Daedalus IV

AAE 251 Final Project

Team R06 “Jason and the Argonauts”

Spring 2019

Youssef Noureddine

Jordan Soberg

Wellington Froelich

Ryan Alcorn

Jason Adams

Tomoki Koike

Abstract—Considering the impending zombie apocalypse, we determined that the best option to aid in the survival of the human species was a series of thermal imaging CubeSATS launched into a polar low Earth orbit by the Daedalus IV rocket designed by our team. By identifying the risks involved, we ensured that those most likely to cause error were mitigated effectively. In addition, we considered the various pros and cons of launch vehicle layout and decided that the Daedalus IV should fly as a two stage rocket with a solid lower stage and a liquid upper stage.

Table of Contents

List of Figures

List of Tables

1 Introduction

Civilization on planet Earth faces the dawn and doom of what we feared the most, a zombie apocalypse. We, the people of West Lafayette have been safe up until now but we do not know how long it will stay that way. In a desperate act to protect mankind from extinction, the mayor of West Lafayette has tasked us, The Argonauts, with designing a rocket capable of deploying multiple CubeSats into low Earth orbit to search for stranded humans using thermal imaging. Then, rescue aircraft will set out to reach nearby humans and recover them to West Lafayette.

The department head of biology has calculated that if there are any stranded survivors we will have approximately one year to find them before they succumb to malnutrition and disease. The Mayor has given us permission to use any resource present within West Lafayette to construct the rocket and has declared a state of emergency, allowing us to perform tasks such as launching rockets from airport, something that is normally not allowed. This report will go in depth into how we plan to honor the mayor’s request with our launch vehicle, the Daedalus IV.

The Daedalus IV is an efficient, easy-to-manufacture spacecraft designed to deliver multiple CubeSats into an orbits that will allow for the discovery of survivors.

2 Needs and Requirements Analysis

2.1 Stakeholders

Stakeholders hold a vested interest in our program. They provide motivation, funding, and manpower. We determined our stakeholders by analyzing implications of our missions in the immediate future and thereafter. As a result, we found the following groups to be concerned with the success of our mission.

- Stranded survivors that remain to be found as they can increase our odds to be able to strike back and survive the zombie apocalypse
- Our children and their ability to survive the zombie apocalypse
- Our team as we will be the ones who design and launch the rockets that will find stranded survivors who will assist in our survival
- Zombies as they pose a threat to the stranded survivors
- The aircraft team as they rely on our results to know where to launch their aircraft

2.2 Needs

After identifying all relevant stakeholders, we then developed a list of all stakeholder needs. This list was broken down into needs for the satellite itself and the launch vehicle. Only after establishing this list would we be able to generate mission requirements. The needs for our launch vehicle and CubeSAT are as follows, determined from our textbook. (Sellers 2004)

Vehicle:

- Deliver the CubeSAT into orbit
- Launch vehicle must be manufactured quickly

● Launch vehicle must utilize common materials such as aluminium, steel, and plastics.

● Launch vehicle must require as few materials as possible to assemble.

● Launch vehicle fairing and structural adaptors can be changed to accommodate multiple CubeSATS

● Launch vehicle must be able to make attitude adjustments

●Launch vehicle must be controllable

CubeSAT:

● The CubeSAT can accurately locate other human survivors

● The CubeSAT should transfer data to Purdue headquarters

● The CubeSAT can withstand solar radiation

● The CubeSAT can withstand thermal expansion or contraction due to the heat of the sun or lack thereof

● The CubeSAT can survey the entirety of North America with an orbital period of at least 1 hour.

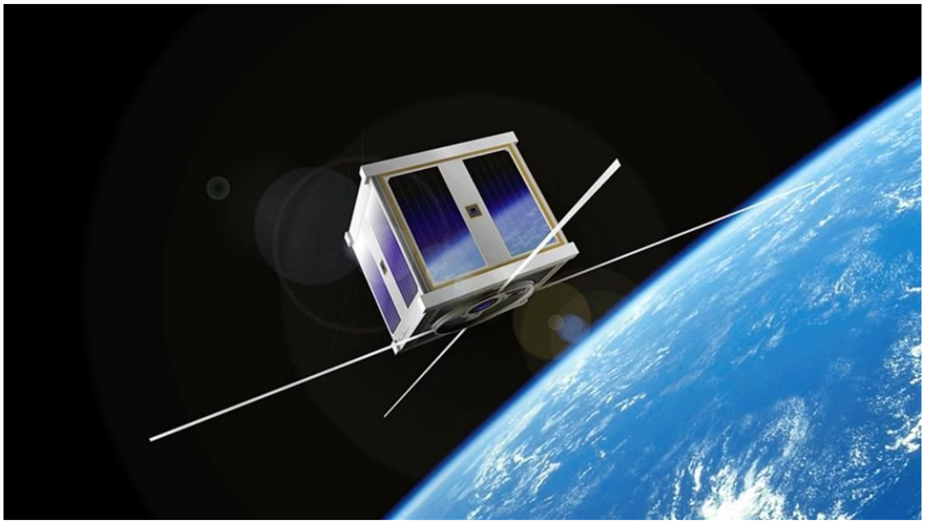


Figure 2.2-1: Example of CubeSat (from website of Sen Corporation Ltd)

Shown above is a CubeSAT, which we have access to for our mission. CubeSATs have standard side length of 10cm.

2.3 Requirements

From our list of vehicle and CubeSAT needs, we then developed mission specific requirements. Requirements, as opposed to needs, must be verifiable, and independent, and complete. Listed below are our completed mission requirements:

Vehicle:

● The vehicle can deliver Cubesats in an orbit that covers our entire survey area: The survey area is tentative because this depends on the range of the aircraft and that will be clear after collaborating with the aircraft team.

● The vehicle has a storage capacity for X number of CubeSats. This is so we can launch multiple CubeSATS.

● The launch vehicle can be built using only X quantity of available resources. This is due to the limited availability of resources

● The launch vehicle takes X amount of time to build. This is to ensure survivors are found as quickly as possible

● The launch vehicle must be able to adjust pitch/yaw/roll. This will allow for attitude adjustment in flight

● The launch vehicle must be able to transmit and receive signals/instructions from ground control to perform specific tasks

● The launch vehicle must launch the CubeSATS high enough so that it would be in orbit for at least a year

CubeSATs:

● Deliver X amount of gigabytes of thermal data to the LAByrinth to locate signs of life

● Provide full coverage of North America in X amount of time

● Protect sensors from X range of temperatures

● Verify accuracy (X%) of thermal signatures to minimize false positives

● The CubeSAT protects sensors from X amount of radiation exposure

2.4 Preliminary Risk Analysis

This section will provide an insight into the risks that we will need to address in order to complete our mission. We looked into the risks and challenges that currently face rockets both inside the atmosphere and out. Using these metrics, we created a table to compare impact level vs likelihood allowing us to understand what effects a certain risk may have.

Risk Matrix:

LIKELIHOOD

7 R[1]	16 R[2]	20 R[12]	23	25
6 R[8], R[10]	13 R[5]	18	22	24
4	10	15 R[7]	19 R[3]	21 R[9], R[14]
2	8	11	14	17 R[4]
1	3 R[13]	5	9	12 R[6], R[11]

CONSEQUENCE

Table 2.4-1: Risk Matrix (Kelly, 2004)

This risk matrix is used to place the importance of each risk on a quantifiable scale. After debating the risks as a group and conducting some research in *Sellers, Ch.3* and *The MIT Technology Review*, we decided to place the risks in the matrix accordingly. (Sellers 2004), (Sauser 2009) To mitigate these we will follow historical mitigation methods below. The color of the box the risk is placed in indicates the level of attention we need to give it. Each R[x] value indicates a certain risk explained in Table 2.4.2 below.

Risks/Hazards	Label	Effects	Historical Mitigation
Drag	R[1]	Causes severe stress on materials and can result in catastrophic fractures.	Design our flight path to escape Earth's atmosphere as quickly as possible.
Atomic Oxygen	R[2]	Rusts the exterior and degrades the electronics	Atomic oxygen can be neutralized with ozone molecules to create non-reactive O2.
Outgassing	R[3]	Molecules of air trapped in the material are released once the craft reaches the vacuum of space. Sensors can be covered by the molecules and electronics can malfunction due to this.	We will bake all our materials in a thermal vacuum prior to installation.
Cold Welding	R[4]	In the vacuum of space, there is no air between different metal parts. This results in two pieces of metal being fused together.	This risk can be mitigated by choosing materials that do not cold-weld and lubricate/coat any moving parts that may cold-weld.
Heat	R[5]	Heat threatens the structure with internal stress due to thermal expansion	We will coat the surface of the rocket with insulating material as well as use materials that will experience a low thermal strain.
Collision	R[6]	Collision endangers the entire mission because of	We will reinforce our rocket and fairing with hardened materials.

		the potential for an object to hit the payload.	
Radiation	R[7]	Heat exposes surfaces to extreme degrees causing them to melt.	We will design thermal control systems that releases heat
Solar Pressure	R[8]	Photons from the sun carry energy that can change the momentum of the spacecraft upon collision.	This can be mitigated by carrying extra fuel for course corrections using attitude thrusters.
Charged Particles	R[9]	Without the atmosphere to protect the sensitive electronics of CubeSATs, charged particles can destroy or disrupt functions. Can give false positives that lead to a waste of resources.	We will mitigate the effects of charged particles by utilizing rad-hardened electronics, spacecraft shielding, and advanced coding techniques.
Cold Temperatures on Take-Off	R[10]	Can cause components to become brittle and fail.	Set legal limits for flight conditions.
Engine Failure	R[11]	Loss of thrust or asymmetric thrust, causing the rocket to lose control due to a large moment.	Use simpler engines when possible such as pressure fed liquid motors or solid motors.
Vibrations from the engines.	R[12]	Thrust oscillations send sound waves up the rocket. This vibration can damage electronics, confuse guidance systems, and causing liquid engines to shutdown. [1]	Lower total mass of the rocket as less thrust needed to lift the rocket means less amplitude in the vibration. Guidance systems can be programmed to compensate for vibrations.
Wildlife damaging the rocket.	R[13]	Wires can be chewed and wildlife can build nests in hidden structures prior to launch. Organic matter can jam moving parts and clog pipes.	Build indoors and hire wildlife management. Minimize launchpad time.

Premature stage ignition	R[14]	Sparks could trigger premature stage ignition	Minimize fire hazards on board and at launch site. Ground rocket on launchpad to reduce static electricity.

Table 2.4-2: Risks, Effects, and Mitigations (Marais, 2019)

Based on the risk analysis table (Table 2.4.1), we determined the following risks were most important:

- R[9]: Charged Particles
- R[12]: Engine Vibrations
- R[14]: Premature Stage Ignition

Thus, we have determined that the most important mitigations are using rad-hardened electronics as well as spacecraft shielding to protect against R[9], reducing total mass to ward off the dangers of R[12], and minimizing fire hazards along with reducing electrical potential to avoid a premature ignition, R[14]. Nonetheless, we took action to mitigate the rest of the risks as even the smallest disruption could spell disaster for a project as complex as this one.

3 Concept Generation, Selection, and Development

In this section we will discuss possible orbits, analyze each orbit type’s advantage and disadvantage and make a decision on which orbit type we will send our CubeSATS on. We will also discuss types of propellants and Engine types and discuss their advantages and disadvantages.

3.1 Concept Generation and Selection

The manner in which we generated the following concepts was to look at past mission types for inspiration. We discovered and researched missions based on polar and geostationary orbits. An example of our polar orbit flight profile is the NOAA’s Joint Polar Satellite System, a 5000 lb satellite launched by a Delta II. One example of a successful geostationary mission in 2018 is the GOES-R mission. NASA sent GOES (Geostationary Operational Environmental Satellite-S) network satellites to provide constant vigil of weather conditions such as hurricanes, tornadoes, flash floods, hail storms and other severe weathers. At a final altitude at 22300 miles, the currently operating satellites provide the US high resolution and meteorologically informative images to earth; therefore, satisfying its geosynchronous mission (Dunber, Jenner 2018).

The third mission type we learned of was changing the inclination of the launch vehicle. We realized this presented another option for deploying CubeSats by sending them into orbits of different inclinations of orbit. However, no historical data on missions with inclination changes was found. This, combined with the overwhelming suggestions to consider other mission types from online forums (GMAT 2011), tainted the mission type from the get-go. However, we still considered its potential.

1. Polar

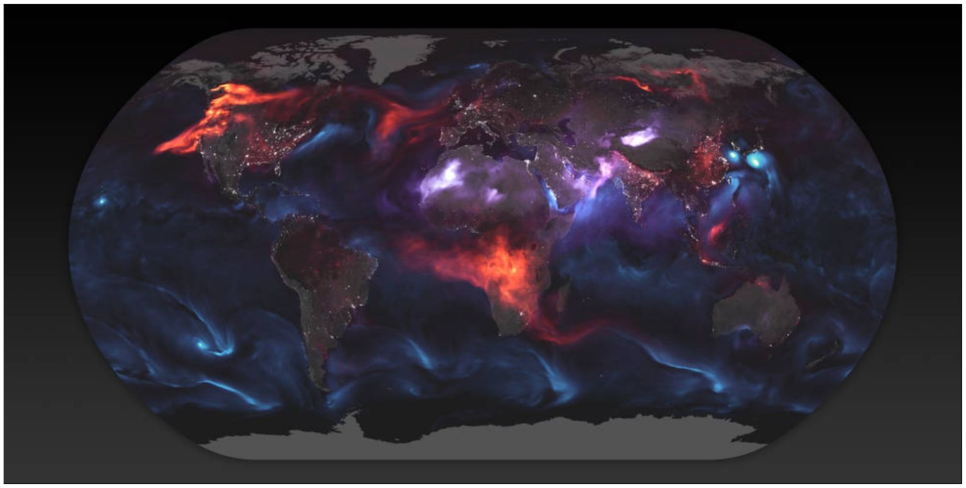
This system is designed to launch satellites into a polar Low Earth Orbit. A polar-orbiting satellite allows for eventual 100% ground cover. This method is efficient in that only one CubeSat is required to achieve full ground cover. Since we will be only conducting one launch (as only one is required for full earth coverage), it would only mean that there are multiple cubesats sharing the same orbit. We may carry 2 or 3 cubesats to create redundancy should one fail or get damaged from the launch.

a. Pros:
   i. A single satellite will eventually survey the entire earth.
   ii. Requires a smaller amount of ΔV than geostationary orbit.
   iii. Since it is a lower orbit than the geostationary orbit, it will provide higher resolutions (this is why climate satellites and NWP, numerical weather prediction models are in this orbit)
   iv. More polar cubesats can be launched at different longitudes to increase coverage rate. This could however cost a lot of ΔV.
b. Cons:
   i. The data updates will be in low frequency (full earth cover would take 24 hours).
   ii. Higher ΔV is required to achieve polar orbit because the rotation of the earth is not contributing to the velocity.
   iii. Might provide repetitious and continuous views of one location

2. Geostationary

A geostationary orbit means that the satellite will orbit at exactly the same rate as the Earth, allowing the satellite to stay stationary overhead a certain area. This is a benefit, since the we are only able to help the survivors near Purdue University and there is no need to have a satellite that is able to view the entire Earth. To maintain geostationary velocity, however, this satellite would require a circular orbit of 35,756 km.

Aforementioned, GOES satellite is able to take a beautiful picture with geostationary orbit like the following. The image below is for meteorological means, and this technology can be applied to our polar orbit CubeSats to provide information to the survivors on earth.



Credits: NASA Earth Observatory image by Joshua Stevens, using GEOS data from the Global Modeling and Assimilation Office at NASA GSFC.

Figure 3.1: Image by GEOS Satellite (Patrinas, 2018)

a. Pros:
   i. The ΔV can be reduced by using the Earth’s rotation.
   ii. Gives a constant view of the area surrounding Purdue (North America)
b. Cons:
   i. It takes more fuel to get CubeSATS to 36,756 km (necessary for a geostationary orbit).
   ii. It takes additional fuel to align the orbit with Indiana for optimal viewing.
   iii. Sensors may not be able to accurately detect body heat 36,756 km away from earth, an altitude that is over 50 times farther than the altitude for a polar orbit of 700 km

3. Multiple inclined orbits

Launch a CubeSat carrier into polar orbit and deploy a cubesat there, then perform a burn to another inclination at the poles which will change the longitude of the polar orbit, and then launch another CubeSat. Repeat this process to launch more CubeSats to cover the Earth faster. Simply we will launch one rocket with a large ΔV place it in one orbit release a CubeSat and perform a plane change into another orbit, release another CubeSat and perform this procedure until we have released all our CubeSats/consumed all our fuel.

This solution was developed because in our apocalyptic scenario, we have a plethora of pre-manufactured CubeSats. This means that we could place more eyes in the sky and cover the Earth faster than other options. This option also collects more data since there are more systems observing the Earth.

a. Pros:
   i. Faster and larger data collection at multiple angles compared to a polar orbit/geostationary orbit.
   ii. ΔV can be reduced by using the Earth's rotation for non-polar orbits
b. Cons:
   i. At 700 km of altitude of orbit the CubeSat is probable to deorbit in 5-7 years but our mission duration is only one year and so we do not expect to use the CubeSats long enough for deorbiting to be a risk.
   ii. The amount of fuel required to perform a burn to another inclination around the Earth makes this option very challenging and fuel inefficient given our scarce fuel supplies.

Discussion

We choose the best possible option based on the following criteria:

1) The orbit can be achieved with minimal ΔV.
2) The vehicle's storage capacity.
3) The launch vehicle's resourcefulness
4) The time in which the vehicle can be built
5) The time the CubeSATS will remain in orbit
6) The time it takes to complete an orbit

Our first attempt at a solution involved a combination of the geostationary concept and the multiple orbits concept. The idea was to focus multiple satellites in orbits that covered areas around Purdue extending to the ranges of the aircraft teams’ designs. We decided that the near-Purdue orbit would be an inefficient use of the multiple satellites because a launch configuration offering complete Earth coverage would vastly improve the odds of finding survivors, as opposed to a solution that can only search around Purdue’s campus. This therefore has a low resourcefulness a criteria we decided upon.

Though the time to deorbit does differ for each option, we decided that the duration is not very important as the lowest orbit we are considering is 700km which has a 5-7 year deorbit time period. Since our mission duration is only a year all orbits meet the time requirement.

One consideration was the need for multiple CubeSats in the same orbit in order to cross reference thermal data between the satellites. Radiation can damage received data, which could result in false positives. Multiple sensors would mitigate this issue.

Our team concluded that the best launch plan consisted of launching multiple satellites at different polar orbit longitudes. This satellite constellation was chosen because multiple polar satellites can survey the entire earth less time than any other configuration. The projected simultaneous launch of 10 iridium satellites (to different orbits) by SpaceX proves that such a configuration is feasible. The longitude of the polar orbit will shift with inclination change burns at the poles. This will result in faster and larger data collection at multiple angles. This orbit plan is visualized in the image below.

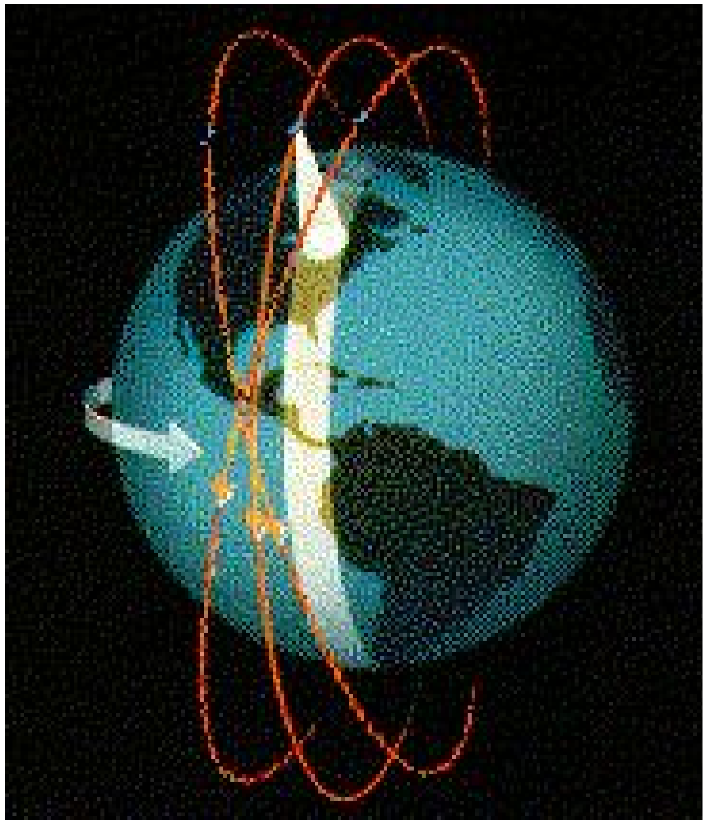


Figure 3.1-3: The Argonauts mission of 3 polar orbit CubeSats, modified by Jordan Soberg (Tablestore, 2019)

3.2 Detailed Concept Development

For a circular polar orbit at 700 km we will need a minimum amount of launch delta V (ΔV_{LEO}), ΔV_{LEO} being the velocity we need to provide the rocket with to maintain a low-earth orbit at a given altitude. This is delta V is given by the equation below:

$$\Delta V_{LEO} = \text{sqrt}\ (\mu / r) = (3.986\times10\hat{}4 / (6.378\times10\hat{}6 + 7\times10\hat{}5)) = 7504 \text{ m/s}$$

Where μ=GM (G being the gravitational constant and M is the mass of the Earth) and r is the distance of the circular orbit from the center of Earth. We will approximate a total ΔV_{LOSS} = 2000 m/s as this is the typical amount for launches.

Since we seek a polar orbit, Earth’s orbital speed will be working against us. However, we argue that since the maximum help earth can provide us with is 463.7m/s from the equator, and the latitude of West Lafayette is 40.4°, Earth’s rotation can only disrupt our polar orbit by 463.7×cos(40.4) = 353m/s. This means that we will have a velocity of 7504 m/s going from pole to pole and a velocity around the equator of 353m/s if we ignore the the velocity of Earth’s rotation.

By ignoring the rotation of Earth and saving 353 m/s ΔV would only put us at a degree separation from a completely polar orbit of arctan(353 / 7004) = 3° only. We believe the benefit of 350 m/s of saved ΔV compensates for a 3° off-polar orbit. To summarize we will just let Earth’s rotation take us where we end up instead of trying to counteract it.

This brings our total delta V to 9504 m/s:

$$\Delta V_{TOTAL} = 7504+2000=9504\text{m/s}$$

After comparing it to the ΔV required to get other launch vehicles into low earth orbit in Table 3 we noticed our 9504 m/s is slightly higher than the other launch vehicles. This is due to us not utilizing earth’s rotation as we are attempting to achieve a polar orbit. However since it is very close to the other ΔV we can say conclude it is safe to conclude we will need approximately 9504m/s. For reference, a table of historical ΔV data is included below, which shows that our requirement is fairly comparable:

Vehicle	Orbit: $h_p \times h_a$ Inclination (deg)	v_{LEO}	Δv_{grav}	$\Delta v_{steering}$	Δv_{drag}	Δv_{rot}*	$\Sigma\Delta v = \Delta v_{prop}$
Ariane A-44L	170 × 170† 7.0	7802	1576	38	135	−413	9138
Atlas I	149 × 607 27.4	7946	1395	167	110	−375	9243
Delta 7925	175 × 319 33.9	7842	1150	33	136	−347	8814
Space Shuttle	~196 × 278 28.5	7794‡	1222	358	107	−395	9086**
Saturn V	176 × 176 28.5	7798	1534	243	40	−348	9267
Titan IV/ Centaur	157 × 463 28.6	7896	1442	65	156	−352	9207

Table 3.2-1: Historical ΔV_{TOTAL} data
Space Propulsion Analysis and Design (Humble, Henry, Larson 2007)

3.3 Engine Selection

In this section, we discuss the selection of a means to propel our payload into orbit (a propulsion system). We considered several different types of engines: liquid, solid, and hybrid. For each option, we came up with pros and cons and finally decided on the best engine type based on what best fits our needs.

Options

1. **Liquid Bi-Propellant**
   a. **Pros:**
      i. Can be fueled with ethanol from local corn and Liquid Oxygen
      ii. High level of control
      iii. Most efficient
   b. **Cons:**
      i. Complex thermodynamic design requirements
      ii. Expensive to design, test, and build
      iii. Tankage requires additional structure work.
      iv. Turbopumps require additional testing as they are failure-prone.
2. **Solid Rocket Grain**
   a. **Pros:**
      i. Much easier to handle and store
      ii. Easier to design
      iii. Easily provides large amounts of thrust required for boost stages.
   b. **Cons:**
      i. Cannot reignite and prevents multi-step maneuver
      ii. Lower specific impulse than liquid bi-propellant
      iii. Dangerous to store as any kind of ignition source will set off the fuel grain.

3. **Hybrid Rocket**
    a. Pros:
        i. Moderate cost
        ii. Simple construction
        iii. Very safe to handle as fuel and oxidizer is separated
    b. Cons:
        i. Hard to refuel
        ii. Intermediate specific impulse
        iii. Low thrust to weight ratio

The following requirements were among the most important considered when determining what type of engine to use for each stage of our launch vehicle:

1. The launch vehicle can be built using only X quantity of available resources. This is due to the limited availability of resources.
2. The launch vehicle takes X amount of time to build. This is to ensure survivors are found as quickly as possible.
3. The launch vehicle must be able to transmit and receive signals/instructions from ground control to perform specific tasks. (such as the ability to throttle the engine)

In terms of manufacturing speed, solid rocket engines are much quicker to build, as additional structure work to the tank and turbopumps to pump liquid propellants would not be necessary.

The ability to throttle our rocket was another essential factor we had to consider. Since the main purpose of the first stage is to achieve enough altitude to start our horizontal burn, attitude adjustment is less essential, as it only needs to attain altitude. Since control is not as important of a need for this stage, we determined that a solid first stage would best match the needs laid out. Historical launch data supports this decision.

Conversely, attitude adjustment was an essential factor in determining the second stage, as we would need to control the burn such that the CubeSats achieve precise and accurate orbits. This is not impossible with a solid engine stage, but putting us into the desired orbit with a single burn after the first stage would require meticulous calculations of several parameters such as the start of the burn time, attitude of the burn, total delta V of the burn and many other values that would take far too long to accurately calculate. Even after all these calculations, a small miscalculation could cause irreversible mistakes and so the team decided that the risks involved into not having control of the vehicle for the second stage were too high and so we opted for a liquid second stage.

Hybrid rockets were not considered for either stage due to the complexity of their design. Because time is one of our most important requirements, it was essential that we develop a simple rocket that could be manufactured cheaply. For this reason we decided the a solid first stage engine and a liquid second stage engine best meets our needs.

3.4 Selection of Number of Stages

In this section we will discuss the number of stages to be used in our rocket. We decided to rate each stage based on each rocket’s initial launch mass, it’s complexity and the type of fuel we could use in each stage (such as issues like launching with liquid fuel leaves toxic waste behind at Purdue which can compromise our mission).

1. **Concept Generation**

We decided that the most available propellants to us and most effective would be solid propellant such as ammonium perchlorate in aluminum bound by rubber and liquid fuel such as hydrazine. Nuclear and electric propulsion are unproven to this point and would take much more R&D than is feasible for this urgent rescue mission.

- a. **Pros:**
    - i. Minimizes the amount of propellant that needs to be loaded onto the rocket as upper stages combined with higher isp. have less propellant mass.

ii. Hypergolic rockets are particularly easy to ignite and restart as their propellants ignite on contact, allowing is to fine-tune the final orbit of our cubesats. However, other liquid rockets can be configured to have multiple ignitions.

b. Cons

i. Total launch mass is higher due to the inefficiencies of propellants.

ii. Solid first stage core booster induces higher vibrations.

iii. High max Q from high launch acceleration, which places a heavy aerodynamic load on the structure.

2. Calculations

Considering the resources available to us we realized that ethanol and liquid methane would be most plentiful in Indiana. The ethanol can be produced by refining corn (RFA, 2019) which is widely available in Indiana. To procure methane we realized that humans produce a lot of methane (CSI, 2016). Given the copious amounts of zombies surrounding the area, we propose that the bodies be collected and stored in a chamber to gather the methane that rises. The ISP provided by liquid methane is in the range of 356-375s in space (Musk, 2016). Therefore we assumed the ISP to be 350s in our calculations.

For the solid fuel we considered the amounts of delta v required of the first stage. We chose the Space Shuttle Solid Rocket Booster which nearly has the Isp of 250 seconds. The booster itself has 237 seconds (Jenkins, 2001), but we can increase efficiency by eking it out through nozzle configurations.

In the following section we will cover the algorithms used to carve out the values for the parameters of our design including propellant masses and inertial masses.

Algorithm

- $\Delta V = \Delta V_1 + \Delta V_2 + \Delta V_3 + \cdots + \Delta V_n \quad \cdots (1)$
- $\Delta V_1 = f_1 \cdot \Delta V, \quad \Delta V_2 = f_2 \cdot \Delta V, \ \cdots\cdots, \Delta V_n = f_n \cdot \Delta V \quad \cdots (2)$
- $f_1 + f_2 + f_3 + \cdots + f_n = 1 \quad \cdots (3)$
- $$m_{\text{initial,stage1}} = m_{\text{pay}} \frac{e^{\frac{\Delta V_1}{g_o I_{\text{sp,solid}}}}(1 - f_{\text{inert,solid}})}{\left(1 - f_{\text{inert,solid}} \cdot e^{\frac{\Delta V_1}{g_o I_{\text{sp,solid}}}}\right)} \cdot \prod \left[\frac{(1 - f_{\text{inert,liquid}}) e^{\frac{\Delta V_n}{g_o I_{\text{sp,liquid}}}}}{1 - f_{\text{inert,liquid}} \cdot e^{\frac{\Delta V_n}{g_o I_{\text{sp,liquid}}}}} \right] \cdots (4)$$

Figure 3.4-2.1: Basic Equations for the calculation

Our calculations determined that the optimal number of stages for our launch vehicle is two. The first stage is a solid rocket motor and the second stage is a restartable liquid motor. The code will first assign arbitrary velocity ratios (f1 … fn in equation 3). Then using the forth equation the weight for each velocity ratio is computed, and the minimum weight along with the optimum velocity ratio corresponding to the minimum weight is identified. The next step is to exchange one of the arbitrary velocity ratios with the optimum value that we previously found and fixate the value. We iterate this until we exchange all the arbitrary velocity ratios with fixed values that correspond with minimum initial masses for each stage. From the optimum velocity ratios that we have obtained we can calculate the initial fuel mass for the rocket using a combination of equation (4) in Figure 3.4-2.1 above and the inert mass ratio. Consequently, we are able to calculate the inert masses as well (Table 3.4-2).

After this computation we obtained the initial, fuel, and inert mass for each 2-, 3-, and 4-stage rocket (Figure 3.4-2.3) From this we can see that the initial mass and propellent mass decreases proportionally to the number of stages. However, the inert mass undergoes a converse trend (Figure 3.4-2.2). With that being said, the inert mass accounting for a portion of the total mass will be greater for the 4-stage and 3-stage rocket than the 2-stage. If this is true, this means that the difference between the initial mass and final mass for the 4- and 3-stage rocket is going to be smaller (i.e. propellant mass expelled), thereby decreasing the amount of delta V attained by the mass difference ($\ln[m_i/m_f]$). Besides this, it is important to note that these computations assume ideal conditions so the values obtained for the 3- and 4-stage rockets are small and impractical to consider. In addition, the more stages constructed the more work and time will have to be devoted to build the rocket. With limited amounts of both in this post-apocalyptic world it would be reasonable to select the 2-stage rocket as most efficient and reliable.

Percent mass of Inert Mass	
2-stage	31.0%
3-stage	38.5%
4-stage	54.7%

Table 3.4-2: Table of percent mass of Inert Mass

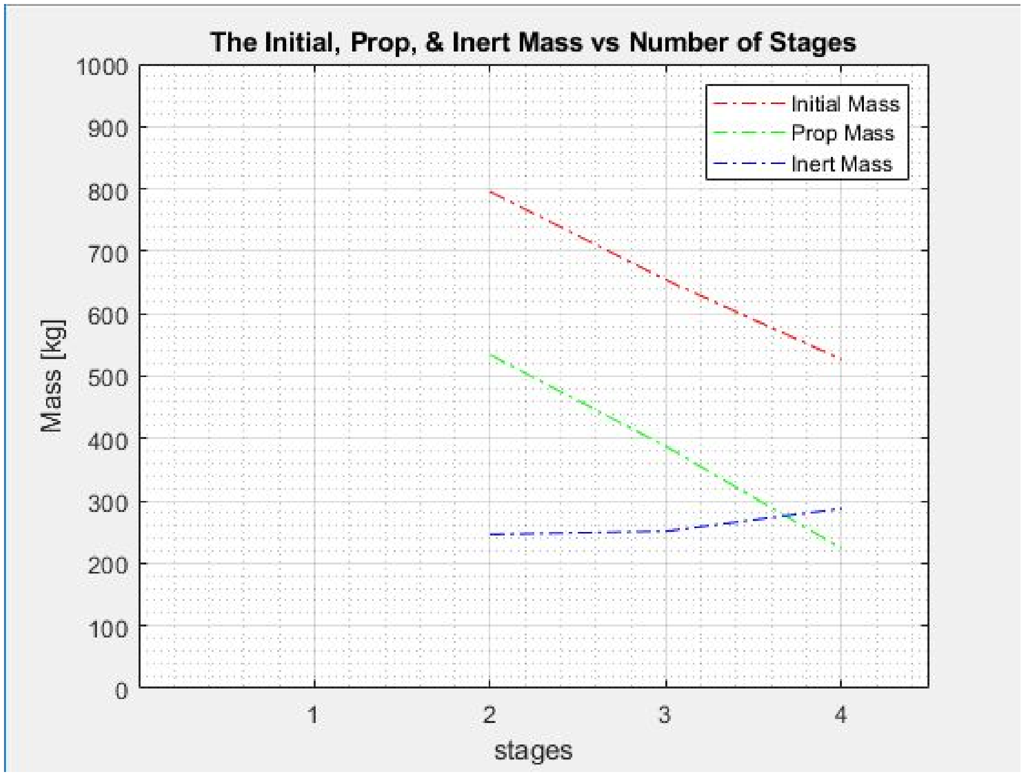


Figure 3.4-2.2: MATLAB plot

Final Results

>> The outputs that we obtain here are initial, inert, and prop masses for our 2, 3, or 4 staged rocket. These values are ideal values, that is the lowest masses we can have while still accomplishing our mission.

(1) 2-stage rocket

- Initial mass: 795.64 kg / 7805.2 N / 1754.7 lbs
- Prop mass: 534.01 kg / 5238.6 N / 1177.7 lbs
- Inert mass: 246.64 kg / 2419.5 N / 543.93 lbs

(2) 3-stage rocket

- Initial mass: 654.17 kg / 6417.4 N / 1442.8 lbs
- Prop mass: 387.60 kg / 3802.4 N / 854.85 lbs
- Inert mass: 251.57 kg / 2467.9 N / 554.83 lbs

(3) 4-stage rocket

- Initial mass: 526.76 kg / 5265.6 N / 1183.8 lbs
- Prop mass: 223.85 kg / 2196.0 N / 493.70 lbs
- Inert mass: 287.91 kg / 2824.4 N / 634.98 lbs

Figure 3.4-2.3: Final Results From Matlab

3. Computer Aided Design

For visualization of the final design, we worked in AutoCAD Inventor 2019 to create a model of the launch vehicle (Figure 3.4-3.1).

Figure 3.4-3.1: CAD Design of The Rocket

From the bottom, we added a diverging nozzle feeding from the Solid Rocket Booster (SRB). The four fins are a design feature to add lift and decrease the losses due to steering. The first stage is designed to hold the calculated amount of propellant plus five percent extra fuel for reserves. The next segment is the coupler that fastens the two stages via explosive bolts. Tucked inside the coupler, we have an optimized liquid engine that is constructed to replicate the RS-88 engine made by Rocketdyne (Farr and Sanders, 2003). This is a temporary design was used because this engine is fueled by ethanol, the most abundant fuel we have in Indiana. After finding out that ethanol fuel does not have sufficient energy density to keep the upper stage mass low, we switched to a liquid methane and liquid oxygen combination to achieve the Isp of 350 seconds. An historical example of the efficiency of liquid methane is the SpaceX raptor engine. Since methane can be made from biological matter, we can turn dead zombies and food waste into fuel. Crowning the vehicle is the payload fairing. The design is modelled after the SpaceX fairing. A pneumatic system opens the capsule from within along the vertical and the halves peel off as the vehicle ditches the first stage (SpaceX News, 2013). The second stage will propel the payload into the final orbit.

4 Conclusions

The impetus for Daedalus IV was the advent of the inevitable zombie apocalypse. The humans surviving in West Lafayette have decided to search for other signs of life that may be in need of immediate assistance. They have come to us, Jason & The Argonauts, to develop a rocket to launch CubeSats fitted with thermal sensors. Above, we laid out our solution for this design challenge. The future of human survival on this planet may depend on the integrity of our proposal of the launch vehicle Daedalus IV. In this final section we review our decisions and our methods to make the Daedalus IV a viable option for the citizens of West Lafayette.

4.1 Design Evaluation

The Daedalus IV will achieve mission success. The design for a two stage, one solid, one liquid rocket is backed by hard math and science as can be seen in our code below. It meets the necessary delta-V requirement can carry up to three CubeSats. Not only is our launch vehicle bound for success, the proposed payload system is as well. A constellation of three satellites will allow us to efficiently scan the earth with built in redundancy. Furthermore, our choices of rocket fuels and construction allows us to produce and launch this rocket as quickly as possible. The only way this could be seen as a failure is if the costs become outrageous and its makes no sense to fly the rocket. Instead, sending planes with thermal imaging systems would get the job done as well, albeit this has limited range.

The final risks we are considering in this design are limited. Should the zombie’s swarm West Lafayette, we have no immediate solution on how to protect the rocket and at that point, we need to fight for our own lives rather than the potential survivors outside of the area. Another risk that we are concerned about is the logistics behind the rocket construction. We want it to proceed as fast as possible, but we do not wish to speed up to the detriment of the integrity of the rocket. This final risk will hopefully be balanced out by supervisors of the project construction.

To summarize, we believe that the Daedalus IV meets the requirements and therefore the needs of the citizens of West Lafayette. We addressed the need for finding survivors by designing a sound rocket to send up thermal imaging cameras into orbit. We weighed all of our options on how to deliver this payload and decided on the delivery of three CubeSats in order to quickly and iteratively scour the Earth.

4.2 Next Steps

Many different steps can be taken from this point forward. If we were given the chance to continue this project for another semester, we would attempt to build and test some prototypes of the mission. The first thing that we could do is make a subscale version of our rocket. We could use fiberglass and epoxy for the structure, and motor that could represent a scaled version of the thrust we need. Next, we could launch a dummy payload that simulated the weight of three CubeSats. From there we could measure the apogee and determine whether our calculations conquer with reality.

Another aspect of the project we could work on if we dedicated another semester, would be building and testing a prototype CubeSat of the same technical capabilities as the ones we hypothetically have at our disposal. To do so we could purchase some Arduino’s and some sheet metal to machine our parts. We would need to learn about signal processing and sending and receiving transmissions. However, there are several methods of doing so and if we have trouble we could ask fellow electrical engineers for some assistance. For the thermal imaging cameras, we could purchase and install them. The primary testing would test the distance from which it could send and receive a signal. The secondary test would be the quality of the results. Since we have three satellites configured in our design, we use them to verify one another and check for accuracy that way.

4.3 Lessons Learned

To design Daedalus IV, we learned about everything it takes to build a rocket starting with mission requirements. We wrote and rewrote our mission needs and requirements on several occasions using the methods we learned in class and improving upon each iteration. This led us down the right path to design. With reasonable constraints in mind, we had to figure out how high we needed to get our payload and into what orbit. We learned about the Vis-Viva equation for orbit determination. Then we learned about calculating the delta-V we needed to get into that orbit and all the forces acting with us and against us. Knowing this information we learned that the there are multiple ways of achieving this launch through staging. For this we derived the equation for the delta-V in terms of the numbers of stages. To build onto this information, we learned that we had a choice of propellants that produced different amounts of thrust and required different quantities. We learned the pros and cons of each. After analyzing the amount of stages we needed we finally selected a solid booster for our initial blast off and a variable liquid engine for the second stage.

On the administrative side of the project, we learned that setting a standard group meeting time will not always be successful. Teammates may be committed to something else for one meeting so less get accomplished. Additional meetings throughout the week will most definitely be required especially when we approach deadlines.

To communicate, we used GroupMe. This worked out well enough although at times some would not respond due to the overwhelming amount of groups they are in. That is why getting each other's numbers or other means of communication is important. It is just another form of redundancy.

Google Docs worked out great for the project. We could see what the other was working on and quickly jump to his location if help was needed. We could also leave comments on the side to address things at a later date.

References

Clark, S. (2015). NASA: “Tracking CubeSats is easy, but many stay in orbit too long”. *Spaceflight Now*. Retrieved from https://spaceflightnow.com/2015/07/30/nasa-tracking-cubesats-is-easy-but-many-stay-in-orbit-too-long/

CSI(2016) “Methane and the Carbon Cycle”, *Climate Science Investigations*, Retrieved from http://www.ces.fau.edu/nasa/module-4/causes/methane-carbon-dioxide.php

Dunbar, Brian, Jenner, Lynn (2018). NASA. “GOES Network Satellite”. Retrieved from https://www.nasa.gov/content/goes

Farr, R. & Sanders, T. (2003). “RS-88 Pad Abort Demonstrator Thrust Chamber Assembly Testing at NASA Marshall Space Space Flight Center, November - December 2003”, *RS-88 PAD Testing at NASA MSFC May 2005, AIAA,* Retrieved from https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20050237017.pdf

Film School Projects (2013). “How Long Would the Zombie Apocalypse Really Last?”. Retrieved from https://filmschoolrejects.com/how-long-would-the-zombie-apocalypse-really-last-fcd71daf61c8/

GMAT Forums. (2011). Retrieved from http://forums.gmatcentral.org/viewtopic.php?f=2&t=605

Humble, R. W., Henry, G. N., & Larson, W. J. (2007). *Space propulsion analysis and design*. New York: McGraw-Hill.

Jenkins, D. (2001) “STS Data Sheet”, *Shuttle Reference Manual*, Retrieved from http://www.spacelaunchreport.com/sts.html

“Joint Polar Satellite System (JPSS).” National Oceanic and Atmospheric Administration. Retrieved from https://www.jpss.noaa.gov/launch.html

Kelly D, Moses, P., E. Roy, W. Malone, Jr. (2004). “Development of Risk Assessment Matrix for NASA Engineering and Safety Center” from https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20050123548.pdf

Lam, Matthew (2019). ENGINE LIST 1. (n.d.). Retrieved from http://www.projectrho.com/public_html/rocket/enginelist.php

Musk, Elon (27 September 2016). "SpaceX IAC 2016 Announcement" (PDF). *Mars Presentation*. SpaceX. Archived from the original (PDF) on 28 September 2016. Retrieved 27 September 2019.

NASA, Retrieved April 24, 2019, from https://history.nasa.gov/conghand/propelnt.htm

Patrinas, Thalia (2018). “Amazing Earth: Satellite Images From 2018”, NASA. Retrieved from: https://www.nasa.gov/feature/amazing-earth-satellite-images-from-2018

“Polar Orbiting Environmental Satellites (POES).” San Francisco State University. Retrieved from http://tornado.sfsu.edu/geosciences/classes/m407_707/Monteverdi/Satellite/PolarOrbiter/Polar_Orbits.html

RFA (2019) “How is Ethanol Made?”, *Renewable Fuels Association*, Retrieved from https://ethanolrfa.org/how-ethanol-is-made/

Sauser, B. (2009). “What’s the Deal with Rocket Vibrations”. *MIT Technology Review*. Retrieved from: https://www.technologyreview.com/s/414364/whats-the-deal-with-rocket-vibrations/

Sellers, J. (2004).“Understanding Space”, *Chapter 3, The Space Environment,* Retrieved from Blackboard Learn Purdue and from file:///C:/Users/welli/Downloads/epdf.tips_understanding-space-an-introduction-to-astronautic.pdf

Sen Corporation Ltd. Retrieved from: https://sen.com/features/cubesats-good-things-that-come-in-small-packages

Spaceflight101(2019) “SpaceX Raptor”, Retrieved from http://spaceflight101.com/spx/spacex-raptor/

Tablestore, A. (2019). *Data - Page 1 — Allen Institute for Artificial Intelligence*. [online] Allenai.org. Available at: https://allenai.org/data/data-all.html [Accessed 25 Apr. 2019].

“Weather Satellites - Software.” *SDR for Mariners”*, sdrformariners.blogspot.com/p/weather-satellites-software.html.

Appendix: Matlab Code

>> this is the function matlab code to figure out the optimum initial mass of n-staged rocket

```
function [m_i_final, fopt_vec, f_holder, m1_vec, m_prop, m_inert] =...
    n_staged_rocketInitialMass_cal(number_of_stages, total_delta_V, f_inert_solid,...
    f_inert_liquid, Isp_solid, Isp_liquid, payload_mass, f1)

% Assigning simpler variable names to the input varialbes for simplification
n = number_of_stages; % The number of stages of the rocket
Vtot = total_delta_V; % The total velocity required for the rocket
f_inert_s = f_inert_solid; % The inert mass ratio of the solid fuel stage
f_inert_l = f_inert_liquid; % The inert mass ratio of the liquid fuel stages
Isp_s = Isp_solid; % The specific impulse of the solid fuel stage
Isp_l = Isp_liquid; % The specific impulse of the liquid fuel stage
m_pay = payload_mass; % The estimated payload
g_o = 9.81; % The gravitational acceleration
f_holder = f1; % Storing f1 array

% Initializing the necessary values and the vectors
fopt = 1;  % The optimized velocity ratio
fopt_vec = zeros(n, 1); % The vector to store the optimized velocity ratio for each stage
m1_min_vec = zeros(n,1); % The vector to store the minimum estimated initial mass for the
                         % corresponding optimized velocity ratio
m1_vec = zeros(size(f1)); % The  vector to store the masses for the corresponding
                          % arbitrary velocity ratios
limit = 1;
% Start Outer loop
for v = 1:n
    % Condition to restrict loop
    if v ~= n
        % Inner loop
        for x = 1:length(f1)
            % Calling the function to calculate the initial mass
            m1 = change_symprod(x, v, total_delta_V, m_pay,...
        f_inert_solid, f_inert_liquid, Isp_solid, Isp_liquid, f1, fopt_vec,...
        number_of_stages);
            % Append the mass value into vector
            m1_vec(x) = m1;
        end % End of inner loop

        % Finding the minimum mass for stage 1
        % Omitting the negative values
        m1_vec = m1_vec(m1_vec > 0);
        % Identifying the minimum value
        m1_min = min(m1_vec);
        % Storing the minimum value into vector
        m1_min_vec(v) = m1_min;
        % The index of this minimum value
        idx_min = find(m1_vec == m1_min);
        % The corresponding f1 value
        f1_min = f1(idx_min);

        % Designating the optimized velocity ratio
```

```
        fopt = f1_min;
        % Storing the optimized velocity ratio
        fopt_vec(v) = fopt;

        % Adjusting f1
        limit = limit - fopt;
        f1 = f1(f1 < limit);

    else
        break;
    end % End of if-else statement

end % End of outer loop

% Appending the remaining n-th optimized velocity ratio to the vector
% Initialize
fopt_n = 1;
for z = 1:(length(fopt_vec)-1)
    fopt_n = fopt_n - fopt_vec(z);

    if z == length(fopt_vec)-1
        fopt_vec(z+1) = fopt_n;
    end % End of if-else statement

end % End of for loop

% Computing the initial mass for the optimized velocity ratios
m_i_final = m_pay;
for i = 1:length(fopt_vec)
    if i == 1
        a = exp(fopt_vec(i) * Vtot / g_o / Isp_s);
        m_i_final = m_i_final * a * (1 - f_inert_s) / (1-f_inert_s*a);
    else
        b = exp(fopt_vec(i) * Vtot / g_o / Isp_l);
        c = ( 1 - f_inert_l * b);
        m_i_final = m_i_final * b * (1 - f_inert_l) / c;
    end % End of if-else statement
end % End of loop

% Computing the total propellent mass of the rocket
% Initializing the mass
m_start = m_pay;
for count = n:-1:1
    if count ~= 1
        P = exp(fopt_vec(count) * Vtot / g_o / Isp_l);
        Q = 1 - f_inert_l;
        R = P - 1;
        S = 1 - f_inert_l * P;
        % Propellent mass for the n-th (n = 2, ...) stage
        m_prop = m_start * R  * Q / S;
        % Get the inert mass
        m_inert = f_inert_l / Q * m_prop;
        % Get the initial mass for the stage
```

```
        m_i_stage = m_pay + m_inert + m_prop;
        % Replace the next starting/initial mass with the m_i_stage
        m_start = m_i_stage;
    else
        P = exp(fopt_vec(count) * Vtot / g_o / Isp_s);
        Q = 1 - f_inert_s;
        R = P - 1;
        S = 1 - f_inert_s * P;
        % Propellent mass for the n-th (n = 2, ...) stage
        m_prop = m_start * R  * Q / S;
        % Get the inert mass
        m_inert = f_inert_s / Q * m_prop;
        % Get the initial mass for the stage
        m_i_stage = m_pay + m_inert + m_prop;
        % Replace the next starting/initial mass with the m_i_stage
        m_start = m_i_stage;
    end
end

% Finally getting the values of the inert mass
m_inert = m_i_final - m_prop - m_pay;

end % End of function
```

>>function that manipulates the product series for the optimization system equation

```
function m_i = change_symprod(inner_count, outer_count, total_delta_V, m_payload, ...
    f_inert_solid, f_inert_liquid, Isp_solid, Isp_liquid, f1, fopt_vec, number_of_stages)
% Preparations
g_o = 9.81; % Gravitational acceleration
Vtot = total_delta_V; % Total velocity required
f_inert_s = f_inert_solid; % inert mass ratio for the solid fuel
f_inert_l = f_inert_liquid; % inert mass ratio for the liquid fuel
Isp_s = Isp_solid; % The specific impulse for the solid fuel
Isp_l = Isp_liquid; % The specific impulse for the liquid fuel
n = number_of_stages; % The number of the stages
m_pay = m_payload; % The estimated payload for the rocket

% Fixiating the velocity ratio as one arbitrary value
f1_k = f1(inner_count);
% Define a system equation to solve
syms k
% Designate the the coefficients to a simpler term for clarity
% For the coefficients for the liquid fuel
A = exp((1 - f1_k * k) * Vtot / g_o / Isp_l);
B = (1 - f_inert_l * A);
% The coefficients for the solid fuel
C = exp(f1_k * Vtot / g_o / Isp_s);
% Define the equation for the product series with the arbitrary velocity ratio
% 'f1' used
eqn = (1 - f_inert_l) * A / B;

% The loop to create the proper equation for the initial mass
% For each 'outer counter', one of the arbitrary 'f1_k' term (a
% value inside the f1 vector) will be replaced with a optimized velocity
% ratio value of 'fopt'. This process will enable to retrieve the
% next optimized velocity ratio value.
if outer_count == 1
    % The initial mass equation for the first optimization process
    m_i = (C * m_pay * (1 - f_inert_s) / (1 - f_inert_s * C)) * symprod(eqn, k, 1, n-1);
else
    % We start obtaining 'fopt' values so we will replace the arbitrary 'f1_k'
    % with the 'fopt'

    % Initialize the equation as 1 to allow to compound for iteration
    eqn_fix = 1;
    for x = 1:(outer_count-1)
        % Taking out one 'fopt' value
        fopt_x = fopt_vec(x);
        if x == 1
            % Replacing the solid fuel component of the equation
            C_opt = exp(fopt_x * Vtot / g_o / Isp_s);
        else
            % Replacing the liquid fuel component of the equation
            A_opt = exp(fopt_x * Vtot / g_o / Isp_l);
            B_opt = (1 - f_inert_l * A_opt);
```

```
            % Fixating the new equation for the product series
            eqn_fix = eqn_fix * A_opt * (1 - f_inert_l) / B_opt;
        end % End of if-else statement

    end % End of inner for loop

    % Combining the solid and liquid fuel components for the new equation
    eqn_fix = eqn_fix * (C_opt * m_pay * (1 - f_inert_s) / (1 - f_inert_s * C_opt));
    % Manipulated eqn
    A_alpha = exp((1 - sum(fopt_vec) - f1_k * k) * Vtot / g_o / Isp_l);
    B_alpha = (1 - f_inert_l * A_alpha);
    A_beta = exp((f1_k) * Vtot / g_o / Isp_l);
    B_beta = (1 - f_inert_l * A_beta);
    eqn_alpha = (1 - f_inert_l) * A_alpha / B_alpha;
    % Defining the new initial mass equation using the 'fopt'
    m_i = eqn_fix * A_beta * (1 - f_inert_l) / B_beta * symprod(eqn_alpha, k, 1,...
        n-outer_count+1);
end % End of outer for loop

end % End of function
```

>> Execution

2-stage

```
% Setting the arbitary velocity ratios to get the optimized velocity ratios
f2 = 0.1:0.001:0.57;
% Calling the fuction to compute the
% 1: The Initial mass of the first stage of the rocket
% 2: The vector containing the optimized velocity ratios for each stage
% 3: The rearranged arbitary velocity ratios to use later for the graphs
% 4: The masses corresponding to the output #3
[m2_stage, fopt_vec2, f2_new, m1_vec2, m_prop2, m_inert2] = ...
    n_staged_rocketInitialMass_cal(2, 9504, 0.1, 0.07, 250, 350, 15, f2);
fprintf('2-stage done');
```

```
2-stage done
```

3-stage

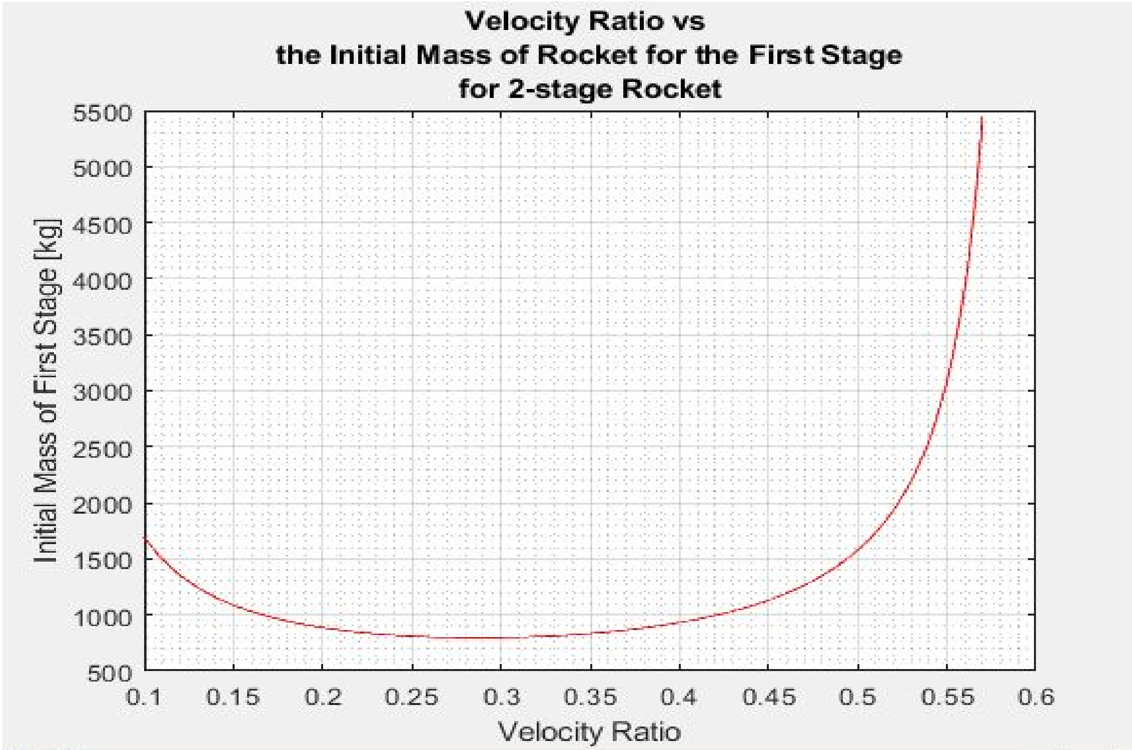
```
f3 = 0.05:0.001:0.35;
[m3_stage, fopt_vec3, f3_new, m1_vec3, m_prop3, m_inert3] = ...
    n_staged_rocketInitialMass_cal(3,9504, 0.1, 0.07, 250, 350, 15, f3);
fprintf('3-stage done');
```

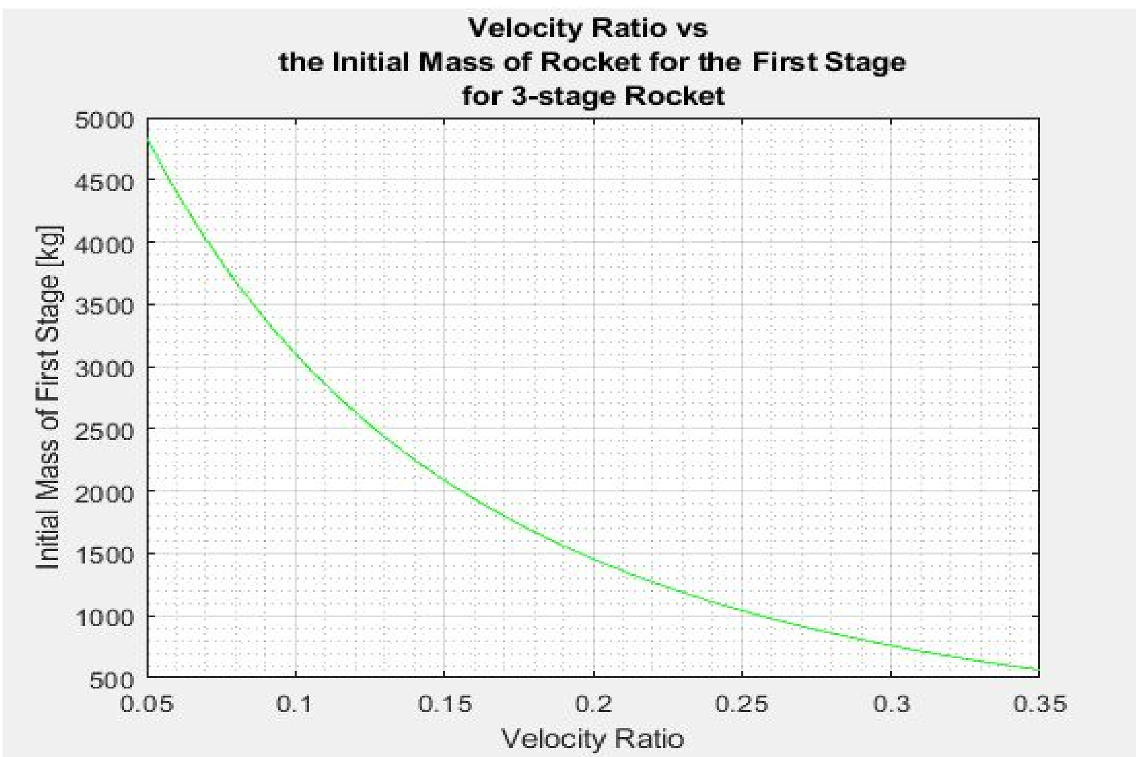
```
3-stage done
```

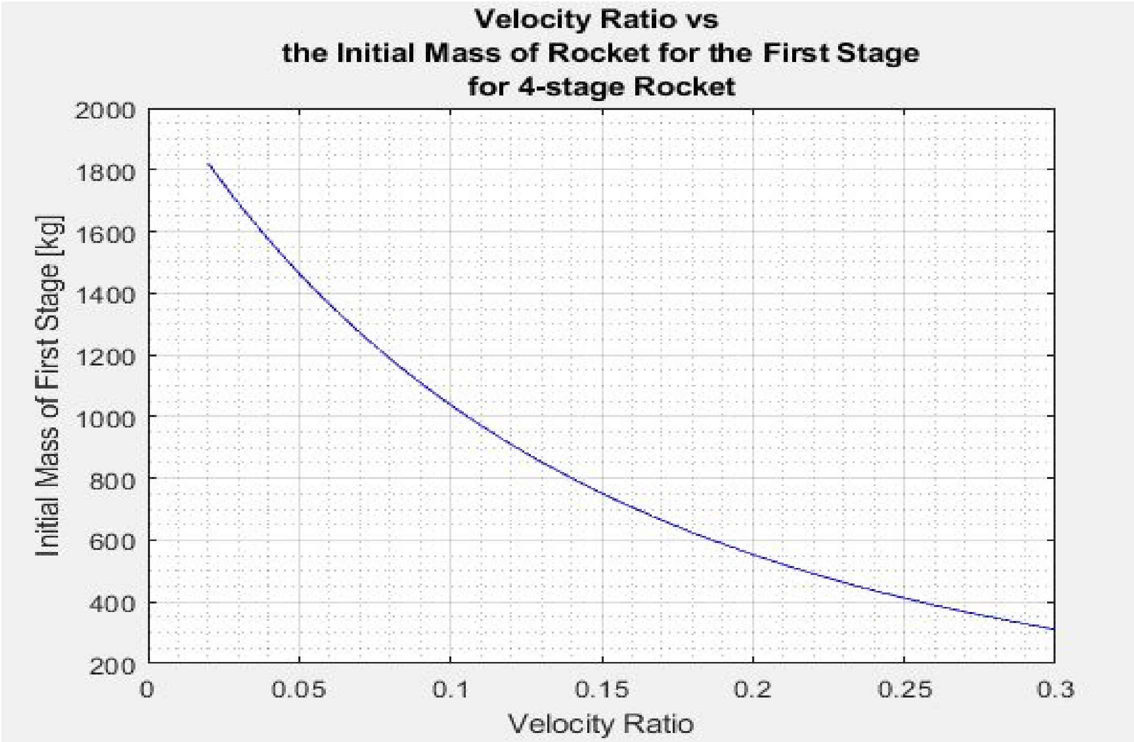
4-stage

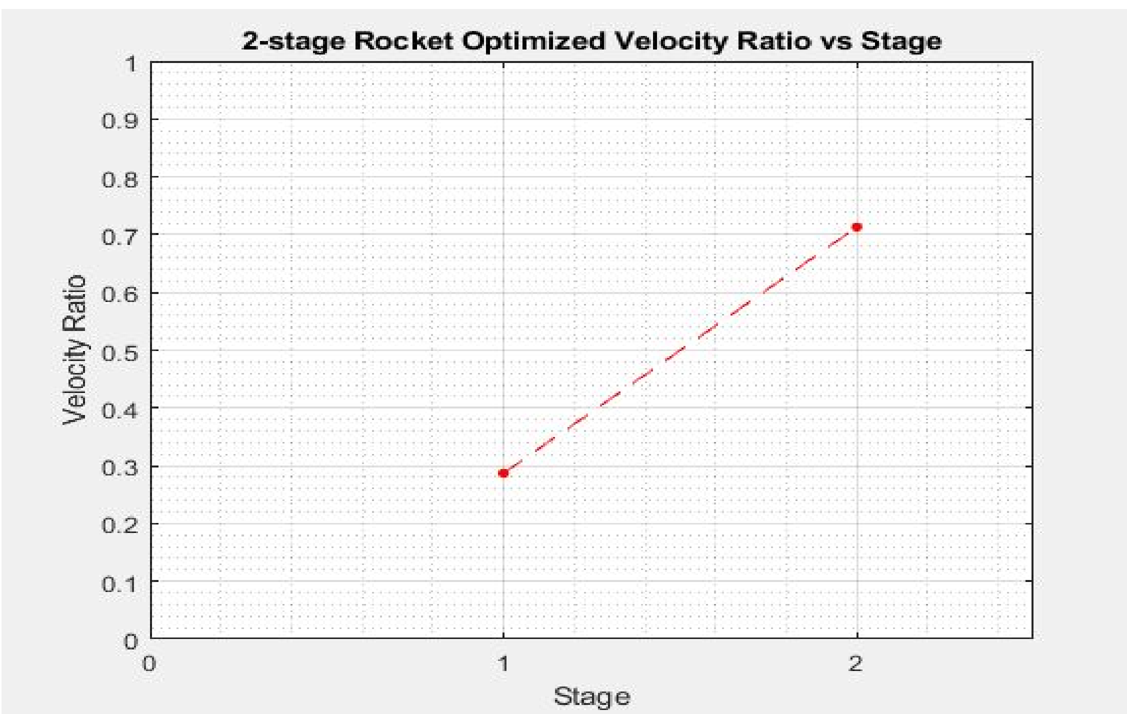
```
f4 = 0.02:0.001:0.3;
[m4_stage, fopt_vec4, f4_new,  m1_vec4, m_prop4, m_inert4] = ...
    n_staged_rocketInitialMass_cal(4, 9504, 0.1, 0.07, 250, 350, 15, f4);
fprintf('4-stage done');
```

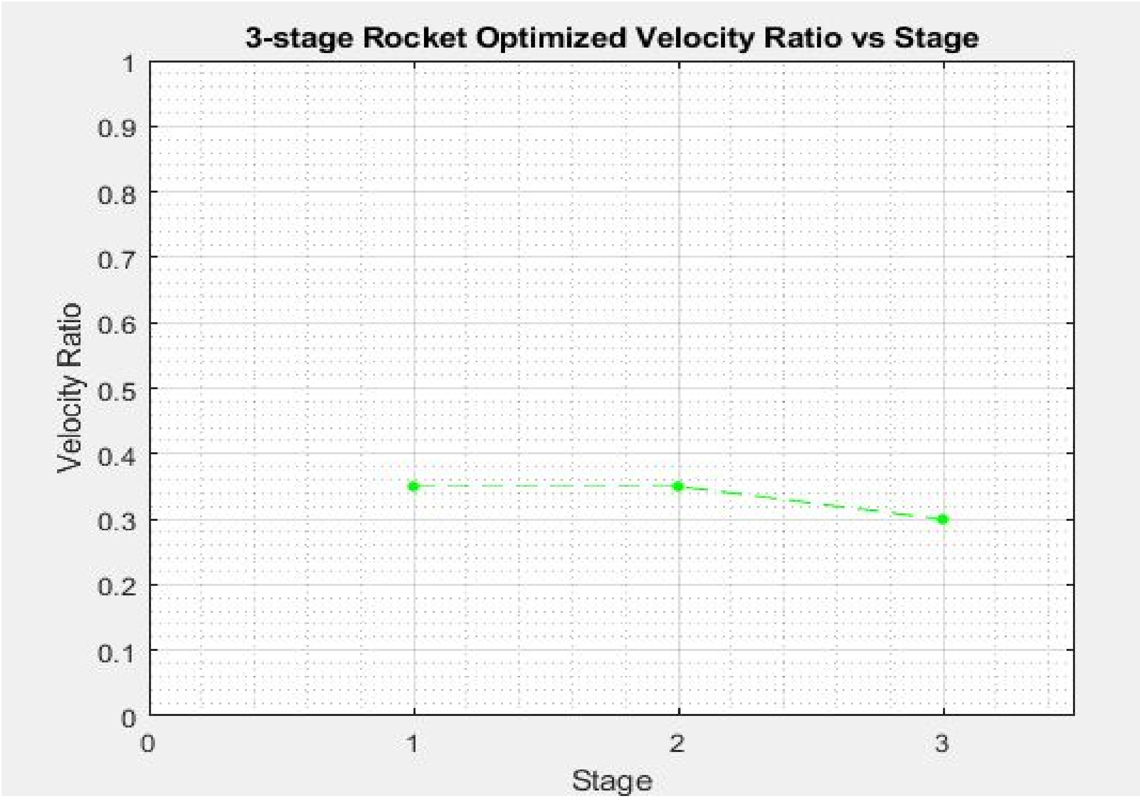
```
4-stage done
```

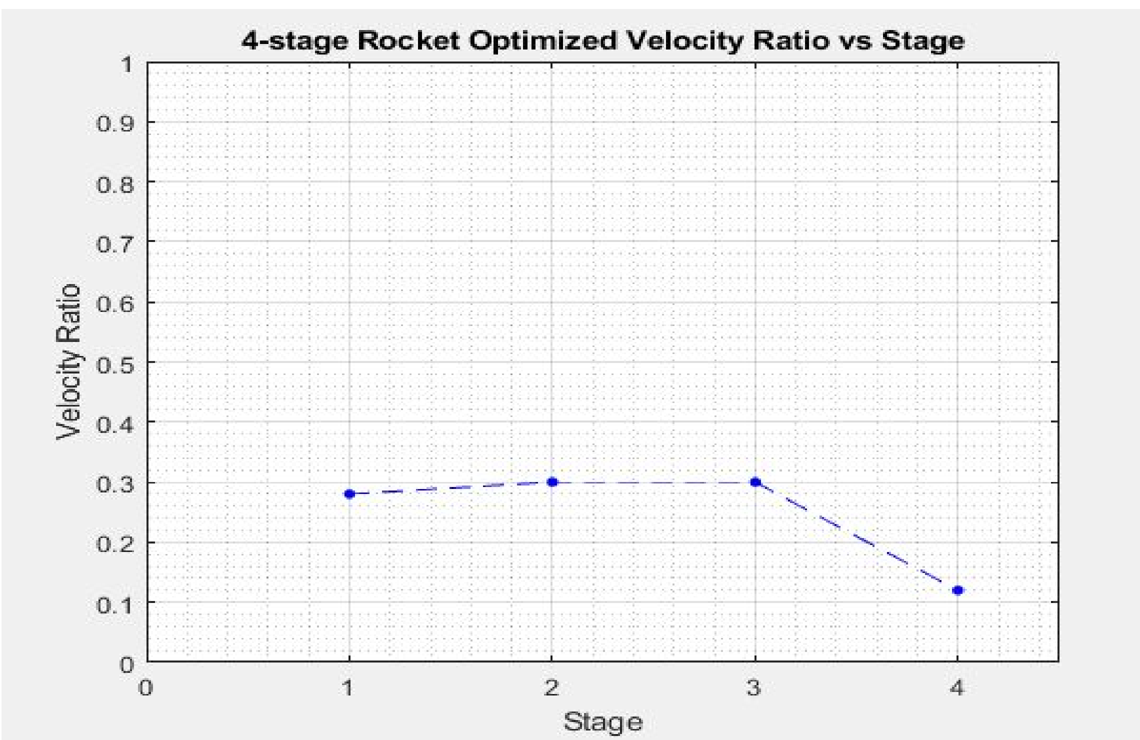

Velocity Ratio vs
the Initial Mass of Rocket for the First Stage
for 2-stage Rocket
Initial Mass of First Stage [kg]
Velocity Ratio
5500
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.55
0.6

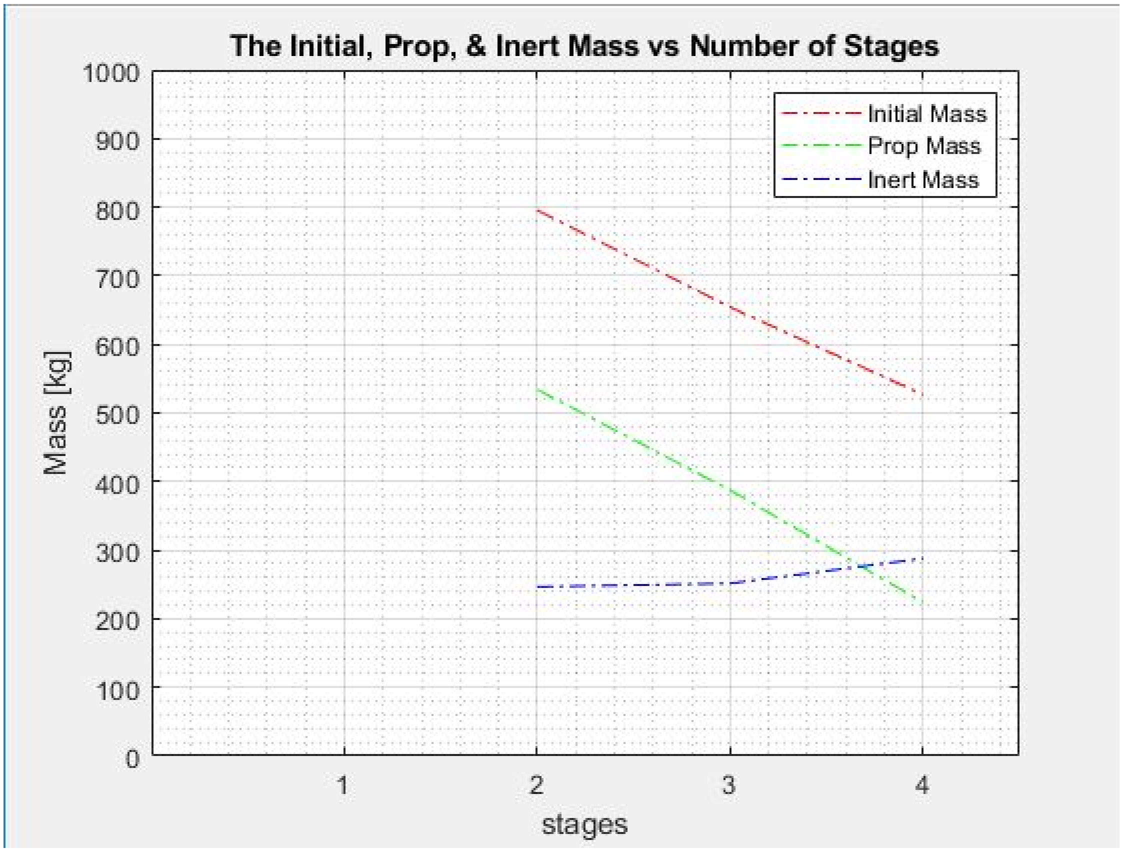

Velocity Ratio vs
the Initial Mass of Rocket for the First Stage
for 3-stage Rocket
Initial Mass of First Stage [kg]
Velocity Ratio
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0.05
0.1
0.15
0.2
0.25
0.3
0.35


Velocity Ratio vs
the Initial Mass of Rocket for the First Stage
for 4-stage Rocket
Initial Mass of First Stage [kg]
2000
1800
1600
1400
1200
1000
800
600
400
200
0
0.05
0.1
0.15
0.2
0.25
0.3
Velocity Ratio


2-stage Rocket Optimized Velocity Ratio vs Stage
Velocity Ratio
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
1
2
Stage


3-stage Rocket Optimized Velocity Ratio vs Stage
Velocity Ratio
Stage
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1
2
3


4-stage Rocket Optimized Velocity Ratio vs Stage
Velocity Ratio
Stage
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1
2
3
4


The Initial, Prop, & Inert Mass vs Number of Stages
Initial Mass
Prop Mass
Inert Mass
Mass [kg]
0
100
200
300
400
500
600
700
800
900
1000
1
2
3
4
stages

Appendix: Weekly Update Emails

1/13

Team R06 got organized this week by finding a common time slot during which we could meet for 2 hours. We also created a GroupMe and shared a Google Drive folder.

Have a great weekend.

1/21

This week we banged out our stakeholders, needs, requirements, and mission statement. We also began work on our VoTW. Sorry this did not send. For some reason I kept it in my drafts and did not hit the send button.

1/27

This week we worked a lot on developing our VotW presentation and broke ground on the report. We also completed our initial risk assessments.

2/1

This week team R06 worked on concept generation.

2/17

This week we spent the majority of our meetings to polish up our VotW as our presentation day is on Thursday. We also added some final touches to the VotW report looking for any ways we could improve it and fixing any grammatical errors. Due to this we have not made as much progress on the team project and for that reason will be meeting up twice next week.

2/26

Apologies for the late update. This week we finished the votw presentation and finally presented it in class. We also fixed all the suggestions made to us from our team's first project report and submitted it in the second team project update.

3/3

This week team R06 worked on the video script, engine selection, and addressing the comments made by professor Marais.

3/10

This week our team began working on the re submission of our VoTW by using the suggestions and feedback given back to us on our report.

3/22

This week we worked on the project update by addressing comments. Also we worked on the VotW report to resubmit it. We will continue to work on our project this week.

3/30

This week we worked on submitting the vehicle of the week report for regrade and the project update.

4/12

This week, we worked extensively on the report. Tomo wrote code at the bottom and we edited the report accordingly. We shall submit it as a PDF to the review today.

4/21

This team our week began recording videos for final project video and also addressed the comments made by both Nicoletta and by the aircraft team reviewing our project.