UCF Student Rocket Launch Initiative

Silver Team

# Design Implementation

Alec Bensema (Aerospace Engineering)
Brian Borgman (Aerospace Engineering)
Rene Estrada (Aerospace Engineering)
Sonja Frohock (Electrical Engineering)
Miguel Moran (Aerospace Engineering)
Zachary White (Aerospace Engineering)

Sponsored by
Aerojet Rocketdyne

Advised by
Gary Dahlke

Submitted April 26, 2023

UCF | Mechanical and Aerospace Engineering
UNIVERSITY OF CENTRAL FLORIDA

## Executive Summary

The UCF Student Launch Initiative was a challenge presented in pursuit of a Return-To-Launch-Site (RLTS) with an application in amateur High-Power Rocketry (HPR). The team was tasked to design, build, and launch a rocket with the capability to be guided back to a designated landing zone near the original launch point. Additionally, the rocket was required to meet certain requirements such as: a minimum altitude, remain below Level 2 HPR standards, and refrain from using combustion-based descent control, or lift generating descent control beyond that of the mass of the rocket. The team devised a solution primarily using a Rogallo wing based on NASA documentation to control the descent and a selectable control scheme allowing for automated control via GPS with a secondary manual control via a radio control receiver and handheld controller. The team was able perform numerous live drop tests to test and validate the flight ability and stability of the Rogallo parachute as well as provide flight simulations in OpenRocket and developed Simulink simulations.

## Table of Contents

## List of Figures

## List of Tables

## Glossary

- COTS – Commercial off the Shelf
- HPR – High Powered Rocketry
- NAR – National Association of Rocketry
- TRA – Tripoli Rocketry Association
- AGL – Above Ground Level
- UCF – University of Central Florida
- IMU – Inertial Measurement Unit
- GPS – Global Positioning System
- GNC – Guidance, Navigation, and Control
- PID – Proportional-Integral-Derivative
- FPV – First Person View
- PWM – Pulse Width Modulation
- Cd – Coefficient of Drag
- LZ – Landing Zone
- $p$ – Atmospheric pressure
- $c_p$ – Pressure coefficient
- g – Gravity
- ρ – Density
- $m_i$ – Initial Mass
- $m_f$ – Final Mass

- $\dot{m}$ – Mass flow rate
- $V_i$ – Initial Velocity
- $lat_R$ , $long_R$ – GPS Latitude and Longitude of Rocket
- $lat_{LZ}$ , $long_{LZ}$ – GPS Latitude and Longitude of Landing Zone
- $S$ – Reference Area (vertical facing area projection)
- M – Molar Mass of Carbon Dioxide (CO2)
- $R_0$ – Gas constant
- $T_0$ – Initial Temperature

# 1. Introduction

The purpose of this report is to generate a comprehensive set of system requirements and measurable criteria that can be used for conceptual design of a HPR system. The goal of this report is to generate a realizable design for a HPR system with the capability to autonomously guide itself during descent. Many university projects and HPR enthusiasts have aimed to solve this problem using hardware solutions such as controllable fins and parafoils.

Traditional HPR recovery systems rely on passive recovery systems, meaning that no moveable hardware is used. A parachute is the primary recovery method on most HPR systems, but these solutions are highly susceptible to environmental factors. Thus, rockets that reach a sufficiently high apogee began implementing dual-deployment systems where a small drogue parachute is deployed at apogee to limit lateral movement for most of the rocket descent phase. A large parachute is still used closer to ground-level to limit descent rate. While these systems increase the predictability of a rocket's landing location, those rockets may land in hazardous areas such as lakes or trees. Therefore, a novel recovery system that can steer a rocket to a specified location would reduce the risk of losing a rocket during recovery.

This report seeks to outline the process of designing a rocket with the following customer requirements. The rocket apogee must meet or exceed 3000 feet AGL. The rocket must not be actively guided during the ascent phase. All active control systems must be contained within the vehicle recovery system and may not produce a thrust-to-weight ratio greater than 0.9 (cannot produce sufficient lift to increase altitude once the recovery system activates). The recovery system should successfully land the rocket within 5 meters of a specific target site. Finally, a FPV recording must begin at a minimum altitude of 500 feet AGL.

Preliminary analysis indicated that the minimum target apogee of 3000 feet AGL will be very difficult to reach. A particularly significant requirement of the system is that the total impulse of the rocket motor must not exceed that of a I-class motor, which is very low for the rockets targeting an apogee above 2500 feet AGL. However, after subsequent analysis of custom, lightweight rocket designs indicated that a target apogee of 4000 feet AGL was theoretically feasible, a high-power rocket was designed to exceed the original customer requirement.

This report outlines the design implementation of a high-power rocket utilizing a guided recovery system and will be presented as follows:

- Project Objectives
- Assessment of Relevant Technologies
- Professional and Societal Implications
- System Requirements
- Concept Development
- Design Analysis
- Final Design Overview
- System Evaluation
- Significant Accomplishments
- Conclusions

# 2. Project Objectives & Scope

The short-term objectives for the final semester of this project are related to each intermediate milestone assignments. The short-term project objectives included:

- Refining system requirements
- Performing comprehensive analysis
- Modeling design concepts

The long-term objectives aided in the completion of this report and are outlined in expected chronological order.

- Modeling a prototype design
- Purchasing materials for prototyping
- Manufacturing and testing a prototype
- Refining a final design based on prototype testing
- Purchasing materials for a final competition rocket
- Manufacturing a final rocket system for competition

# 3. Assessment of Relevant Existing Technologies and Standards

## 3.1. Propulsion Systems

In high-power rocketry, the propulsion technology is well established and mature and is broken down into classes based on total impulse as described in Table 1 below. These classes are well established and have allowed rocketeers to make informed decisions on what motor they should buy or are allowed to buy. To be considered a “High Power” motor, the motor must have no less than 320 Newton-seconds of total impulse, or a low H-class motor. Below this in classes A-G are low power motors which can commonly be found in hobby store rockets and mid-power motors.

In the high-power class, there are 3 levels that correspond to certifications given to rocketeers upon completing certain requirements. The awarding of the Level 1, 2, and 3 certifications are governed and executed by the two largest amateur rocket organizations, the National Association of Rocketry (NAR) and the Tripoli Rocketry Association (TRA). For the UCF Student Launch Initiative, we are limited to a Level 1 motor thus allowing us up to 640 Ns of total impulse, or an I-class motor [1].

**Table 1: Rocket Motor Classifications [2]**

| Impulse Class | Impulse Limit (Ns) | Category |
|---|---|---|
| **A** | 2.5 | Low Power |
| **B** | 5 | |
| **C** | 10 | |
| **D** | 20 | |
| **E** | 40 | Mid Power |
| **F** | 80 | |
| **G** | 160 | |
| **H** | 320 | High Power (Level 1) |
| **I** | 640 | |
| **J** | 1280 | High Power (Level 2) |
| **K** | 2560 | |
| **L** | 5120 | |
| **M** | 10240 | High Power (Level 3) |
| **N** | 20480 | |
| **O** | 40960 | |

Motors also come in two primary use cases, single use and reloadable. Single use motors need to be completely replaced after each flight but incur a lower up-front cost.

Reloadable motors have the benefit of being able to reuse the motor hardware after each flight but incur a higher upfront cost for the more robust and durable hardware. For someone expecting to fly many flights with the same hardware, this upfront cost is well worth it and quickly surpasses the cost per flight of a single use motor.

## 3.2. Rocket Flight Stability

Stability is one of, if not the most important consideration when designing a high-powered rocket. The purpose of this deliverable is to define stability, outline how it is measured, and how to enhance the stability of an unguided high-power rocket. According to the National Association of Rocketry (NAR), imperfections in manufacturing, gusts of wind, or off-center thrust will impart rotating forces on the rocket during flight [3]. Stability is the ability of a rocket to restore itself to its original orientation (trim condition) after a perturbation [4].

Rocket stability is measured by the distance between the center of gravity and the center of pressure. Ideally this distance is about one or two times the length of the diameter of the body tube. This will provide the aerodynamic forces present at the center of pressure a long enough moment arm to counteract any rotational forces acting around the center of gravity in flight. Placing fins on the back of the rocket moves the center of pressure towards that end, and the larger the fins, the further back the center of pressure will move since the aerodynamic forces on that end are greater than the nose. The location of the center of gravity can be manipulated by adding or removing weight from the nose of the rocket.

The thrust to weight ratio of a rocket is the ratio of the force of the weight of the rocket on the pad to the force generated by the thrust of its motor. For the sake of stability, the thrust to weight ratio is not as important to consider as the speed of the rocket when it exits the rail at launch. The thrust to weight ratio should be high enough that by the time the rocket leaves the support rail and is in unguided flight it has accelerated to a speed where the aerodynamic forces on the rocket are strong enough to create the restoring force that will act against any disturbances.

## 3.3. Flight Computer

Flight computers consist of multiple sensors vital to both the flight and recovery of the rocket. The sensors are used during the flight of the rocket including barometer/altimeter, accelerometer, gyroscope, and timers. These timers are used with sensors to make sure certain events happen and guarantee a safe flight. For example, the altimeter watches for apogee to be met before deploying a drogue chute. They can log and transmit certain data for a flight such as the altitude of apogee needed in this challenge. The microcontroller takes in all the data from the sensors and will be used to program the recovery system. It will collect data from an altimeter and GPS. This will then be read and used by the program to recover the rocket. Altimeters work by measuring the ambient air pressure around the rocket as well as temperature to determine its altitude using the barometric pressure equation.

**Equation 1: Relation of Static Atmospheric Pressure and Altitude**

$$p = p_0\left(1 - \frac{Lh}{T_0}\right)^{\frac{gM}{RL}} = p_0\left(1 - \frac{gh}{c_p T_0}\right)^{\frac{c_p M}{R_0}} \approx p_0 exp\left(-\frac{ghM}{T_0 R_0}\right)$$

**Table 2: Constants for Pressure-Altitude Relation**

| Parameter | Value |
|---|---|
| $p_0$ | 101325 Pa |
| L | 0.00976 K/m |

| | |
|---|---|
| $c_p$ | 1004.68506 J/kgK |
| $T_0$ | 288.16 K |
| g | 9.80665 m/s2 |
| M | 0.02896968 kg/mol |
| $R_0$ | 8.314462618 J/molK |

Another necessary sensor is an IMU which is used basically as a compass for the recovery system. It includes a gyroscope, accelerometer, and magnetometer. Gyroscopes find the angular rate to determine if it is tilted by using a mass seeking equilibrium. Accelerometers measure the inertial acceleration and magnetometers measure the strength of magnetic fields. The last sensor to consider is GPS which uses satellites to triangulate either a 2D or 3D position including altitude. The camera interfaces with a raspberry pi to process and store the footage. The camera will record the entire flight to view it later and transmit it to the ground.

## 3.4. Automated Recovery Systems

The recovery subsystem of a high-power rocket is integral to overall system performance. In amateur rocketry, developing a functional recovery system is an often-overlooked component of rocket design. Organizations such as the National Association of Rocketry (NAR) and the Tripoli Rocketry Association (TRA) maintain guidelines for recovery systems in their safety guidelines. While passive technologies (e.g., tumble recovery or aerobraking) exist to serve this purpose, active recovery systems are nearly required for high-power rocketry.

NAR and TRA define active recovery systems as the deployment of a primary recovery device that changes the physical configuration and drastically reduces the vertical descent rate of a high-power rocket. Active recovery methods include parachutes, streamers, helicopter devices, and RC control. These systems significantly reduce the vertical descent rate of rockets, immensely improving ground safety. However, the lateral movement of a rocket is highly variable in most common active recovery systems. The uncertain downward trajectory of high-power rockets using active systems necessitates the design of rockets utilizing novel recovery methods. These improved recovery systems use human or autonomous descent control to land at a specified target.

Autonomous recovery systems provide some significant benefits over RC systems. Eliminating the requirement for an on-board antenna presents greater material flexibility and avoids additional test flights to verify radio transmission range. On the other hand, autonomous flight control poses separate design challenges, notably the development of a closed-loop control system. Software must be developed independently for specific systems, but pre-written software libraries exist for COTS flight computer components.

High-power rocketry projects from other universities have utilized custom C++ classes paired with GPS software libraries to intuitively track ground-level altitude, latitude, and longitude. Interfacing an array of sensors, such as GPS, barometric pressure sensors, and accelerometers, with a proportional-integral-derivative (PID) control system can effectively guide the main parachute during recovery.

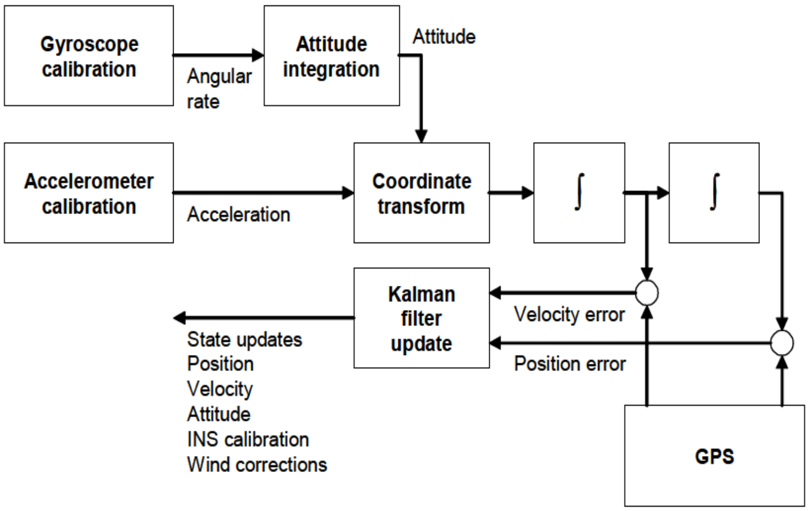


**Figure 1: GNC Algorithm for an Automated Parachute Recovery System**

## 3.5. Recovery System Deployment

A parachute is the primary method used to safely land a high-powered rocket. Lower-power rockets can get away with only using a single, main parachute since the wind will not affect the rocket for as long at a lower altitude. However, many high-powered rockets will use two parachutes, a drogue parachute and main parachute. When a rocket uses these multiple parachutes, it is called dual deployment. The biggest advantage of a dual deployment system is the rocket will not drift as far, which is helpful when attempting to land on a target.

Many high-power rockets utilize ejection charges to separate the rocket components and release the internal recovery parachutes. The most common type of ejection charges used in high powered rockets is black powder. The black powder charge can be built into the rocket motor, or it can be packed into an electronic system that can ignite at a specified parameter. Recovery systems can have multiple ejection charges if a dual deployment system is in use to release a drogue parachute before the main parachute.

The shock cord is an elastic material inside the rocket that holds together the pieces of body tube separated from the ejection charge. Typical materials for the shock cord are elastic, rubber, nylon, and Kevlar. Braided Kevlar shock cord has a high tensile strength for its weight and is heat resistant. However, the cord stretches less than elastic cords and can result in a larger force acting on the rocket.

### 3.6. Nose Cone

The nose cone is an essential part of a rocket as it is the leading section of the vehicle. The main function of the nose cone is to minimize the aerodynamic drag on the flight vehicle. The geometry of the nose cone greatly impacts the capability of the rocket. There are two main types of drag effecting a subsonic rocket, pressure drag and skin friction drag. Pressure drag is dependent on the cross-sectional area and shape of the object. To reduce the pressure drag, the cross sectional area must be made smaller. Skin friction drag is caused by a fluid as it moves over an object, in this case, the fluid is air. A technical study composed by NASA in 1974 found the skin friction component of drag is the largest contributor to the total drag in subsonic flows at about 45%. To reduce skin friction drag the surface of the object can be smoothed to delay the formation of turbulent boundary layers.

## 4. Professional and Societal Considerations

A rocket capable of returning to a specified launch site has become a desirable design objective for commercial spacecraft manufacturers in the past decade. Rocket boosters on launch vehicles such as Falcon 9 and, recently, Starship seek to reduce the cost of spaceflight by eliminating the need to produce multiple stages for every test. Furthermore, reusable rocket boosters reduce the time required between flights, which expedites full-system analysis and testing.

Reusable rocketry components in commercial applications also reduce the amount of space debris produced during flight. The low Earth and geostationary orbitals are currently at risk from overcrowding of space debris due to their importance in satellite communication. Space debris also increases the likelihood of collisions with spacecraft and orbital vehicles, potentially damaging critical systems that society currently relies on.

Designing a small-scale guided recovery system provides the team with the necessary skills to apply the knowledge of reusable rocketry systems in professional applications.

## 5. System Requirements and Design Constraints

The users of the system conceptualized in this report include HPR hobbyists and enthusiasts who would benefit from an improved recovery system. Since these users may conduct multiple flights with the same rocket, a high-power rocket with a reliable and predictable recovery system would significantly reduce catastrophic failures between flights.

On a conceptual level, commercial space industry manufacturers would benefit greatly from the rockets with steerable descent systems because individual rockets modules could be reused between missions. Thus, representatives from Aerojet Rocketdyne have chosen to sponsor this project to prepare college students with the skill necessary to aid the development a commercially viable autonomous rocket recovery system. For this project to maintain the goal of creating a guided descent control system, the relevant sponsors have outlined to following list of needs. The key user needs considered in this report are the minimum apogee, propulsion hardware limit, and expected landing accuracy.

**Table 3: Prioritized List of User Needs**

| Needs |
|---|
| **Vehicle apogee SHALL be less than 3000ft Above Ground Level (AGL).** |
| **Vehicle SHALL NOT use any active guidance during ascent** |
| **Vehicle SHALL NOT use any combustion-based descent control** |
| **Active descent components SHALL be stowed in Vehicle Recovery Subsystem (VRS) until deployment of VRS** |
| **If a human controlled active descent method is used, the Vehicle SHALL be "aimed" via the on-board First-Person Video (FPV)** |
| **The Active descent control thrust-to-weight ratio SHALL be less than 0.9** |
| **Vehicle SHALL transmit FPV NLT 500 AGL** |
| **Vehicle hardware SHALL stay below Level 2 NAR certification** |
| **Vehicle SHALL land within 10m diameter landing zone.** |

# 6. System Concept Development

## 6.1. Propulsion

To choose a motor, the primary requirement of concern was the 3,000ft apogee requirement since the motor was the main component responsible of achieving this goal. In pursuit of satisfying the requirements a phased approach was followed to explore the HPR trade space and identify motors for further study utilizing an idealized model to simulate motor performance, then utilize a more accurate model to simulate the motors of interest to determine a ranked list of motors, then finally to iterate the assumed geometric properties of within the model to produce a volume of results to cover a range of geometric combinations thus establishing a well-defined performance expectation

### *6.1.1. Initial Propulsion Analysis*

The initial propulsion analysis utilized an idealized model described in 7.1.1 to establish where in the overall trade space of HPR motors our motor solution could be found. As such, a variety of motors were chosen to represent the high-power impulse classes (H & I) as well as a G-class motor to prove our solution needed to be a high-power motor. The results of the initial analysis are summarized in Table 4 below and very clearly show our motor solution needed to be an I-class motor, and further, a high-end I class motor.

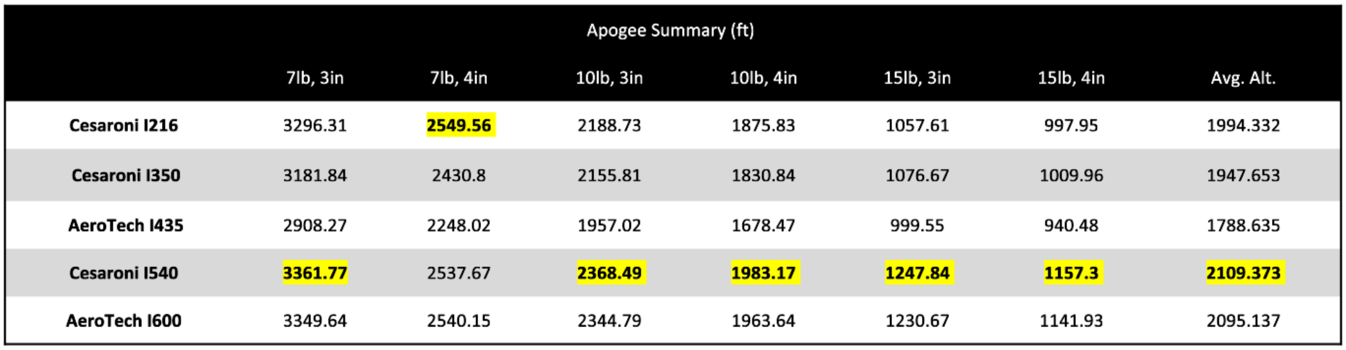
**Table 4: Initial Propulsion Analysis Results**

| Motor | Burnout Velocity (ft/s) | Burnout Altitude (ft) | Unpowered Ascent (ft) | Apogee Altitude (ft) |
|---|---|---|---|---|
| **Cesaroni G50** | 13 | 18 | 3 | 21 |
| **AeroTech H178** | 169 | 142 | 443 | 585 |
| **Cesaroni I125** | 377 | 538 | 2205 | 2743 |
| **Cesaroni I540** | 447 | 265 | 3111 | 3375 |

### *6.1.2. Secondary Propulsion Analysis*

The second iteration of the propulsion, described in Section 7.1.2, utilized a more accurate model and varied geometric/mass properties to simulate a series of selected high-power rocket motors to identify a ranked list of motors to use. The results of this analysis are shown in Table 5 below with the Cesaroni I540 and Aerotech I600 coming in as the two best choices for our rocket based primarily on their ability to reach the highest apogee.

**Table 5: Secondary Propulsion Analysis Results**



| | Apogee Summary (ft) | | | | | | |
|---|---|---|---|---|---|---|---|
| | 7lb, 3in | 7lb, 4in | 10lb, 3in | 10lb, 4in | 15lb, 3in | 15lb, 4in | Avg. Alt. |
| **Cesaroni I216** | 3296.31 | **2549.56** | 2188.73 | 1875.83 | 1057.61 | 997.95 | 1994.332 |
| **Cesaroni I350** | 3181.84 | 2430.8 | 2155.81 | 1830.84 | 1076.67 | 1009.96 | 1947.653 |
| **AeroTech I435** | 2908.27 | 2248.02 | 1957.02 | 1678.47 | 999.55 | 940.48 | 1788.635 |
| **Cesaroni I540** | **3361.77** | 2537.67 | **2368.49** | **1983.17** | **1247.84** | **1157.3** | **2109.373** |
| **AeroTech I600** | 3349.64 | 2540.15 | 2344.79 | 1963.64 | 1230.67 | 1141.93 | 2095.137 |

### *6.1.3. Final Propulsion Analysis*

The final iteration of the propulsion analysis, described in Section 7.1.3, embedded the Secondary Analysis model in a series of loops to allow for the assumed properties to be varied and thus provide a range of performance possibilities for a large variety of combination of total rocket mass, drag coefficient, and motor. These results were then plotted against the Cesaroni I540 since it was identified as the “most ideal” motor. It was during this phase the final motor was selected as the Cesaroni I470 based on availability and price compared to the Cesaroni I540 being identified based on its altitude at apogee. The results of runs comparing the Cesaroni I540 (blue data) vs AeroTech I600 (green data) vs Cesaroni I470 (red data) Apogee, time to Apogee, and max velocity are shown below in **Error! Reference source not found.**, **Error! Reference source not found.**, and Figure 4: I540 (blue) vs I600 (green) vs I470 (red) Max Velocity respectively. In choosing the Cesaroni I470, we lost about 12% in apogee altitude and about 15% in max velocity, however these performance losses were made up for by the motor being locally available allowing us to avoid excessive Hazardous Material shipping charges and shipping delays due the inherent risk in shipping rocket motors.

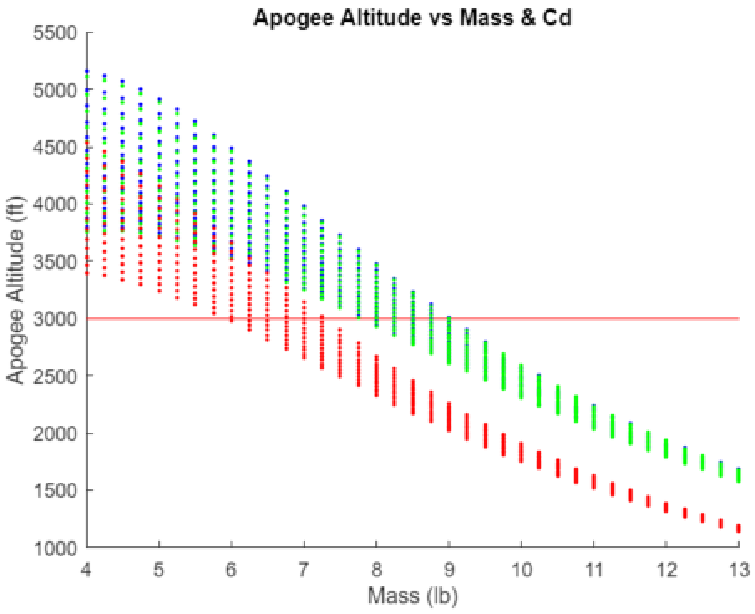


Figure 2: I540 (blue) vs I600 (green) vs I470 (red) Apogee Altitude

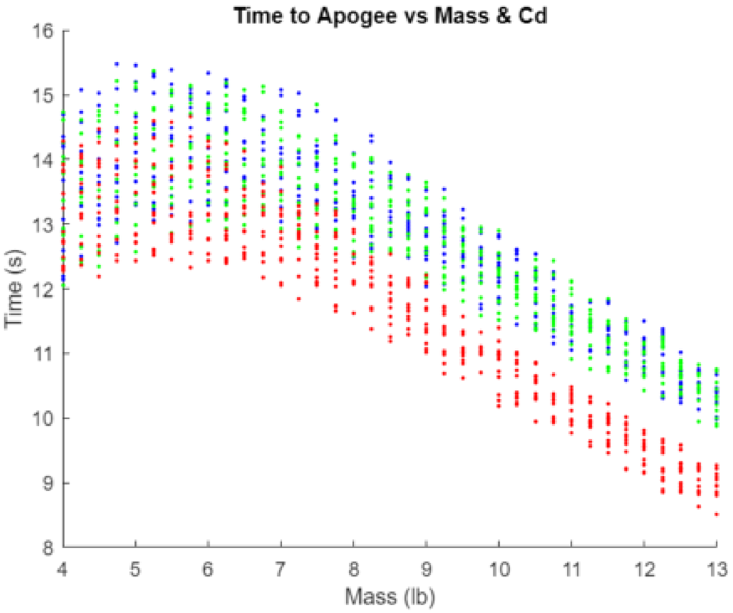


Figure 3: I540 (blue) vs I600 (green) vs I470 (red) Time to Apogee

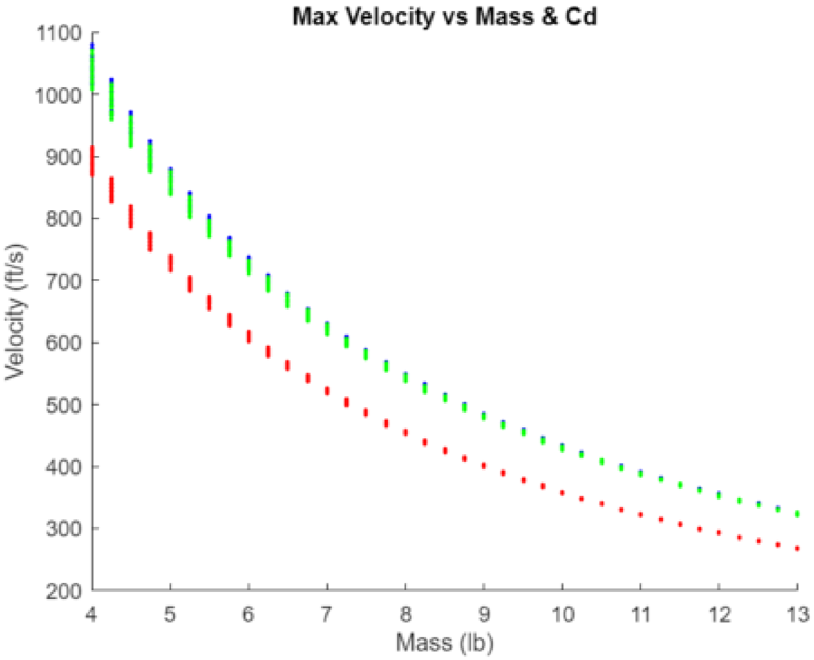


**Figure 4: I540 (blue) vs I600 (green) vs I470 (red) Max Velocity**

It can be seen that the Cesaroni I540 and AeroTech I600 are a very close match, and both would have appropriate choices. However, due to the aforementioned price considerations in shipping rocket motors compared to buying from a local dealer, the Cesaroni I470 was purchased as the motor for flight and the Cesaroni Pro-38X external motor hardware was sourced from our advisor.

## 6.2. Guidance, Navigation, and Control

The Guidance, Navigation, and Control (GNC) subsystem design was primarily driven by the landing accuracy requirement and was constrained by the capabilities and limitations of the flight computer. The GNC algorithms were designed and developed later in the lifecycle of this project causing the overall design to be reduced to control a one-dimensional scheme based purely on flight heading. A diagram of the inputs and outputs of the system is shown **Error! Reference source not found.** in below.

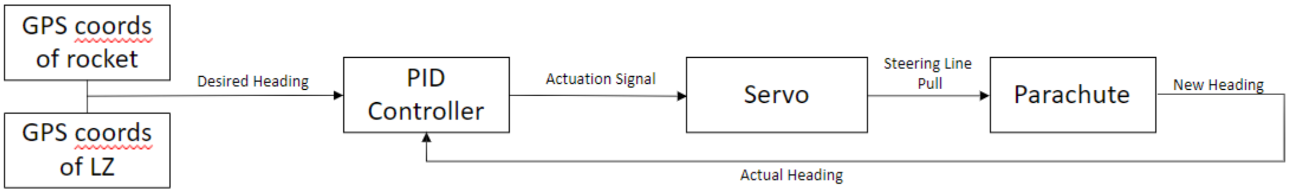


**Figure 5: GNC Input/Output Loop**

### *6.2.1. Guidance*

In this design, guidance was established based on a calculated "desired" bearing from the rocket to the target via GPS coordinates. The GPS coordinates of the center of the landing zone were to be input during preparation at the launch area and the real-time coordinates of the rocket were to be queried from the onboard GPS receiver. These 2 points would then be input into Equation 2 below to calculate a bearing to the landing zone [5]. This bearing would then be used as the setpoint for the control loop.

**Equation 2: GPS Coordinates to Bearing**

$$\gamma = \tan^{-1}(y, x)$$

$$y = \sin(long_{LZ} - long_R) * \cos(lat_{LZ})$$

$$x = \cos(lat_R) * \sin(lat_{LZ}) - \sin(\mathrm{lat}_\mathrm{R}) * \cos(lat_{LZ}) * \cos(long_{LZ} - long_R)$$

### *6.2.2. Navigation*

Navigation would be primarily handled by using the onboard GPS receiver to continually feed the rocket's current location and calculate the current heading for comparison to the desired bearing in the control loop. Since the GNC environment was chosen to be dimensionally reduced, only the angle from North of the velocity vector was of concern and the magnitude was neglected.

### *6.2.3. Control*

The control scheme selected was a closed loop PID control scheme. Due to the complexity of the system dynamics and transfer functions, PID control gave us the best chance of arriving at stable and effective control solution. The control loop consisted of the setpoint established by via GPS, the PID controller, the Servo motor, and the Parachute. Simulink was used as the primary method to model and test the control loop to establish stability and ability and is pictured in Figure 6 below.

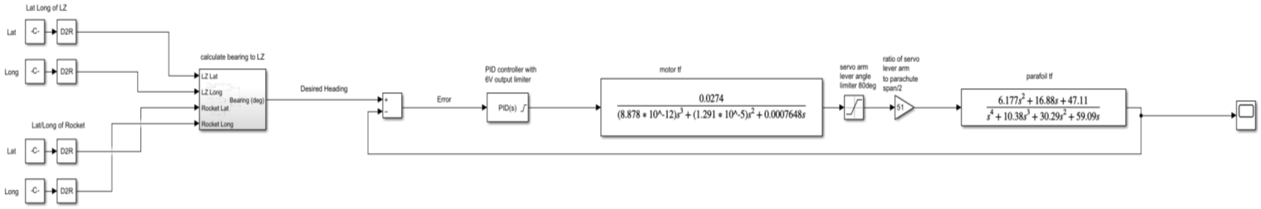


**Figure 6: Simulink Control Loop**

**Equation 3: Servo Motor Transfer Function**

$$tf_{servo} = \frac{0.0274}{(8.878 * 10^{-12})s^3 + (1.291 * 10^{-5})s^2 + 0.0007648s}$$

**Equation 4: Parafoil Transfer Function**

$$tf_{parafoil} = \frac{6.177s^2 + 16.88s + 47.11}{s^4 + 10.38s^3 + 30.29s^2 + 59.09s}$$

The servo motor transfer function, Equation 3, was sourced via the University of Michigan [6] and the transfer function for the parachute, Equation 4, was based on parafoil lateral heading dynamics to build a transfer function. [7] A transfer function for a parafoil was chosen since there was not a good equivalent transfer function available based on a Rogallo wing, as well as an assumption that a parafoil and Rogallo wing design are similar enough to allow for a good initial model with the understanding that a testing regime and iterative adjustments would be required to better represent a Rogallo wing's dynamics. Additionally, an output limiter was placed in the controller at +/- 6 since the servo motor had a maximum operating voltage of 6V and another limiter was placed after the servo to model the maximum degrees of the turn for the servo arm at +/- 50°.

The first system to be analyzed was a continuous system in the S-plane. This provided a good opportunity to set a system baseline based on previous coursework and knowledge even though the system would need to be at least partially discretized to accurately model the influence of the digital flight computer. The PID coefficients were chosen by first utilizing the embedded PID Tuner application to create an initial tune, then manually iterating the values to produce an optimal response. The final PID tunings and system parameters are shown below in Table 6 as well as a sample response to a 180 degree turn in Figure 7.

**Table 6: Continuous System Parameters**

| System Parameters | Value |
|---|---|
| **P** | 2718 |
| **I** | 3444 |
| **D** | 0.0022 |
| **N** | 15 |
| **Rise Time** | 0.491s |
| **Settling Time** | 123s (2%) / ~2.5s (5%) |
| **Overshoot** | 12.4% |
| **Gain Margin** | ~5.5dB @ 0.295 rad/s |
| **Phase Margin** | 57.9 deg @ 2.78 rad/s |

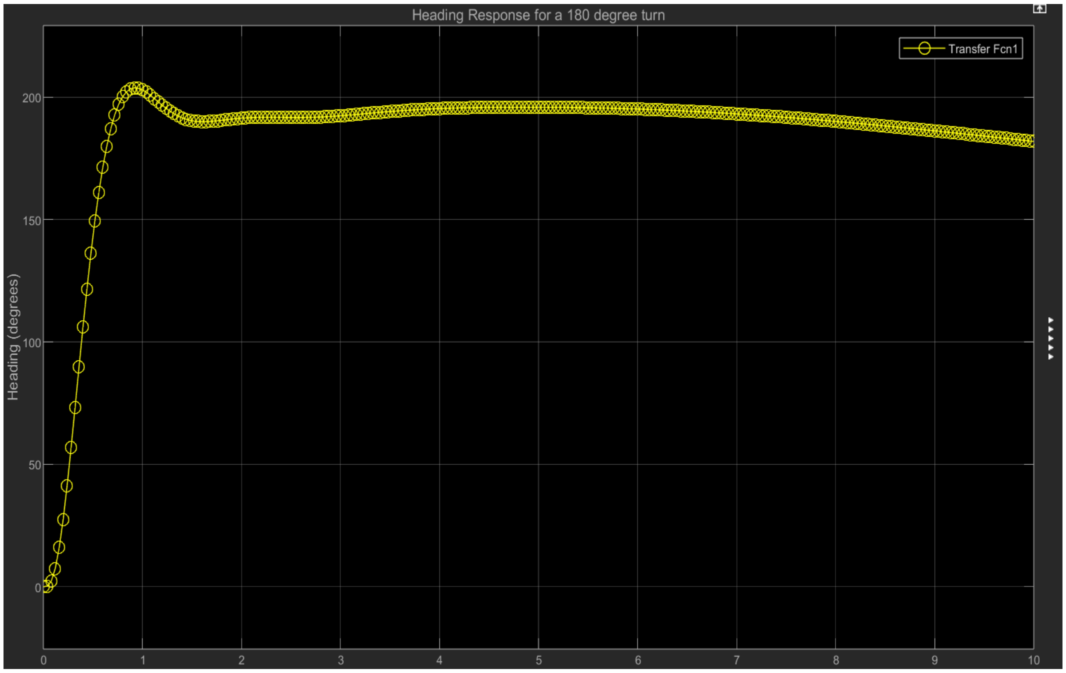


**Figure 7: Continuous System 180° Turn Time Response**

In an effort to discretize the system, the PID parameters, the servo transfer function, and parachute transfer function were all transformed from the S-plane to the Z-plane via the Tustin Method in MATLAB. These parameters and transfer functions were then rebuilt in Simulink to follow a similar process as the continuous system. This process was unsuccessful within our time constraints due the nature of Z-transformations inherently increasing instability in the system causing the overall system to become unstable an oscillatory significantly beyond the bounds of our system. A successful discretization of the system would present a great opportunity for follow-on work to enhance the system.

## 6.3. Airframe

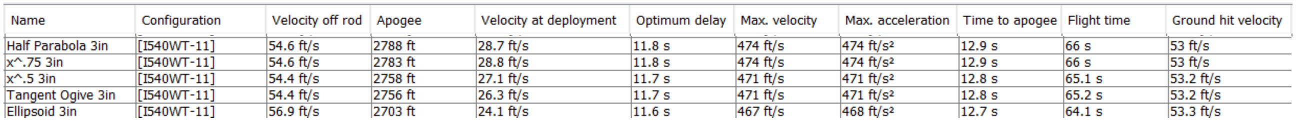
### *6.3.1. Nose Cone*

The nose cone is one of the most important aerodynamic aspects of a rocket for drag resistance. With a goal of reaching 3,000ft apogee, the nose cone needed to have the lowest drag possible. One of the main design choices needed to be made was to make the nose cone with PLA 3D printing or purchase one commercially. As seen in Table 7, the 3D printed nose cone was the better choice due to the geometry being fully customizable while also being the same cost or cheaper than the commercial counterparts.

| Criteria | Importance | 3D Printed | Commercially Bought |
|---|---|---|---|
| **Cost** | 25% | 4 | 4 |
| **Weight** | 25% | 2 | 4 |
| **Customizable Design** | 45% | 5 | 2 |
| **Ability to Simulate** | 5% | 5 | 1 |
|  | 100% | 4 | 2.95 |

**Table 7: Weighted Nose Cone Manufacturing Matrix**

With the customizable aspect of the 3D printed nose cone, two programs were used for simulations. OpenRocket was used initially to estimate the altitude the rocket will reach, and secondly, a more detailed simulation with ANSYS Fluent to calculate the drag force, drag coefficient, and Mach number. The results of the OpenRocket simulation with a generic rocket can be seen below in Table 8.



| Name | Configuration | Velocity off rod | Apogee | Velocity at deployment | Optimum delay | Max. velocity | Max. acceleration | Time to apogee | Flight time | Ground hit velocity |
|---|---|---|---|---|---|---|---|---|---|---|
| Half Parabola 3in | [I540WT-11] | 54.6 ft/s | 2788 ft | 28.7 ft/s | 11.8 s | 474 ft/s | 474 ft/s² | 12.9 s | 66 s | 53 ft/s |
| x^.75 3in | [I540WT-11] | 54.6 ft/s | 2783 ft | 28.8 ft/s | 11.8 s | 474 ft/s | 474 ft/s² | 12.9 s | 66 s | 53 ft/s |
| x^.5 3in | [I540WT-11] | 54.4 ft/s | 2758 ft | 27.1 ft/s | 11.7 s | 471 ft/s | 471 ft/s² | 12.8 s | 65.1 s | 53.2 ft/s |
| Tangent Ogive 3in | [I540WT-11] | 54.4 ft/s | 2756 ft | 26.3 ft/s | 11.7 s | 471 ft/s | 471 ft/s² | 12.8 s | 65.2 s | 53.2 ft/s |
| Ellipsoid 3in | [I540WT-11] | 56.9 ft/s | 2703 ft | 24.1 ft/s | 11.6 s | 467 ft/s | 468 ft/s² | 12.7 s | 64.1 s | 53.3 ft/s |

**Table 8: OpenRocket Initial Simulation**

The half parabola nose cone geometry reached the highest altitude with this simulation of all variables kept constant except for the geometry of the nose cone. The x^.75 power series geometry was a very close second.

With the initial simulation complete, a more detailed analysis was completed with ANSYS Fluent. A 2D model of the nose cone was created in SOLIDWORKS and imported into

ANSYS Fluent to perform a computational fluid analysis. The result of this analysis is shown in Table 9 below.

| L = 12in | D = 3in | Cd | Cd Cone + Body | D (N) |
|---|---|---|---|---|
| | Ellipsoid | 0.0337 | 0.0347 | 631.03 |
| | Tangent Ogive | 0.0289 | 0.0299 | 542.81 |
| | Half Parabola | 0.0260 | 0.0269 | 487.85 |
| | x^.5 | 0.0267 | 0.0276 | 500.90 |
| | x^.75 | 0.0257 | 0.0266 | 481.50 |

**Table 9: ANSYS Fluent Nose Cone Analysis**

This secondary analysis shows the half parabola does not have the lowest drag coefficient or produce the least drag force, but the x^.75 power series does. It was determined the altitude of the rocket was more important than the drag coefficient.

| Criteria | Importance | Ellipsoid | Tangent Ogive | Half Parabola | x^.5 Power | x^.75 Power |
|---|---|---|---|---|---|---|
| **Max Altitude** | 50% | 1 | 2 | 5 | 3 | 4 |
| **Drag Coefficient** | 45% | 1 | 2 | 4 | 3 | 5 |
| **Ease of Manufacturing** | 5% | 5 | 5 | 5 | 5 | 5 |
| | 100% | 1.2 | 2.15 | 4.55 | 3.1 | 4.5 |

**Table 10: Nose Cone Geometry Matrix**

With the decision to create the nose cone with 3D printing and a geometry selected, a finalized model could then be created. The simulation process can be found in section 7 and the geometry model in section 8.

# 7. Design Analysis

## 7.1. Propulsion

### *7.1.1. Initial Propulsion Analysis*

In an effort to explore the trade space of possible HPR motors to use, an initial idealized propulsion analysis was conducted. This initial analysis was conducted using MATLAB based on Tsiolkovsky's Rocket Equation and basic ballistic motion physics to produce an estimate of a motor's basic flight characteristics such as apogee, time to apogee, and max velocity while making some key geometric assumptions such that the total mass was 10lbs, the flight was perfectly vertical, and no effects by drag.

**Equation 5: Tsiolkovsky's Rocket Equation**

$$\Delta V = \frac{F_{thrust}}{\dot{m}} \ln \left(\frac{m_i}{m_i - m_f}\right)$$

**Equation 6: Ballistic Motion Height Equation**

$$\Delta y = \frac{V_i^2}{2g}$$

### *7.1.2. Secondary Propulsion Analysis*

The secondary motor performance analysis utilized a Simulink model and MATLAB code to increase the accuracy and realism of the simulation. With multiple candidate motors identified, the thrust curves, propellant masses, and burn times were imported into MATLAB and fed into the Simulink. The Simulink model, shown in Figure 8 below, utilized lookup tables to model the atmospheric density, mass flow rate, and thrust curve for each rocket, feedback loops including altitude and velocity for the drag calculation, and a double integration block to continuously compute the velocity and altitude based on a 2nd order differential acceleration equation derived from a vertical force balance on the rocket shown in Equation 7 below.

**Equation 7: Rocket Acceleration**

$$\ddot{x} = \frac{F_{thrust}}{m} - g - \left(\frac{\rho \dot{x} S C_d}{2m}\right)$$

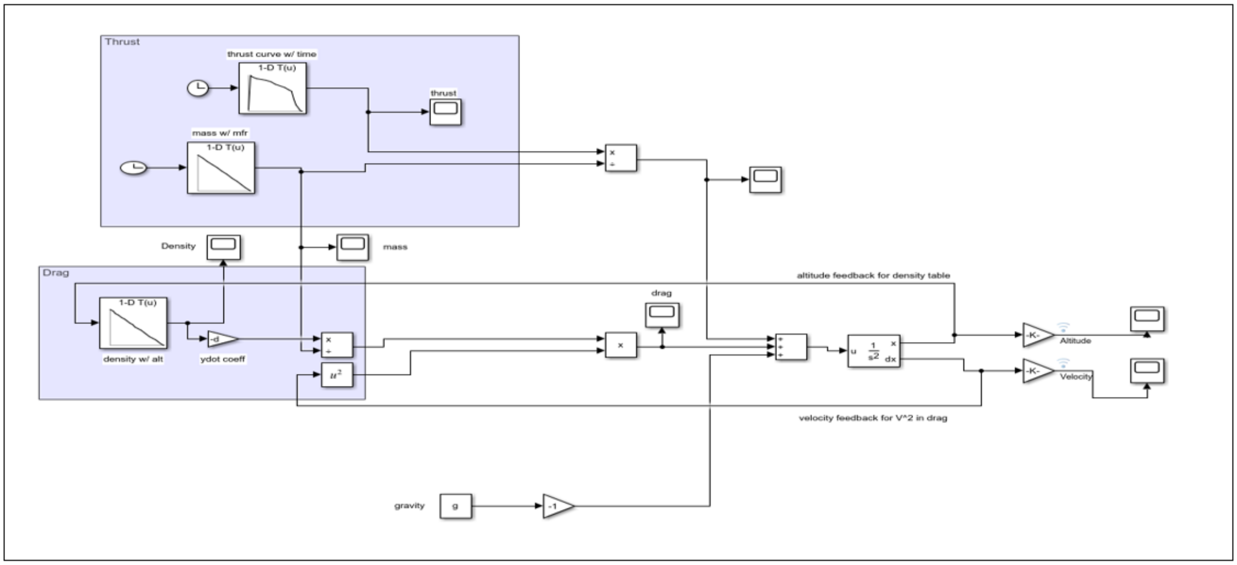


**Figure 8: Simulink Propulsion Model**

The model was then verified using provided flight data from other HPR flights, of which the results are summarized in Table 11, to establish the validity of the model.

**Table 11- Propulsion Model Verification Results**

| Motor | Model Apogee (ft) | Actual Apogee (ft) | Percent Error |
|---|---|---|---|
| **I125** | 2718 | 2745 | 1% |
| **J381** | 3444 | 3309 | 4% |

### 7.1.3. Final Propulsion Analysis

The final iteration of the propulsion analysis utilized the same model described above in 7.1.2, however the accompanying MATLAB code was edited to place the Simulink model within a series of “for” loops to allow for continuous iterations over multiple chosen parameters including total mass and drag coefficient for each finalist motor. The MATLAB code is presented in Appendix H: MATLAB Code. The total mass was allowed to vary from 4lb to 13lb with 0.25lb increments to present 37 mass points and the drag coefficient was allowed to vary from 0.5 to 0.8 with 0.025 increments to present 13 drag points. This resulted in 481 data points per motor and 1,443 total data points presented to fully encompass the realm of possibilities of our rocket. This technique allowed for a very detailed look into the possible performance of a motor, and comparison of multiple motors, while our actual rocket design was still being finalized.

## 7.2. Airframe

### 7.2.1. Nose Cone

To create the nose cone geometry that can achieve the highest altitude, analysis needed to be done. Seven typically used geometries were selected to run simulations to determine the best for this case. OpenRocket was used as an initial tool with a generic rocket created and all variables kept constant, besides for the nose cone. In this simulation, the half parabola nose cone geometry reached the highest altitude.

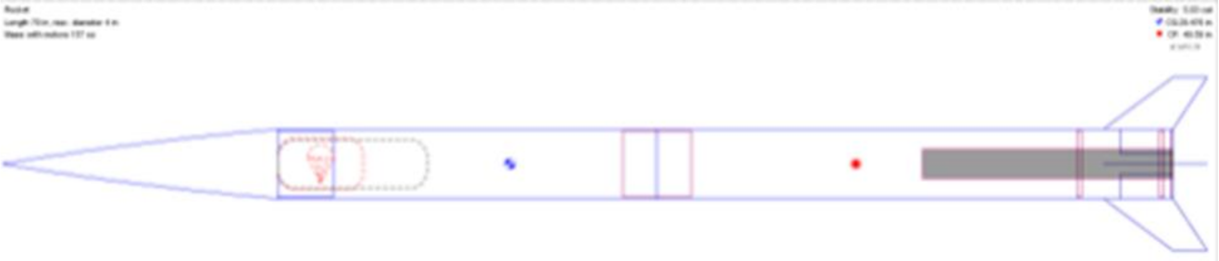


**Figure 9: OpenRocket Generic Rocket**

A 2D simulation was of the top five performing nose cone geometries were then run using ANSYS Fluent to confirm if the results from the OpenRocket simulation are correct. SolidWorks was used to model the geometry of the nose cones and then imported into ANSYS Fluent for simulation. Part of the body was included at the end of the nose cone to simulate how the air will flow after it passes the end of the cone.

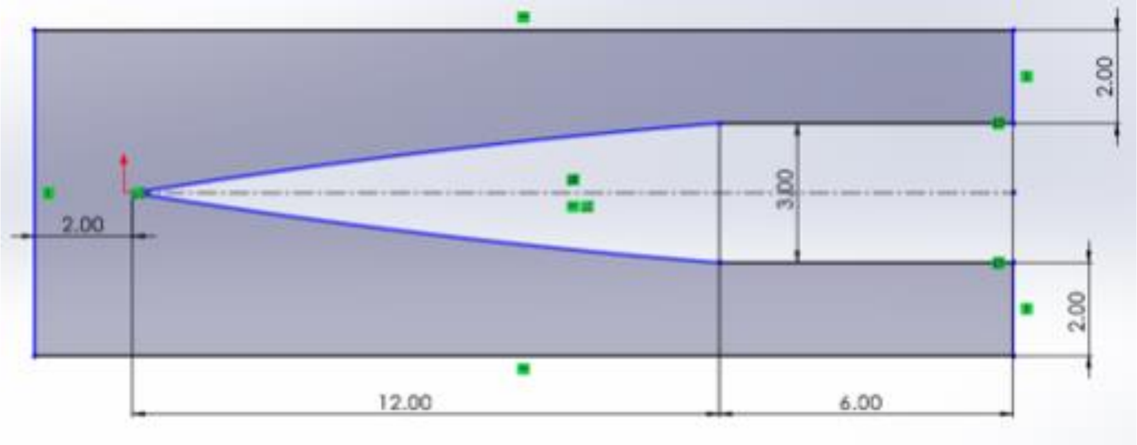


**Figure 10: 3in Half Parabola Nose Cone**

A mesh was generated from the imported geometry and an inflation layer was made surrounding the edge of the nose cone and body tube.

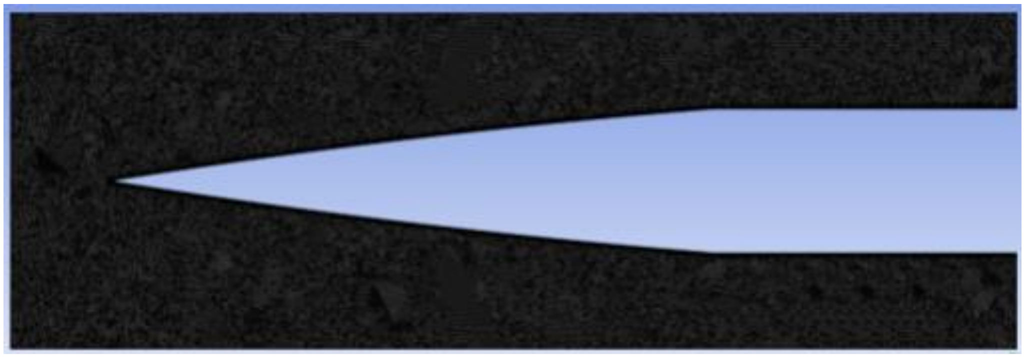


**Figure 11: Half Parabola Mesh**

The simulations were done using the worst-case initial conditions with the maximum velocity and the lowest pressure the rocket will experience during flight. After the simulations were completed, the half parabola geometry had the lowest drag coefficient and confirmed the results from the OpenRocket simulation. The Mach number contour can be seen below in Figure 12.

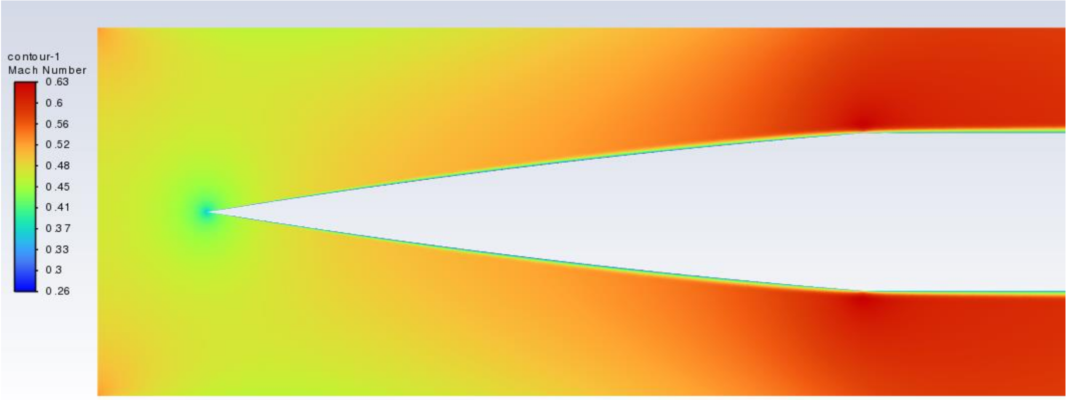


**Figure 12: Mach Number Contour**

Knowing the Mach number around the nose cone is vital to the structural integrity of the air frame. If the Mach number is above 1, shocks can form. Behind the shocks, the pressure, temperature, and density increase drastically. Specifically, the increase in temperature can be detrimental to parts of the rocket not meant to withstand much heat. In this case there are no shocks so there is no worry of the structural integrity failing in flight.

### *7.2.2. Fin Design*

Fins play a very important role in the stability of a rocket. Stability is defined by calipers, 1 caliper (cal) is equal to the diameter of the body tube. A good stability will have the center of pressure 1.5 – 2.0 cal behind the center of gravity, or 1.5 – 2 times the diameter of the body tube. Due to their location, small changes in the shape and size can have drastic effects on stability. Finding stability can be simple with the help of OpenRocket. This will tell you the location of the center of gravity and center of pressure, and stability of the rocket. Once a rocket is created in OpenRocket, the fin shape can be altered to create the necessary stability.

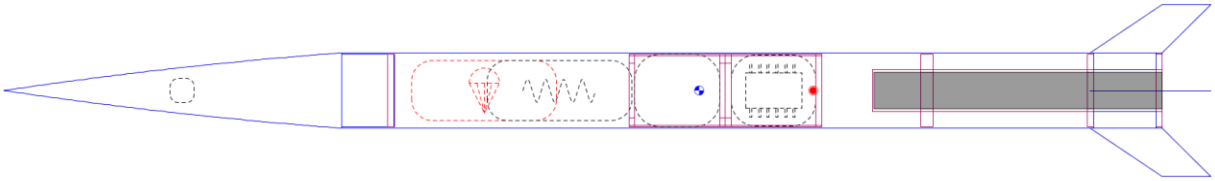


**Figure 13: OpenRocket Model**

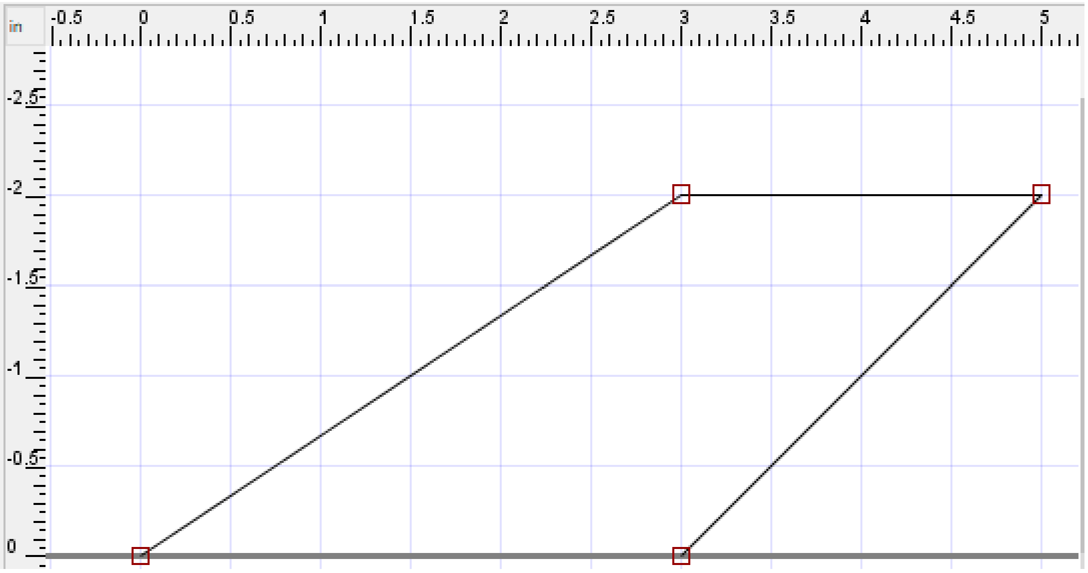
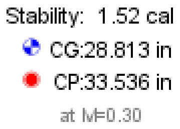


**Figure 14: OpenRocket Fin Geometry**



Stability: 1.52 cal
CG:28.813 in
CP:33.536 in
at M=0.30

**Figure 15: OpenRocket Stability**

The figures above show the model created in OpenRocket, the fin geometry creation, and the stability, center of gravity, and center of pressure. Editing the shape of the fin is as simple as dragging the points into the size and shape wanted.

## 7.3. Parachute Design

The Rogallo wing has characteristics that would prove to be the best steerable recovery system compared with circular parachutes or other recovery systems. The dynamics are more defined than other vectoring methods as well as cost efficient for testing purposes. It deals with good traction of 2.0 - 2.5 with good directional stability. Traction shows the proportion of the maximum tractive force to the normal force when the amount of available friction restricts the maximum tractive force between a body and a surface.

Quick Prototyping and Experimentation was implemented in this process to verify the sizing of the parachute needed to withstand the current rocket weight at slightly under 5lbs. For the Rogallo Parachute, our approach was to purchase a COTS parachute with a similar design and then implement drop tests at the UCF Libra parking garage with 1lb, 3lb, and 5lb weights attached through each test run. After 5-7 drop test on day 1, it was simple to deduce that the parachute was not large enough to withstand our current weight requirement thus a larger parachute needed to be manufactured. In Figure 16, the construction of the parachute is designed by Ripstop nylon fabric, fishing line with a tensile strength of 80lbs for the shroud lines, and bands to hold the leading edge in place. This was enlarged from a 32in parachute to a 50in canopy, thus increasing the drag as well as being able to perform a stable flight test with larger weight.

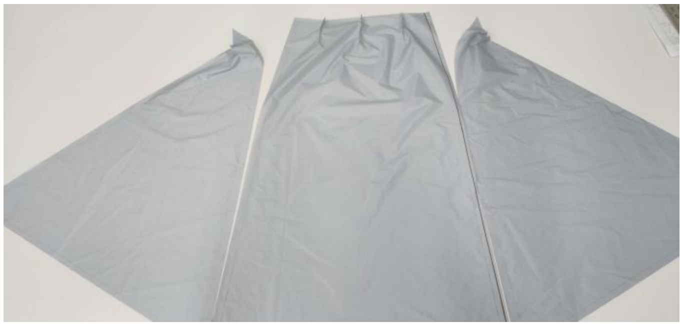


**Figure 16: Fabric Cutouts for Rogallo Paraglider**

## 7.4. Flight Computer

The prototyping for the computer consisted of things that were overlooked prior to integration with the whole system. The first issue addressed was the 5V required input voltage of the Raspberry Pi battery which differed from the 3.7V supplied by the battery. This required an addition of a voltage regulator booster board. This was not only needed due to the voltage requirements of the Raspberry Pi but also mitigated inconsistencies during battery discharge. The voltage drop across a battery decreases as the battery discharges, and the fluctuating power supplied by the battery could cause the Raspberry Pi to suddenly boot loop.

To ensure that the program made was working as designed, the logging of the sensor data circuit was built to visually show this. It consisted of a switch and LED the data would save if the switch was on and would save to a text file when it was switched off. The LED would remain on for the entire duration it was logging data as a visual representation of the code. However, it was not necessary in the final design once the code was proven to do it by itself.

Another circuit was made to protect the system from large current draw when lighting the E-match for the parachute deployment. This circuit consisted of a resistor, capacitor, and a MOSFET switch. The resistor would control the amount of current the battery would draw at any given time and the capacitor would hold the current until the MOSFET was switched on. It would then discharge the current to light the E-match without drawing more current from the system. If too much current is drawn from the system, it could cause the system to momentarily shut down and lose all data.

The calculations leading to the specifications of the circuit were used from Ohm's Law. Since the circuit was being supplied 5V and a resistor of 20 ohms was selected, no more than 250mA could be drawn from the system at once. However, after testing the E-match it was found it did not pull enough current from the system for the circuit to be necessary.

The last prototyping done was for the battery. As the system increased in size, the Raspberry Pi would boot loop because too much current was being drawn by the system to start. Which was tested by running the whole system and adding in a multimeter to test the current, which confirmed the need for a bigger battery. This led to the need for a bigger battery. To save space 2 3500mAh batteries were connected in parallel. These prototypes were necessary in creating a well-tested system ready for integration.

## 8. Final Design and Engineering Specifications

A comprehensive system design was developed from data obtained from prototype simulations and ground testing. A design of each subsystem was finalized individually before full system integration.

## 8.1. Parachute

The geometry of the Twin-Keel Rogallo Parawing parachute—more specifically, the size, shape, and angle of the two keels that run along its length—is a crucial design factor. To ensure a safe and steerable descent, the chute should be stable and avoid spinning uncontrollably.

The span of the Twin-Keel Parawing is intentionally expanded to ensure that the parachute can withstand a heavier weight with slower descent rates. During testing of the initial parachute, it was discovered a larger parachute was essential to safely slow the rocket with the estimated total weight. The increased surface area of a larger parachute would increase the drag on the system, creating more time in the air to navigate to the target location. The larger parachute is designed with 36 fishing lines with a tensile strength of 80lb. The lines are distributed to 6 lines on each outer keel and 12 lines on each row down the center canopy attachment. The outer keel has a 0.50-inch seam, and the inner keel has a 0.75-inch seam. This seam creates a more defined leading edge. Bands are also sewed on the leading edge to provide a weight for forward direction.

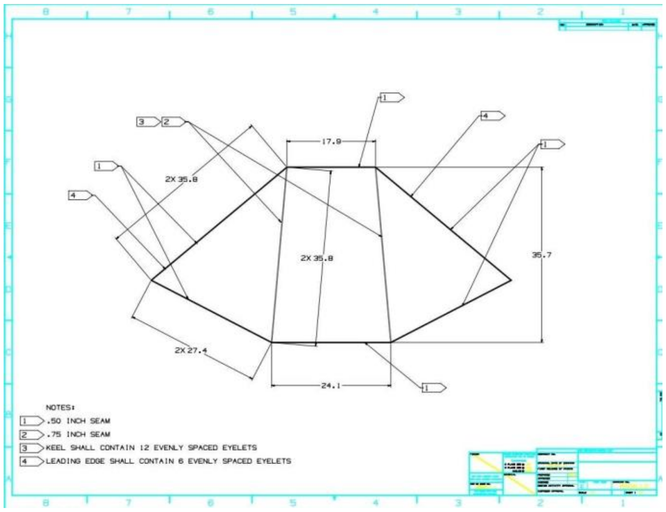

8
7
6
5
4
3
2
1
3
2
1
4
1
1
4
17.9
2X 35.8
2X 35.8
35.7
2X 27.4
24.1
1
NOTES:
1 .50 INCH SEAM
2 .75 INCH SEAM
3 KEEL SHALL CONTAIN 12 EVENLY SPACED EYELETS
4 LEADING EDGE SHALL CONTAIN 6 EVENLY SPACED EYELETS

**Figure 17: Paraglider Design Drawing**

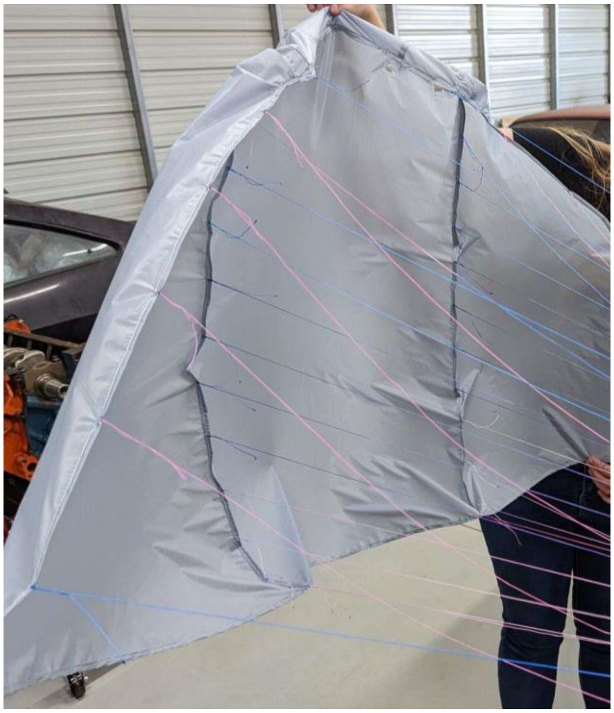


Figure 18: Final Paraglider Design

## 8.2. Airframe

The finalized nose cone design can be seen in the figures below. A Deltamaker 3D printer with Hatchbox PLA filament was used to print the nose cone.

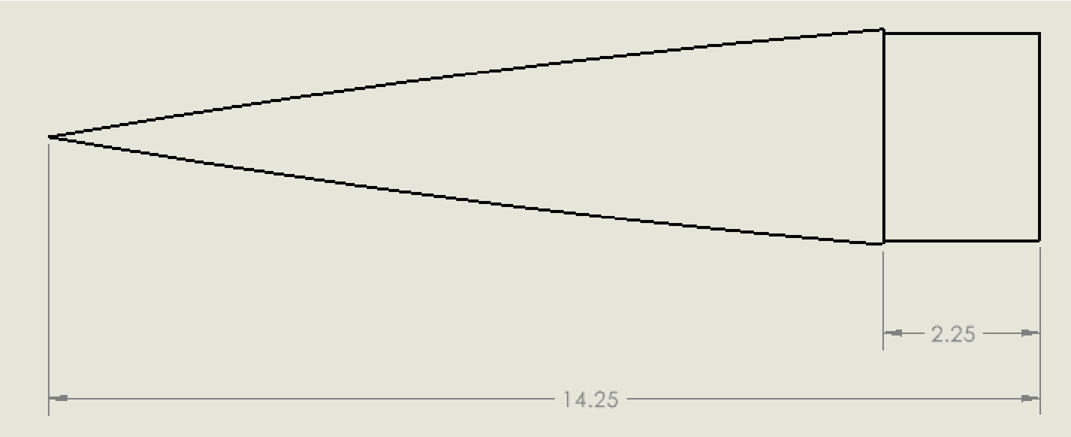


**Figure 19: Final Nosecone Design Drawing**

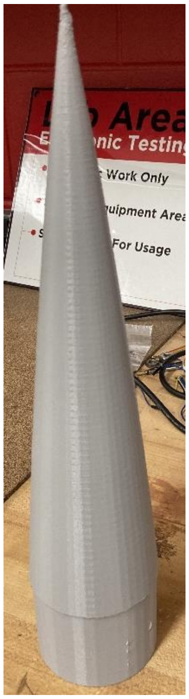


**Figure 20: Final Nosecone Design**

The fin design below was modeled in SOLIDWORKS and a laser cutter was used to cut the shapes out of 1/4in birch plywood.

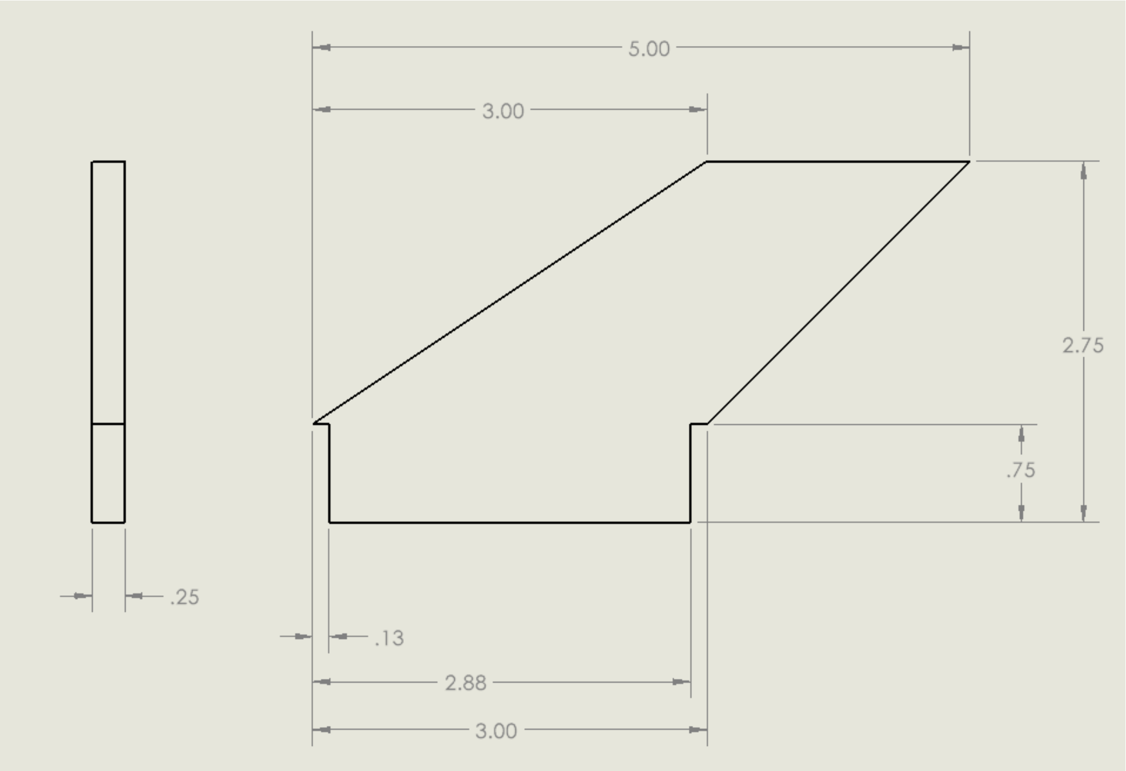


**Figure 21: Final Fin Design Drawing**

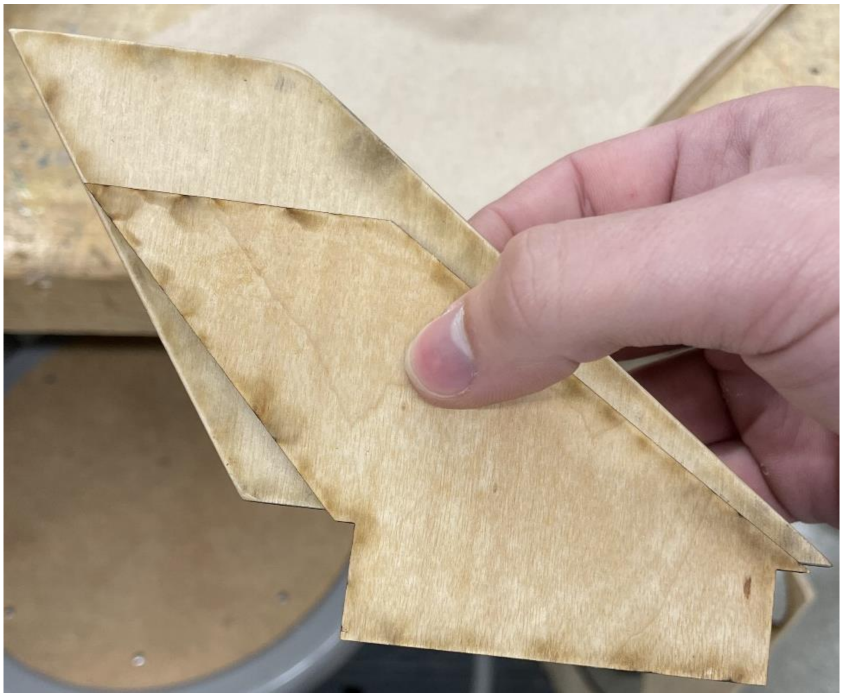


**Figure 22: Final Fin Design**

The airframe assembly consists of the nose cone, body tube, centering rings, motor mount, and fins. The nose cone friction fits into the top of the body tube. Centering rings were epoxied to the motor mount to keep the motor centered in the body tube. The centering rings were then epoxied to the body tube to keep permanently in place. Slots were cut into the body tube to insert the fins. Epoxy was added to the fins where it contacted the motor mount internally, and the body tube externally.

**Figure 19: Final Fin Design**

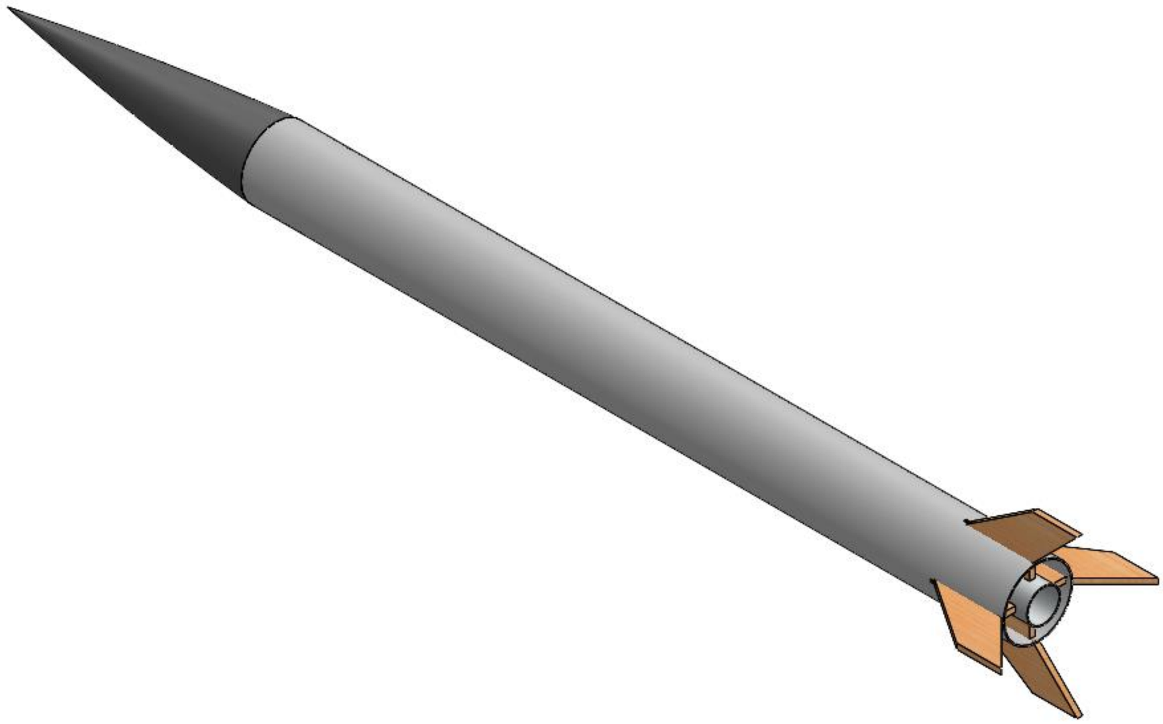


**Figure 23: Exterior Isometric View of Final Rocket Design**

## 8.3. Flight Computer

The final design of the flight computer consisted of the Raspberry Pi as the central control unit. The Arduino was connected to collect data from the GPS and barometric pressure sensor and send data to the Raspberry Pi via serial ports. The video transmitter was soldered to the composite video port of the Raspberry Pi and would transmit the video from the Raspberry Pi camera to the video receiver screen.

A 5V DC bus was connected to the Raspberry Pi to connect 3 servo cables. One cable went straight to the servo the other two servo cables connected to channels 2 and 6 on the RC receiver. These were to switch between automatic and manual control. The final piece of the final design was a MOSFET switch. It would wait for the code to open the switch and allow 5V to go directly to the E-match effectively lighting it.

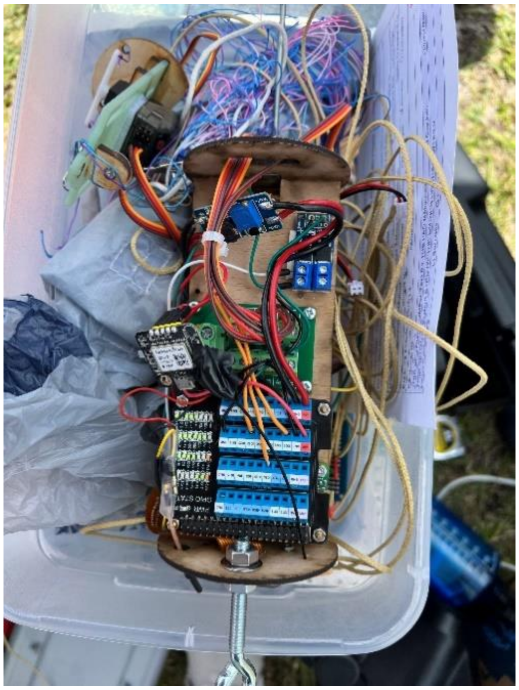


**Figure 21: Final Flight Computer Design**

# 9. System Evaluation

Due to significant time restraints, the system designed was not able to meet the system requirements established for this project. However, the design should be theoretically capable of satisfying all desired objects listed in Table 3 of Section 4. This section will outline ground testing performed on manufactured components as well as the results of the final competition rocket flight. The ground testing primarily sought to evaluate the performance of the Rogallo paraglider and flight computer since those subsystems introduced the greatest amount of complexity.

Most ground tests conducted for this project consisted of dropping the parachute from a high location. Although no specific requirement quantified the desired velocity of decent, the descent rate was evaluated qualitatively from video footage. The original COTS parachute purchased for this project was quickly determined to be insufficient for the final rocket due to its inability to safely carry a 3lb payload. Therefore, testing was conducted on a larger canopy with 3lb and 5lb payload weights.

The parachute was also evaluated on its ability to maintain lateral direction. Each support line was cut to a specific length depending on the canopy dimensions, so the parachute would retain its shape when fully inflated. The canopy shape was morphed using two control lines attached to a single servo, and the remaining support lines were routed to a single confluence with knotted nylon cords. Minor adjustments were made to the length of each control line and set of support lines between tests until the parachute travelled along a constant vector.

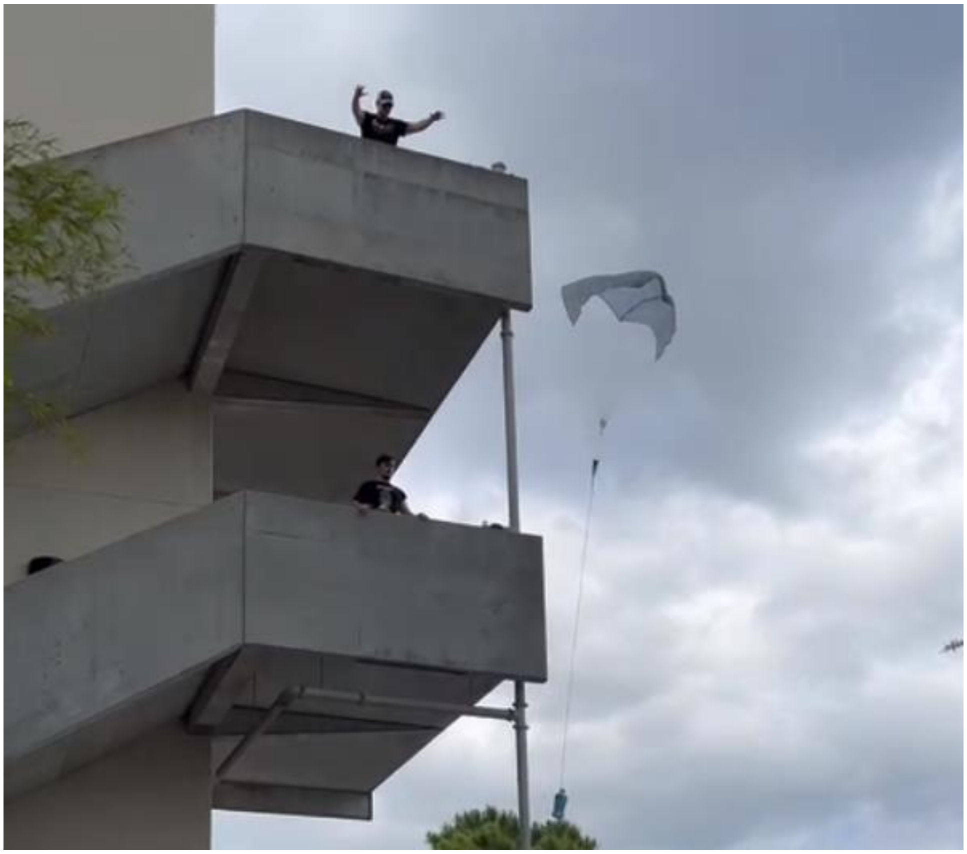
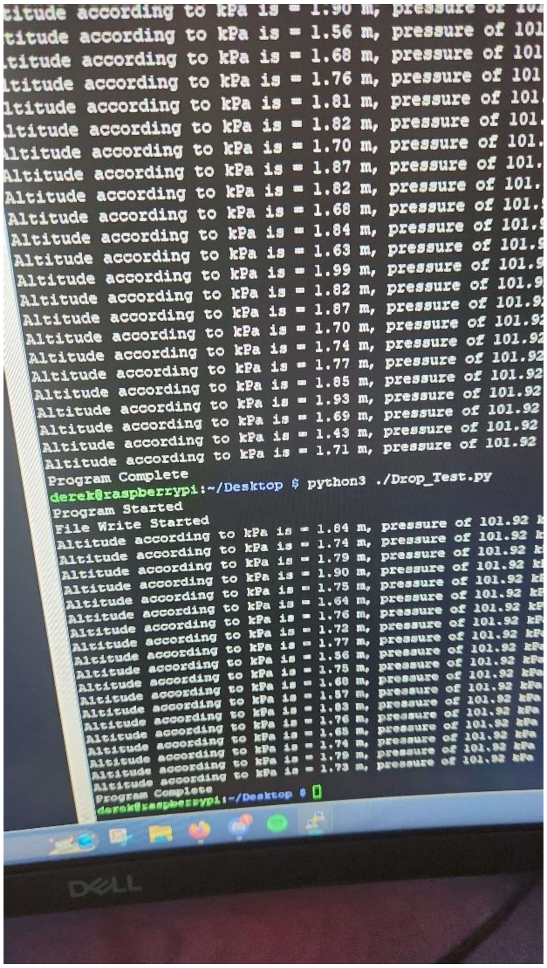


**Figure 24: Image of a Successful Drop Test**

Many tests were done on the flight computer as more parts were integrated into the system.  As each part was integrated the multimeter was used to ensure good connections before applying power. The code was made into sections where it tested the certain part being integrated before adding it into the complete master code. The sensors were tested by taking the input from both the GPS and the barometric pressure sensor and testing the validity of the values as the system moves. The whole system was walked up the parking garage 4 floors to ensure the barometric pressure sensor accurately reported the height.



```
titude according to kPa is = 1.90 m, pressure of 10
titude according to kPa is = 1.56 m, pressure of 101
titude according to kPa is = 1.68 m, pressure of 101
ltitude according to kPa is = 1.76 m, pressure of 101
ltitude according to kPa is = 1.81 m, pressure of 101.
ltitude according to kPa is = 1.82 m, pressure of 101.
ltitude according to kPa is = 1.70 m, pressure of 101.
Altitude according to kPa is = 1.87 m, pressure of 101.
Altitude according to kPa is = 1.82 m, pressure of 101.
Altitude according to kPa is = 1.68 m, pressure of 101.
Altitude according to kPa is = 1.84 m, pressure of 101.
Altitude according to kPa is = 1.63 m, pressure of 101.
Altitude according to kPa is = 1.99 m, pressure of 101.9
Altitude according to kPa is = 1.82 m, pressure of 101.9
Altitude according to kPa is = 1.87 m, pressure of 101.9
Altitude according to kPa is = 1.70 m, pressure of 101.92
Altitude according to kPa is = 1.74 m, pressure of 101.92
Altitude according to kPa is = 1.77 m, pressure of 101.92
Altitude according to kPa is = 1.85 m, pressure of 101.92
Altitude according to kPa is = 1.93 m, pressure of 101.92
Altitude according to kPa is = 1.69 m, pressure of 101.92
Altitude according to kPa is = 1.43 m, pressure of 101.92
Altitude according to kPa is = 1.71 m, pressure of 101.92
Program Complete
derek@raspberrypi:~/Desktop $ python3 ./Drop_Test.py
Program Started
File Write Started
Altitude according to kPa is = 1.84 m, pressure of 101.92 k
Altitude according to kPa is = 1.74 m, pressure of 101.92 k
Altitude according to kPa is = 1.79 m, pressure of 101.92 k
Altitude according to kPa is = 1.90 m, pressure of 101.92 k
Altitude according to kPa is = 1.75 m, pressure of 101.92 k
Altitude according to kPa is = 1.64 m, pressure of 101.92 kP
Altitude according to kPa is = 1.76 m, pressure of 101.92 kP
Altitude according to kPa is = 1.72 m, pressure of 101.92 kP
Altitude according to kPa is = 1.77 m, pressure of 101.92 kP
Altitude according to kPa is = 1.56 m, pressure of 101.92 kPa
Altitude according to kPa is = 1.75 m, pressure of 101.92 kPa
Altitude according to kPa is = 1.68 m, pressure of 101.92 kPa
Altitude according to kPa is = 1.57 m, pressure of 101.92 kPa
Altitude according to kPa is = 1.83 m, pressure of 101.92 kPa
Altitude according to kPa is = 1.76 m, pressure of 101.92 kPa
Altitude according to kPa is = 1.65 m, pressure of 101.92 kPa
Altitude according to kPa is = 1.74 m, pressure of 101.92 kPa
Altitude according to kPa is = 1.79 m, pressure of 101.92 kPa
Altitude according to kPa is = 1.73 m, pressure of 101.92 kPa
Program Complete
derek@raspberrypi:~/Desktop $ 
```

**Figure 25: Serial data being sent from the Arduino to the Raspberry Pi**

The main test points were the transmitters, servo, and the E-match as the only hardware directly connected outside of the flight computer. The first tests were done on the servo themselves as proof of concept. When it is connected directly to the Raspberry Pi it can be turned based on a hardware PWM output.

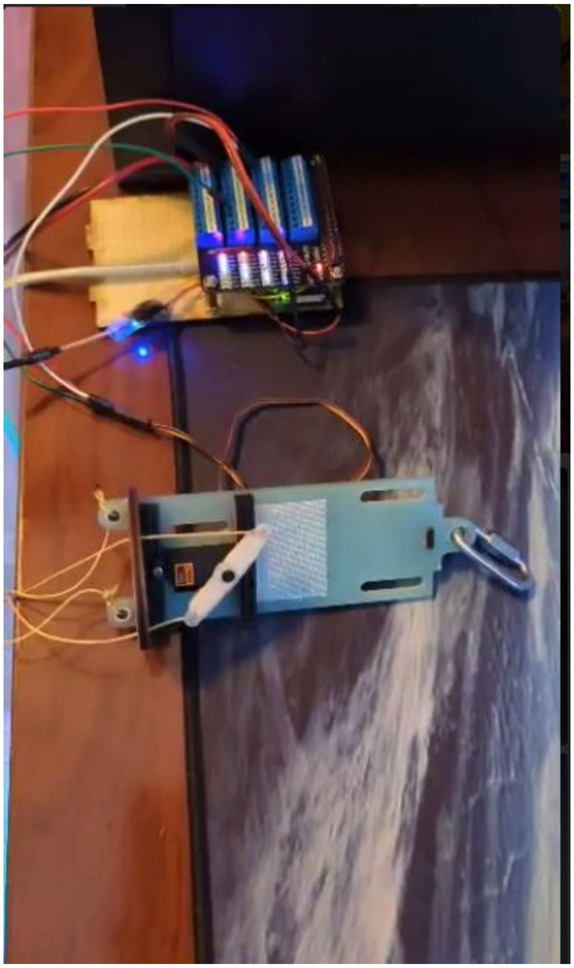


Figure 26: The servo being controlled directly by the Raspberry Pi

The video transmitter was range tested by walking away in a straight line and using google maps to find the distance between the two points. The RC transmitter was tested by controlling the servo and switching between manual and automated. The E-match was tested by connecting it to the MOSFET switch and activating the code to send 5V to it in order to ignite it.

**Figure 27: E-match testing**

## 10. Significant Accomplishments and Open Issues

The final system was designed to guide a rocket to an intended landing site, and preliminary testing indicated that this objective could be met with the proposed system. Time constraints hindered the ability to adequately perform tests with a fully integrated system. However, some significant accomplishments were made during the design process.

The final rocket design weighed just under five pounds which resulted in a theoretical apogee of over 4000ft AGL far greater than the target minimum of 3000ft. Furthermore, a custom parachute was designed, manufactured, and tested in less than a week. The eventual system errors most likely would have occurred in preliminary flight testing, but the proposed system performed far above the expectations of aerodynamic performance.

Despite the successful project objectives, the time constraints resulting from product procurement issues heavily impacted flight computer development. A full system test could not be performed in time for launch, so the electronics ultimately failed in the field. Analysis of components recovered from the launch was conducted to assess the failures that occurred on launch day.

- The high outdoor temperatures at the launch site may have caused certain components to overheat, and proper covering of electronics during field testing would have mitigated the impact of ambient conditions. However, if the electronics were tested, interfaced, and properly secured within the rocket before launch day, the electronics would never need to be exposed to sunlight.
- A voltage step up was used to ensure the voltage entering the Raspberry Pi remained a constant 5V throughout flight computer operation. However, after conducting a flight computer fit test, the voltage reading across the voltage regulator was only 3.3V, showing that the Raspberry Pi was not sending or receiving proper voltage signals. The lack of voltage regulation most likely led to a failure of the servo controlling the parachute as well.
- Some wires were most likely harmed during fit testing of the flight computer. The specific wires that failed send proper signals were the camera ribbon cable, the custom USB cable, and soldered 22-gauge voltage regulator wires. If more time was available to modify the flight computer chassis, more cable grommets could have been incorporated into the design to mitigate crimping.

The following design modifications and advice are offered considering the identified failures of the rocket's structure and flight computer. Additional design considerations are also provided if further flight tests were to be conducted.

a) Enhance the control system: The autonomous functionality of the steerable recovery system was not properly tested during flight. With such a precise system, rigorous ground and flight testing is necessary to fully characterize the weaknesses of the system and improve performance for future flights

b) Make the system more durable: While the materials used in the final design were primarily chosen for their light weight, the simulated apogee would indicate that some materials could be replaced with stronger alternatives, especially in the flight computer. The improved rigidity would mitigate the wire failures observed prior to launch.

The rocket launch was ultimately a failure without a functional flight computer, but some components were recovered from the rocket after flight. Although most of the components were destroyed upon landing, the parachute remained fully intact with no visible damage.

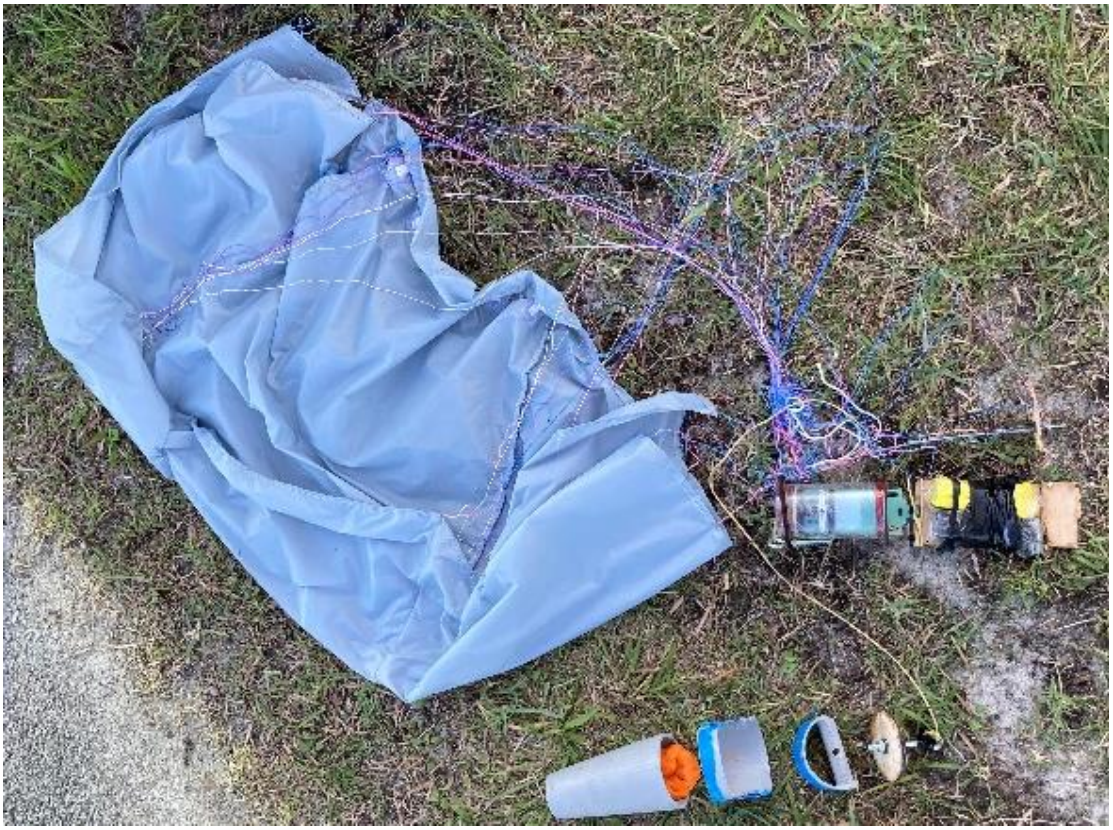


**Figure 28: Recovered Components of Rocket Post-Flight**

## 11. Conclusions and Recommendations

In theory, the design laid out above should be able to meet the requirements outlined in the UCF Student Rocket Launch Initiative challenge statement. A fully integrated launch test with all subsystems functioning normally was not performed. The system could reach altitudes high above 3000 feet, the chosen recovery system did not violate any of the thrust to weight constraints, and the data collected during NASA's Gemini program shows the viability of Rogallo wings for guiding a descending payload to a desired landing area. Had there been ample time for testing of the individual subsystems as well as multiple opportunities to test the fully integrated rocket, it may well have been able to complete the challenge.

For the future, an extensive testing program is key to finding the hidden flaws in a design. Eight months of work by six senior level engineering students from a reputable university went into this project. Many hand calculations were performed, simulations were run, and countless hours went into writing procedures for the day of the launch. All of this hard work produced a very good machine, but not a perfect one. Rigorous testing over a longer period of time would have allowed the team to find flaws and iterate the design to better meet the requirements of the challenge.

# References

"Rocket Motor Basics," Apogee Components, [Online]. Available:
1] https://www.apogeerockets.com/Rocket-Motor-Basics-Quick-Start-Guide?pg=quickside.

"Standard Motor Codes," National Association of Rocketry, [Online]. Available:
2] https://www.nar.org/standards-and-testing-committee/standard-motor-codes/.

"Rocket Stability," National Association of Rocketry, 2014. [Online]. Available:
3] https://www.nar.org/nar-products/rocket-stability/.

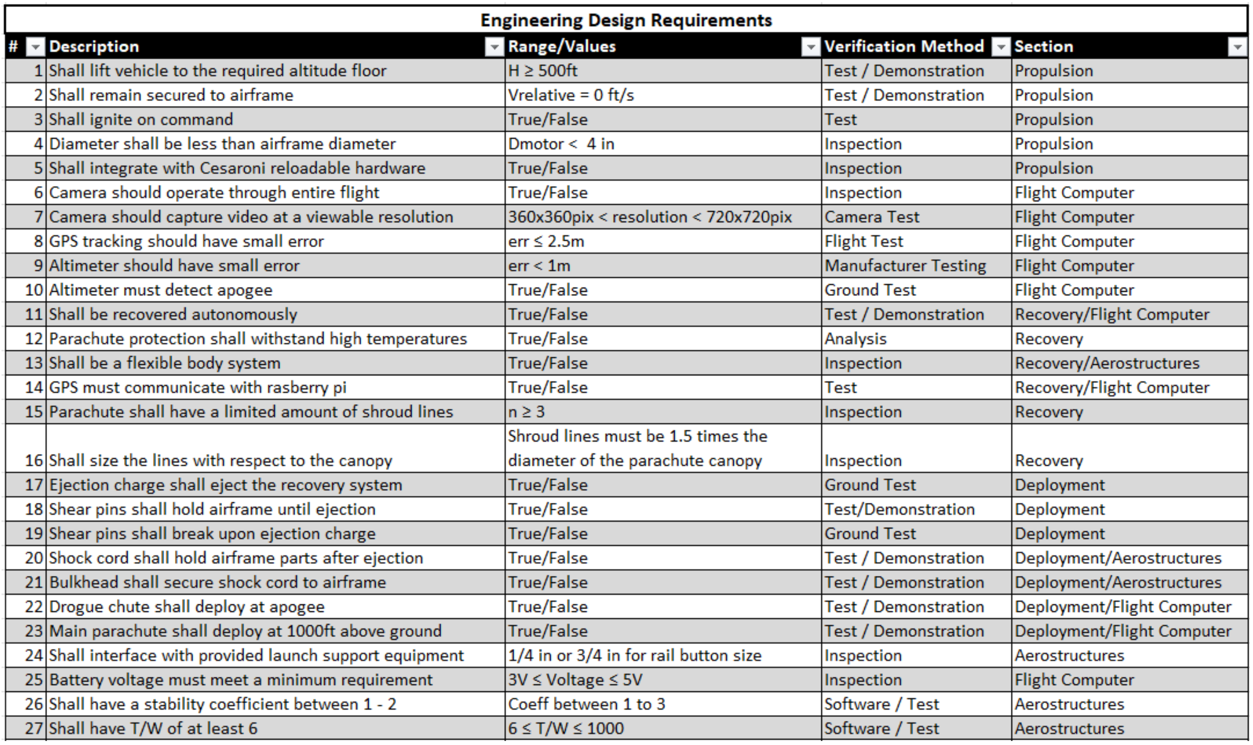
"Rocket Stability," NASA, [Online]. Available: https://www.grc.nasa.gov/www/k-
4] 12/rocket/rktstab.html.

"How to Calcualte bearing between two coordinates," 09 01 2023. [Online].
5] Available: https://mapscaping.com/how-to-calculate-bearing-between-two-coordinates/.
[Accessed 20 03 2023].

"Controls Tutorials for MATLAB & Simulink, DC Motor Position: System Modeling,"
6] [Online]. Available:
https://ctms.engin.umich.edu/CTMS/index.php?example=MotorPosition&section=SystemModeling. [Accessed 21 March 2023].

J. Umenberger and A. H. Goktogan, "Guidance, Navigation, and Control of a Small-
7] Scale Paramotor," in *Australasian Conference on Robotics and Automation*, Wellington, New
Zealand, 2012.

# Appendix A: Product Specifications



Engineering Design Requirements

| # | Description | Range/Values | Verification Method | Section |
|---|---|---|---|---|
| 1 | Shall lift vehicle to the required altitude floor | H ≥ 500ft | Test / Demonstration | Propulsion |
| 2 | Shall remain secured to airframe | Vrelative = 0 ft/s | Test / Demonstration | Propulsion |
| 3 | Shall ignite on command | True/False | Test | Propulsion |
| 4 | Diameter shall be less than airframe diameter | Dmotor < 4 in | Inspection | Propulsion |
| 5 | Shall integrate with Cesaroni reloadable hardware | True/False | Inspection | Propulsion |
| 6 | Camera should operate through entire flight | True/False | Inspection | Flight Computer |
| 7 | Camera should capture video at a viewable resolution | 360x360pix < resolution < 720x720pix | Camera Test | Flight Computer |
| 8 | GPS tracking should have small error | err ≤ 2.5m | Flight Test | Flight Computer |
| 9 | Altimeter should have small error | err < 1m | Manufacturer Testing | Flight Computer |
| 10 | Altimeter must detect apogee | True/False | Ground Test | Flight Computer |
| 11 | Shall be recovered autonomously | True/False | Test / Demonstration | Recovery/Flight Computer |
| 12 | Parachute protection shall withstand high temperatures | True/False | Analysis | Recovery |
| 13 | Shall be a flexible body system | True/False | Inspection | Recovery/Aerostructures |
| 14 | GPS must communicate with rasberry pi | True/False | Test | Recovery/Flight Computer |
| 15 | Parachute shall have a limited amount of shroud lines | n ≥ 3 | Inspection | Recovery |
| 16 | Shall size the lines with respect to the canopy | Shroud lines must be 1.5 times the diameter of the parachute canopy | Inspection | Recovery |
| 17 | Ejection charge shall eject the recovery system | True/False | Ground Test | Deployment |
| 18 | Shear pins shall hold airframe until ejection | True/False | Test/Demonstration | Deployment |
| 19 | Shear pins shall break upon ejection charge | True/False | Ground Test | Deployment |
| 20 | Shock cord shall hold airframe parts after ejection | True/False | Test / Demonstration | Deployment/Aerostructures |
| 21 | Bulkhead shall secure shock cord to airframe | True/False | Test / Demonstration | Deployment/Aerostructures |
| 22 | Drogue chute shall deploy at apogee | True/False | Test / Demonstration | Deployment/Flight Computer |
| 23 | Main parachute shall deploy at 1000ft above ground | True/False | Test / Demonstration | Deployment/Flight Computer |
| 24 | Shall interface with provided launch support equipment | 1/4 in or 3/4 in for rail button size | Inspection | Aerostructures |
| 25 | Battery voltage must meet a minimum requirement | 3V ≤ Voltage ≤ 5V | Inspection | Flight Computer |
| 26 | Shall have a stability coefficient between 1 - 2 | Coeff between 1 to 3 | Software / Test | Aerostructures |
| 27 | Shall have T/W of at least 6 | 6 ≤ T/W ≤ 1000 | Software / Test | Aerostructures |

## Appendix B: System Evaluation Plan

It is important to note that dropping objects from a parking garage can be dangerous and should only be done in a controlled environment with proper safety measures in place. With that in mind, here is a possible test procedure for dropping a parachute from a parking garage multiple times to make shroud line corrections:

1. Choose a parking garage with a flat, open area on the ground level. Ensure that there are no obstructions in the drop zone, such as cars or pedestrians.
2. Obtain a parachute with adjustable shroud lines and a known weight of 1lb.
3. Attach the parachute to a secure object at the top of the parking garage using a strong and reliable anchor point.
4. Verify that the parachute is correctly stowed and prepared for use.
5. Open the parachute from the parking garage's top and watch it fall to the ground.
6. Take note of the parachute's behavior as it descends, including if it spins or turns and where it lands.
7. Adjust the shroud lines as necessary to correct any issues observed during the descent.
8. Repeat steps 5-7 multiple times, gradually increasing the weight of the object attached to the parachute to 3lbs and then 5lbs.
9. After each drop, observe and record the behavior of the parachute and make any necessary shroud line adjustments.
10. Once the parachute can consistently deploy without issue with a 5lb weight attached, the testing is complete.

To guarantee precise and trustworthy results, it is crucial to document all observations and modifications made throughout the testing procedure. Additionally, during the testing procedure, all safety precautions should be followed, including making sure the drop zone is open and secure and that all individuals participating are suitably qualified and equipped for the work at hand.

# Appendix C: User Manual

To connect to the raspberry pi it needs to be connected to a solid internet connection to SSH into it using the IP address. The other option is to use a monitor with an HDMI in to view the screen while connecting a mouse and keyboard to interface with it. Once it is connected a command line needs to be opened in order to run the code. At this point the video should be visible on the video receiver.  The RC signal is then verified by switching to manual and controlling the servo. Switch controller back to automatic and it is ready to be secured to the rocket

Lay the parachute on a flat surface with the lines facing up. Fold one wing tip over to the middle of the leading edge of the parafoil, and then do the same for the other to form a diamond shape. Fold the parachute in half along the length of the keel and check to make sure that the parachute lines are inside the parachute. Fold the top parachute wing in a zig zag pattern, flip the parachute over, and repeat the zig zag pattern for the other wing. Next roll up the fabric of the canopy in the direction of the support lines. Place your insulator (Dog barf, Nomex blanket, etc.) inside the body tube of the rocket, followed by the parachute, and then affix the nose cone to the body of the rocket.

Screw the motor into the motor retainer and insert the assembly into the rocket. Install the motor retainer to ensure that the motor does not fall out during flight. Take care not to have the motor near any sources of heat to avoid unintended ignition of the propellant.

Slide the rail buttons of the rocket onto the launch rail. Pay attention to hand position to avoid pinching skin. Adjust the launch tower to the desired angle. Insert the igniter into the motor as far as it will go, then install the igniter retainer. Only after the igniter is installed can its wires be hooked up to the source of electricity. If the igniter is hooked up to a source of electricity before being inserted into the motor, there is a risk of unintended ignition.

## Appendix D: Cost Analysis and Manufacturability Analysis

**Table 12: Make-Buy Analysis of High-Power Rocket Components**

| Part | Manufacturing Cost | COTS Purchase Cost |
|---|---|---|
| FPV Camera | $53.94 | $89.99 |
| Flight Computer | $348.24 | $358.31 |
| Nose Cone | $24.99 | $27.64 |
| Fins + Centering Rings | $26.29 | $23.74 |
| Parachute | $41.94 | $120.75 |
| Total | **$495.40** | **$620.43** |

## Appendix E: Expense Report

**Table 13: Full High-Power Rocket Expense Report**

| Part | Total | Subsystem |
|---|---|---|
| 3.00" (75mm) Airframe Tubing | $46.61 | Airframe |
| 3.00" (75mm) Airframe Tubing | $34.30 | Airframe |
| 1/2 in. Metal Offset Clip (8-Piece) | $1.76 | Airframe |
| 3/8 in. x 4 in. Zinc-Plated Eye Bolt with Nut | $1.28 | Airframe |
| M8-1.25 Stainless Steel Metric Hex Nut | $1.25 | Airframe |
| 0.85 fl. oz. Epoxy | $14.96 | Airframe |
| M8 Zinc-Plated Split Lock Washers | $1.25 | Airframe |
| TBS Unify Pro32 Nano 5G8 V1.1 500mW Video Transmitter | $61.14 | Flight Computer |
| 32" Gliding Parachute Canopy | $168.00 | Recovery |
| Coupler for 75mm high power LOC tubes | $5.42 | Flight Computer |
| SMALL NYLON SHEAR PINS - 20 PACK | $3.52 | Airframe |
| 300 lb test strength braided cord | $22.50 | Airframe |
| 4.3" FPV Monitor 48CH 480 x 272 LCD Reciever Monitor Auto Search with OSD Built-in Battery+Sunshade Hood for RC FPV Quadcopter | $52.99 | Flight Computer |
| Crayola Modeling Clay in Bold Colors, 2lbs | $8.59 | Airframe |
| HiLetgo 12V 1 Channel Relay Module With Optocoupler Isolation Support High or Low Level Trigger | $5.89 | Flight Computer |

| | | |
|---|---|---|
| iFlight 5.8g FPV Antennas 150mm RHCP RP-SMA Male Antenna for Long Range Quadcopter Drone (Pack of 2) | $16.99 | Flight Computer |
| 9KM DWLIFE Nylon Ripstop Fabric, Grey 60x196 Inch, 40D Waterproof, Lightweight, Windproof, Bulk Fabric for Kite, Tent, Flag, Bag, Tarp Cover, Outdoor DIY Project | $27.95 | Recovery |
| Miscellaneous Fasteners | $11.80 | Airframe |
| HATCHBOX 1.75mm Cool Gray PLA 3d Printer Filament, 1 KG Spool, Dimensional Accuracy +/- 0.03mm, 3D Printing Filament | $24.99 | Airframe |
| Glass | $14.89 | Recovery |
| Cessaroni i470 White Thunder | $75.00 | Propulsion |
| Cessaroni 4-Grain Reloadable Motor Casing | $81.31 | Propulsion |
| 1/8 in. x 48 in. x 32 in. DPI Markerboard Panel | $26.77 | Flight Computer |
| 1/4 in. x 2 ft. x 4 ft. Birch Plywood | $23.74 | Airframe |
| Arduino Nano BLE Sense Rev2 with headers | $43.50 | Flight Computer |
| HiLetgo 3pcs Nano V3.0 3.0 Controller Terminal Adapter Expansion Board Nano IO Shield Simple Extension Plate for Arduino Mano AVR ATMEGA328P | $8.79 | Flight Computer |
| GPS Module Receiver,Navigation Satellite Positioning NEO-6M (Arduino GPS, Drone Microcontroller, GPS Receiver) Compatible with 51 Microcontroller STM32 Arduino UNO R3 with Antenna High Sensitivity | $11.99 | Flight Computer |
| Raspberry Pi Zero WH 512 MB | $47.00 | Flight Computer |
| GeeekPi Raspberry Pi GPIO Screw Terminal Block Breakout Board HAT with GPIO Status LED, Raspberry Pi GPIO Expansion Board Breakout Module for Raspberry Pi 4B/3B+/3B/2B/B+/Pi Zero W/Pi Zero 2 W | $18.99 | Flight Computer |

| | | |
|---|---|---|
| Dorhea MT3608 DC-DC Step Up Boost Power Converter 2A Module Adjustable Step Up Voltage Regulator Board Voltage 2-24V to 5V-28V Output Voltage Mico USB (Pack of 10) | $9.99 | Flight Computer |
| LoveRPi MicroUSB to USB 4 Port Black OTG Hub for Raspberry Pi Zero | $6.99 | Flight Computer |
| Amazon Basics High-Speed Mini-HDMI to HDMI TV Adapter Cable (Supports Ethernet, 3D, and Audio Return) - 6 Feet | $8.79 | Flight Computer |
| Arducam 8MP IMX219 Camera Module for Raspberry Pi, with Low Distortion 105°(D) FOV M12 Lens, Compatible with Raspberry Pi 4 Model B, Pi 3/3B+, Pi Zero 2W and More | $23.99 | Flight Computer |
| Spater Micro USB Sync Cable for Samsung, HTC, Motorola, Nokia, Android, and More (5 Pack) (Black) | $6.98 | Flight Computer |
| OONO 30Amp 48V 2x6 Position Terminal Block Distribution Module | $12.99 | Flight Computer |
| V TELESKY MOS FET Board 10Pcs High-Power FET Trigger Switch DC 5V-36V 15A(Max 30A) 400W FET Driver,PWM Adjustment Electronic Switch Control Board | $15.99 | Flight Computer |
| KBT 3.7V 3500mAh Li-Polymer Battery: 933871 Lipo Rechargeable Lithium-ion Replacement Batteries with PH 2.54 JST Connector, PH1.25/2.0 JST Connector for Replacement | $16.99 | Flight Computer |

**Table 14: Subsystem Expense Report**

| Subsystem Total | Proportion | Subsystem |
|---|---|---|
| $196.55 | 20% | Airframe |
| $402.18 | 42% | Flight Computer |
| $210.84 | 22% | Recovery |
| $156.31 | 16% | Propulsion |
| **$965.88** | | |

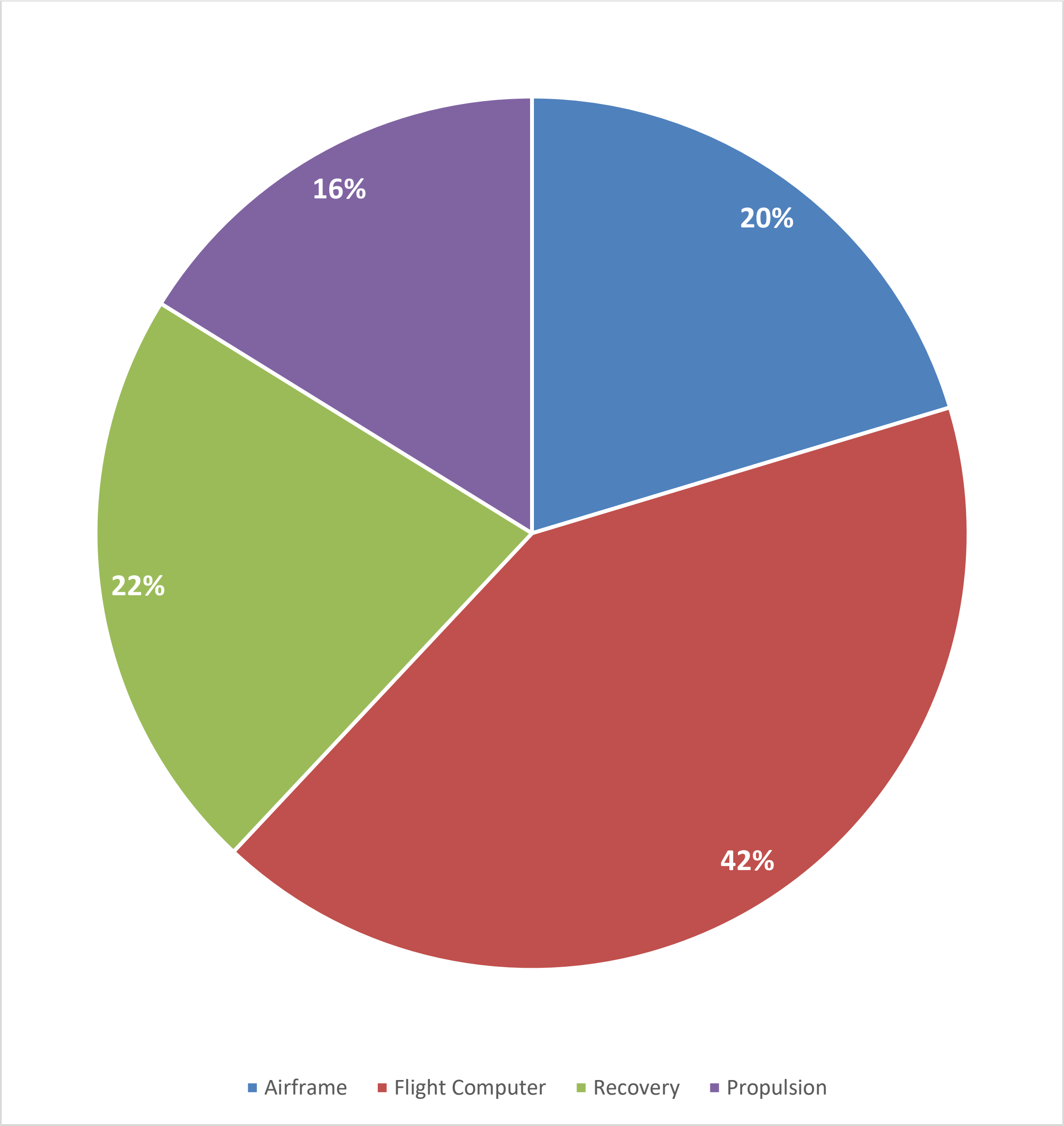


**Figure 29: Proportional Funding of Each Subsystem**

## Appendix F: List of Manuals and Other Documents

- Apogee Components Gliding Parachute Design Manual

## Appendix G: Design Competencies

Table 15: ABET Aeronautical Design Competency Matrix

| AERONAUTICAL | Critical/Main contributor | Strong contributor | Necessary but not a primary contributor | Necessary but only a minor contributor | Only a passing reference | Not Included in this Design Project |
|---|---|---|---|---|---|---|
| Aerodynamics | | 🟥 | | | | |
| Aerospace Materials | | | | 🟥 | | |
| Flight Mechanics | | 🟥 | | | | |
| Propulsion | | | 🟥 | | | |
| Stability & Control | 🟥 | | | | | |
| Structures | | | 🟥 | | | |

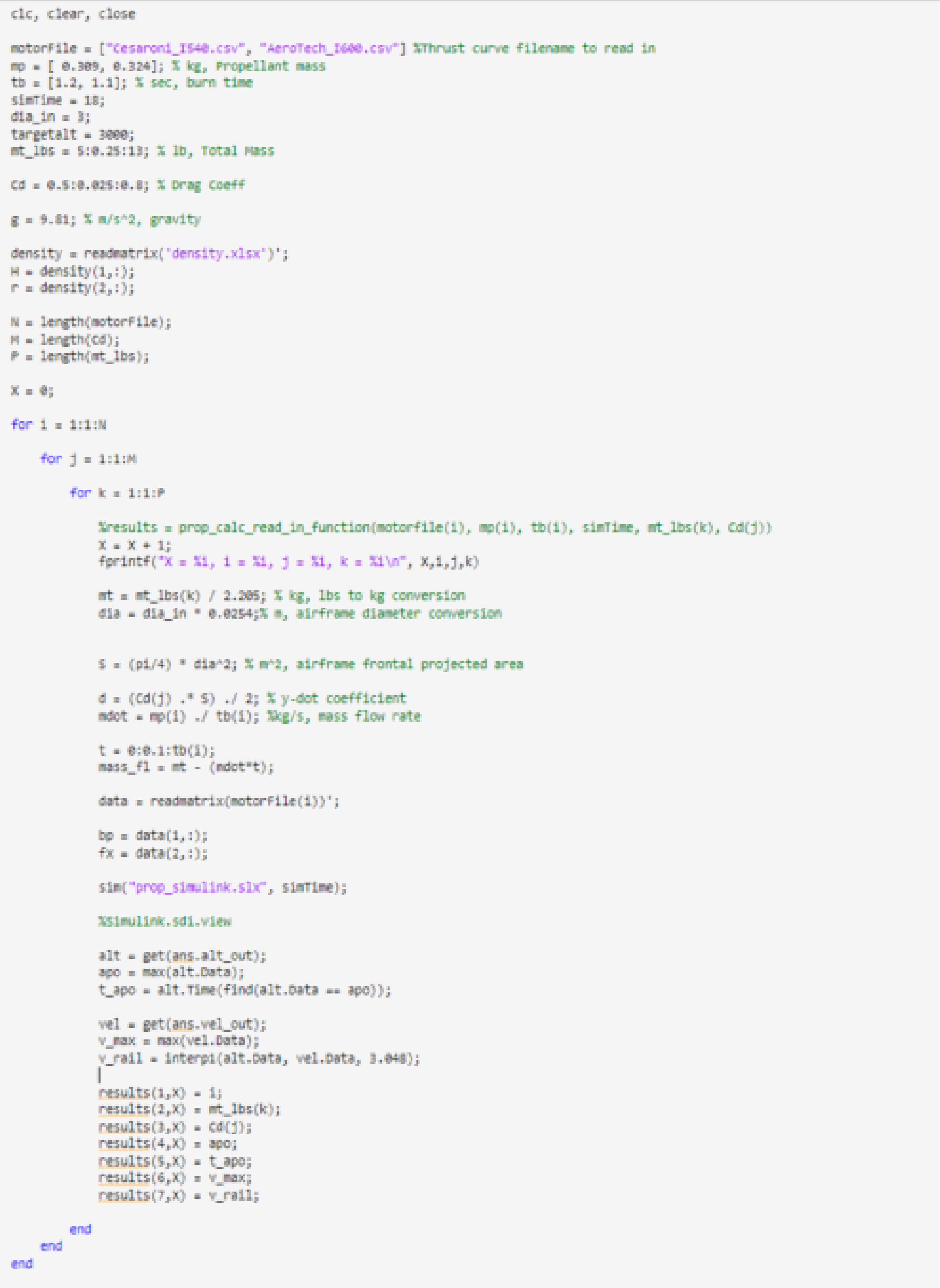
Table 16: ABET Astronautical Design Competency Matrix

| Astronautical | Critical/Main contributor | Strong contributor | Necessary but not a primary contributor | Necessary but only a minor contributor | Only a passing reference | Not Included in this Design Project |
|---|---|---|---|---|---|---|
| Aerospace Materials | | | | 🟥 | | |
| Attitude Determination & Control | 🟥 | | | | | |
| Orbital Mechanics | | | | | | 🟥 |
| Rocket Propulsion | 🟥 | | | | | |
| Space Environment | | | | | | 🟥 |
| Space Structures | | | | | | 🟥 |
| Telecommunications | 🟥 | | | | | |

Table 17: ABET Competency Lookup Table

| Topic | Criticality to Project | Section | Comments |
|---|---|---|---|
| Aerodynamics | Strong | 3.6, 6.3.1, 7.2.1 | |
| Aerospace Materials | Necessary but minor | 3.6, 7.3 | |
| Flight Mechanics | Strong | 6.2 | |
| Propulsion | Necessary but minor | 3.1 | |
| Stability & Controls | Critical | 3.2, 3.4, 6.2.3, 7.3 | |
| Structures | Necessary | 3.6, 6.3 | |
| Attitude Determination & Control | Critical | 6.2.1, 6.2.2 | |
| Rocket Propulsion | Critical | 6.1, 7.1 | |
| Telecommunications | Critical | 3.3, 7.4, 8.3 | |

# Appendix H: MATLAB Code



```
clc, clear, close

motorFile = ["Cesaroni_I540.csv", "AeroTech_I600.csv"] %Thrust curve filename to read in
mp = [ 0.309, 0.324]; % kg, Propellant mass
tb = [1.2, 1.1]; % sec, burn time
simTime = 18;
dia_in = 3;
targetalt = 3000;
mt_lbs = 5:0.25:13; % lb, Total Mass

Cd = 0.5:0.025:0.8; % Drag Coeff

g = 9.81; % m/s^2, gravity

density = readmatrix('density.xlsx')';
H = density(1,:);
r = density(2,:);

N = length(motorFile);
M = length(Cd);
P = length(mt_lbs);

X = 0;

for i = 1:1:N

    for j = 1:1:M

        for k = 1:1:P

            %results = prop_calc_read_in_function(motorfile(i), mp(i), tb(i), simTime, mt_lbs(k), Cd(j))
            X = X + 1;
            fprintf("X = %i, i = %i, j = %i, k = %i\n", X,i,j,k)

            mt = mt_lbs(k) / 2.205; % kg, lbs to kg conversion
            dia = dia_in * 0.0254;% m, airframe diameter conversion


            S = (pi/4) * dia^2; % m^2, airframe frontal projected area

            d = (Cd(j) .* S) ./ 2; % y-dot coefficient
            mdot = mp(i) ./ tb(i); %kg/s, mass flow rate

            t = 0:0.1:tb(i);
            mass_fl = mt - (mdot*t);

            data = readmatrix(motorFile(i))';

            bp = data(1,:);
            fx = data(2,:);

            sim("prop_simulink.slx", simTime);

            %Simulink.sdi.view

            alt = get(ans.alt_out);
            apo = max(alt.Data);
            t_apo = alt.Time(find(alt.Data == apo));

            vel = get(ans.vel_out);
            v_max = max(vel.Data);
            v_rail = interp1(alt.Data, vel.Data, 3.048);

            results(1,X) = i;
            results(2,X) = mt_lbs(k);
            results(3,X) = Cd(j);
            results(4,X) = apo;
            results(5,X) = t_apo;
            results(6,X) = v_max;
            results(7,X) = v_rail;

        end
    end
end
```