminary test, the UAV would be dropped from 10 feet to ensure it passes this required specification.

6.1.1 Initial Drop Test

The first drop test was conducted from a height of 4 feet indoors. This drop test was inclined more towards testing structural integrity, so all internal electronics except for the battery were replaced with weights so that they were not damaged. There were some considerations made in terms of the parts that would receive more damage. The landing legs and the base of the structure should be the parts that get affected the most because those parts have direct contact with the ground. Detailed steps for the preliminary test are seen below.

1. The prototype was assembled: Attach laser cut Delrin sides to each other with bolts, nuts, and l-brackets. Attach legs to the base with screws. Attach arms to laser cut walls with bolts and nuts. Finally, attach the base to the Delrin sides with bolts.

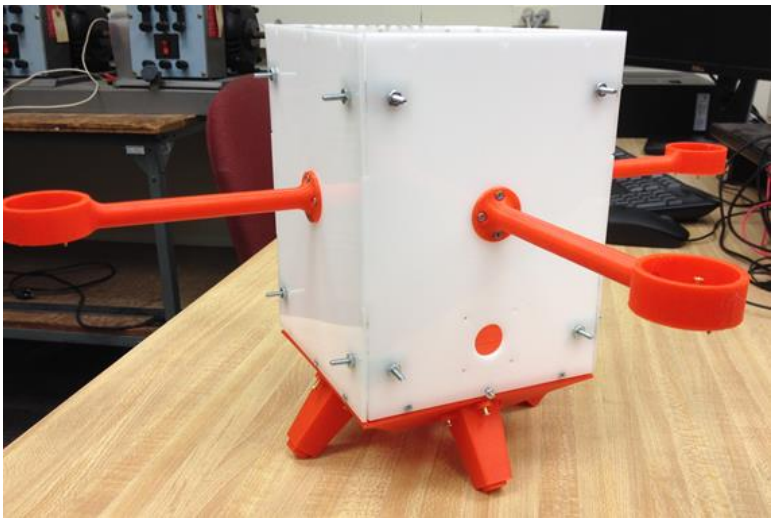


Figure 49: Assembled Prototype

2. Place equivalent weight remaining (of 2 kg required) carefully inside the structure.

Figure 50: Additional Weight Added

3. Set up various cameras to capture a video of drop test and test cameras for functionality.
4. Raise the assembly UAV structure to a height of approximately 4 feet.

Figure 51: Measured Height (4 feet)

5. Drop the quadrotor and record footage of the falling structure.
6. Analyze the quadrotor and videos of the drop test.
7. Refine test for next drop test.

6.1.2 Initial Drop Test Analysis

The initial drop test resulted more favorably than initially hypothesized. The team had expected all four legs to break off of the hull base. However, the base sustained minimal damage that was easily repairable with a hot glue gun. One leg was cleanly cut off from the hull base, which was simple to fix as well. The one surprising fracture was that one of the motor housing arms snapped off without any contact. However, this as well was repaired so that the second drop test could be completed for further analysis. Finally, the last surprising result of this test was the strength of the Delrin walls of the hull. The team anticipated that cracks would propagate up the side walls. Due to the material thickness of 3/32 inches and the strong bolts, nuts, and l-brackets holding everything together, the only sustained fractures were to a few 3D printed parts.

6.1.3 Final Drop Test

As previously mentioned, the initial drop test determined to be a passed test because only minor damages were observed. The legs did a much better job at distributing the impact shock throughout the entire hull than the team had expected. With that in mind, the second drop test was performed. In light of the initial drop test, the team expected the extra 7 feet to create more potential energy which would then convert to kinetic energy which would finally be turned into an impact force. This means that there was a much higher probability of something more important breaking. Thus, the team expected all of the legs to sustain damage as well as the hull base. It was also probable that the laser cut material would sustain fractures from the fall. Considering how the arm broke the first time, the team expected more arms to break this time. Detailed steps are primarily unchanged, as listed again below.

1. Assemble prototype.
2. Emplace equivalent weight remaining (of 2 kg required).
3. Set up cameras to capture footage.
4. Raise assembly to a height of approximately 10 feet. The height was modified to 17 feet for this test due to finding a suitable location.

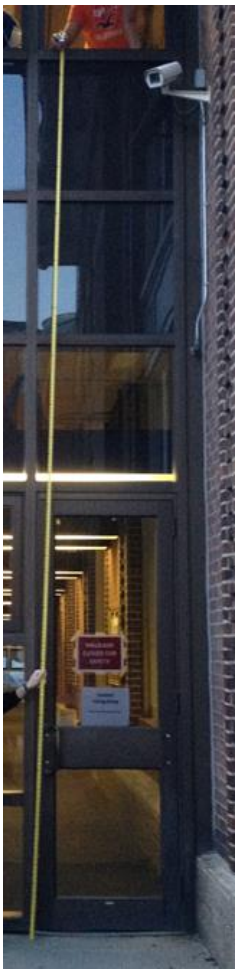


Figure 52: Final Drop Test Location (17 feet)

5. Drop the quadrotor and record footage of the falling structure.
6. Analyze the quadrotor and videos of the drop test.

6.1.4 Final Drop Test Analysis

The final drop test essentially went as expected. The addition of 14 feet gave the quadrotor much more potential energy that caused more parts to break upon impact. Three landing legs snapped squarely off of the structure from the base due to the material thickness and angle. Three motor housing arms also snapped squarely off of the structure and did not sustain any damage other than that. The hull base completely broke off from the structure and was critically damaged. Surprisingly, the Delrin held together fairly well, only sustaining a few, small cracks. However, it was enough to need to laser cut new material for the next assembly. Therefore, the final drop test demonstrated that the quadrotor is able to sustain falls from minimal heights, but will not succeed at surviving a fall of above 10 feet. Potential reasons for this include that the structure rotated as it fell and thus the landing legs were unable to distribute the shock evenly to the structure. Also, the uneven landing created sharp angles that caused the arms and legs to snap at their vital connections.

Thus, if the quadrotor was actually flying it can be assumed that it would be able to survive a minimal fall, but unlikely that it would survive a higher distance fall.

6.2 Flight Tests

6.2.1 Introduction

Flight tests were conducted to ensure that the UAV would be able to autonomously fly without any problems. Thrust tests were conducted by individually testing the motors on a test stand. The thrust tests simply demonstrated that the motors will rotate at the given input velocity without issues such as overheating. The stability test used two motors mounted on the UAV to test the PD control of the roll and pitch axes. A PD control was chosen because it is a simple, linear controller that would be efficient for the quadrotor model. There is no integral component like the traditional PID control technique because a steady state error does not occur in the system and the integral component would only make the system more complex. The stability test was conducted by pinning two of the drone's arms without motors to a metal structure that allowed for rotation along that axis. The other two arms had motors with PD control and the drone went to equilibrium after adjusting the K values. A perturbation was then introduced to the system by pushing one side down and the test determined that the system was stable.

6.2.2 Flight Tests and Analysis

Flight Test #1

This flight test was done in the lab with 2 students each holding wires from the UAV while it was attempting to stabilize. Two other students stood off to the side to record footage of the test. The thrust was gradually increased until the quadrotor briefly lifted off of the ground. However the drone immediately became unstable and fell sideways to the ground. No damage was sustained during this flight test.

Flight Test #2

In this flight test there were no wires or cables involved. The drone was free and located in the middle of a classroom with a high ceiling. Again the students stood far away to take videos of the test. The thrust was gradually increased and vibrations were seen in the motor arms with increasing frequency. Instead of lifting off the ground however, the drone fell over. This was due to the team using only 50% power for the thrust values. However, the falling signified that the control values needed to be adjusted. So in the next flight tests the drone was hung from the ceiling to test the stability values.

Flight Test #3

Flight Test #3 Procedures and Hypothesis

This test attempted to demonstrate whether or not the drone is capable of stable flight. The hypothesis for this test is that the drone will successfully be able to fly a few feet high and then descend with no issues. The motor and propeller physics predicts that the thrust produced should be more than enough to successfully lift the device to a satisfactory altitude, hover, and then

land. The landing legs with springs should prevent any damage to the hull or sensor module. Furthermore, the cameras and other sensors should be able to produce live data feed to the user ground station. Procedures are seen below for the test.

1. Using rubber cord, carefully attached the UAV to the ceiling by making knots on the UAV and the ceiling with the rubber cord. Leave enough slack so that the drone is slightly above the floor, but ensure that the slack created from flying will enter the hull so as not to get tangled with the propellers.
2. Using the remote controller, increase the propeller thrust until the drone starts to lift off the ground.
3. Continue increasing the thrust so that the drone starts accelerating in the upward vertical condition.
4. Within 1 to 2 feet of the ceiling, reduce thrust to equal the UAV's weight so that it hovers.
5. Carefully reduce thrust so that the quadrotor begins its descent.
6. Continue slowly decreasing thrust and then when close to the floor either slightly increase or maintain current thrust for a safe landing.
7. Inspect the drone and record observations and results.

Figure 53: Flight Test #3 set up

Flight Test #3 Analysis

An elastic rubber cord was used to hang the drone from the ceiling for this flight test. A wire harness was tied through the holes at the top of the drone and the rubber cord was tied to the harness and the ceiling. The drone was left to hang freely and was adjusted to make it level. After ensuring a safe environment, the thrust values were gradually increased. However, after a little increase in thrust the quadrotor became unstable and started to overshoot about several of its axes. The team suspected that this in part was due to the elasticity of the cord used, so another flight test was conducted to take out that factor.

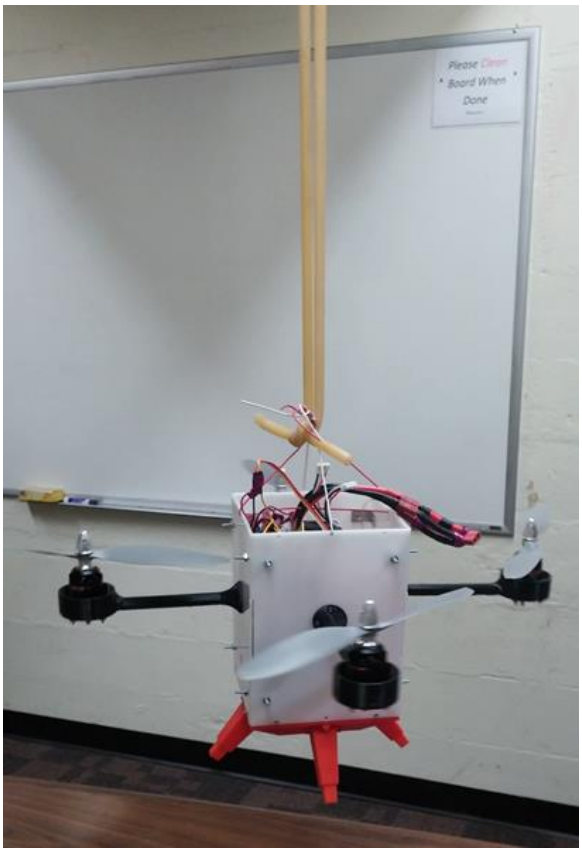


Figure 54: Flight Test #3 close up

Flight Test #4

For flight test number 4, wire was used in place of the elastic rubber cord. This took the possible problem with elasticity out of the test. The test stand was the same as flight test #3. The drone hung freely from the ceiling by the wire and the test was conducted. However, the same results as flight test #3 occurred but without as much rapid movement. The thrust was increased, and after reaching a certain velocity, the quadrotor became unstable and overshot about multiple axes. This demonstrated that even without elasticity as a factor the control values still needed to be updated.

Flight Test #5

Flight test 5 was performed in the same manner as flight test 4. However, upon increasing the thrust, the wire snapped and the drone flew off and crashed into a nearby wall. The arms broke off of the drone as well as the legs and the base. The Delrin surprisingly survived but two motor controllers did not survive. Thus, this test caused the team to need to repair and reorder several parts and push off the next flight tests for a while. It was considered to attempt the future flight tests in a more stable environment to prohibit any more possible delays.

Flight Test #6

Due to the issues created by the previous flight test, several components needed to be replaced. New arms were 3D printed, a new base was laser cut out of Delrin, and new legs were 3D printed. It was determined that the test stands were not adequately isolating the issue of where the instability was coming from. Thus, the team created a test stand that enabled the quadrotor to rotate on only the pitch and roll axes. The stand holds the drone from its base, so it has to be held upright until its thrust is strong enough to carry its own weight. The test stand was created out of Vex metal. It was expected that the test would isolate the axis that was causing the instability. However, after increasing the thrust the drone had similar results to the previous tests, with the quadrotor becoming unstable in both axes and flailing around. Thus, it was decided that testing about a single axis was needed again to determine the correct stability constants.

Flight Test #7

The test stand was modified so that the drone could only rotate about one axis. It was tested several times with different iterations of control values. Some caused the drone to not stabilize itself at all, while others overcorrected too much. One set of values was good until a certain velocity where it overcorrected way too much. Thus, it was decided that if the hovering velocity was determined, an appropriate range of velocities could be determined that would assist the setting of the control values. This would ensure that the controls would only begin to become unstable as the velocity approached maximum and minimum speeds. A new test stand was needed to test hovering and to determine the takeoff and hovering velocities.

Figure 55: Flight Test #7

Flight Test #8

In order to determine an approximate hovering velocity, a vertical test stand was created. This test stand restricted the drone to one degree of freedom, vertical translation. The stand was built out of long poles and a wooden block secured to the top of the poles to prevent the drone from flying off of the mechanism. Four holes were drilled into the base of the UAV and the wooden block to place the poles through. The poles were hammered deep into the ground and then the quadrotor was placed on the poles. Finally the wooden block was placed on top with extra supports in case the drone was able to knock the wood off. The complete test stand is seen below. It was expected that a basic stability would be achieved and the drone would be able to hover by varying the thrust with an RC controller. When the drone's thrust was being slowly increased, the drone did not rise off of the ground. However after the thrust on the controller was placed back to zero, the drone flew up and violently hit the wooden support, breaking a few propellers and arms. It was determined that there was either a delay in the data transmission or an error with the code.

The code was updated and tested to account for any potential errors that would cause the problem above and the broken parts were replaced. The max thrust during this new test was set to half of the possible max thrust to help ensure safety. After attempting the test again, the same issue occurred. The drone took off after the thrust value had returned to zero and the drone hit the wooden support, again breaking a few of the UAV arms and propellers. This proved that even with the updated code the problem was still present. After this problematic test, other hovering flight tests procedures were considered, but this was the last flight test due to time constraints.

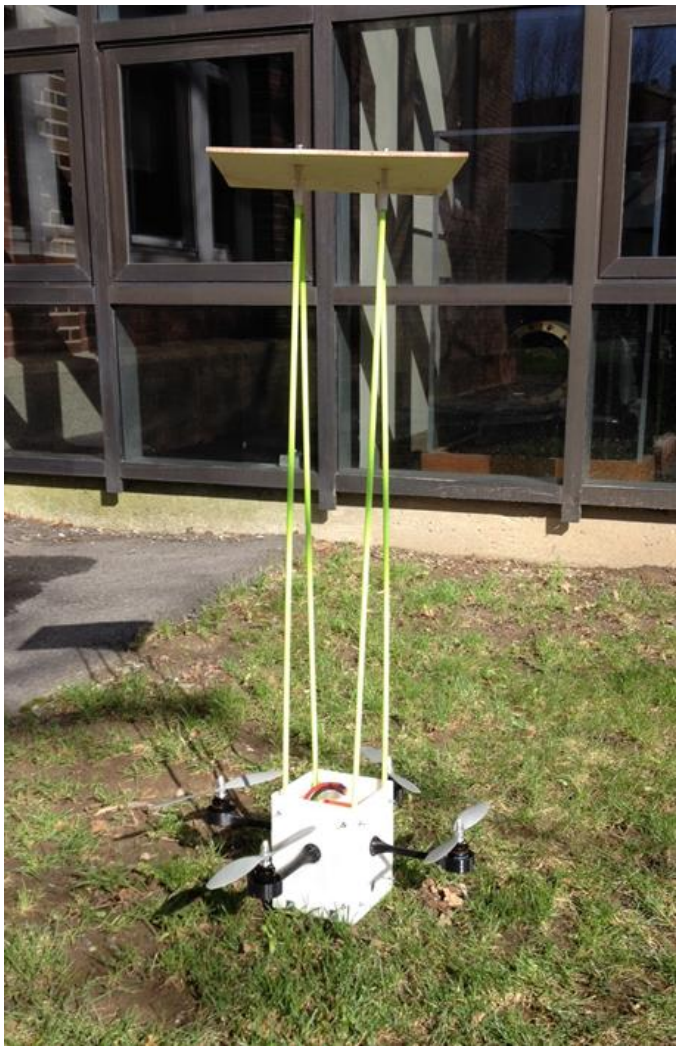


Figure 56: Flight Test #8

6.2.3 Flight Test Conclusions

The flight and stability tests were done to determine if the quadrotor would be able to successfully fly. The drone clearly had the potential to fly to one hundred feet, as seen by the impressive thrust power demonstrated in flight test 8, however the drone was never able to achieve stable flight. This in part was due to coding and erroneous control values, but also due to a misunderstanding of the controls model. The simulation for the quadrotor was successful because it took into account a perfect model with bidirectional motors. When changing the motors in the simulation to no longer be bidirectional, the simulation was unstable like the real quadrotor. The realization that the model was no longer linear occurred during flight test #7 and that is why the model was attempted to be linearized along the hovering velocity in flight test #8. There still may be possible coding errors as shown by flight test #8, but after linearizing the model, stability may be possible.

6.3 Flight Time Test

A flight time test was conducted to ensure that the quadrotor would be able to fly for more than 10 minutes. A single 2500mah battery and a single motor and propeller were used to determine how long the battery could last at approximately hovering velocity. From flight test #8, it can be seen

that the quadrotor can gain altitude with less than half of the max thrust. The single motor was sent to rotate at a continuous rate of half the maximum velocity until the battery was seen to drop less than 10.3 volts. The battery was charged completely before administering the test to be 12.6 volts. An online stopwatch application was used to record the time before a battery alarm went off to ensure that 10.3 volts was present. At approximately 23.5 minutes, the alarm went off. With only a single motor moving and a single 2500mah battery, a 5000mah battery would last only about 11.1 minutes. The chosen 10000mah battery would last the full 23.5 minutes. This test only estimated the quadrotor flight time at hovering velocity rather than moving around. It is assumed that increasing the velocities of the motors causes an exponential decrease in flight time because of the nonlinear relationship between velocity and force, so a 5000mah battery may fall short of the desired 10 minute flight time.

7. Results, Conclusions, and Recommendations

7.1 Results

This project lasted the entire school year and involved a lot of research, testing, and analysis. At the beginning of the project, the team created initial project specifications as detailed in Section 2.1 that served as guidelines for the MQP. A review was done to see what specifications the project fulfilled and which ones it did not meet.

1. The drone was not able to reach an altitude of 100 feet. Although several flight tests were done, the drone was not stable enough to safely attempt reaching this altitude. This was also not fulfilled due to time constraints.
2. The drone was able to meet the weight requirements. The total drone was approximately 1.8 kg and the power from the motors offered enough thrust for a 0.5 kg payload.
3. The drone met the requirement to have four propellers and four motors.
4. The drone met the requirement to have a landing device or landing legs.
5. The 10 foot drop test was conditionally met. The landing gear and rods did not survive the fall, but the sensor module was protected and remained undamaged.
6. The requirement to have at least one camera for image processing was met, even though network programming wasn't completed for image streaming. Images could be processed, but programming changes will need to be done in the future for a better performance.
7. The drone conditionally met the requirement to have a battery life of at least 15 minutes. The flight time test demonstrated that a 10000mah battery could last for around 23 minutes at hovering velocity, but it is unknown how a variable velocity flight time test would behave.
8. The drone conditionally met the requirement to have the ability to sense its location and altitude. On board there was a GPS and an IMU to measure position and attitude. The IMU had very accurate precision of roll and pitch angles, but the yaw angle had an occasional error of around 5 degrees. The GPS is only reliable in open environments because of a large variance. An ultrasonic sensor was bought but not implemented to be used for takeoff and landing.
9. The drone conditionally met the requirement to transmit data wirelessly. As of the completion of the project, the software allowed for limited transmission.
10. The drone met the requirement to not exceed $1000 in cost. The total cost of the drone was approximately $490.
11. The drone met the requirement to be easy to carry and deploy, friendly, and intuitive. It was less than 4 cubic feet and was put together with well-placed screws and nuts.
12. The drone did not meet the waterproof requirement. Although the body of the drone intended to account for waterproofing, enough research wasn't completed to fully apply waterproofing.

Figure 57 summarizes the key specifications that this drone met or failed to meet.

Specification	Goals	Met?	Comments
Altitude	100 ft.	Not Met	Not stable enough to attempt reaching max. height
Weight	2 kg	Met	Entire drone approximately 1.8 kg
Payload	0.5 kg	Met	Max. thrust capacity of 4 kg allows for 0.5 kg payload
Size	< 4 ft^3	Met	Total size approximately 3.52 ft^3 (0.1 m^3)
Cost	< $1000	Met	Total cost approximately $490
Imaging	2 Streamed Cameras	Conditionally	Networking problems with ground station
Weather Protection	Waterproof	Not Met	Exposed openings throughout entire drone
Drop Test	10 ft.	Conditionally	Undamaged sensor module; damaged landing gear and rods
Landing Device	Impact Reducing	Conditionally	Impact reduction partially accomplished by 4 spring legs
Communications	Real Time Control	Not Met	Networking problems with ground station
Position and Orientation	GPS and IMU	Met	Fully operational location sensing and attitude control

Figure 57: Project Specification Results Summary

7.2 Conclusions

The final results showed that not all of the specifications and goals set at the beginning of the project were met. Although fully autonomous flight was not achieved, the team made significant development towards creating a low-cost, lightweight unmanned aerial vehicle capable of surveillance with its implementation of a camera system. Several design iterations were completed to create a quadrotor best suited for the project goals and the final prototype demonstrated potential for success with the project goals. As described earlier in Section 6, there were several tests performed to ensure the designed prototype would successfully fly. If more time was available, the team would have been able to design a more robust controls system that would ensure stable flight. Furthermore, the team's work on a user-friendly and intuitive GUI would promote the surveillance aspect of the quadrotor.

Overall, an unmanned aerial vehicle was fully realized in this project. The quadrotor demonstrated it had the power to fly to a high altitude, but lacked the stability. Further iterations of this project would ensure its future success.

7.3 Recommendations

Several lessons were learned during the course of this project and thus the team has several recommendations. As always, a lighter vehicle is desired to maximize power efficiency that at the same time is capable of carrying a heavier payload, and also allow for a longer battery lifetime. Another consideration would be to use variable pitch propellers which offer easier control because of their linear relationship to force. From a controls aspect, more sensors would be helpful for more precision or capabilities. A lidar sensor or other depth sensor could be implemented to apply obstacle avoidance and path planning. Encoders for the motors are also possible to be used to help ensure that motor velocities are acting appropriately. With the encoders, the thrust could be controlled more effectively. Although more sensors are useful, it would also need a larger processor and could increase price. Improving the vision system would definitely help the operator utilize the drone for surveillance applications. A few more recommendations would be to make the drone weatherproof as well as developing the code to make potential for swarm technology.

The router chosen allows for multiple drones, but this wasn’t implemented. After performing different tests, it can be mentioned that having a more effective testing configuration and environment that don't necessarily strain the drone on a single axis by increasing the weight on some arms but not others will provide better results. This can be partially accomplished by using more reliable materials to reduce frictional forces on the test stand. Taking these recommendations into account will facilitate the realization of a fully functional and autonomous UAV capable of surveillance.

Appendices

Appendix A: Authorship

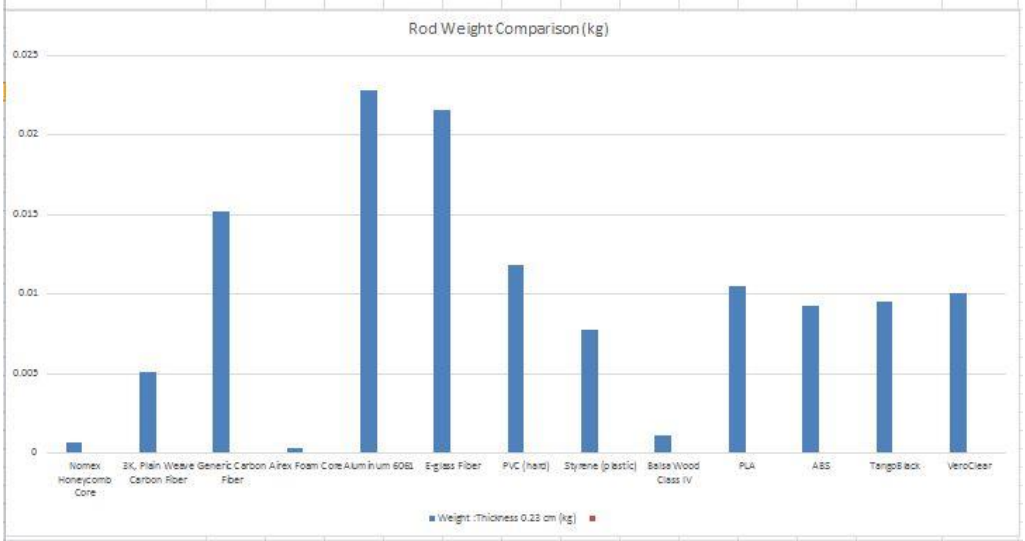
Content	Author
A term	
Background: History of UAVs	Steven Guayaquil, Arianna Niro
Background: Basic Flight Physics	Andrew Gallagher, Antonio Puzzi
Background: Physics of Parachutes	Andrew Gallagher, Antonio Puzzi
Background: Physical Limitations	Steven Guayaquil
Background: System Power	Steven Guayaquil
Background: System Sensors	Ben McIntyre
Background: Materials	Arman Uygur
Background: Motors	Antonio Puzzi
Background: Propellers	Antonio Puzzi
Background: Batteries	Arianna Niro
Background: Sensors	Ben McIntyre
Background: Microcontrollers	Ben McIntyre
Background: Future of UAVs	Arman Uygur
Proofreading and Formatting	Arianna Niro
B term	
Part Selection: Materials	Arman Uygur
Part Selection: Router	Steven Guayaquil
Part Selection: Motors and Propellers	Antonio Puzzi
Part Selection: Battery	Ben McIntyre, Arianna Niro
Part Selection: Motor Speed Controller	Antonio Puzzi
Part Selection: DC to DC Converter	Antonio Puzzi
Part Selection: Previous MQP Components	Ben McIntyre, Arianna Niro
Design: Introduction	Andrew Gallagher
Design: Initial Concept Designs	Andrew Gallagher
Design: Project Specifications	Andrew Gallagher, Arianna Niro
Design: Prototype 1 Design	Andrew Gallagher

Design: CAD Modeling and 3D Printing	Andrew Gallagher
Design: GUI Design	Steven Guayaquil
Design: Controls and Simulation	Ben McIntyre
Prototype 1: Lessons Learned	Andrew Gallagher
Appendices B-F	Arman Uygur
Proofreading and Formatting	Arianna Niro
C term	
Prototypes	Andrew Gallagher, Steven Guayaquil, Arianna Niro, Antonio Puzzi
Final Systems Components Selection	Ben McIntyre
System Tests	Andrew Gallagher, Antonio Puzzi
Future Work and Alternative Recommendations	Arman Uygur
Appendices G-H	Arman Uygur
Proofreading and Formatting	Arianna Niro
D term	
Flight Tests	Andrew Gallagher
Final Prototype	Antonio Puzzi
Landing Gear	Arman Uygur
Appendices I-J	Arman Uygur
Appendices K-M	Ben McIntyre
Results, Conclusions, and Recommendations	Andrew Gallagher, Steven Guayaquil, and Ben McIntyre
Section Updates from A-C Term	Arianna Niro
Proofreading and Formatting	Andrew Gallagher, Steven Guayaquil, and Arianna Niro
Final Overview	All

Appendix B: Rod Calculations and Decision Matrices

Rod Weight Calculations for Various Materials

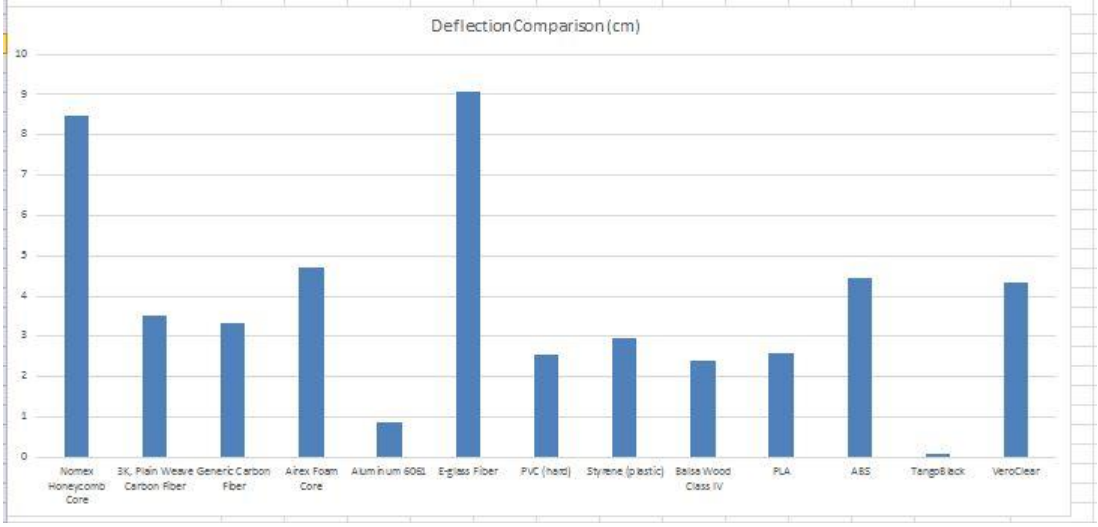
	Nomex Honeycomb Core	3K, Plain Weave Carbon	Generic Carbon Fiber	Airex Foam Core	Aluminum 6061	E-glass Fiber	PVC (hard)	Styrene (plastic)	Balsa Wood Class IV	PLA	ABS	TangoBlack	VeroClear
Density (g/cm^3)	0.0769	0.6	1.7992	0.04	2.7	2.5466	1.4	0.914	0.13	1.24	1.1	1.13	1.19
Weight : Thickness 0.23 cm (kg	0.0006	0.0051	0.0152	0.0003	0.0228	0.0215	0.0118	0.0077	0.0011	0.0105	0.0093	0.0095482	0.010055



rod-radius(cm)	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7
rod-length (cm)	10	10	10	10	10	10	10	10	10	10	10	10	10
rod_inner_radius (cm)	0.47	0.47	0.47	0.47	0.47	0.47	0.47	0.47	0.47	0.47	0.47	0.47	0.47

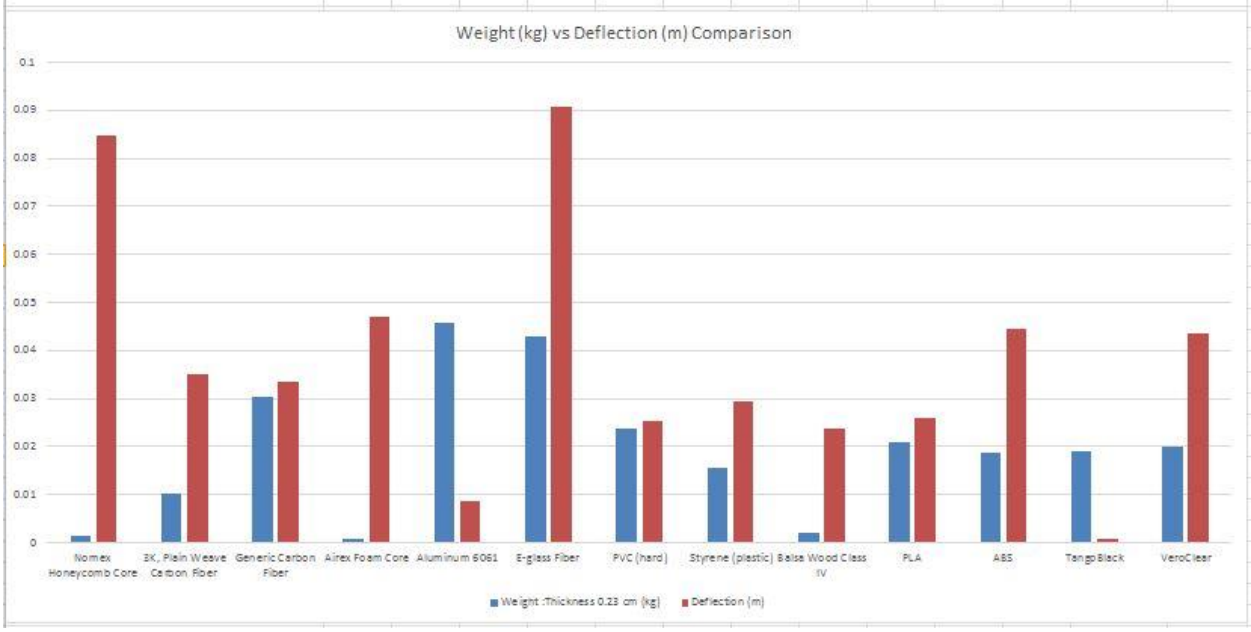
Rod Deflection Calculations for Various Materials

	Nomex Honeycomb Core	3K, Plain Weave Carbon	Generic Carbon Fiber	Airex Foam Core	Aluminum 6061	E-glass Fiber	PVC (hard)	Styrene (plastic)	Balsa Wood Class IV	PLA	ABS	TangoBlack	VeroClear
Young's/Tensile Modulus E (Gp	0.0112	227.53	227.53	0.028	68.9	72.395	3	2.9	6	3.5439	1.5	2.5	2.5
Tensile Strength (Gpa)	0.0005	4.2058	3.999	0.0007	0.31	3.4474	0.04	0.045	0.075	0.0484	0.035	0.0012	0.057
Bending Stress (Gpa) at FS=2	0.0002	2.1029	1.9995	0.0003	0.155	1.7237	0.02	0.0225	0.0375	0.0242	0.0175	0.0006	0.0285
Material Specific - Gamma	0.0149	0.0062	0.0059	0.0082	0.0015	0.0159	0.0044	0.0052	0.0042	0.0046	0.0078	0.00016	0.0076
rod-radius(cm)	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7
rod-length (cm)	20	20	20	20	20	20	20	20	20	20	20	20	20
Deflection (cm)	8.4877	3.5209	3.3478	4.6915	0.857	9.0703	2.5397	2.9557	2.381	2.6014	4.4444	0.0914286	4.342857



Rod Deflection versus Weight Comparison for Various Materials

	Nomex Honeycomb Core	3K, Plain Weave Carbon	Generic Carbon Fiber	Airex Foam Core	Aluminum 6061	E-glass Fiber	PVC (hard)	Styrene (plastic)	Balsa Wood Class IV	PLA	ABS	TangoBlack	VeroClear
Weight : Thickness 0.23 cm (kg)	0.0013	0.0101	0.0304	0.0007	0.0456	0.043	0.0237	0.0154	0.0022	0.021	0.0186	0.0190964	0.02011
Deflection (m)	0.0849	0.0352	0.0335	0.0469	0.0086	0.0907	0.0254	0.0296	0.0238	0.026	0.0444	0.0009143	0.043429



Rod Thickness Calculations – Stress Analysis

Thickness Calculation - Stress Analysis													
	Nomex Honeycomb Core	3K, Plain Weave Carbon	Generic Carbon Fiber	Airex Foam Core	Aluminum 6061	E-glass Fiber	PVC (hard)	Styrene (plastic)	Balsa Wood Class IV	PLA	ABS	TangoBlack	VeroClear
Bending Stress (Pa) at FS=2	249935	2.103E+09	2E+09	344738	1.6E+08	1.7E+09	2E+07	22500000	37500000	2E+07	1.8E+07	600000	28500000
Lenghth (m)	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
r_outer (m)	0.007	0.007	0.007	0.007	0.007	0.007	0.007	0.007	0.007	0.007	0.007	0.007	0.007
F_maxThrust (N) -Motor	25.8	25.8	25.8	25.8	25.8	25.8	25.8	25.8	25.8	25.8	25.8	25.8	25.8
Term A	2.4E-09	2.401E-09	2.4E-09	2.4E-09	2.4E-09	2.4E-09	2.4E-09	2.4E-09	2.4E-09	2E-09	2.4E-09	2.4E-09	2.4E-09
Term B	1.84E-07	2.187E-11	2.3E-11	1.3E-07	3E-10	2.7E-11	2.3E-09	2.04E-09	1.23E-09	2E-09	2.6E-09	7.66E-08	1.61E-09
r_inner (cm)	#NUM!	0.6984005	0.69832	#NUM!	0.67729	0.69805	0.31743	0.434685	0.585429	0.473	#NUM!	#NUM!	0.529712
r_outer(cm)	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7
thickness (cm)	#NUM!	0.0015995	0.00168	#NUM!	0.02271	0.00195	0.38257	0.265315	0.114571	0.227	#NUM!	#NUM!	0.170288

Rod Decision Matrix

Decision Matrix -Rods							
Material Name	**Stress Analysis**	**Deflection**	**Weight**	**Impact Strength***	**Cost**	**Water Resistivity**	**Total**
Nomex Honeycomb Core	Fail						
3K, Plain Weave Carbon Fiber	Pass	3	2	2	5	2	2.85
Generic Carbon Fiber	Pass	3	4	2	4	2	3.2
Airex Foam Core	Fail						
Aluminum 6061	Pass	1	5	3	3	3	2.7
E-glass Fiber	Pass	5	5	3	4	2	4.35
PVC (hard)	Pass	2	4	4	2	2	2.7
Styrene (plastic)	Pass	2	3	4	2	2	2.45
Balsa Wood Class IV	Pass	2	2	5	1	5	2.45
PLA	Pass	2	3	3	2	2	2.35
ABS	Fail						
TangoBlack	Fail						
VeroClear	Pass	3	3	4	2	2	2.85
*This property is the resistivity of material against a collision							

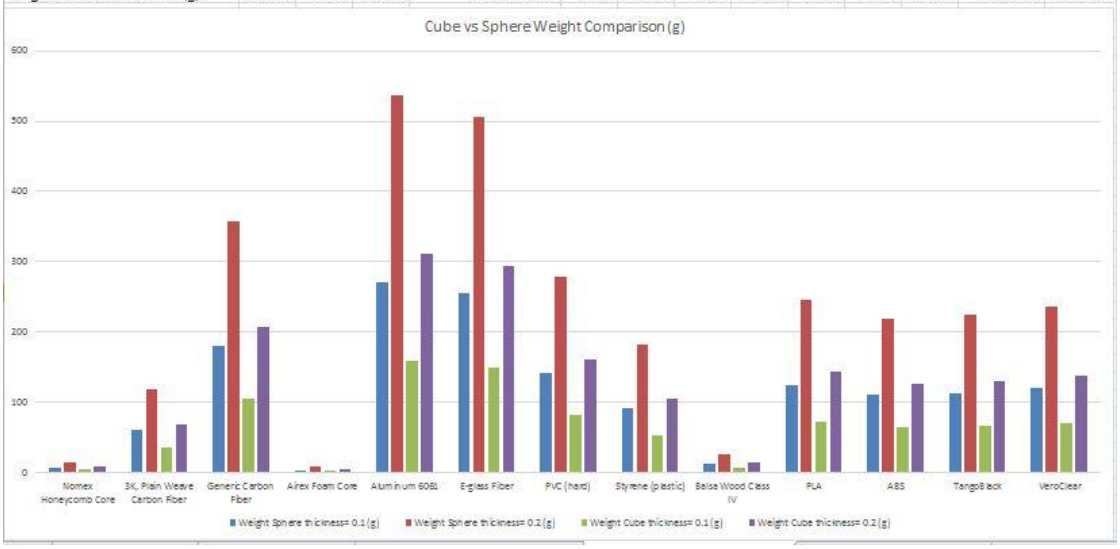
Rod Weight Distribution of Factors for Decision Matrix

Weight Distribution					
Deflection	**Weight**	**Impact Strength***	**Cost**	**Water**	**Total**
0.4	0.25	0.1	0.15	0.1	1

Appendix C: Hull Calculations and Decision Matrices

Hull Cube versus Sphere Design Comparison for Various Materials

	Nomex Honeycomb Core	3K, Plain Weave Carbon	Generic Carbon Fiber	Airex Foam Core	Aluminum 6061	E-glass Fiber	PVC (hard)	Styrene (plastic)	Balsa Wood Class IV	PLA	ABS	TangoBlack	VeroClear
Density (g/cm^3)	0.0769	0.6	1.7992	0.040046158	2.7	2.5466	1.4	0.914	0.13	1.24	1.1	1.13	1.19
Weight Sphere thickness= 0.1(g)	7.7357	60.366	181.02	4.029035453	271.65	256.21	140.85	91.957	13.079	124.76	110.67	113.68906	119.7256
Weight Sphere thickness= 0.2 (g)	15.3	119.39	358.01	7.968540511	537.26	506.72	278.58	181.87	25.868	246.74	218.88	224.8518	236.7908
Weight Cube thickness= 0.1(g)	4.5217	35.285	105.81	2.355034483	158.78	149.76	82.331	53.751	7.645	72.922	64.689	66.45304	69.98152
Weight Cube thickness= 0.2 (g)	8.8625	69.158	207.38	4.615880402	311.21	293.53	161.37	105.35	14.984	142.93	126.79	130.24832	137.1642



Hull Decision Matrix

Decision Matrix -Middle Part					
Material Name	**Weight**	**Impact Strength***	**Cost**	**Water Resistivity**	**Total**
Nomex Honeycomb Core	1	2	4	2	2.2
3K, Plain Weave Carbon Fiber	2	2	5	2	2.9
Generic Carbon Fiber	4	2	4	2	3.4
Airex Foam Core	1	2	4	2	2.2
Aluminum 6061	5	3	3	3	3.8
E-glass Fiber	5	3	4	2	4
PVC (hard)	4	4	2	2	3.2
Styrene (plastic)	3	4	2	2	2.8
Balsa Wood Class IV	2	5	1	5	2.6
PLA	3	3	2	2	2.6
ABS	3	3	2	2	2.6
TangoBlack	3	4	2	2	2.8
VeroClear	3	4	2	2	2.8
*This property is the resistivity of material against a collision					

Hull Weight Distribution of Factors for Decision Matrix

Weight Distribution				
Weight	**Impact Strength***	**Cost**	**Water**	**Total**
0.4	0.2	0.3	0.1	1

Appendix D: Motor Decision Matrix, Parameters, and Weight Distribution of Factors for Decision Matrix

Motor Decision Matrix

Decision Matrix -Motor								
Motor Names	**Weight**	**KV Value**	**Cost**	**Thrust**	**RPM**	**Curent**	**User Reviews**	**Total**
Turnigy Park250 Brushless Outrunner 2200kv	4	4	2	4	4	3	1	3.275
Turnigy A1405-3000KV Indoor Brushless Motor	1	3	1	3	3	4	4	2.825
AX 1806C 2100kv Brushless Micro Motor	5	4	2	3	3	3	3	3.3
C2020 Micro brushless Outrunner 3500kv	3	3	1	2	2	1	2	1.925
C05M Micro brushless outrunner 11000kv	1	1	3	1	1	3	5	2.05

Motor Parameter Values

Motor Parameters							
Motor Names	**Weight**	**KV Value**	**Cost**	**Thrust**	**RPM**	**Current**	**User Reviews**
Turnigy Park250 Brushless Outrunner 2200kv	14 g	2200	$10.58	0.18 kg	15400	7A	*****
Turnigy A1405-3000KV Indoor Brushless Motor	5 g	3000	$9.97	0.34 kg	21000	6A	**
AX 1806C 2100kv Brushless Micro Motor	26 g	2100	$11.09	0.42 kg	23100	7A	***
C2020 Micro brushless Outrunner 3500kv	11 g	3500	$9.99	1.20 kg	38850	10A	****
C05M Micro brushless outrunner 11000kv	4.6 g	11000	$14.76	11.92 kg	122100	7A	*

Motor Weight Distribution of Factors for Decision Matrix

Weight Distribution							
Weight	**KV Value**	**Cost**	**Thrust**	**RPM**	**Curent**	**User Reviews**	**Total**
0.15	0.1	0.1	0.225	0.1	0.225	0.1	1

Appendix E: Battery Decision Matrix, Parameters, and Weight Distribution of Factors for Decision Matrix

Battery Decision Matrix

Decision Matrix -Battery					
Battery Names	Weight	Capacity	Cost	Length	Total
Rhino 4900mAh 3S1P 11.1v 20C Lipoly Pack	3	3	2	5	3.4
ZIPPY Flightmax 8000mAh 3S1P 30C	5	1	4	5	3.3
Turnigy 5000mAh 3S 25C Lipo Pack	4	3	3	4	3.5
ZIPPY Flightmax 3000mAh 3S1P 20C	2	5	1	3	3.35
ZIPPY Flightmax 2500mAh Transmitter Pack (Futaba/JR	1	5	1	1	2.6
Polyquest 6100mAh 3S1P LP3 20C Aircraft Lipo Pack	4	2	5	5	3.55
Turnigy nano-tech 4000mah 3S 40~80C Lipo Pack	3	4	3	3	3.4

Battery Parameter Values

Battery Parameters				
Battery Names	Weight	Capacity	Cost	Length
Rhino 4900mAh 3S1P 11.1v 20C Lipoly Pack	352 g	4900 mAh	$22.49	165 mm
ZIPPY Flightmax 8000mAh 3S1P 30C	644 g	8000 mAh	$47.03	169 mm
Turnigy 5000mAh 3S 25C Lipo Pack	412 g	5000 mAh	$28.08	146 mm
ZIPPY Flightmax 3000mAh 3S1P 20C	239 g	3000 mAh	$14.78	137 mm
ZIPPY Flightmax 2500mAh Transmitter Pack (Futaba/JR	146 g	2500 mAh	$15.99	100 mm
Polyquest 6100mAh 3S1P LP3 20C Aircraft Lipo Pack	427 g	6100 mAh	$64.34	164 mm
Turnigy nano-tech 4000mah 3S 40~80C Lipo Pack	321 g	4000 mAh	$35.57	136 mm

Battery Weight Distribution of Factors for Decision Matrix

Weight Distribution				
Weight	Capacity	Cost	Length	Total
0.25	0.4	0.1	0.25	1

Appendix F: Material Properties

	Nomex Honeycomb Core[10]	3K, Plain Weave Carbon Fiber[25]	Generic Carbon Fiber[8]	Airex Foam Core[26]	Aluminum 6061[27]	E-glass Fiber[28]	PVC[29]	Styrene[30]	Balsa Wood Class IV[31]
Density (g/cm^3)	0.076888624	0.6	1.799194	0.040046158	2.7	2.546551	1.4	0.914	0.13
Young's/Tensile Modulus E (GPa)	0.01121777	227.5270499	227.5269906	0.027992715	68.9	72.39495155	3	2.9	6
Tensile Strength (GPa)	0.00049987	4.205803043	3.998959228	0.000689476	0.31	3.447378645	0.04	0.045	0.075
Assumed Factor of Safety	2	2	2	2	2	2	2	2	2
Bending Stress (GPa)	0.000249935	2.102901521	1.999479614	0.000344738	0.155	1.723689323	0.02	0.0225	0.0375

[25] Fibre Glast Developments Co. *3K, Plain Weave Carbon Fiber Product Data Sheet*. N.p.: Fibre Glast Developments, n.d. *Fibre Glast*. Web. 17 Oct. 2013. <http://cdn.fibreglast.com/downloads/00095-C.pdf>.

[26] DragonPlate Company. *Carbon Fiber Airex Foam Core*. N.p.: DragonPlate, n.d. *DragonPlate Company*. Web. 17 Oct. 2013. <http://dragonplate.com/docs/DPSpecAirexCore.pdf>

[27] ASM Aerospace Material Specfication Metals Inc. *Aluminum 6061-T6; 6061-T651 Data Sheet*. N.p.: ASM Aerospace Material Specfication Metals, n.d. Web. 17 Oct. 2013. <http://asm.matweb.com/search/SpecificMaterial.asp?bassnum=MA6061t6>

[28] Carbonfibertubeshop. *Properties of Carbon Fiber Tubes*. N.p.: Carbonfibertubeshop, n.d.*Tube Properties*. Web. 17 Oct. 2013. <http://www.carbonfibertubeshop.com/tube%20properties.html>.

[29] PVC Org. *PVC Strength*. N.p.: PVC Org, n.d. *PVC*. Web. 17 Oct. 2013. <http://www.pvc.org/en/p/pvc-strength>.

[30] *Overview of Materials for Styrene Acrylonitrile (SAN), Molded*. *Overview of Materials for Styrene Acrylonitrile (SAN), Molded*. N.p., n.d. Web. 17 Oct. 2013. <http://www.matweb.com/search/DataSheet.aspx?MatGUID=b19565721c534077911ecf643c7cfc94>.

[31] Matbase. *Balsa Class 4*. N.p.: Matbase, n.d. *Chemical, Mechanical, Physical and Environmental Properties of Materials*. Web. 17 Oct. 2013. <http://www.matbase.com/material-categories/composites/polymer-matrix-composites-pmc/wood/class-4-wood-slightly-durable/material-properties-of-balsa-wood.html#properties>.

Appendix G: Landing Gear Spring Suggestions

Spring MC078-0205

Metric Compression Springs

Double click on above image to view full picture

Additional Information	
Deflection at Load P (mm)	4.1
Free Length (mm)	20.5
Hole Diameter (mm)	14.6
Length, L1 (mm)	16.4
Load, P (newtons)	518.77
Outside Diameter (mm)	14
Part Number	MC078-0205
Rate, R (N/mm)	127.49
Rod Size (mm)	7.8
Solid Height (mm)	15.4
Stainless Load P	454.23
Stainless Rate	110.79
Wire Diameter (mm)	2.8

Spring MC060-0162

Metric Compression Springs

Double click on above image to view full picture

Additional Information	
Deflection at Load P (mm)	3.3
Free Length (mm)	16.2
Hole Diameter (mm)	11.6
Length, L1 (mm)	12.9
Load, P (newtons)	337.35
Outside Diameter (mm)	11
Part Number	MC060-0162
Rate, R (N/mm)	100.03
Rod Size (mm)	6
Solid Height (mm)	12.1
Stainless Load P	286.86
Stainless Rate	86.93
Wire Diameter (mm)	2.2

Spring MC075-180

Metric Compression Springs

Double click on above image to view full picture

Additional Information

Deflection at Load P (mm)	6.8
Free Length (mm)	18
Hole Diameter (mm)	12.5
Length, L1 (mm)	11.2
Load, P (newtons)	317.74
Outside Diameter (mm)	12
Part Number	MC075-0180
Rate, R (N/mm)	46.58
Rod Size (mm)	7.5
Solid Height (mm)	11
Stainless Load P	275.25
Stainless Rate	40.48
Wire Diameter (mm)	2

Spring MC060-0230

Metric Compression Springs

Double click on above image to view full picture

Additional Information

Deflection at Load P (mm)	5.3
Free Length (mm)	23
Hole Diameter (mm)	11.6
Length, L1 (mm)	17.7
Load, P (newtons)	337.35
Outside Diameter (mm)	11
Part Number	MC060-0230
Rate, R (N/mm)	63.55
Rod Size (mm)	6
Solid Height (mm)	16.5
Stainless Load P	292.69
Stainless Rate	55.22
Wire Diameter (mm)	2.2

Appendix H: Landing Gear Calculations MATLAB Code

```
%Acceleration due to gravity - m/s^2
g=9.81;
%Mass - kg
m=1.5;
%Height assumptions - m
h=[1,3,5,8,10];
%Initial Potential Energy - J
PE_initial=m*g.*h;
Total_Energy1=PE_initial;
%initial velocity assumed as 0 - m
v_initial1=0;
%Initial Kinetic Energy - J
KE_initial1=0.5*m*v_initial1;
%Impact velocity -m/s^2
v_impact=sqrt(2*g*h);
%Net work done - J
W_net1=0.5*m*(v_impact.^2-v_initial1.^2);
%Distance traveled after impact - m
d=[0.02,0.05,0.1,0.15,0.2];
%Average impact force - N
F_average_impact1=[W_net1/d(1);W_net1/d(2);W_net1/d(3);W_net1/d(4);W_net1/d(5
)];

%-------------------------------------------------------------------------------
---------%
%New Calculations with V_initial= 2m/s ; 5m/s ; 8m/s and 12m/s
v_initial2=1;
v_initial3=2;
v_initial4=3;
v_initial5=5;

KE_initial2=0.5*m*v_initial2^2;
KE_initial3=0.5*m*v_initial3^2;
KE_initial4=0.5*m*v_initial4^2;
KE_initial5=0.5*m*v_initial5^2;

Total_Energy2=PE_initial+KE_initial2;
Total_Energy3=PE_initial+KE_initial3;
Total_Energy4=PE_initial+KE_initial4;
Total_Energy5=PE_initial+KE_initial5;

v_impact2=sqrt(2*Total_Energy2/m);
v_impact3=sqrt(2*Total_Energy3/m);
v_impact4=sqrt(2*Total_Energy4/m);
v_impact5=sqrt(2*Total_Energy5/m);

W_net2=0.5*m*(v_impact2.^2-v_initial2.^2);
W_net3=0.5*m*(v_impact3.^2-v_initial3.^2);
W_net4=0.5*m*(v_impact4.^2-v_initial4.^2);
W_net5=0.5*m*(v_impact5.^2-v_initial5.^2);

F_average_impact2=[W_net2/d(1);W_net2/d(2);W_net2/d(3);W_net2/d(4);W_net2/d(5
)];
```

```
F_average_impact3=[W_net3/d(1);W_net3/d(2);W_net3/d(3);W_net3/d(4);W_net3/d(5
)];
F_average_impact4=[W_net4/d(1);W_net4/d(2);W_net4/d(3);W_net4/d(4);W_net4/d(5
)];
F_average_impact5=[W_net5/d(1);W_net5/d(2);W_net5/d(3);W_net5/d(4);W_net5/d(5
)];

%-------------------------------------------------------------------------
---------%
%Net Impact Force Calculation
F_net_impact1=F_average_impact1.*cosd(50);
F_net_impact2=F_average_impact2.*cosd(50);
F_net_impact3=F_average_impact3.*cosd(50);
F_net_impact4=F_average_impact4.*cosd(50);
F_net_impact5=F_average_impact5.*cosd(50);



%-------------------------------------------------------------------------
---------%
%Damping Force Calculation
%Net Impact Force, taking the maximum force calculated above
NIT=F_net_impact1(1,5);
%Different stroke diameter sizes
d_stroke=[15,18,22,25,28];
%Total Energy, taking the maximum energy calculated above
TE=Total_Energy5(5);
%Energy of Stroke
E_stroke=(TE/4)+(NIT/4).*d_stroke/1000;
%Correction factor
cf=2;
%Damping Force
DF=(E_stroke./d_stroke)*1000*cf;
```

Appendix I: Future Work and Alternative Recommendations

As a part of alternative future work, the team came up with 3 different UAV designs that can be implemented during future MQPs. The first design is a HexaCopter model, which consists of a middle hull, 6 small-size motors (10000 rpm) with 8x4 propellers, landing gear, and a top structure that connects all of the parts together while ensuring mechanical robustness.

Figure 1 shows the final design after the simulation and modeling. The initial top frame design had some modifications based on the results of simulations. The main top frame was intended to be manufactured with PLA. However, the displacement calculations show that PLA has approximately 1.027 mm deformation on the motor hulls whereas acrylic has 6.816e-1 mm (Appendix J). From the strain analysis point of view, changing the material decreases the strain concentrations by around 30 percent. The final decisions were made based on the weight analysis and the most feasible was to use Delrin plastic, which has similar material behaviors as acrylic but is lighter. The remaining body parts can be made of PLA. The approximate structural weight is estimated to be 695.71 grams for the given materials. Thrust is calculated to be around 3.12 kg, but it can be increased to 7.62 kg using 10x5 propellers.

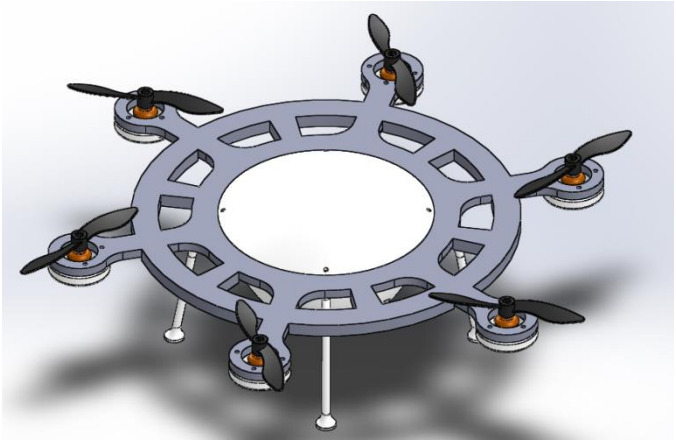


Figure 1: HexaCopter Isometric View

As another alternative future work, the second design is a PentaCopter model. This model consists of a large motor (18000 rpm) with 10x5 propellers or bigger, 4 small-size motors (9000 rpm) with 6x3 propellers, 4 battery casings, 4 electronic board casings and a top structure that connects all of the parts together while ensuring mechanical robustness. The aim is to lift the drone using the big motor located in the middle and control the pitch, yaw and altitude using the small motors.

Figure 2 below shows the final design after the simulation and modeling. The top frame design is based on a square model that connects all motors, battery casings, and other structural parts. The battery casings will also serve as landing gear. Due to airflow and aerodynamics, the electronic boards and wires can't be placed in the middle seen in the HexaCopter model. The middle hull's propeller needs to push the air from the top to bottom, so additional electronic board casings were placed on the top. The top frame's design changed many times based on simulations and the square frame is found to optimize performance in this limited setting. As with the HexaCopter model, Delrin could be used for the top frame, with the rest of the body made of PLA. The approximate

structural weight is estimated to be 910.30 grams for the given materials. Thrust is calculated to be around 4.66 kg.

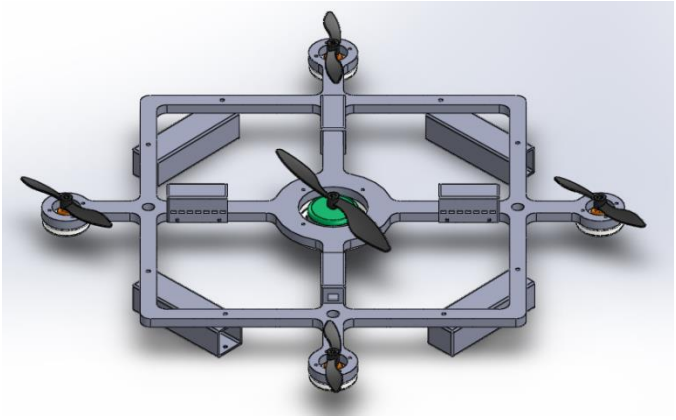


Figure 2: PentaCopter Top View

The last alternative future design is a TriCopter model, which consists of 3 small-size motors (10000 rpm) with 10x5 propellers, 3 battery/electronic board casings and 1 middle hull for wires, extra electronic components including cameras, a GPS, the IMU, etc., and top structure that connects all parts together while ensuring mechanical robustness.

Figure 3 shows the final design after the simulation and modeling. The top frame design is based on a triangle model; it connects all the motors and the battery casing component. The electronics casing will also serve as landing gear. The top frame's design changed many times based on simulations, and a triangle frame is found to be a competitive option. Unlike the previous two designs, the top structural model is made of PLA due to the unfeasibility of using a heavy material like Delrin for a 3-motor drone. The approximate structural weight is estimated to be 598.88 grams for the given materials. Thrust is calculated to be around 3.81 kg.

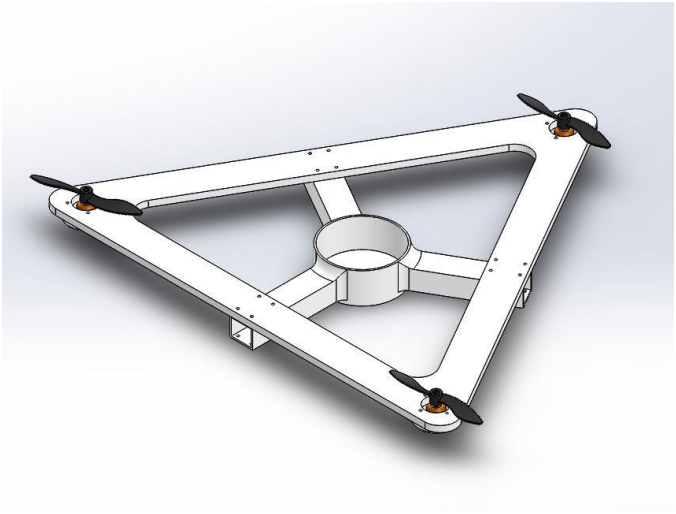
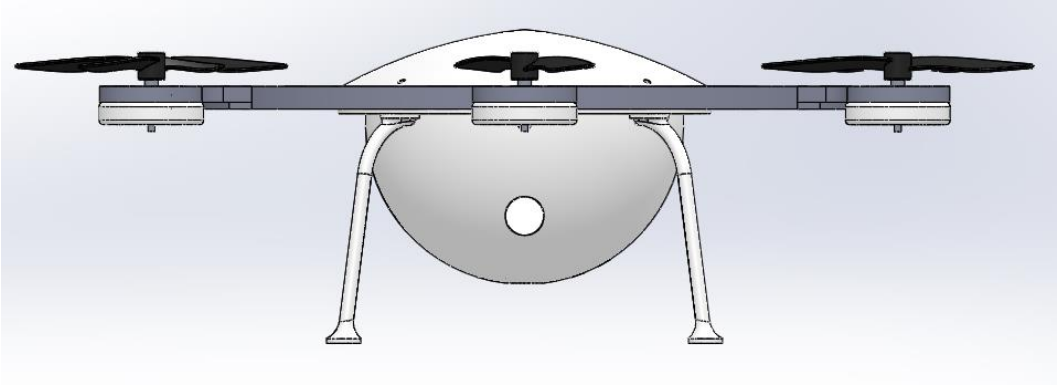


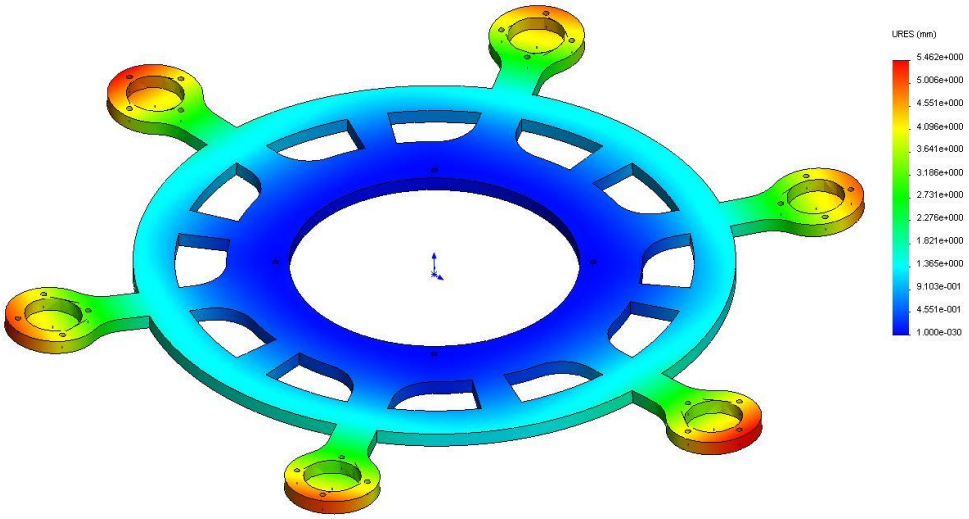
Figure 3: TriCopter Isometric View

Appendix J: Multirotor Design Models

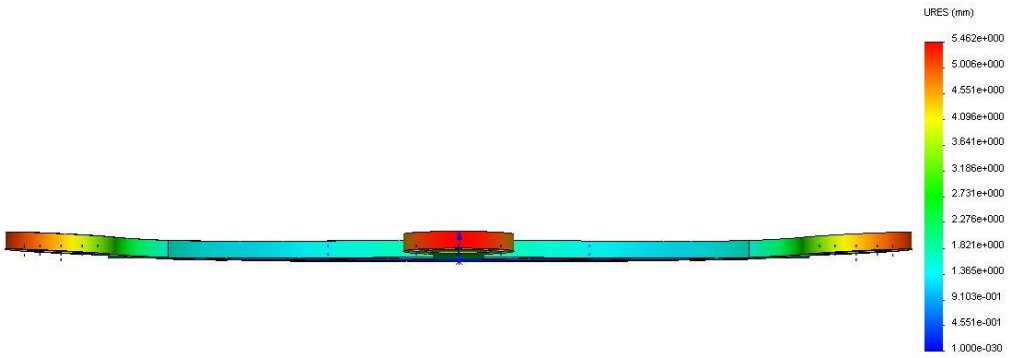
HexaCopter Side View



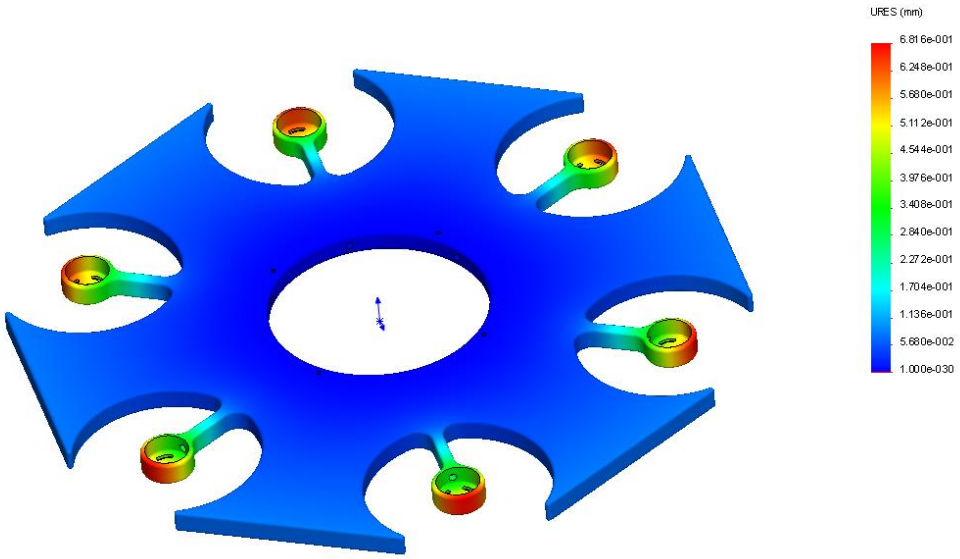
HexaCopter Displacement Simulation Isometric View



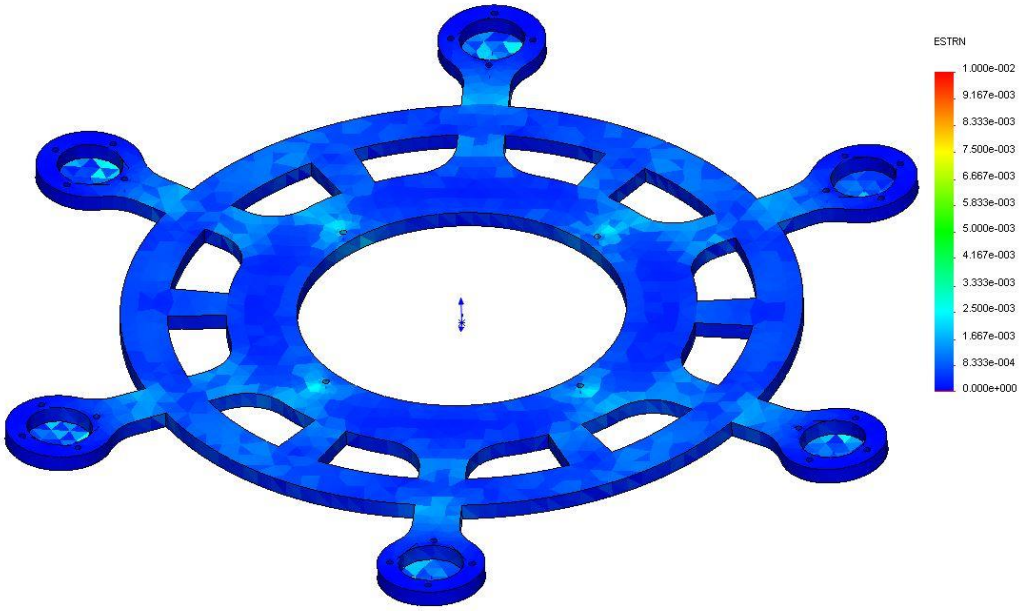
HexaCopter Displacement Simulation Front View



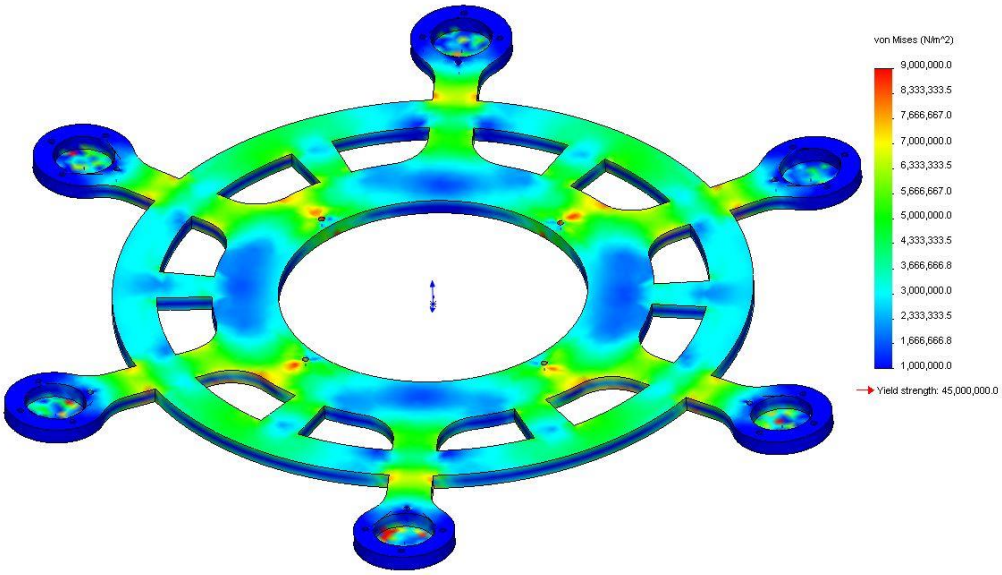
HexaCopter Displacement Simulation and Initial Simulation Isometric View



HexaCopter Strain Simulation Isometric View



HexaCopter Stress Simulation Isometric View



Appendix K: Quadrotor Simulation MATLAB Code

```
clear all; clc;

%Testing variables...
ExampPDz = 0;
ExampleZ = 0;
ExampZd = 0;
ExampZdd = 0;
ExampYd = 0;
ExampYdd = 0;
ExamPDphi = 0;
fore =0;
stop = 0;
allU1 = 0;
allU2 = 0;
allU3 = 0;
allU4 = 0;
allO = 0;
allXd = 0;
allYd = 0;
allZd = 0;
allzvd = 0;
allyvd = 0;
allzad = 0;
allyad = 0;

%Quadrotor constants
d = .00001;
b = .0001;
l = .3;
m = 1.6;
Ixx = .01;
Iyy = .01;
Izz = .03;
g = 9.81;
Jr = .0005;

%Angular velocities for each rotor
O1 = 0;
O2 = 0;
O3 = 0;
O4 = 0;

M1 = O1;
M2 = O2;
M3 = O3;
M4 = O4;


%Global position and orientation
X = 0;
Y = 0;
Z = 0;
theta = 0;
phi = 0;
```

```
psi = 0;

allTheta = theta;
allPhi = phi;
allPsi = psi;
allX = X;
allY = Y;

%Global velocity and acceleration
Xd = 0;          Xdd = 0;
Yd = 0;          Ydd = 0;
Zd = 0;          Zdd = 0;
thetaD = 0;      thetaDd = 0;
phiD = 0;        phiDd = 0;
psiD = 0;        psiDd = 0;

%Desired position and orientation
%zDesire = 10;
%yDesire = -2;
%xDesire = 6;
thetaDesire = 0;        % 90 > theta and phi > -90
phiDesire = 0;
psiDesire = 0;

%PID / PD values
KpZ = 50;
KiZ = 5;
KdZ = 30;

errSum = 0;
errYSum = 0;
errXSum = 0;

KpY = 2;
KiY = 0;
KdY = 1;

KpX = 2;
KiX = 0;
KdX = 1;

KpTh = .8;
KdTh = 1;

KpPsi = 2;
KdPsi = 1;

KpPhi = .8;
KdPhi = 1;


%Simulation/Animation variables
myaxes = axes('xlim', [-15 15],'ylim',[-15 15],'zlim',[0 20]);
view(3);
```

```
grid on;
xlabel('x');
ylabel('y');
zlabel('z');

[xsphere ysphere zsphere] = sphere();
[xcylinder ycylinder zcylinder] = cylinder([.1 .1]);

h(1) = surface(.4*xsphere+1, 0.4*ysphere,.4*zsphere);
h(2) = surface(.4*xsphere-1, 0.4*ysphere,.4*zsphere);
h(3) = surface(.4*xsphere, 0.4*ysphere+1,.4*zsphere);
h(4) = surface(.4*xsphere, 0.4*ysphere-1,.4*zsphere);
h(5) = surface(zcylinder,xcylinder,ycylinder);
h(6) = surface(-zcylinder,xcylinder,ycylinder);
h(7) = surface(ycylinder,zcylinder,xcylinder);
h(8) = surface(ycylinder,-zcylinder,xcylinder);

combinedobject = hgtransform('parent', myaxes);
set(h, 'parent', combinedobject)
drawnow

transform = makehgtform('translate', [X Y Z],'xrotate', phi,'yrotate',
theta,'zrotate',psi);
set(combinedobject,'matrix',transform);

%Variables needed for simulation
Tf = 4;
dt = .01;
%PD error
pzError = 0;
pyError = 0;
pxError = 0;
pthError = 0;
ppsiError = 0;
pphiError = 0;

%Quintic Trajectory
to = 0;
tf = Tf;
A = [1 to to^2 to^3 to^4 to^5; 0 1 2*to 3*to^2 4*to^3 5*to^4; 0 0 2 6*to
12*to^2 20*to^3; 1 tf tf^2 tf^3 tf^4 tf^5; 0 1 2*tf 3*tf^2 4*tf^3 5*tf^4; 0 0
2 6*tf 12*tf^2 20*tf^3];
AI = inv(A);
zPos = [ 0; Zd; Zdd; 12; 0; 0 ];
xPos = [ 0; Xd; Xdd; 6; 0; 0 ];
yPos = [ 0; Yd; Ydd; -8; 0; 0 ];
xA = AI*xPos;
yA = AI*yPos;
zA = AI*zPos;


for i=0:dt:Tf

    %Quintics
xDesire = xA(1) + xA(2)*i + xA(3)*i^2 + xA(4)*i^3 + xA(5)*i^4 +xA(6)*i^5;
```

```
yDesire = yA(1) + yA(2)*i + yA(3)*i^2 + yA(4)*i^3 + yA(5)*i^4 +yA(6)*i^5;
zDesire = zA(1) + zA(2)*i + zA(3)*i^2 + zA(4)*i^3 + zA(5)*i^4 +zA(6)*i^5;
xVelDesire = xA(2) + 2*xA(3)*i + 3*xA(4)*i^2 + 4*xA(5)*i^3 +xA(6)*5*i^4;
zVelDesire = zA(2) + 2*zA(3)*i + 3*zA(4)*i^2 + 4*zA(5)*i^3 +zA(6)*5*i^4;
yVelDesire = yA(2) + 2*yA(3)*i + 3*yA(4)*i^2 + 4*yA(5)*i^3 +yA(6)*5*i^4;
zAccDesire = 2*zA(3) + 6*zA(4)*i + 12*zA(5)*i^2 +zA(6)*20*i^3;
yAccDesire = 2*yA(3) + 6*yA(4)*i + 12*yA(5)*i^2 +yA(6)*20*i^3;
xAccDesire = 2*xA(3) + 6*xA(4)*i + 12*xA(5)*i^2 +xA(6)*20*i^3;


  %PID for Z
  zError = (zDesire - Z);
  Pz = KpZ * zError;
  errSum = errSum + (zError * dt);
  Iz = KiZ * errSum;
  Dz = KdZ * (zError-pzError)/dt;
  pzError = zError;
  PDz = Pz + Iz + Dz;


  %PD for Y
  yError = (yDesire - Y);
  PDy = KpY*yError + KdY*(yError - pyError)/dt;
  errYSum = errYSum + (yError * dt);
  Iy = KiY * errYSum;
  pyError = yError;
  phiDesire = -(PDy+Iy);
  if phiDesire > .725
      phiDesire = .725;
  elseif phiDesire < -.725
      phiDesire = -.725;
  end;

  %PD for X
  xError = (xDesire - X);
  PDx = KpX*xError + KdX*(xError - pxError)/dt;
  errXSum = errXSum + (xError * dt);
  Ix = KiX * errXSum;
  pxError = xError;
  thetaDesire = (PDx+Ix);
  if thetaDesire > .725
      thetaDesire = .725;
  elseif thetaDesire < -.725
      thetaDesire = -.725;
  end;


  %PD for theta
  thError = (thetaDesire - theta);
  Ptheta = KpTh * thError;
  Dtheta = KdTh * (thError-pthError)/dt;
  pthError = thError;
  PDtheta = Ptheta + Dtheta;

  %PD for phi
```

```
phiError = (phiDesire - phi);
Pphi = KpPhi * phiError;
Dphi = KdPhi * (phiError-pphiError)/dt;
pphiError = phiError;
PDphi = Pphi + Dphi;

%PD for psi
psiError = (psiDesire - psi);
Ppsi = KpPsi * psiError;
Dpsi = KdPsi * (psiError-ppsiError)/dt;
ppsiError = psiError;
PDpsi = Ppsi + Dpsi;

U1 = ((PDz * m) / (cos(phi)*cos(theta)));
U2 = PDphi;
U3 = PDtheta;
U4 = PDpsi;


  % Limiting by the max motor speed
if U1 > 2075
  U1 = 2075;
elseif U1 < -2075
  U1 = -2075;
end;

if U2 > 20
  U2 = 20;
elseif U2 < -20
  U2 = -20;
end;

if U3 > 20
  U3 = 20;
elseif U3 < -20
  U3 = -20;
end;

if U4 > 20
  U4 = 20;
elseif U4 < -20
  U4 = -20;
end;


% motor forces calculated by inverting U = Mf
O1 = U1/(4*b) - U3/(2*b*l) - U4/(4*d);
O2 = U1/(4*b) - U2/(2*b*l) + U4/(4*d);
O3 = U1/(4*b) + U3/(2*b*l) - U4/(4*d);
O4 = U1/(4*b) + U2/(2*b*l) + U4/(4*d);
Omega = real(-sqrt(O1) + sqrt(O2) - sqrt(O3) + sqrt(O4));


phiDd = (Iyy-Izz/Ixx)*thetaD*psiD - (Jr/Ixx)*thetaD*Omega + (l/Ixx)*U2;
thetaDd = (Izz-Ixx/Iyy)*phiD*psiD + (Jr/Iyy)*phiD*Omega + (l/Iyy)*U3;
```

```
  psiDd = (Ixx-Iyy/Izz)*phiD*thetaD + (1/Izz)*U4;

  phiD = phiD + phiDd*dt;
  psiD = psiD + psiDd*dt;
  thetaD = thetaD + thetaDd*dt;
  phi = phi + phiD*dt;
  psi = psi + psiD*dt;
  theta = theta + thetaD*dt;

  Xdd = (U1/m) * ( cos(phi)*sin(theta)*cos(psi) + sin(phi)*sin(psi) );
  Ydd = (U1/m) * ( cos(phi)*sin(theta)*sin(psi) - sin(phi)*cos(psi) );
  Zdd = -g + ((U1/m) * cos(phi)*cos(theta));


  Zd = Zd + Zdd*dt;
  Xd = Xd + Xdd*dt;
  Yd = Yd + Ydd*dt;
  X = X + Xd*dt;
  Y = Y + Yd*dt;
  Z = Z + Zd*dt;


  transform = makehgtform('translate', [X Y Z],'xrotate', phi,'yrotate',
theta,'zrotate',psi);
  set(combinedobject,'matrix',transform);

  pause(dt);


%original plotting technique

allX = [allX; X];
allY = [allY; Y];
allXd = [allXd; xDesire];
allYd = [allYd; yDesire];
allZd = [allZd; zDesire];

end


%plot X, Y, Z position (desired and actual)
t = dt:dt:Tf;
figure();
subplot(2,2,1)
plot(allX,'r');
hold;
plot(allXd);
title('X');
subplot(2,2,2)
plot(allY,'r');
hold;
plot(allYd);
title('Y');
subplot(2,2,3)
```

```
plot(ExampleZ,'r');
hold;
plot(allZd);
title('Z');
```

Appendix L: Basic Arduino to Raspberry Pi Communication MATLAB Code

```
#include <SPI.h>

const int ledPin = 13;
double piSerTotal = 0;
boolean hasDec;
byte numByte;

void setup()
{
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
  Serial.setTimeout(500);
}

void loop()
{
  if (Serial.available())
  {
    char serial[20] = "00000000000000000000";
    piSerTotal = 0;
    hasDec = false;
    numByte = Serial.readBytesUntil('end',serial,20);
    for( int i = 0; i < numByte; i++) {
      if ( serial[i] == '.') {
        hasDec = true;
        i = numByte;
      }
    }
    if (hasDec == false) {
      serial[numByte] = '.';
    }

    piSerTotal = atof(serial);
    Serial.println(piSerTotal);
  }
}
```

Appendix M: Arduino Stability Control MATLAB Code

```
/* WIRING

  Arduino Uno - Red - 5V
              White - Gnd
              Yellow - A4
              Blue - A5

  IMU Connector - Red, White, Blue, Yellow  starting from Red,Black,Blue,Green
*/

#include <Wire.h>
#include <Servo.h>
#include <PPM.h>

PPM ppm(2);

#define MAX_SIGNAL 2000
#define MIN_SIGNAL 760
#define MOTOR_PIN_FRONT 11    //Clockwise
#define MOTOR_PIN_RIGHT 5     //CCW
#define MOTOR_PIN_BACK 3      //Clockwise
#define MOTOR_PIN_LEFT 10     //CCW


Servo motorFront;
Servo motorRight;
Servo motorBack;
Servo motorLeft;

#define ACCEL_ADR (0x30 >> 1)
#define MAG_ADR (0x3c >> 1)
#define GYRO_ADR 0x68
#define ACCEL_CTRL_REG1 0x20
#define ACCEL_CTRL_REG2 0x21
#define ACCEL_CTRL_REG3 0x22
#define ACCEL_CTRL_REG4 0x23
#define ACCEL_CTRL_REG5 0x24
#define ACCEL_OUT_X_L 0x28

#define MAG_CRA_REG 0x00
#define MAG_CRB_REG 0x01
#define MAG_MR_REG 0x02
#define MAG_OUT_X_H 0x03
```

```
#define GYRO_SMPLRT_DIV 0x15
#define GYRO_INT_CFG 0x17
#define GYRO_PWR_MGM 0x3E
#define GYRO_OUT_X_H 0x1D


int16_t accX, accY, accZ;
int16_t gyroX, gyroY, gyroZ;

double accXangle, accYangle, accZangle;
double gyroXangle, gyroYangle, gyroZangle;
double MagX, MagY, MagZ;
double compAngleX, compAngleY, compAngleZ;
double cos_x, sin_x, cos_y, sin_y, YawEst, YawEst2;

double thError, theta, Ptheta, Itheta, Dtheta, PIDtheta;
double phiError, phi, Pphi, Iphi, Dphi, PIDphi;
double yawError, yaw, Pyaw, Dyaw, PDyaw;
double pphiError = 0;
double pthError = 0;
double pyawError = 0;
double thetaDesire = 180;
double phiDesire = 180;
double yawDesire = 90;
double KpTh = 2;    //1 has oscillation at max speed
double KiTh = .1;
double KdTh = .3;
double errThSum = 0;
double KpPhi = 2; //1 has oscillation at max speed
double KiPhi = .1;
double KdPhi = .3;
double errPhiSum = 0;
double KpYaw = 0;
double KiYaw = 0;
double KdYaw = 0;
double errYawSum = 0;
double motorF, motorB, motorR, motorL;
double temp = 0;
double timeDiff = 0;
double timeDiff2 = 0;
int throttle;

long timer;
long timer2;
uint8_t i2cData[14]; // Buffer for I2C data
```

```
void setup() {
  Wire.begin();
  Serial.begin(9600);

  // Motor Calibration
  motorFront.attach(MOTOR_PIN_FRONT);
  motorRight.attach(MOTOR_PIN_RIGHT);
  motorBack.attach(MOTOR_PIN_BACK);
  motorLeft.attach(MOTOR_PIN_LEFT);
  motorFront.writeMicroseconds(MAX_SIGNAL);
  motorRight.writeMicroseconds(MAX_SIGNAL);
  motorBack.writeMicroseconds(MAX_SIGNAL);
  motorLeft.writeMicroseconds(MAX_SIGNAL);
  delay(2000);
  motorFront.writeMicroseconds(MIN_SIGNAL);
  motorRight.writeMicroseconds(MIN_SIGNAL);
  motorBack.writeMicroseconds(MIN_SIGNAL);
  motorLeft.writeMicroseconds(MIN_SIGNAL);
  delay(3000);


  // Initialize IMU
  writeTo(ACCEL_ADR, ACCEL_CTRL_REG1, 0x37);
  writeTo(ACCEL_ADR, ACCEL_CTRL_REG4, 0xD0);
  writeTo(MAG_ADR, MAG_MR_REG, 0x00);
  writeTo(GYRO_ADR, GYRO_PWR_MGM, 0x80);
  writeTo(GYRO_ADR, GYRO_SMPLRT_DIV, 0x07);
  writeTo(GYRO_ADR, GYRO_INT_CFG, 0x00);

  delay(100);

  byte buffer[6];
  readFrom(ACCEL_ADR, (ACCEL_OUT_X_L | (1 << 7)), 6, buffer);
  accX = (int)(buffer[0] << 8 | buffer[1]);
  accY = (int)(buffer[2] << 8 | buffer[3]);
  accZ = (int)(buffer[4] << 8 | buffer[5]);
  readFrom(MAG_ADR, MAG_OUT_X_H, 6, buffer);
  MagX = (int)(buffer[0] << 8 | buffer[1]);
  MagY = (int)(buffer[2] << 8 | buffer[3]);
  MagZ = (int)(buffer[4] << 8 | buffer[5]);

  accXangle = (atan2(accX,sqrt(pow(accY,2) + pow(accZ,2)))+PI)*RAD_TO_DEG;
  accYangle = (atan2(accY,sqrt(pow(accX,2) + pow(accZ,2)))+PI)*RAD_TO_DEG;
```

```
  // Define hard iron values on magnetometer
  MagX = MagX + 10;
  MagY = MagY + 20;
  MagZ = MagZ +90;

  compAngleX = accXangle;
  compAngleY = accYangle;

  cos_x = cos(compAngleX*3.14159/180);
  sin_x = sin(compAngleX*3.14159/180);
  cos_y = cos(compAngleY*3.14159/180);
  sin_y = sin(compAngleY*3.14159/180);
  YawEst = atan2( -(MagY*cos_y + MagZ*sin_y ) , MagX*cos_x + MagY*sin_x*sin_y+
MagZ*sin_x*cos_y) *180/PI;
  compAngleZ = YawEst;

  timer2 = millis();
  timer = micros();
}

void loop() {

  IMUFunction();
  MotorControls();

}

//Function for reading num bytes from addresses on an I2C device
void readFrom(byte device, byte fromAddress, int num, byte result[]) {
  Wire.beginTransmission(device);
  Wire.write(fromAddress);
  Wire.endTransmission();
  Wire.requestFrom((int)device, num);

  int i = 0;
  while(Wire.available()) {
    result[i] = Wire.read();
    i++;
  }
}

//Function for writing a byte to an address on an I2C device
void writeTo(byte device, byte toAddress, byte val) {
  Wire.beginTransmission(device);
  Wire.write(toAddress);
  Wire.write(val);
```

```
  Wire.endTransmission();
}

void MotorControls () {

  // With tele-op RC controller
  throttle = ppm.getChannel(2);
  if (throttle < 60 && throttle > 0) {
    throttle = map(throttle, 60, 0, 800, 1900);
  }
  else {
     throttle = 760;
  }

  /*
  int throttle = ppm.getChannel(2);
  if(throttle < 60 && throttle > 0) {
    throttle = 1350;   //need to find value to linearize at (hovering speed?)
  }
  else {
    throttle = 760;
  }
  */

  //Offsets on sensor angles
  theta = compAngleX+2.5;
  phi = compAngleY-1;
  yaw = compAngleZ;

  timeDiff = ((double)(millis()-timer2))/1000;
  //PD for theta
  thError = (thetaDesire - theta);
  errThSum = errThSum + (thError*timeDiff);
  Ptheta = KpTh * thError;
  Itheta = KiTh * errThSum;
  Dtheta = KdTh * (thError-pthError)/(timeDiff);
  pthError = thError;
  PIDtheta = Ptheta + Dtheta; //+ Itheta;

  //PD for phi
  phiError = (phiDesire - phi);
  errPhiSum = errPhiSum + (phiError*timeDiff);
  Pphi = KpPhi * phiError;
  Iphi = KiPhi * errPhiSum;
  Dphi = KdPhi * (phiError-pphiError)/(timeDiff);
  pphiError = phiError;
```

```
PIDphi = Pphi + Dtheta; //+ Iphi;

//PD for yaw
yawError = (yawDesire - yaw);
Pyaw = KpYaw * yawError;
Dyaw = KdYaw * (yawError-pyawError)/(timeDiff);
pyawError = yawError;
PDyaw = Pyaw + Dyaw;
timer2 = millis();

PDyaw = 0;
//throttle will become PDz
motorF = throttle + PIDphi - PDyaw;   //U1 - U3 - U4;
motorR = throttle - PIDtheta + PDyaw;     //U1 - U2 + U4;
motorB = throttle - PIDphi - PDyaw;   //U1 + U3 - U4;
motorL = throttle + PIDtheta + PDyaw;     //U1 + U2 + U4;

if (motorF < 760) {
  motorF = 760;
}
else if (motorF > 2000) {
  motorF = 2000;
}
if (motorB < 760) {
  motorB = 760;
}
else if (motorB > 2000) {
  motorB = 2000;
}
if (motorR < 760) {
  motorR = 760;
}
else if (motorR > 2000) {
  motorR = 2000;
}
if (motorL < 760) {
  motorL = 760;
}
else if (motorL > 2000) {
  motorL = 2000;
}

motorFront.writeMicroseconds(motorF);
motorBack.writeMicroseconds(motorB);
motorRight.writeMicroseconds(motorR);
motorLeft.writeMicroseconds(motorL);
```

```
  /*
  Serial.print(phi);Serial.print("\t");
  Serial.print(theta); Serial.print("\t");
  Serial.print(yaw);Serial.print("\t");

  Serial.print("\n");
  */

  /*
  Serial.print(compAngleX);Serial.print("\t");
  Serial.print(compAngleY); Serial.print("\t");
  Serial.print(compAngleZ);Serial.print("\t");
  Serial.print(YawEst);Serial.print("\t");
  */


  Serial.print(motorF);Serial.print("\t");
  Serial.print(motorB);Serial.print("\t");
  Serial.print(motorR);Serial.print("\t");
  Serial.print(motorL);Serial.print("\t");


  //Serial.print(throttle);
  Serial.print("\n");


}

void IMUFunction() {

  byte buffer[6];
  readFrom(ACCEL_ADR, (ACCEL_OUT_X_L | (1 << 7)), 6, buffer);
  accX = (int)(buffer[0] << 8 | buffer[1]);
  accY = (int)(buffer[2] << 8 | buffer[3]);
  accZ = (int)(buffer[4] << 8 | buffer[5]);
  readFrom(GYRO_ADR, GYRO_OUT_X_H | (1 << 7), 6, buffer);
  gyroX = (int)(buffer[0] << 8 | buffer[1]);
  gyroY = (int)(buffer[2] << 8 | buffer[3]);
  gyroZ = (int)(buffer[4] << 8 | buffer[5]);
  readFrom(MAG_ADR, MAG_OUT_X_H, 6, buffer);
  MagX = (int)(buffer[0] << 8 | buffer[1]);
  MagY = (int)(buffer[2] << 8 | buffer[3]);
  MagZ = (int)(buffer[4] << 8 | buffer[5]);
```

```
  accXangle = (atan2(accX,sqrt(pow(accY,2) + pow(accZ,2)))+PI)*RAD_TO_DEG;
  accYangle = (atan2(accY,sqrt(pow(accX,2) + pow(accZ,2)))+PI)*RAD_TO_DEG;
  double gyroXrate = (double)gyroX/131.0;
  double gyroYrate = -((double)gyroY/131.0);
  double gyroZrate = (double)gyroZ/131.0;

  cos_x = cos((compAngleX)*3.14159/180);
  sin_x = sin((compAngleX)*3.14159/180);
  cos_y = cos((compAngleY)*3.14159/180);
  sin_y = sin((compAngleY)*3.14159/180);
  timeDiff2 = ((double)(micros()-timer))/1000000;
  compAngleX = (0.5*(compAngleX+(gyroXrate*timeDiff2)))+(0.5*accXangle); // Calculate the
angle using a Complimentary filter
  compAngleY = (0.5*(compAngleY+(gyroYrate*timeDiff2)))+(0.5*accYangle);

  //Hard iron offsets
  MagX = MagX + 10;
  MagY = MagY + 20;
  MagZ = MagZ + 90;

  YawEst=atan2( -(MagY*cos_y + MagZ*sin_y ) , MagX*cos_x + MagY*sin_x*sin_y+
MagZ*sin_x*cos_y) *180/PI;
  YawEst2=atan2(-MagY, MagX) *180/PI;
  //compAngleZ = (0.93*(compAngleZ+(gyroZrate*timeDiff2)))+(0.07*YawEst);
  compAngleZ = (0.5*(compAngleZ+(gyroZrate*timeDiff2)))+(0.5*YawEst);


  timer = micros();

  /*
  Serial.print(MagX);Serial.print("\t");
  Serial.print(MagY); Serial.print("\t");
  Serial.print(MagZ);Serial.print("\t");
  */

  /*
  Serial.print(compAngleX);Serial.print("\t");
  Serial.print(compAngleY); Serial.print("\t");
  Serial.print(compAngleZ);Serial.print("\t");
  Serial.print(YawEst2);Serial.print("\t");
  */

  //Serial.print("\r\n");
}
```