Booster Rocket Design Analysis

AAE 562 Final Report

Joseph Marsh

Mar 27, 2024

Oregon State University
College of Engineering

DISCLAIMER

To aid this project, I used generative AI to help with accessing and citing peer reviewed sources, formatting code, and formatting LaTeX.

ABSTRACT

The report will compare six total designs for a new booster rocket. We will include two liquid engine designs, two solid rocket motor designs, and two hybrid rocket designs. Under the following constraints – ideal velocity change of 2300 meters per second, maximum 100 second burn time, 100,000-kilogram payload, and a three-meter diameter – we must pick the ideal design to move forward with.

In the preliminary design consideration, many variables, such as specific impulse (Isp), characteristic velocity (c*), specific heat ratio (gamma), oxidizer/fuel ratio(O/F), and thrust coefficient (Cf), were found using NASA's CEA tool. When using NASA's CEA tool, we used a few common inputs to ensure consistency between designs. We assumed optimal expansion at sea level, an initial chamber pressure of 1,000 PSIA, and iterated the optimal oxidizer/fuel ratio in increments of 0.1. Additionally, we completed our analysis in a frozen NFZ 2 scenario.

For the liquid and solid rockets, we determined optimal oxidizer/fuel ratio. For the hybrid design, we determined optimal average oxidizer/fuel ratio and the overall oxidizer/fuel shift over the course of the burn. For all designs, we determined the average specific impulse at sea level, necessary motor dimensions, nozzle dimensions, propellant mass specifics, propellant tank dimensions. Additionally, we plotted the trajectory of the rocket, including altitude, velocity, and acceleration throughout the burn.

All mathematical analysis was completed in Jupyter Notebook.

TABLE OF CONTENTS

Contents

DISCLAIMER 1
ABSTRACT 2
TABLE OF CONTENTS 3
1 Liquid Rocket Engine 2
1.1 Assumptions 2
1.2 Liquid Engine 1: Liquid Oxygen (LO2) and RP-1 2
1.2.1 Required Design Variables 2
1.2.2 Plotting Trajectories 5
1.3 Liquid Engine 2: Liquid Oxygen (LO2) and Liquid Hydrogen(LH2) 6
1.3.1 Required Design Variables 6
1.3.2 Plotting Trajectories 9
2 Solid Rocket Motor 10
2.1 Assumptions 10
2.2 Solid Rocket Motor 1: Aluminum Fuel/Ammonium Perchlorate Oxidizer with Butadiene (HTPB) Binder 10
2.2.1 Required Design Variables 10
2.2.2 Plotting Trajectories 13
2.3 Solid Rocket Motor 2: Magnesium Fuel/Ammonium Nitrate Oxidizer with Butadiene(HTPB) Binder 14
2.3.1 Required Design Variables 14
2.3.2 Plotting Trajectories 17
3 Hybrid Rocket Engine 19
3.1 Assumptions 19
3.2 Hybrid Engine 1: HTPB Fuel/LOX Oxidizer 19
3.2.1 Required Design Variables 19
3.2.2 Plotting OF Shift and Other Useful Quantities 22
3.2.3 Plotting Trajectories 23
3.3 Hybrid Engine 2: N2O Oxidizer/Paraffin Fuel 24
3.3.1 Required Design Variables 24
3.3.2 Plotting OF Shift and Other Useful Quantities 27
3.3.3 Plotting Trajectories 28
4 Design Comparisons 29
4.1 Liquid Rocket Engine Design Comparisons 29
4.2 Solid Rocket Motor Design Comparisons 30
4.3 Hybrid Motor Design Comparisons 32
4.4 Final Design Decision 33
5 REFERENCES 35
6 APPENDICES 38

1 Liquid Rocket Engine

1.1 Assumptions

- Our average thrust will be exactly equal to the minimum thrust required for the rocket, equal to two times the total propellant mass.
- Our design parameters follow optimal expansion at sea level.
- Our final trajectory parameters are based on a model excluding drag losses, but including gravitational losses.
- Values from CEA are taken as accurate values and are used in design analysis.
- Our propellant volume is identical to the volume of the tank they inhabit.
- Propellant tanks are cylindrical.
- Tank diameters are exactly at the constraint limit.
- Tank thickness is negligible enough to not cause a considerable impact on overall tank diameter.
- Tanks are made of aluminum.
- Mass fraction of our rocket is 0.9 .
- Initial chamber pressure is 1000 PSIA.
- Mach number in the chamber is 0.1 .
- Mach number at the throat is 1.
- Characteristic lengths are the mean of characteristic length ranges provided in lecture.
- Nozzle is conical with opening angle of 15 degrees.

1.2 Liquid Engine 1: Liquid Oxygen (LO2) and RP-1

For our first propellant combination, we chose liquid oxygen and RP-1. This is one of the most common liquid propellants, and is used in rockets such as the SpaceX Falcon 9, and some older Atlas rockets [1]. It is a great combination of efficiency and simplicity. Since it is so widely used, the impacts of LOX/RP-1 are well known. Additionally, RP-1 is easily stored at room temperature, so there are no extra requirements for storing RP-1 cryogenically.

1.2.1 Required Design Variables

CEA Output		
Variable	**Value**	**Units**
C*	1807.2	Meters per second
Gamma (γ)	1.15	Unitless
Optimal Oxidizer/Fuel Ratio	2.3	Unitless
Specific Impulse	289.4	Seconds
Thrust Coefficient (C_f)	1.57	Unitless

This liquid propellant combination has a reasonably low optimal oxidizer/fuel ratio, and a reasonably high specific impulse, which indicates promise for our first rocket design rocket design.

Propellant Parameters		
Variable	**Value**	**Units**
LOX Density	1141	Kilograms per Cubic Meter
RP-1 Density	804.6	Kilograms per Cubic Meter
Characteristic Length (L*)	1.15	Meters
Total Propellant Mass	145018	Kilograms
Oxidizer Mass	101073	Kilograms
Fuel Mass	43945	Kilograms
Total Burn Time	80.34	Seconds

Liquid Oxygen was found from [2], and RP-1 Density was found from [3], and taken at 293 Kelvin. Characteristic length was determined from the average characteristic lengths provided in class for the LOX/RP-1 combination.

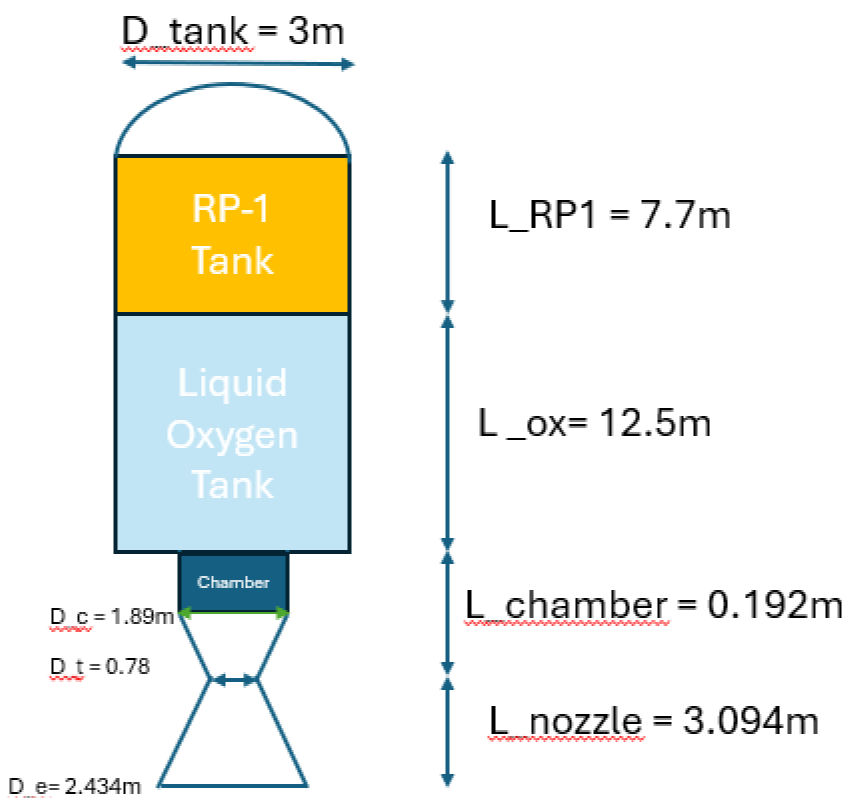
Engine and Nozzle Dimensions		
Variable	**Value**	**Units**
Throat Diameter	0.776	Meters
Exit Diameter	2.434	Meters
Chamber Diameter	1.899	Meters
Minimum Required Chamber Length	0.192	Meters
Nozzle Length	3.094	Meters

All our parameters are within required parameters. Our minimum chamber length seems quite low. However, the chamber length generally does not need to be long, which means we can increase the true chamber length. Nozzle length is reasonable, given the size and thrust required of the booster we're tasked with designing.

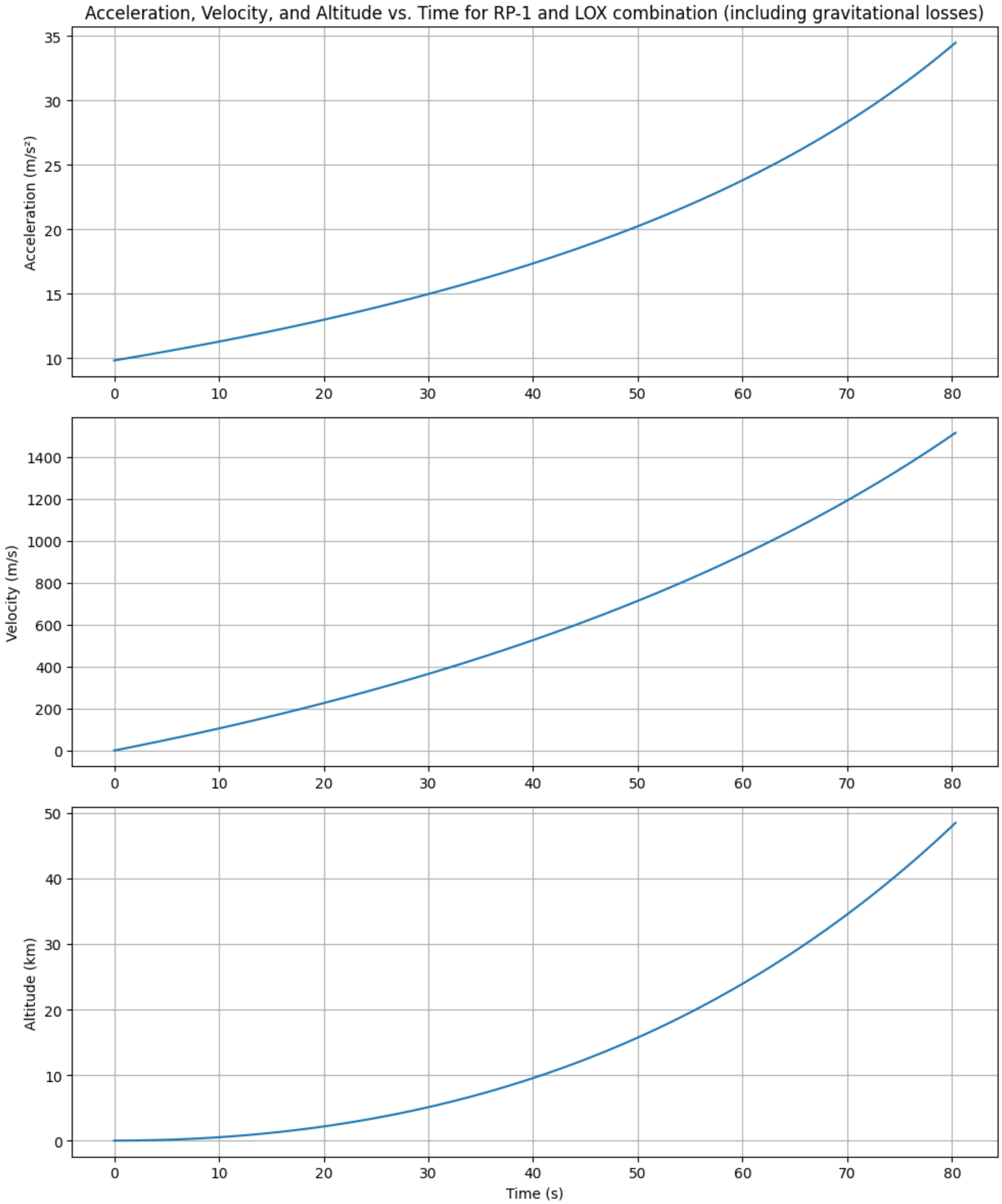
Overall Parameters and Dimensions		
Variable	**Value**	**Units**
Ultimate Tensile Strength	950	MPA
Booster Structural Mass	16113	Kilograms
Tank Diameter	3	Meters
Oxidizer Tank Volume	88.6	Cubic Meters

Fuel Tank Volume	54.6	Cubic Meters
Total Tank Volume	143.2	Cubic Meters
Oxidizer Tank Length	12.53	Meters
Fuel Tank Length	7.73	Meters
Total Tank Length	20.26	Meters
Required Tank Thickness	73.5	Millimeters

Our required tank thickness is low enough that our tank diameter assumption is reasonable. It is less than 2% of the total Tank Diameter. Additionally, our total Tank length is reasonable, if not a little low, given our Thrust requirements. This suggests that if we need to increase our delta V requirements and keep our total tank diameter equal, we can simply elongate the tanks, which will allow more propellant and thus, more total delta V.



1.2.2 Plotting Trajectories



Velocity here only goes to around 1500 meters per second. It is reasonable for the velocity to not reach 2300 since We are also considering gravitational losses. Longer burn times mean more gravitational losses.

1.3 Liquid Engine 2: Liquid Oxygen (LO2) and Liquid Hydrogen(LH2)

For our second propellant combination, we chose liquid oxygen and liquid hydrogen. I chose this combination because it has the highest theoretical specific impulse performance of any liquid engine oxidizer/fuel combination [1]. Since hydrogen is the lightest element, it takes up very little mass. However, its major drawback is that liquid hydrogen is extremely difficult to store. Hydrogen needs to be stored at temperatures at or below twenty degrees Kelvin.

1.3.1 Required Design Variables

CEA Output		
Variable	**Value**	**Units**
C*	2427.9	Meters per second
Gamma (γ)	1.23	Unitless
Optimal Oxidizer/Fuel Ratio	3.3	Unitless
Specific Impulse	386.9	Seconds
Thrust Coefficient (C_f)	1.563	Unitless

The specific impulse provided by the propellant combination is the highest theoretical propellant specific impulse for our given rocket. This means that if the rest of the engine works, and there are no other issues, this is a fantastic rocket choice.

Propellant Parameters		
Variable	**Value**	**Units**
LOX Density	1141	Kilograms per Cubic Meter
LH2 Density	70.85	Kilograms per Cubic Meter
Characteristic Length (L*)	0.85	Meters
Total Propellant Mass	91855	Kilograms
Oxidizer Mass	70494	Kilograms
Fuel Mass	21362	Kilograms
Total Burn Time	87.94	Seconds

Liquid oxygen was found from [2], and liquid hydrogen density was found from [4]. Characteristic length was determined from the average of characteristic lengths provided in class for the LOX/LH2 combination. Total burn time is well under our required 100 second burn time limit. Total propellant mass is less than 100,000 kilograms, which is fantastic for our design.

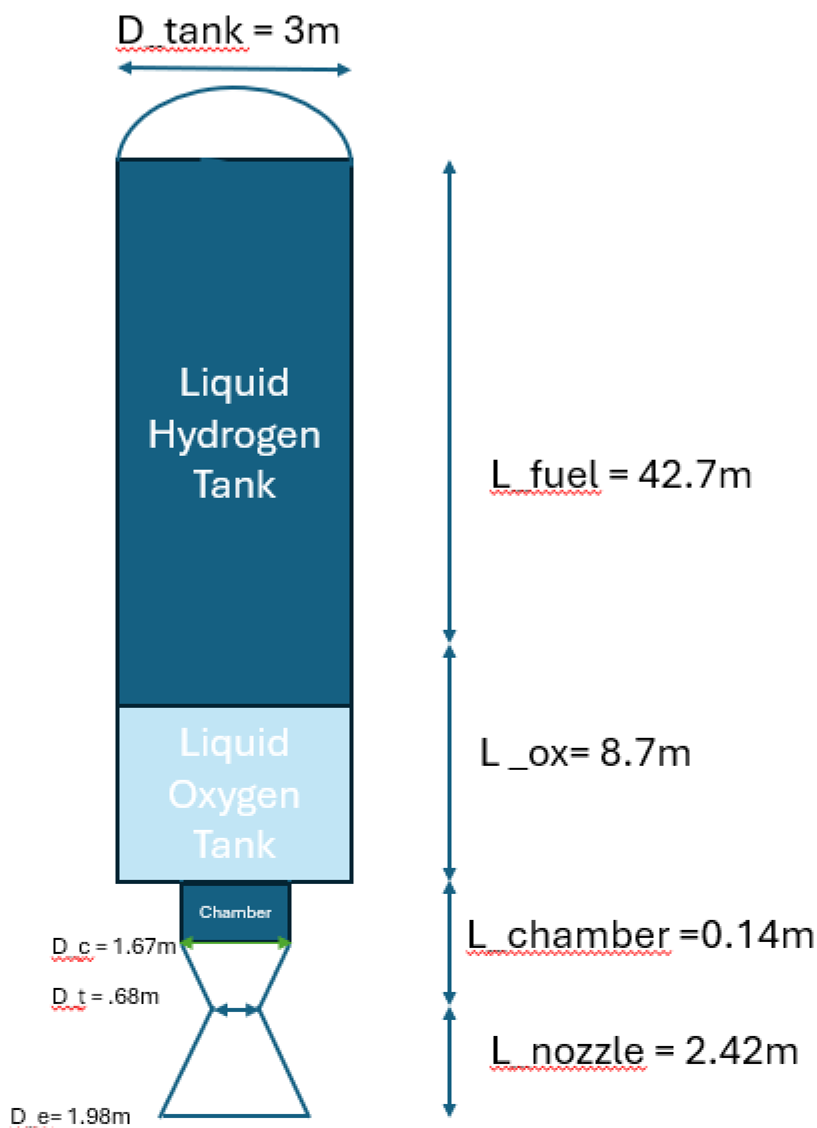
Engine and Nozzle Dimensions		
Variable	**Value**	**Units**
Throat Diameter	0.684	Meters

Exit Diameter	1.980	Meters
Chamber Diameter	1.667	Meters
Minimum Required Chamber Length	0.143	Meters
Nozzle Length	2.418	Meters

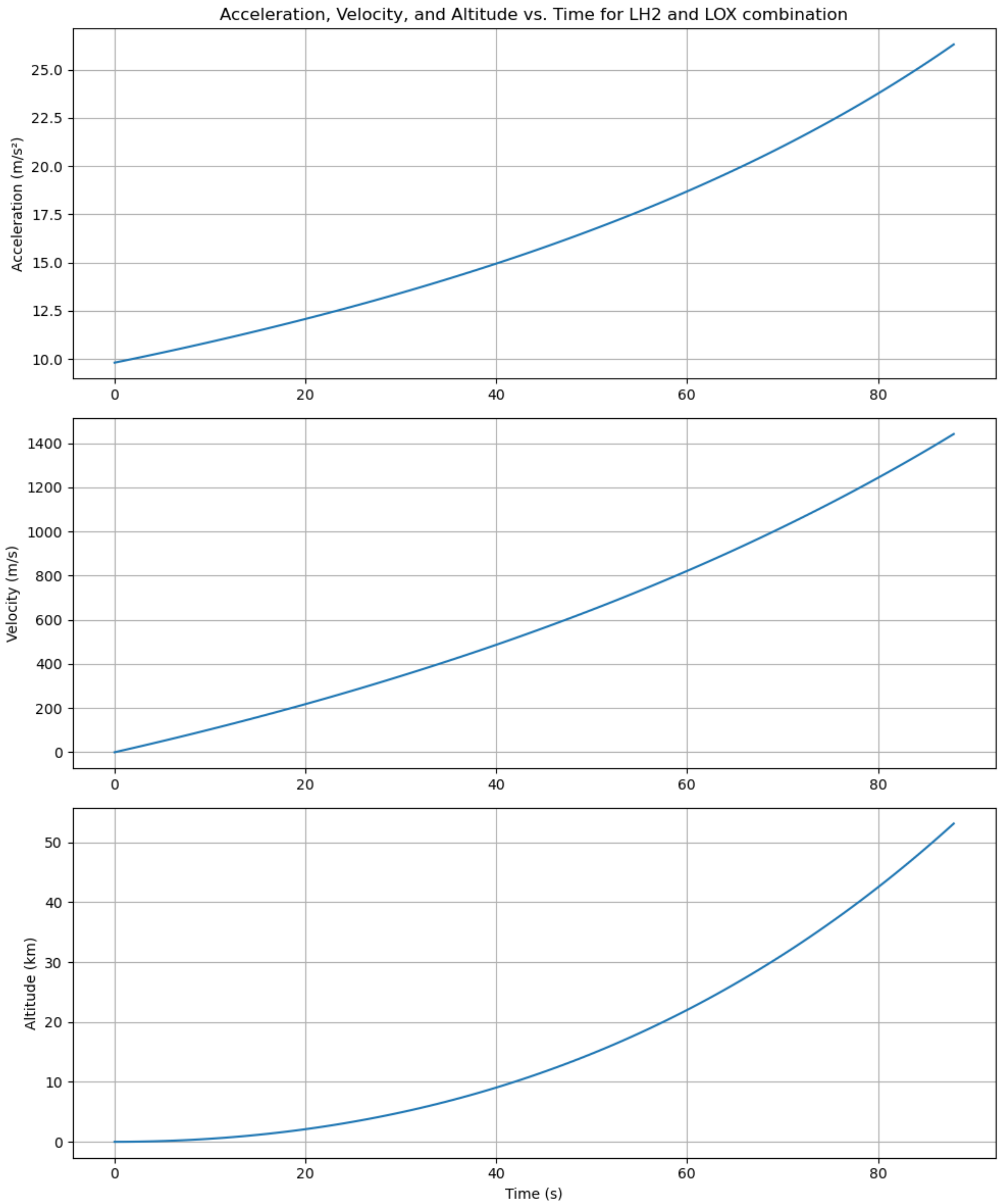
Again, all our parameters are within required parameters. Our minimum chamber length seems quite low. However, the chamber length generally does not need to be long, which means we can increase the true chamber length. Nozzle length is reasonable, given the size and thrust required of the booster we're tasked with designing.

Tank Parameters and Dimensions		
Variable	**Value**	**Units**
Ultimate Tensile Strength	950	MPA
Booster Structural Mass	10206	Kilograms
Tank Diameter	3	Meters
Oxidizer Tank Volume	61.8	Cubic Meters
Fuel Tank Volume	301.5	Cubic Meters
Total Tank Volume	363.3	Cubic Meters
Oxidizer Tank Length	8.7	Meters
Fuel Tank Length	42.7	Meters
Total Tank Length	51.4	Meters
Required Tank Thickness	56.54	Millimeters

Our tank thickness is again well within the limits of our assumption. It is again within 2% of total tank diameter. Our booster structural mass is lowest of any of the six designs. However, the total tank length could be a potential issue. At 51 meters, the tanks will take up the large part of the booster itself, leaving very little space for the payload.


D_tank = 3m
Liquid Hydrogen Tank
L_fuel = 42.7m
Liquid Oxygen Tank
L _ox= 8.7m
Chamber
D_c = 1.67m
L_chamber =0.14m
D_t = .68m
L_nozzle = 2.42m
D_e= 1.98m

1.3.2 Plotting Trajectories



Velocity here only goes to 1450 meters per second. It is reasonable for the velocity to not reach 2300 since We are also considering gravitational losses. Longer burn times mean more gravitational losses.

2 Solid Rocket Motor

2.1 Assumptions

- Our average thrust will be exactly equal to the minimum thrust required for the rocket, equal to two times the total propellant mass.
- Our design parameters follow optimal expansion at sea level.
- Our final trajectory parameters are based on a model excluding drag losses, but including gravitational losses.
- Values from CEA are taken as accurate values and are used in design analysis.
- Burn grain is cylindrical.
- Chamber thickness is negligible enough to not cause a considerable impact on overall chamber diameter.
- Chamber is made of aluminum.
- Mass fraction of our rocket is 0.9 .
- Initial chamber pressure is 1000 PSIA.
- Mach number in the chamber is 0.1 .
- Mach number at the throat is 1.
- Nozzle is conical with opening angle of 15 degrees.
- Outer port diameter is equal to the total chamber diameter.
- Inner port diameter is equal to half of outer port diameter.
- Fuel regression follows $\dot{r_b} = 0.04\, P_c^{0.3}$ where chamber pressure is in PSIA and $\dot{r_b}$ is in in/s

2.2 Solid Rocket Motor 1: Aluminum Fuel/Ammonium Perchlorate Oxidizer with Butadiene (HTPB) Binder

For both solid rocket motor combinations, I chose HTPB as the binder. This is because it is one of the most used binder designs and is used in NASA's SLS booster [15]. I chose Aluminum as my fuel and ammonium perchlorate as my oxidizer for the first solid rocket motor. This is because I wanted to see the performance parameters of NASA's Space Launch System rocket booster [15]. Additionally, I chose a low binder mass percentage of the fuel. I wanted to see how having a low binder mass would impact the performance of the rocket.

2.2.1 Required Design Variables

CEA Output		
Variable	**Value**	**Units**
C*	1633.2	Meters per second
Gamma (γ)	1.174	Unitless
Optimal Oxidizer/Fuel Ratio (CEA)	3.0	Unitless

Calculated Optimal Oxidizer/Fuel Ratio	20.0	Unitless
Specific Impulse	267.05	Seconds
Thrust Coefficient (C_f)	1.604	Unitless

Specific impulse is slightly lower compared to the liquid engine designs, which makes sense as solid rocket motors generally have a lower specific impulse at the benefit of simplicity.

Propellant Parameters		
Variable	**Value**	**Units**
Ammonium Perchlorate Density	1950	Kilograms per Cubic Meter
HTPB Density	614.9	Kilograms per Cubic Meter
Aluminum Density	2712	Kilograms per Cubic Meter
Aluminum Fuel Percentage	85	Percent
Total Propellant Mass	166744	Kilograms
Oxidizer Mass	125058	Kilograms
Fuel Mass	35433	Kilograms
Binder Mass	6252	Kilograms
Total Burn Time	78.1	Seconds

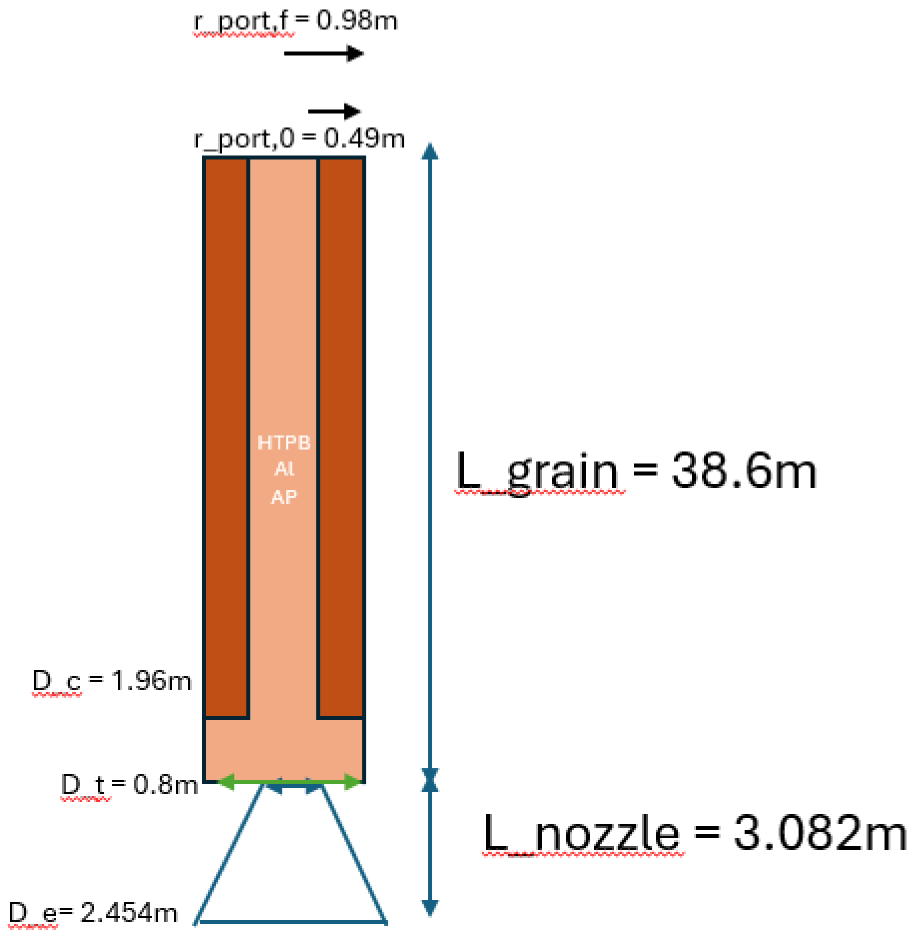
Ammonium Perchlorate density was found from [5], aluminum density was found from [6], and HTPB density was found from [7]. Aluminum fuel percentage was determined as a propellant parameter, testing a low binder mass percentage of overall propellant mass. Total burn time is comparable to liquid engine burn times. Total propellant mass is similar to liquid engine design 1, but substantially higher than liquid engine design 2.

Engine and Nozzle Dimensions		
Variable	**Value**	**Units**
Throat Diameter	0.803	Meters
Exit Diameter	2.454	Meters
Chamber Diameter	1.961	Meters
Initial Port Radius	0.49	Meters
Final Port Radius	0.981	Meters
Nozzle Length	3.082	Meters

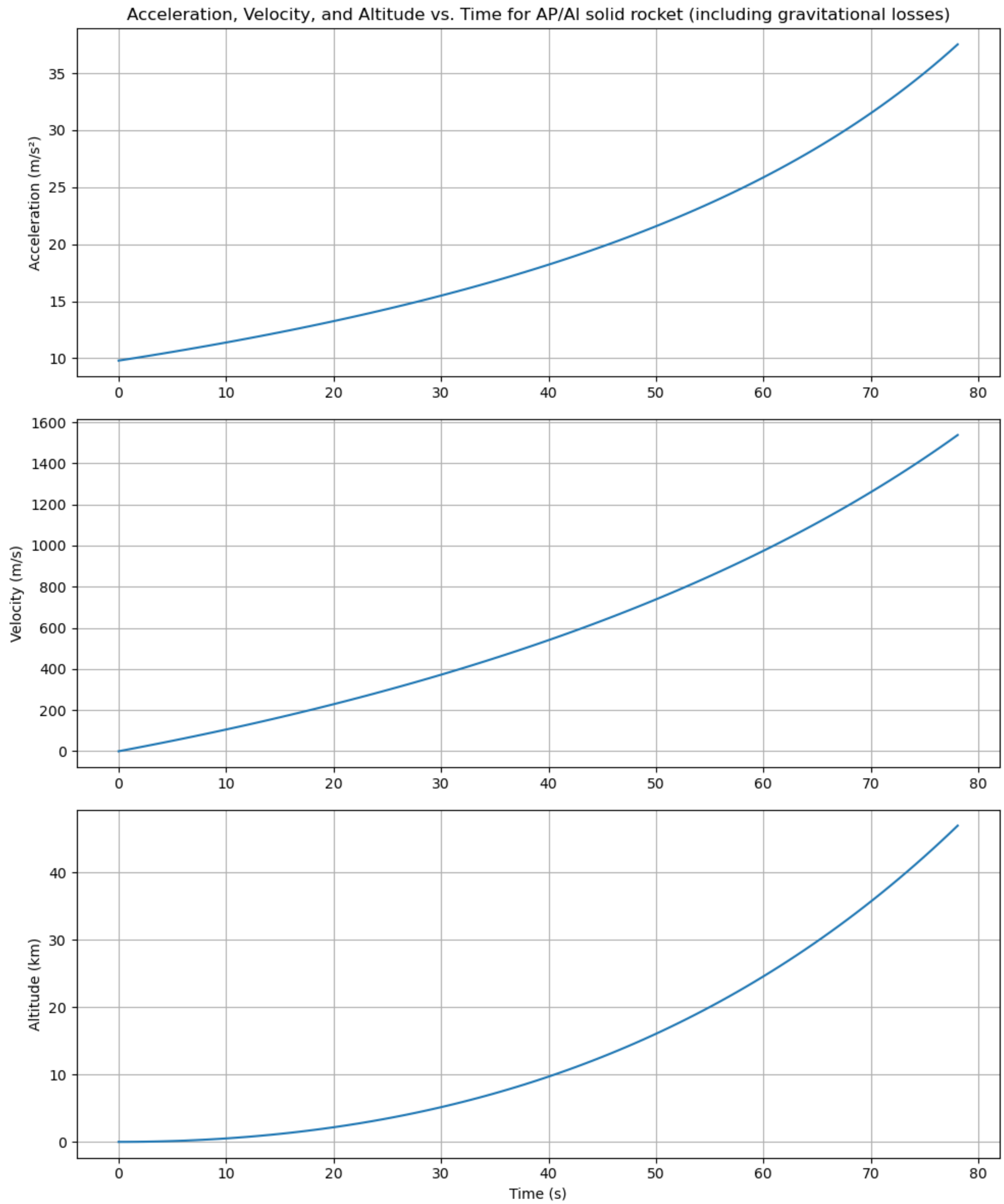
Initial port radius is equal to half of the final port radius, as stated in our assumptions. Additionally, our final port radius is equal to the radius of our chamber. Our throat radius is very slightly smaller than the initial port radius, which makes sense as the port radius should always be bigger than the throat radius. Nozzle length is similar to all other designs. Exit diameter is well within the 3 meter constraint. Nozzle length is comparable to all other nozzle lengths.

Overall Parameters and Dimensions		
Variable	**Value**	**Units**
Ultimate Tensile Strength	950	MPA
Booster Structural Mass	18527	Kilograms
Grain Volume	87.37	Cubic Meters
Grain Length	38.57	Meters
Initial Burn Area	121.1	Square Meters
Final Burn Area	234.6	Square Meters
Required Chamber Thickness	46.63	Millimeters

Total Grain length seems quite long, which indicates that similar to the liquid design 2, our chamber (tank in liquid design) would take up the vast majority of the booster, leaving little space for the payload. Our tank thickness is again well within the limits of our assumption. It is within 5% of total tank diameter.



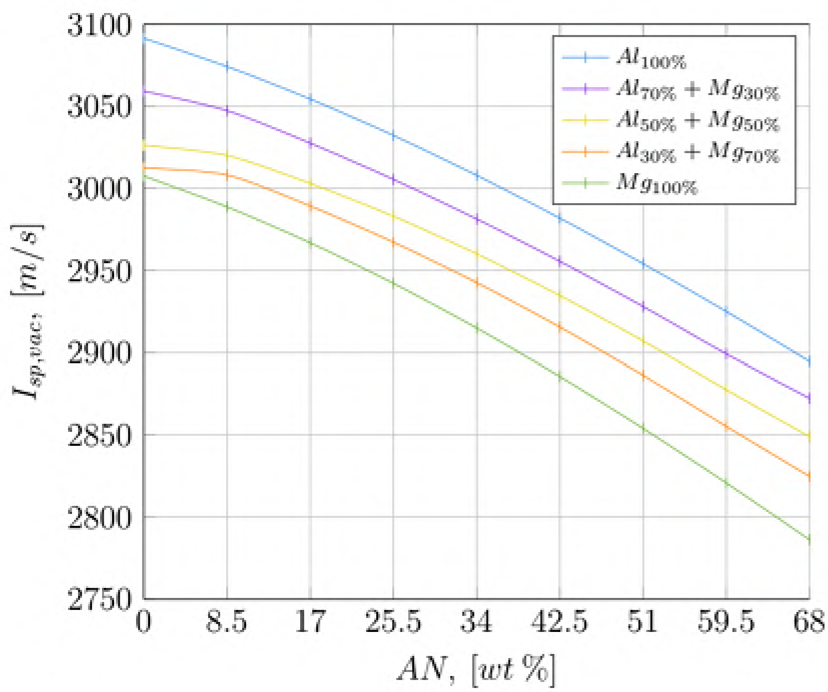
2.2.2 Plotting Trajectories



Velocity here only goes to 1550 meters per second. It is reasonable for the velocity to not reach 2300 since We are also considering gravitational losses. Longer burn times mean more gravitational losses.

2.3 Solid Rocket Motor 2: Magnesium Fuel/Ammonium Nitrate Oxidizer with Butadiene(HTPB) Binder

For both solid rocket motor combinations, I chose HTPB as the binder. This is because it is one of the most used binder designs and is used in NASA's SLS booster. In contrast to the first solid rocket motor design, I made the mass of the binder half of the total fuel mass. I wanted to see how increasing the mass of the binder to an extreme amount impacted the performance of the rocket. I chose magnesium fuel and ammonium nitrate because I wanted to make a greener rocket that had less negative environmental impacts. Additionally, I wanted to see the difference between the magnesium and aluminum as fuels, as I was curious how the impacts of [22] would truly impact our design choices.



2.3.1 Required Design Variables

CEA Output		
Variable	**Value**	**Units**
C*	1816.2	Meters per second
Gamma (γ)	1.243	Unitless
Optimal Oxidizer/Fuel Ratio (CEA)	2.3	Unitless
Calculated Optimal Oxidizer/Fuel Ratio	4.6	Unitless
Specific Impulse	286.5	Seconds
Thrust Coefficient (C_f)	1.547	Unitless

Specific impulse is higher than expected, especially since the binder is such a large part of overall fuel mass. Is it possible that HTPB could be used as its own solid propellant?

Propellant Parameters		
Variable	**Value**	**Units**
Ammonium Nitrate Density	1720	Kilograms per Cubic Meter
HTPB Density	614.9	Kilograms per Cubic Meter
Magnesium Density	1738	Kilograms per Cubic Meter
Magnesium Fuel Percentage	50	Percent
Total Propellant Mass	147636	Kilograms
Oxidizer Mass	102828	Kilograms
Fuel Mass	22354	Kilograms
Binder Mass	22354	Kilograms
Total Burn Time	72.0	Seconds

Ammonium Nitrate density was found from [8], Magnesium density was found from [9], and HTPB density was found from [7]. Magnesium fuel percentage was determined as a propellant parameter, testing a high binder mass percentage of overall propellant mass. Total burn time is comparable to liquid engine burn times. Total propellant mass is similar to liquid engine design 1, but substantially higher than liquid engine design 2.

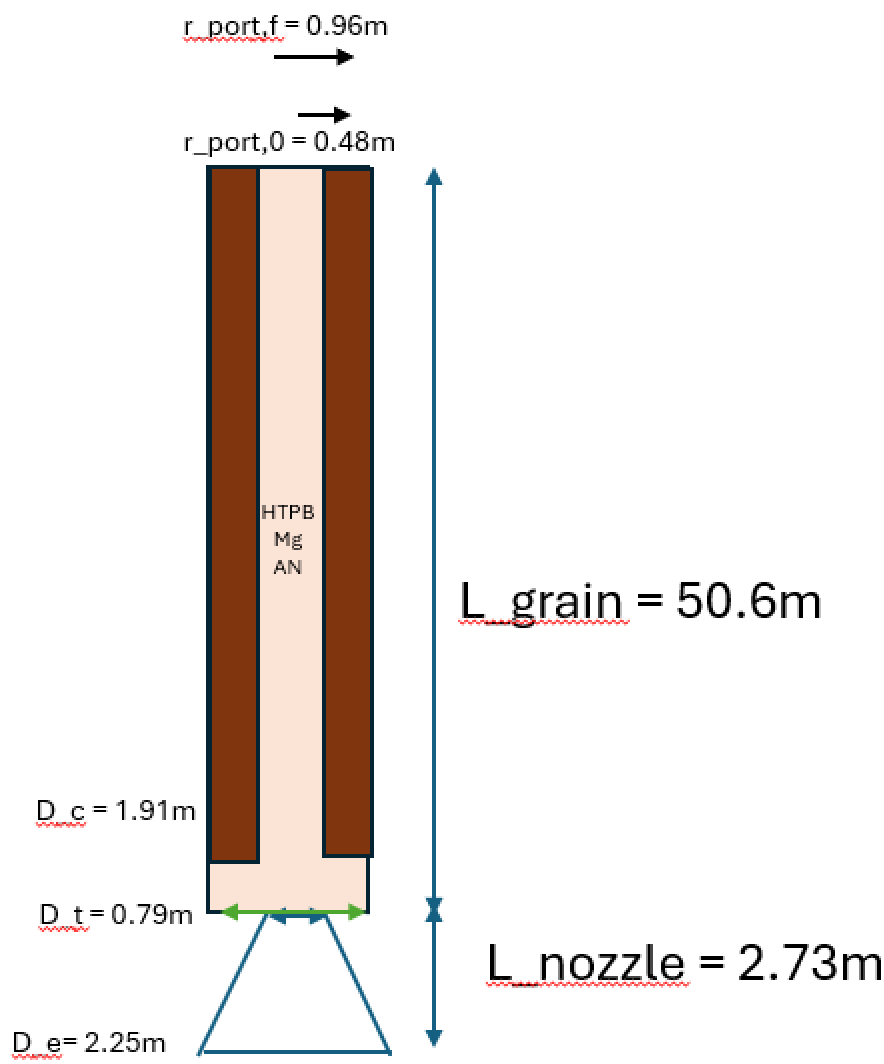
Engine and Nozzle Dimensions		
Variable	**Value**	**Units**
Throat Diameter	0.786	Meters
Exit Diameter	2.247	Meters
Chamber Diameter	1.913	Meters
Initial Port Radius	0.478	Meters
Final Port Radius	0.957	Meters
Nozzle Length	2.726	Meters

Initial port radius is equal to half of the final port radius, as stated in our assumptions. Additionally, our final port radius is equal to the radius of our chamber. Our throat radius is very slightly smaller than the initial port radius, which makes sense as the port radius should always be bigger than the throat radius. Nozzle length is similar to all other designs. Exit diameter is well within the 3 meter constraint.

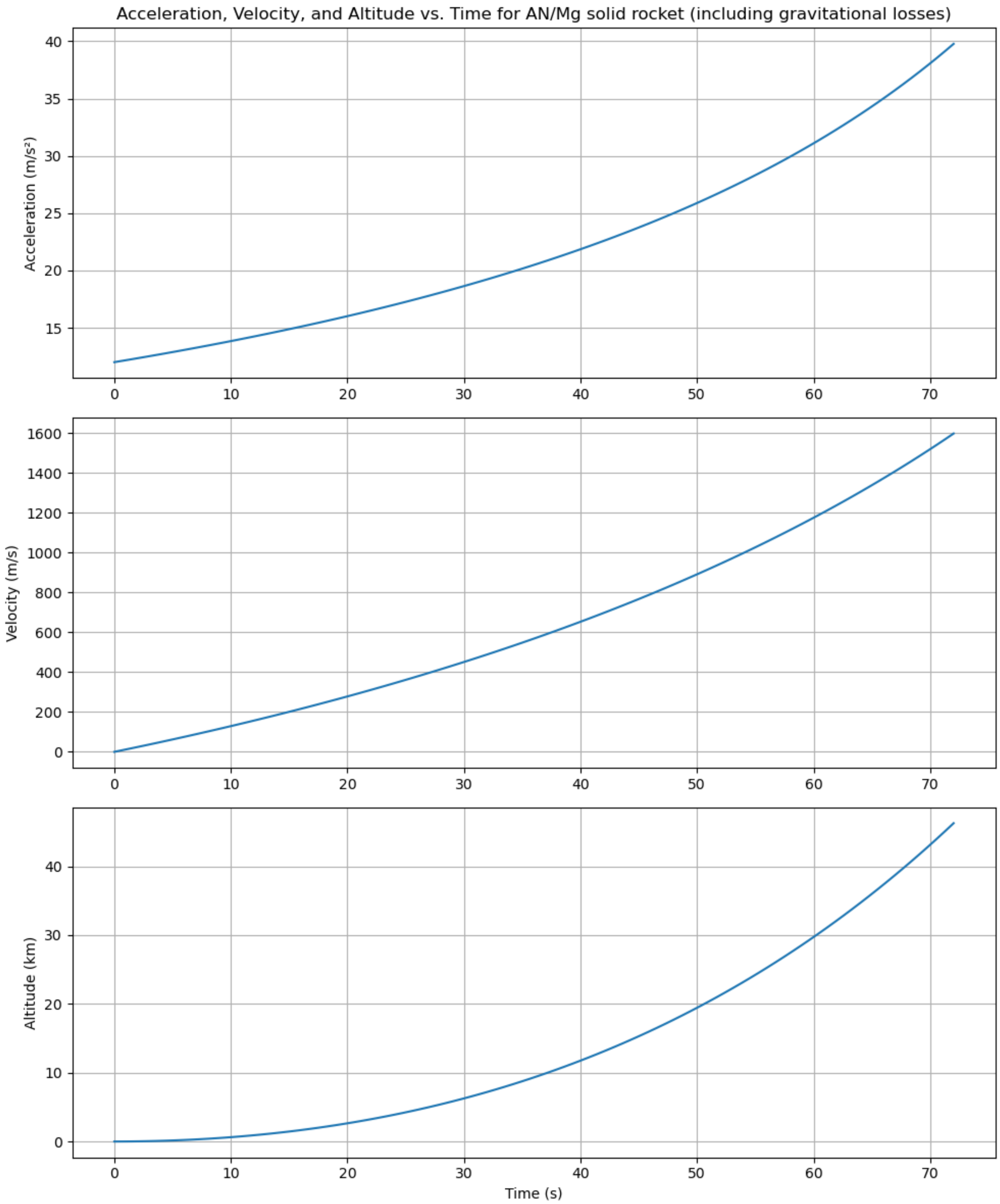
Overall Parameters and Dimensions		
Variable	**Value**	**Units**

Ultimate Tensile Strength	950	MPA
Booster Structural Mass	16392	Kilograms
Grain Volume	109.0	Cubic Meters
Grain Length	50.56	Meters
Initial Burn Area	154.1	Square Meters
Final Burn Area	301.0	Square Meters
Required Chamber Thickness	72.89	Millimeters

Total Grain length seems quite long, which indicates that similar to the liquid design 2, our chamber (tank in liquid design) would take up the vast majority of the booster, leaving little space for the payload. Our tank thickness is again well within the limits of our assumption. It is within 5% of total tank diameter.



2.3.2 Plotting Trajectories



Velocity here only goes to 1450 meters per second. It is reasonable for the velocity to not reach 2300 since We are also considering gravitational losses. Longer burn times mean more gravitational losses.

3 Hybrid Rocket Engine

3.1 Assumptions

- Our average thrust will be exactly equal to the minimum thrust required for the rocket, equal to two times the total propellant mass.
- Our oxidizer mass flow rate is constant.
- Our design parameters follow optimal expansion at sea level.
- Our final trajectory parameters are based on a model excluding drag losses, but including gravitational losses.
- Values from CEA are taken as accurate values and are used in design analysis.
- Our propellant volume is identical to the volume of the tank they inhabit.
- Propellant tanks are cylindrical.
- Tank diameters are exactly at the constraint limit.
- Tank thickness is negligible enough to not cause a considerable impact on overall tank diameter.
- Tanks are made of aluminum.
- Mass fraction of our rocket is 0.9 .
- Initial chamber pressure is 1000 PSIA.
- Mach number in the chamber is 0.1 .
- Mach number at the throat is 1.
- Nozzle is conical with opening angle of 15 degrees.
- Outer port diameter is equal to the total chamber diameter.
- Inner port diameter is equal to half of outer port diameter.
- Fuel regression follows $\dot{r} = 0.1\dot{}\, G_{ox}{}^{0.68}$ where $G_{ox}$is in $\frac{lb}{s \cdot in^2}$ and $\dot{r_b}$ is in $\frac{in}{s}$.
- Burn grain is cylindrical.

3.2 Hybrid Engine 1: HTPB Fuel/LOX Oxidizer

For our first hybrid propellant combination, we chose HTPB as our fuel and Liquid oxygen as the oxidizer. We chose this combination because it has a great balance between performance and it isn't toxic to the environment. However, the LOX is cryogenic, which could make it hard to store.

3.2.1 Required Design Variables

CEA Output		
Variable	**Value**	**Units**
C*	1807.6	Meters per second

Gamma (γ)	1.252	Unitless
Average Oxidizer/Fuel Ratio	2.1	Unitless
Average Specific Impulse	288.6	Seconds
Thrust Coefficient (C_f)	1.565	Unitless

Average specific impulse is quite high in this design. It is one of the highest of the six chosen propellant combinations. Additionally, our Oxidizer/Fuel ratio found in CEA will be simply used as the average O/F ratio thoughout the rest of the hybrid rocket engine design.

Propellant Parameters		
Variable	**Value**	**Units**
LOX Density	1141	Kilograms per Cubic Meter
HTPB Density	614.9	Kilograms per Cubic Meter
Total Propellant Mass	145749	Kilograms
Oxidizer Mass	98733	Kilograms
Grain Mass	47015	Kilograms
Oxidizer/Fuel Shift	0.146	Unitless
Total Burn Time	65.6	Seconds

Liquid Oxygen was found from [2], and HTPB Density was found from [7], and taken at 293 Kelvin. Total oxidizer/fuel shift is quite low for this combination, suggesting the design parameters will not change much over the course of the burn. Burn time is also quite low for this design. Total propellant mass is similar to what Isee for most other designs.

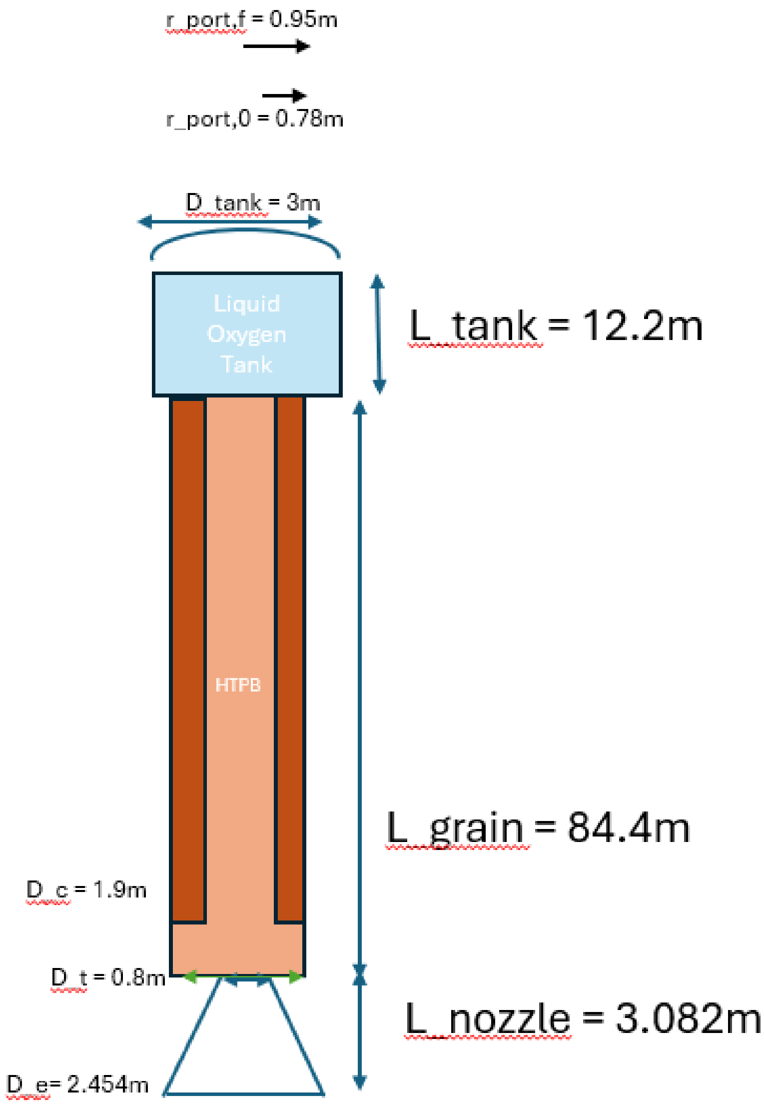
Engine and Nozzle Dimensions		
Variable	**Value**	**Units**
Throat Diameter	0.779	Meters
Exit Diameter	2.207	Meters
Outer Port Radius	0.947	Meters
Inner Port Radius	0.78	Meters
Grain Length	84.4	Meters
Nozzle Length	2.67	Meters

Our throat radius is very slightly smaller than the initial port radius, which makes sense as the port radius should always be bigger than the throat radius. Nozzle length is similar to all other designs. Exit diameter is well within the 3 meter constraint. Nozzle length is comparable to all other nozzle lengths, but is slightly shorter, since the throat diameter to exit diameter difference is smaller than most designs. Our grain length is extremely high, indicating this design is completely unfeasible for the design want.

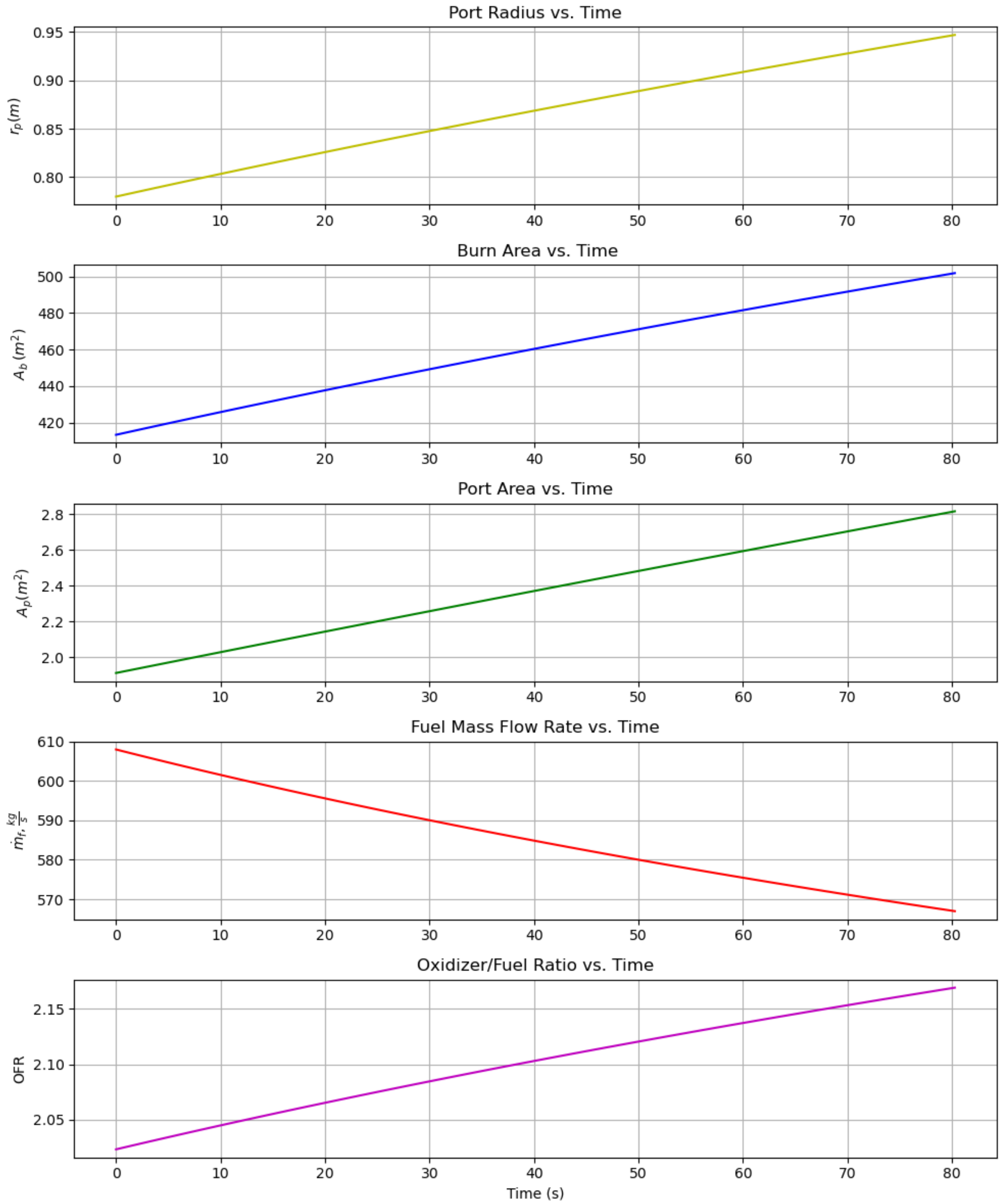
Overall Parameters and Dimensions

Variable	Value	Units
Ultimate Tensile Strength	950	MPA
Booster Structural Mass	16113	Kilograms
Tank Diameter	3	Meters
Oxidizer Tank Volume	86.53	Cubic Meters
Oxidizer Tank Length	12.24	Meters
Required Tank Thickness	73.77	Millimeters

Since grain length is nearly 90 meters, having our oxidizer tank length at nearly 12 meters makes the entire booster over 100 meters long. This is longer than the entirety of the Falcon 9 rocket [13]. Required tank thickness is comparable to other designs.

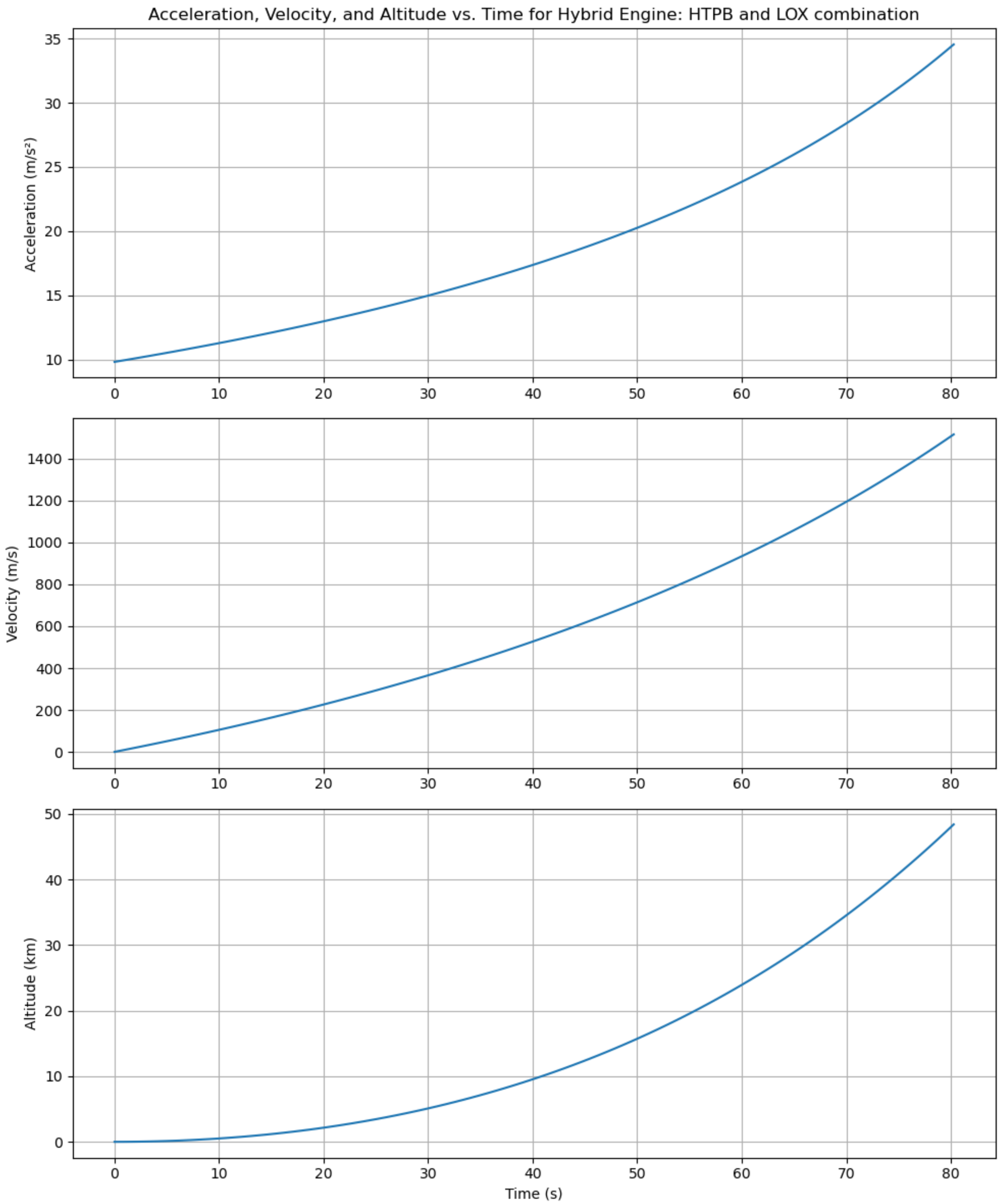


3.2.2 Plotting OF Shift and Other Useful Quantities



WE can see the oxidizer/fuel ratio increases over time, but it does not increase much. Since the total mass flow rate only decreases by around 40 kilograms per second, it makes sense the total O/F ratio does not change much.

3.2.3 Plotting Trajectories



Velocity here only goes to 1450 meters per second. It is reasonable for the velocity to not reach 2300 since We are also considering gravitational losses. Longer burn times mean more gravitational losses.

3.3 Hybrid Engine 2: N2O Oxidizer/Paraffin Fuel

For the second hybrid engine design, we chose nitrous oxide and paraffin. Again, these are clean so they are not extremely toxic to the environment. The benefit to this fuel is that it does not need to be cryogenically stored, which gives a huge advantage of simplicity over the first hybrid engine design.

3.3.1 Required Design Variables

CEA Output		
Variable	**Value**	**Units**
C*	1435.3	Meters per second
Gamma (γ)	1.262	Unitless
Average Oxidizer/Fuel Ratio	8.4	Unitless
Average Specific Impulse	228.9	Seconds
Thrust Coefficient (C_f)	1.564	Unitless

Average specific impulse is very low in this design. Because of this, it may not be fully viable as a propellant combination. However, the previous design had a completely unreasonable length, so we will see about this one.

Propellant Parameters		
Variable	**Value**	**Units**
N2O Density	1220	Kilograms per Cubic Meter
Paraffin Density	930	Kilograms per Cubic Meter
Total Propellant Mass	222928	Kilograms
Oxidizer Mass	199212	Kilograms
Grain Mass	23715	Kilograms
Oxidizer/Fuel Shift	0.679	Unitless
Total Burn Time	73.37	Seconds

Nitrous Oxide Density was found from [11], and Paraffin Density was found from [12], and taken at 293 Kelvin. Total burn time is similar to values WE see in other rockets. Total propellant mass is substantially higher than other rockets, which indicates that we will likely need a more sturdy tank.

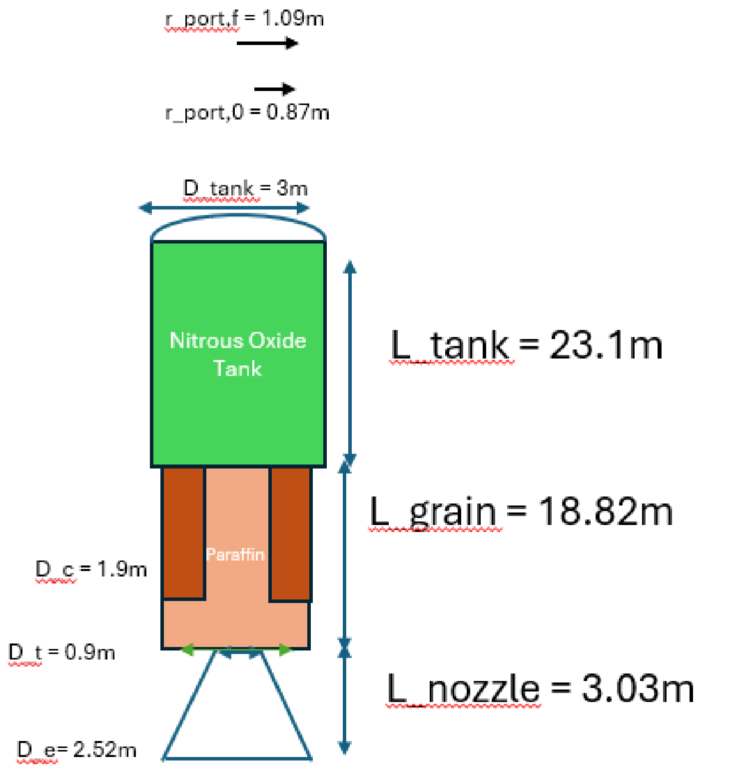
Engine and Nozzle Dimensions

Variable	Value	Units
Throat Diameter	0.897	Meters
Exit Diameter	2.521	Meters
Outer Port Radius	1.091	Meters
Inner Port Radius	0.871	Meters
Grain Length	18.82	Meters
Nozzle Length	3.03	Meters

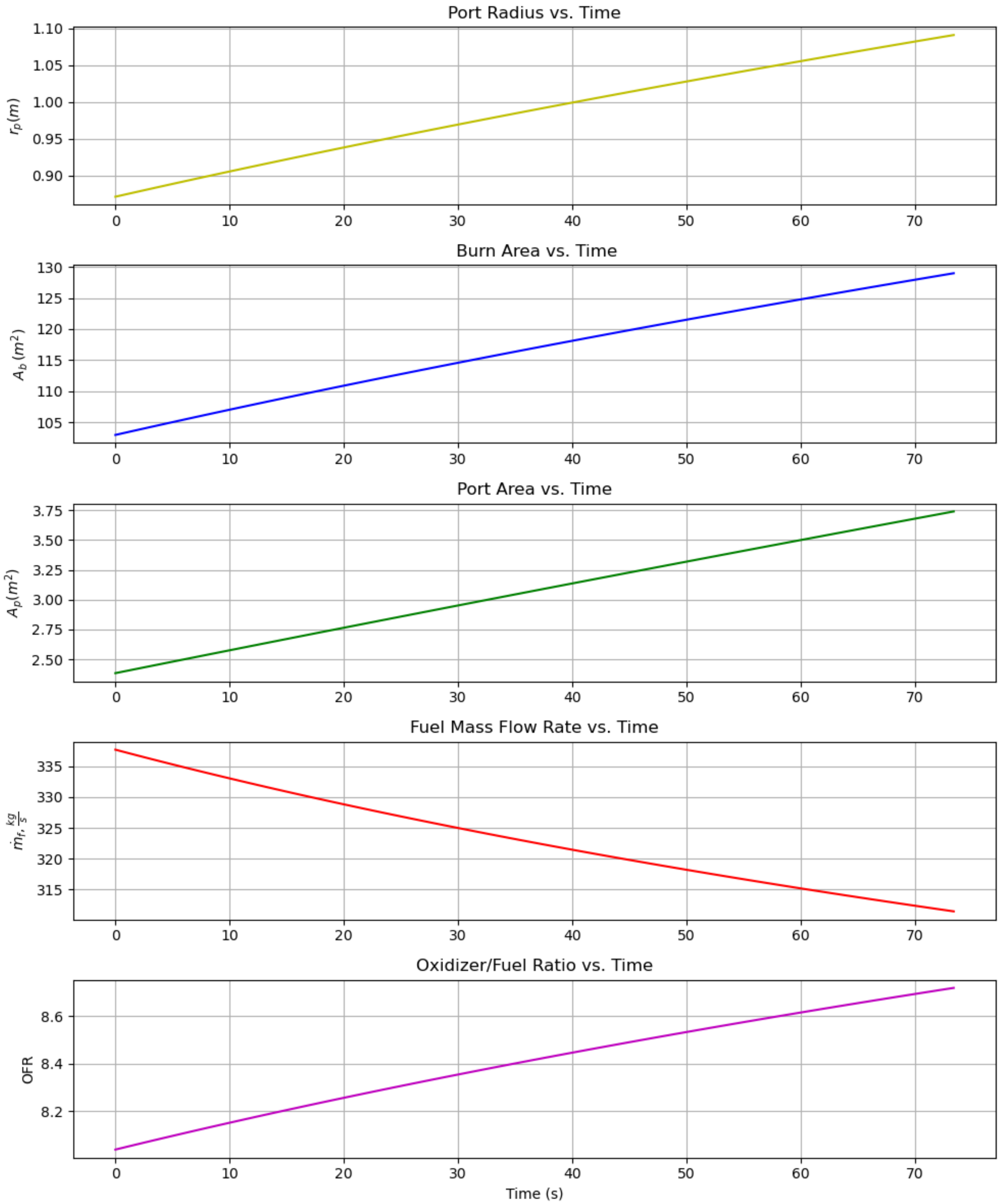
Our throat radius is very slightly smaller than the initial port radius, which makes sense as the port radius should always be bigger than the throat radius. Nozzle length is similar to all other designs. Exit diameter is well within the 3 meter constraint. Nozzle length is comparable to all other nozzle lengths, but is slightly shorter, since the throat diameter to exit diameter difference is smaller than most designs. Grain length is much more reasonable in this design.

Overall Parameters and Dimensions		
Variable	**Value**	**Units**
Ultimate Tensile Strength	950	MPA
Booster Structural Mass	24769	Kilograms
Tank Diameter	3	Meters
Oxidizer Tank Volume	163.29	Cubic Meters
Oxidizer Tank Length	23.10	Meters
Required Tank Thickness	99.3	Millimeters

As expected, our tank thickness is high as expected. Since our total mass of our oxidizer is high, our total pressure inside the tank will be high, and since we're using the same diameter and material for all designs, it makes sense our tank thickness would be the highest of all designs. Additionally, our tank length is reasonable given the length of the grain. It is still longer than WE would want, but substantially better than design 1.

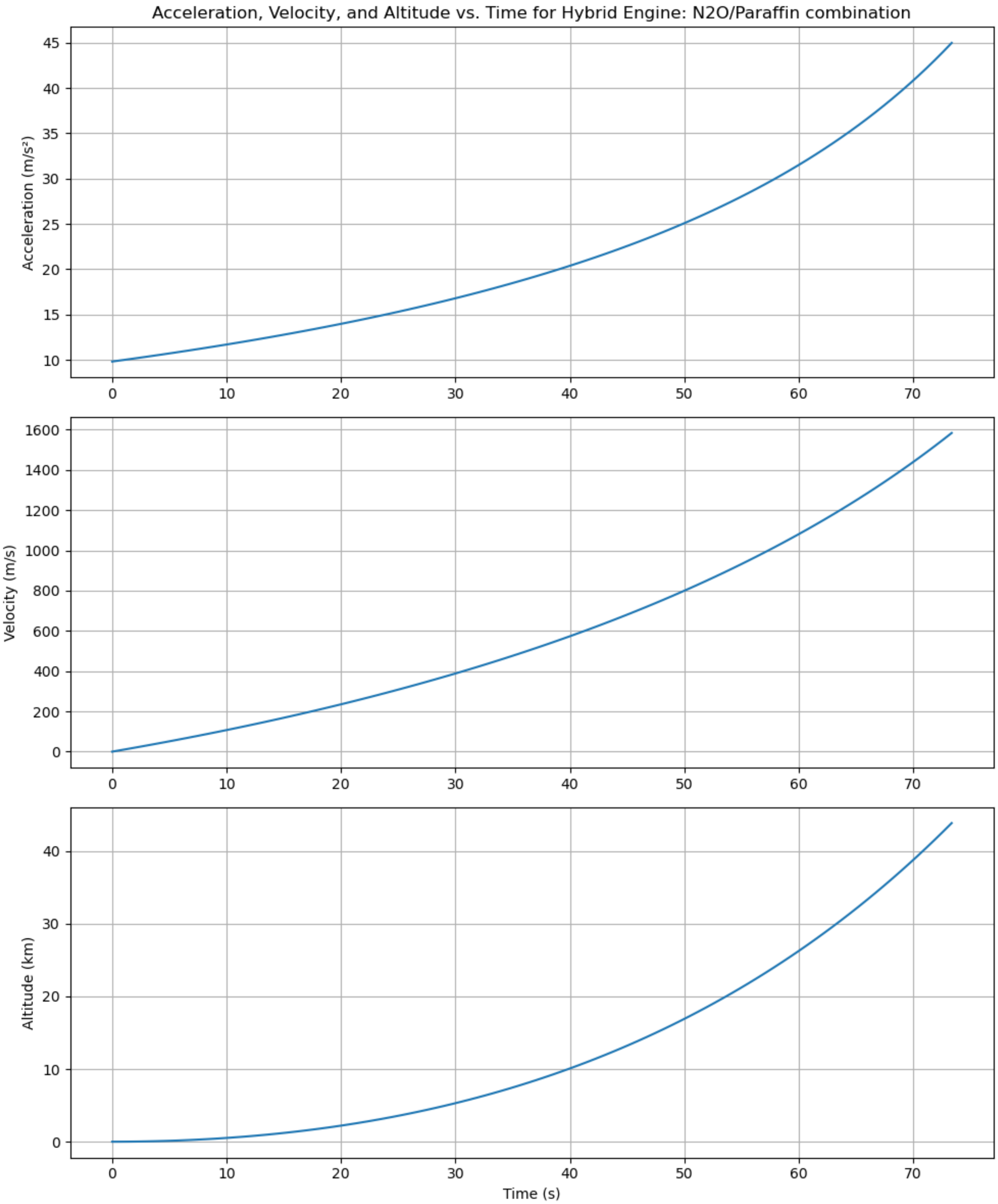

r_port,f = 1.09m
r_port,0 = 0.87m
D_tank = 3m
Nitrous Oxide Tank
L_tank = 23.1m
L_grain = 18.82m
Paraffin
D_c = 1.9m
D_t = 0.9m
L_nozzle = 3.03m
D_e= 2.52m

3.3.2 Plotting OF Shift and Other Useful Quantities



With the nitrous oxide/paraffin propellant combination, we see a much more drastic shift in oxidizer/fuel ratio. Even though fuel mass flow rate does not chance substantially in percentage, it does change substantially in the ratio of the oxidizer to fuel. Since there's so much more nitrous oxide than paraffin, we see a much greater change in O/F ratio.

3.3.3 Plotting Trajectories



Velocity here only goes to 1450 meters per second. It is reasonable for the velocity to not reach 2300 since We are also considering gravitational losses. Longer burn times mean more gravitational losses.

4 Design Comparisons

4.1 Liquid Rocket Engine Design Comparisons

At first glance, the hydrogen rocket seems like the clear favorite here. It has the highest specific impulse of any of the six options. However, issues arise when we look at the propellant tank sizes. In the LOX/RP-1 design, the required propellant tank length nearly one third of the required propellant tank length of the LOX/LH2 design. One potential way to combat this issue is to increase the oxidizer/fuel ratio in the LOX/LH2 design. If we increase the total oxidizer/fuel ratio of the LOX/LH2 propellant combination, we will lower our specific impulse slightly, but we will substantially lower total required fuel tank size.

One thing to note in our design is the environmental impacts of each design. Liquid oxygen and liquid hydrogen are one of the cleanest energy sources. Since the product of the combustion is largely water vapor, it is not harmful to the environment whatsoever. However, the liquid oxygen/RP-1 combination produces carbon dioxide in large amounts. In small amounts, carbon dioxide is not harmful to the environment and is even naturally present within our atmosphere. However, large amounts of atmospheric carbon dioxide has been proven to contribute to climate change, which causes potential issues for the LO2/RP-1 combination. In contrast, RP-1/LO2 combinations are much safer than many other propellant choices[17], which makes both LO2/LH2 and RP-1/LO2 combinations viable when considering environmental safety.

One other major design consideration is the storage. Since both propellants use liquid oxygen, they will both need some part stored cryogenically. However, RP-1 does not need to be cryogenically stored, and hydrogen does. On top of this, liquid oxygen has a much higher boiling point than hydrogen, so the hydrogen fuel tank storage would be much more challenging than the liquid oxygen fuel tank storage. [18]

Additionally, hydrogen fuel often has the potential to be extremely reactive when it comes to mixing with oxygen in a gaseous state. Because of this, rocket engineers must be extremely careful with the amount of hydrogen and oxygen coming out of the fuel tanks at any point in time. One potential risk of hydrogen specifically is the sloshing of liquid hydrogen fuel in the tank during the propulsive maneuvers. As seen in Figure 2, hydrogen fuel becomes unstable during flight. According to [16], “Appling a retention thrust […] could cause the liquid to become unsettled again. Therefore, it is necessary to maintain the settling thrust until the motion of the liquid is stable before performing the thrust switch.” This is a necessary design parameter to focus on, as sloshing can have major impacts on required tank thickness, and fuel flow rate, potentially voiding consistent oxidizer/fuel flow rates assumed.

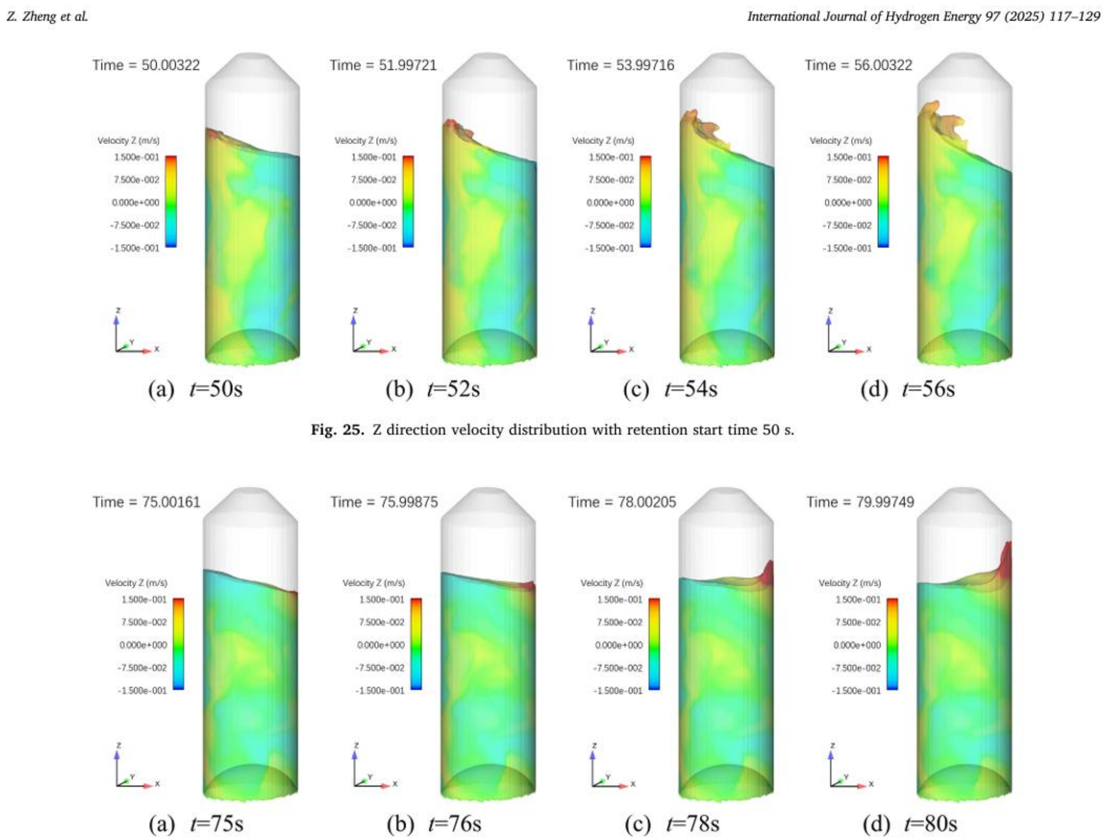


Figure 2. Z-direction velocity distribution with retention start time 50 s. *Reproduced from Zheng et al. [16], International Journal of Hydrogen Energy, 2025.*

Despite these factors, I still recommend moving forward with the liquid oxygen and liquid hydrogen design. The performance of the liquid hydrogen/liquid oxygen design is incredibly high compared to the performance of the liquid oxygen/RP-1 design. Since we're contracting for NASA, we're focused on making the rocket work the best it possibly can. Because liquid hydrogen and liquid oxygen provide such a powerful combination, I suggest moving forward with this combination.

4.2 Solid Rocket Motor Design Comparisons

Both solid rocket motors offer reasonable specific impulses, and the specific impulse offered by each design is very similar. Because of this, we must look elsewhere when making our solid rocket motor design decision. If both are equal and we go off performance alone, we would likely go with the aluminum fuel/ammonium perchlorate oxidizer and butadiene (HTPB) binder. This is because the aluminum/AP design offers slightly better performance than the magnesium/AN design.

When looking at the required grain lengths and total grain areas, we see the magnesium fuel/ammonium nitrate oxidizer with butadiene binder design requires a much longer grain length. Because of this, it could potentially impact the allowable payload volume if we need to keep the total rocket within certain design parameters.

After further research, the aluminum/ammonium perchlorate design is one of the most widely used solid propellant designs. However, aluminum seems to substantially differ in temperature as it moves through the chamber. Figure 3 shows the potential extreme temperature change as aluminum combusts and moves through the combustion chamber. However, this example is shown using aluminum oxide, which is not the same propellant combination in our designs.

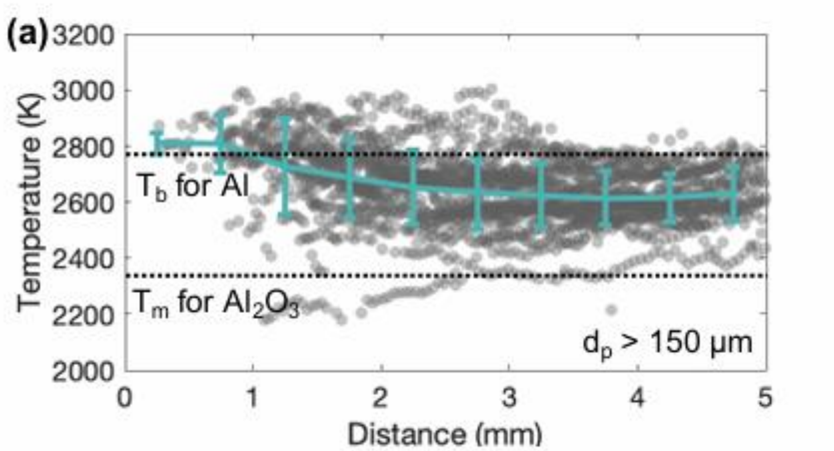


Figure 3. Temperature distribution of aluminum combustion particles as a function of distance through combustion. *Reproduced from Marsh et al. [19], Combustion and Flame*

The final design consideration comes from the environmental impacts of ammonium perchlorate/aluminum and ammonium nitrate/magnesium. First, I'll start with ammonium perchlorate and aluminum. This combination produces many potentially harmful products. First, it produces carbon dioxide and carbon monoxide, which are common greenhouse gases, but this is not the only potentially harmful outcome of combustion between aluminum and AP. Aluminum and AP produce hydrogen chloride, which is extremely toxic [29]. Because of this, many researchers and rocket engineers have begun to use ammonium nitrate as a substitute for ammonium perchlorate. It still produces the greenhouse gases that ammonium perchlorate produces, but it doesn't produce extremely toxic hydrogen chloride. [29] is a full case study of using ammonium nitrate rather than ammonium perchlorate, and testing whether it is viable. Figure 4 demonstrates that ammonium nitrate, while having slightly lower performance characteristics than ammonium perchlorate, is still viable for solid rocket engines.

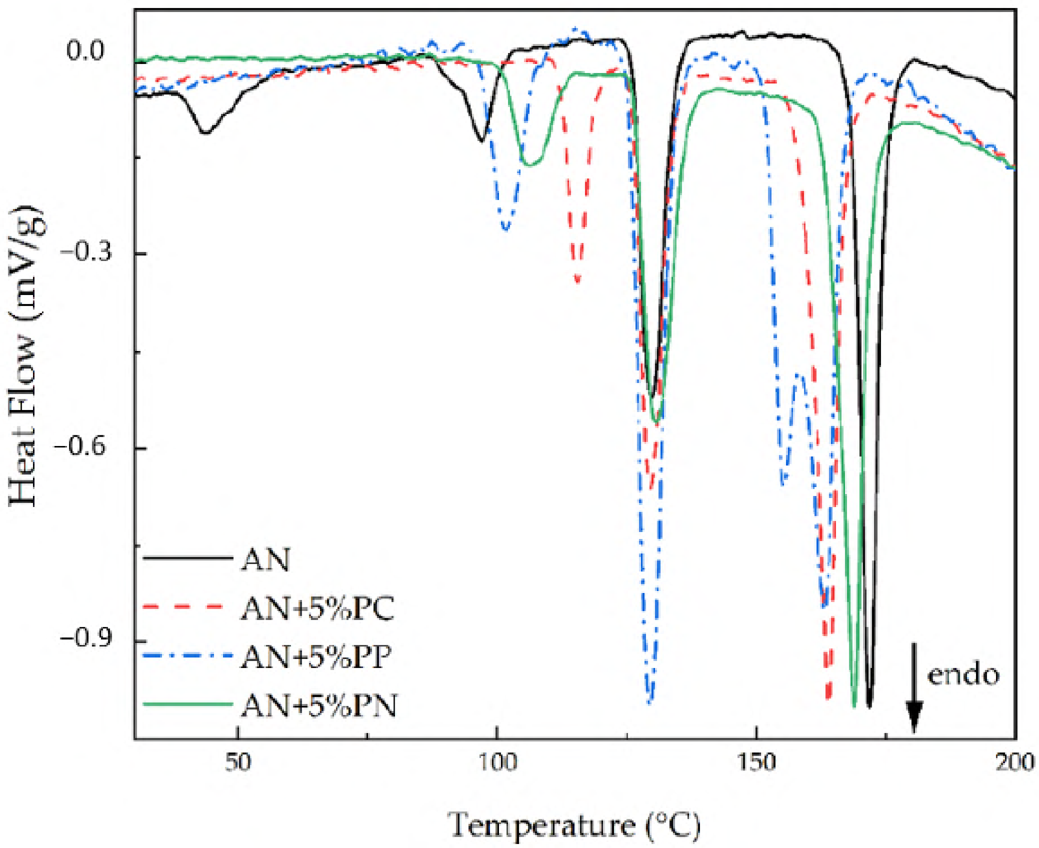


Figure 4. Heat flow as a function of temperature for ammonium nitrate (AN) with different additives. *Reproduced from [29], Processes*, Vol. 9, No. 12, 2021, Article 2201.

My final design decision largely comes from the environmental impacts of ammonium nitrate. As it is an engineer's duty to ensure the work they produce meaningfully helps society, I cannot in good conscience suggest the ammonium perchlorate/aluminum fuel option. Even though ammonium nitrate and magnesium have slightly lower performance parameters, I will still choose ammonium nitrate and magnesium because the products of the propellant combination are much less harmful to the environment [27].

4.3 Hybrid Motor Design Comparisons

Our first hybrid motor design is the HTPB fuel and liquid oxygen. At first glance, our specific impulse is reasonably high, suggesting a promising rocket design. However, upon further inspection, we see the length of the fuel grain is extremely long, which allows very little room for the rest of the rocket. For this reason, this hybrid motor design is not feasible to continue with, unless we are fine with having a rocket over 100 meters tall. One major possibility for altering this design would be to swap the HTPB for a paraffin fuel. If switched out, it could allow the burn rate to substantially increase, and would make this design much more appealing, as hybrid motors excel in simplicity and safety factors[30]. They have similar characteristics to solid rockets, but with the added benefit of being able to alter the total thrust at any given time.

Our second design is nitrous oxide (oxidizer) and paraffin (fuel). While it does not have the abysmal length parameters the liquid oxygen and HTPB design had, it has extremely poor performance parameters. Generally, you want a rocket to have above 250 seconds for specific impulse, so having a specific impulse at sea level in the mid-220s is not desirable.

Upon consideration of environmental factors, it is clear that both designs would not have significant negative environmental impacts[29]. For example, nitrous oxide and paraffin produce outputs seen in figure 5. None of these outputs are particularly harmful, other than producing greenhouse gases, such as carbon dioxide and nitrogen.

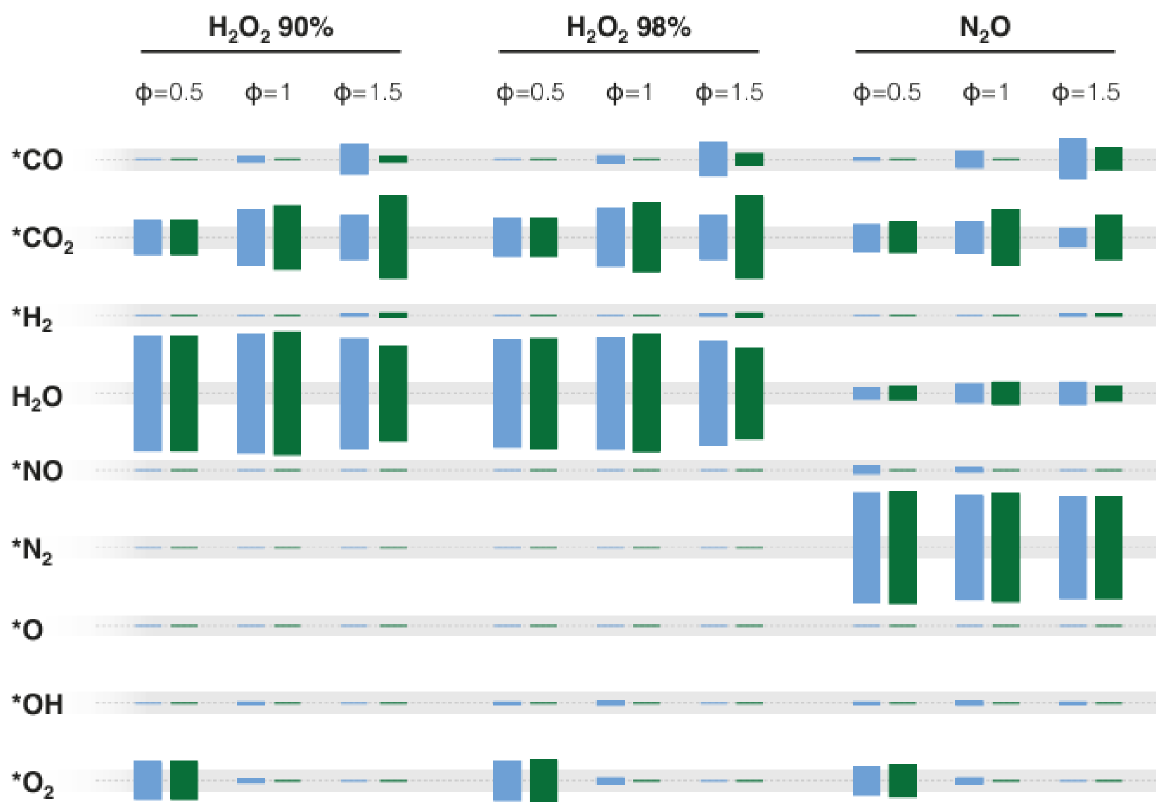


Figure 5. Emission species comparison for different oxidizers (H_2O_2 90%, H_2O_2 98%, and N_2O) at varying equivalence ratios (ϕ). *Reproduced from Paravan et al. [29], Aerospace, Vol. 10, No. 7, 2023, pp. 572.*

Upon final consideration, I will be moving forward with the nitrous oxide/paraffin design. Even though the performance parameters are extremely low, the required chamber and tank diameters from the alternative are abysmal. It is worth noting that neither of these designs are particularly strong, so they will likely be outclassed by the other two engine types in the final design decision.

4.4 Final Design Decision

My final design decision depends on the desired outcome of the rocket. I've decided to stray from the hybrid motor design, as it doesn't offer any substantial benefits over the solid and liquid rockets other than a chance at design simplification. If we're focused on simplicity and are highly constrained on time and resources, I would recommend the solid rocket ammonium perchlorate/aluminum fuel. However, if we need a high performing rocket, I suggest one of the liquid rocket engine designs.

Overall, I recommend the liquid rocket engine with liquid oxygen and liquid hydrogen. While there are many challenges in storing the propellants, the performance of the rocket is unmatched. It provides nearly 100 seconds more of specific impulse (over 30%) than even the second-best performing design displayed. Additionally, the product of the propellant combustion is not harmful to the environment in quantities displayed in a liquid rocket engine.

On top of this, the liquid rocket engines are much simpler than hybrid rockets, which allows another layer of insurance for the rocket to work. Despite not being as simple as a solid rocket, the liquid rocket engine has been used by NASA multiple times in the past, and we can use the expertise of these functional experts to aid in any design challenges relating to cryogenics and propulsion.

5 REFERENCES

[1] NASA, "Materials for Liquid Propulsion Systems," NASA,
URL: https://ntrs.nasa.gov/api/citations/20160008869/downloads/20160008869.pdf

[2] MatWeb, "ASM Material Data Sheet (P641)," MatWeb,
URL: https://asm.matweb.com/search/SpecificMaterial.asp?bassnum=MTP641

[3] Massachusetts Institute of Technology (MIT), "Oxygen Safety Gram," MIT Environmental Health & Safety,
URL: https://ehs.mit.edu/wp-content/uploads/2020/01/safety_gram_6_OXYGEN.pdf

[4] National Institute of Standards and Technology (NIST), "Publication," NIST,
URL: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=832303

[5] Chemical Engineering Transactions, "Article in CET Journal (Vol. 86, 2021)," AIDIC,
URL: https://www.cetjournal.it/cet/21/86/038.pdf

[6] National Center for Biotechnology Information (NCBI), "Ammonium Perchlorate," PubChem,
URL: https://pubchem.ncbi.nlm.nih.gov/compound/Ammonium-perchlorate

[7] The Engineering Toolbox, "Metal Alloys - Densities,"
URL: https://www.engineeringtoolbox.com/metal-alloys-densities-d_50.html

[8] National Center for Biotechnology Information (NCBI), "Boron," PubChem,
URL: https://pubchem.ncbi.nlm.nih.gov/element/Boron#section=Melting-Point

[9] National Center for Biotechnology Information (NCBI), "Ammonium Nitrate," PubChem,
URL: https://pubchem.ncbi.nlm.nih.gov/compound/Ammonium-nitrate#section=Density

[10] National Center for Biotechnology Information (NCBI), "Magnesium," PubChem,
URL: https://pubchem.ncbi.nlm.nih.gov/compound/magnesium

[11] National Center for Biotechnology Information (NCBI), "Nitrous Oxide," PubChem,
URL: https://pubchem.ncbi.nlm.nih.gov/compound/nitrous-oxide

[12] National Institute of Standards and Technology (NIST), "Compositional Data,"
URL: https://physics.nist.gov/cgi-bin/Star/compos.pl?ap213

[13] Space Exploration Technologies Corp. (SpaceX), "Falcon 9,"
URL: https://www.spacex.com/vehicles/falcon-9/

[14] Sutton, G. P., and Biblarz, O., *Rocket Propulsion Elements*, 9th ed., John Wiley & Sons, Hoboken, NJ, 2017.

[15] NASA, "Space Launch System Solid Rocket Booster,"
URL: https://www.nasa.gov/reference/space-launch-system-solid-rocket-booster/

[16] Zheng, Z., Xu, A., Jiang, W., Wang, B., Sun, P., Li, P., and Huang, Y., "Simulation of Sloshing and Settling Behavior of Liquid Hydrogen in an Insulated Tank during Coastal Period," *International Journal of Hydrogen Energy*, Vol. 97, 2025, pp. 117–129.

[17] Blondel-Canepari, L., Sarritzu, A., and Pasini, A., "A Holistic Approach for Efficient Greener In-Space Propulsion," *Acta Astronautica*, Vol. 223, 2024, pp. 435–447.

[18] National Center for Biotechnology Information (NCBI), "Oxygen," PubChem,
URL: https://pubchem.ncbi.nlm.nih.gov/element/Oxygen.

[19] Marsh, A. W., Wang, G. T., Heyborne, J. D., Guildenbecher, D. R., and Mazumdar, Y. C., "Time-Resolved Size, Velocity, and Temperature Statistics of Aluminum Combustion in Solid Rocket Propellants," *Combustion and Flame*.
URL: https://www.sciencedirect.com/science/article/pii/S1540748920305940.

[20] Xue, X., Ye, Q., and Yu, Y., "Thermal Safety of Solid Rocket Motor with Complex Charge Structure," *Applied Thermal Engineering*, Vol. 222, 2023, Article 119932.

[21] Tüzün, F. N., "The Effect of Aluminum and Ammonium Perchlorate Content on Temperature Sensitivity, Mechanical Properties and Performance of HTPB Propellants," *Transactions of the Japan Society for Aeronautical and Space Sciences*, Vol. 50, No. 169, 2007, pp. 175–181.
URL: https://www.jstage.jst.go.jp/article/tjsass/50/169/50_169_175/_pdf.

[22] Paravan, C., Dossi, S., Maggi, F., Galfetti, L., Vorozhtsov, A. B., Zhukov, I., and Sokolov, S., "Reduction of Solid Propellants Environmental Impact: Al-Mg Mechanically Activated Fuels and AP-PSAN Blends," *9th European Conference for Aeronautics and Space Sciences (EUCASS)*, 2022.
URL: https://www.eucass.eu/doi/EUCASS2022-4852.pdf.

[23] "Develop a Low Environmental Impact Propellant," *The Forefront of Space Science*, Japan Aerospace Exploration Agency (JAXA),
URL: https://www.isas.jaxa.jp/e/forefront/2008/hori/index.shtml.

[24] Slavinskaya, N., Wiegand, M., and Riedel, U., "The Potential Impact of Solid Rocket Propellants on the Atmosphere: An Overview," *European Conference for Aero-Space Sciences (EUCASS)*, 2011.
URL: https://www.eucass.eu/component/docindexer/?id=3899&task=download.

[25] Levi, S., Signoriello, D., Gabardi, A., Molinari, M., Galfetti, L., DeLuca, L. T., Cianfanelli, S., and Klyakin, G. F., "Metallized Solid Rocket Propellants Based on AN/AP and PSAN/AP for Access to Space," *Progress in Propulsion Physics*, Vol. 1, 2009, pp. 97–108.
URL: https://www.eucass-proceedings.eu/articles/eucass/pdf/2009/01/eucass1p97.pdf.

[26] "AP-Free Composite Propellants as Replacement for Conventional AP-Propellants," Fraunhofer Publica,
URL: https://publica.fraunhofer.de/bitstreams/dde917f6-300b-47c8-8392-5b505f554f92/download.

[27] Dîrloman, F., Ţigănescu, T., Rotariu, T., Ungureanu, C., Zecheru, T., and Sava, I., "Eco-Oxidizers for Composite Propellants: Ammonium Nitrate and Ammonium Dinitramide," *Journal of Military Technology*,
URL:
https://jmiltechnol.mta.ro/6/05_D%C3%8ERLOMAN%2C%20%C8%9AIG%C4%82NESCU%2C%20ROTARIU%2C%20UNGUREANU%2C%20ZECHERU%2C%20SAVA-min.pdf.

[28] Paravan, C., Dossi, S., Maggi, F., Galfetti, L., Vorozhtsov, A. B., Zhukov, I., and Sokolov, S., "Reduction of Solid Propellants Environmental Impact: Al-Mg Mechanically Activated Fuels and AP-PSAN Blends," *9th European Conference for Aeronautics and Space Sciences (EUCASS)*, 2022.
URL: https://www.eucass.eu/doi/EUCASS2022-4852.pdf.

[29] "Performance and Sensitivity Properties of Solid Heterogeneous Propellants Based on Ammonium Nitrate," *Processes*, Vol. 9, No. 12, 2021, Article 2201.
URL: https://www.mdpi.com/2227-9717/9/12/2201.

[30] "Experimental Evaluation of HTPB/Paraffin Fuel Blends for Hybrid Rocket Engines," *ScienceDirect*, 2021. URL: https://www.sciencedirect.com/science/article/abs/pii/S0010218021001012.

6 APPENDICES

Part 1: Liquid Rocket engine.

First, we're going to go through our liquid rocket engine parameters. We first need to determine our fuel/oxidizers.

For my first propellant combination, I decided to go with liquid oxygen as my oxidizer and RP-1 as my fuel. For my second propellant combination, I went with liquid oxygen as my oxidizer and Liquid Hydrogen as my fuel.

LO2/RP-1

A/B) O/F Ratio and Specific Impulse To find the optimal Oxidizer to fuel ratio, we can use the CEA tool and iterate through various oxidizer/fuel ratios given our propellant combinations to find the highest value of Oxidizer to fuel ratio.

To do this, I iterated through CEA from O/F = 2 to O/F = 4 in 0.1 increments. From there, I found the following values:

```
          THEORETICAL ROCKET PERFORMANCE ASSUMING FROZEN COMPOSITION
                                AFTER POINT 2

 Pin =  1000.0 PSIA
 CASE = _______________

             REACTANT                    WT FRACTION      ENERGY      TEMP
                                          (SEE NOTE)     KJ/KG-MOL      K
 FUEL        RP-1                         1.0000000    -24717.700    298.150
 OXIDANT     O2(L)                        1.0000000    -12979.000     90.170

 O/F=    2.30000  %FUEL= 30.303030  R,EQ.RATIO= 1.480724  PHI,EQ.RATIO= 1.4807

                 CHAMBER   THROAT     EXIT
 Pinf/P            1.0000   1.7406   68.050
 P, BAR            68.947   39.610   1.0132
 T, K             3572.65  3364.04  1705.13
 RHO, KG/CU M    5.1796 0 3.1972 0 1.6134-1
 H, KJ/KG         -818.62 -1530.55 -4845.32
 U, KJ/KG        -2149.74 -2769.46 -5473.29
 G, KJ/KG        -42088.6 -40390.7 -24542.4
 S, KJ/(KG)(K)    11.5516  11.5516  11.5516

 M, (1/n)          22.316   22.577   22.577
 Cp, KJ/(KG)(K)    4.9913   4.5174   1.8774
 GAMMAs            1.1488   1.1493   1.2440
 SON VEL,M/SEC     1236.6   1193.3    883.9
 MACH NUMBER        0.000    1.000    3.211

 PERFORMANCE PARAMETERS

 Ae/At                      1.0000   8.3322
 CSTAR, M/SEC               1807.2   1807.2
 CF                         0.6603   1.5703
 Ivac, M/SEC                2231.5   3059.1
 Isp, M/SEC                 1193.3   2837.9

 MASS FRACTIONS

 *CO             0.45324   *CO2            0.24201   COOH            0.00002
 *H              0.00106   HCO             0.00002   HO2             0.00002
 *H2             0.01075   H2O             0.26345   H2O2            0.00001
 *O              0.00226   *OH             0.02235   *O2             0.00480

  * THERMODYNAMIC PROPERTIES FITTED TO 20000.K
```

$$Optimal\, O/F = 2.3$$

To find the optimal specific impulse, we simply need to divide the exit Isp value given by 9.806 to get into units of seconds. This results in:

$$I_{sp,sl} = 289.4\, seconds$$

C) Engine and Nozzle Dimensions

Using CEA, we were also able to determine our Gamma, Thrust Coefficient, and C*.

First, let's find the area of the throat and the area of the exit for our liquid engine.

To find the throat area, we can use the following equation:

$$A_t = \frac{T}{p_0 \, C_F}$$

We are given a minimum thrust value, so for our iteration of throat area, we can use that minimum thrust value given.

From here, we can use the following equation to calculate the area at the exit:

$$A_e \; = \; A_t \cdot \left(\frac{2}{\gamma + 1} \right)^{\frac{1}{\gamma - 1}} \left(\frac{p_0}{p_e} \right)^{\frac{1}{\gamma}} \sqrt{ \left[\frac{\gamma - 1}{\gamma + 1} \left(1 \; - \; \left(\frac{p_0}{p_e} \right)^{\frac{1-\gamma}{\gamma}} \right) \right]^{-1} }$$

Now, we can calculate the area of the chamber using the following equation:

Our Mach number at the throat is by definition 1, and our mach number in the chamber we will assume to be 0.1.

$$A_c \; = \; A_t \cdot \frac{M_t}{M_c} \sqrt{ \left(\frac{1 \; + \; \frac{\gamma - 1}{2} \, M_c^2}{1 \; + \; \frac{\gamma - 1}{2} \, M_t^2} \right)^{\frac{\gamma + 1}{\gamma - 1}} }$$

Next, we want to find the length of our chamber. To do this, we must use the characteristic length table given in class to determine the characteristic length for our propellant combination. For LOX/RP-1 combination the min L* values are between 1-1.3 so I will take the middle value at 1.15cm.

Now, we can use the equation:

$$L_c = \frac{L^* \, A_t}{A_c}$$

Finally, to find our nozzle length, we will assume a standard conical nozzle with an opening angle of 15 degrees. We can use the following equation to determine our required nozzle length.

$$L_{nozzle} = \frac{D_e + D_t}{2 \cdot tan(15\backslash\text{degree})}$$

D) Propellant Mass, Propellant Tank Size and Thickness

to calculate the propellant mass, we can use the rocket equation

$$\Delta V = I_{\text{sp}} \, g_0 \, \ln\!\left(\frac{m_{\text{prop}} + m_{\text{payload}} + m_{\text{structure}}}{m_{\text{payload}} + m_{\text{structure}}} \right)$$

Rearranging gives us:

$$m_{\text{prop}} = \left(m_{\text{payload}} + m_{\text{structure}}\right)\left(e^{\frac{\Delta V}{I_{\text{sp}}\,g_0}} - 1\right)$$

Now, we can use our knowledge about the mass fraction of our booster:

$$\lambda = 0.9 = \frac{m_{prop}}{m_{sctucture} + m_{prop}}$$

$$m_{prop} = 9\,m_{structure}$$

We can then use a system of linear equations solver in Python to determine our true Propellant and Structural Masses.

Next, we can calculate the fuel and oxidizer mass fractions since we know $m_{ox} = \mathbf{2.3}m_{fuel}$

Since we know they both add to the total propellant mass, we can say

$$m_{fuel} = \frac{m_{prop}}{OFR + 1}$$

and

$$m_{ox} = \frac{m_{prop}}{\frac{1}{OFR} + 1}$$

To find the propellant tank size, we must find the densities of our fuel and our oxidizer. I found the density of liquid oxygen at 1atm from the following MIT publication: https://ehs.mit.edu/wp-content/uploads/2020/01/safety_gram_6_OXYGEN.pdf

and the density of RP-1 at 293 Kelvin from this government publication. https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=832303

To find the required volume of the tank, we can make a couple quick assumptions. Our first assumption is that our tanks are cylindrical tanks vertically stacked. Second is that the volume of our propellant initially should equal the inside volume of the tank.

We know that our tank diameter is constrained to 3 meters so we will assume the outer diameter of the tank is 3 meters.

To find the total volume of both tanks, we can do the following:

$$V_{tank} = V_{RP-1} + V_{LOX} = m_{ox}\rho_{LOX} + m_{fuel}\rho_{RP-1}$$

Now that we have the volume of the tank, we want to find the following: Tank Length, and Tank thickness

To find tank length for each, we can use the following equations:

$$V_{LOX} = \frac{\pi D_{tank}^2 h_{LOX}}{4}$$

$$V_{RP1} = \frac{\pi D_{tank}^2 h_{RP1}}{4}$$

Since we assume the total diameter of our tank is 3 meters, we can easily calculate the height of our tank.

Rearranging each equation results in:

$$h_{LOX} = \frac{4\pi V_{LOX}}{D_{tank}^2}$$

$$h_{RP1} = \frac{4\pi V_{RP1}}{D_{tank}^2}$$

Next, we can use tank thickness. We can use the following equation:

$$t = \frac{P \cdot D_{tank}}{2\,\sigma_w}$$

To find ultimate tensile strength, we can assume our material is titanium, and using the following site: https://asm.matweb.com/search/SpecificMaterial.asp?bassnum=MTP641

We find our ultimate tensile strength of titanium to be 950 GPA. Now, we can divide by 1.25 to get our stress to be used for our tank thickness equation.

Next, to find our pressure in the tank, we must use the equations for static pressure, as well as the fact that ambient pressure acts an opposing pressure on the tank, and the bottom tank will also experience a pressure from the top tank. In this case, I've chosen the bottom tank to be the RP-1 tank.

Our tank pressure will be:

$$P_{LOX} = P_u + \rho_{LOX} \cdot (g_0 + a_{max}) \cdot h_{LOX} - P_a$$

$$P_{RP1} = P_u + \rho_{RP1} \cdot (g_0 + a_{max} \cdot h_{LH2} + P_{RP1} - P_a$$

where $P_u = \frac{T}{A_{tank}}$

finally, plugging in all variables results in:

$$t_{LOX} = \frac{P_{LOX} \cdot D_{tank}}{2\,\sigma_w}$$

$$t_{RP1} = \frac{P_{RP1} \cdot D_{tank}}{2\,\sigma_w}$$

Since we want to go with the larger of the two thicknesses to make the tank thicknesses equal, we will opt for the thickness required by the liquid oxygen tank since it has a greater pressure requirement.

E) Plotting Altitude, Velocity, and Acceleration

First, we must determine our thrust, mass flow rate, and burn time.

We know our minimum average thrust must be twice our booster liftoff weight, so we will assume that as our average thrust through all mission designs.

So,

$$T_{avg} = 2 \cdot m_{booster,0}$$

Next, we must determine our average mass flow rate.

We can find our mass flow rate using this equation:

$$\dot{m_{avg}} = \frac{T_{avg}}{c^* \, C_F}$$

and finally, we can determine our burn time using the following equation:

$$t_{bo} = \frac{m_{prop}}{\dot{m_{avg}}}$$

We must get a set of differential equations that we can solve as a function of time as shown below:

We know gravity as a function of altitude, so now we can use the following differential equations to numerically solve for the altitude, velocity, and acceleration as a function of time. Since we're only neglecting drag losses, and not gravitational losses, our real final velocity will be substantially lower than the ideal velocity gain.

$$g(h) \;=\; G_0 \left(\frac{R_{\text{earth}}}{R_{\text{earth}} + h}\right)^2$$

$$\frac{dv}{dt} = -\frac{I_{\text{sp}}\, g_0}{m}\,\frac{dm}{dt} \;-\; g(h)$$

$$\frac{dh}{dt} = v$$

$$\frac{dm}{dt} = -\dot{m} = -\frac{m_p}{t_b}$$

In [121…

```
#Initializing variables (LOX/RP-1)
import numpy as np
from matplotlib import pyplot as plt
from scipy.interpolate import UnivariateSpline
from scipy.integrate import solve_ivp
from scipy.optimize import fsolve
```

```
#Non-Unique Parameters
g0 = 9.806 #m/s^2
deltav = 2300 #m/s
lam = 0.9
Pc0 = 6894760 #Pascals
Pa = 101325 #Pa
Pe = Pa
m_payload = 100000 #kg
D_tank = 3 #m
MachT = 1
MachC = 0.1
radius_earth = 6378388
sigma_w = 750000000 #Pa

#Unique Parameters
Lstar = 1.15 #meters
rho_lox = 1141 #kg/m^3
rho_rp1 = 804.59 #kg/m^3


#from CEA
cstar = 1807.2 #m/s
gamma = 1.15
OFR = 2.3
Isp=2837.9/g0 #s
C_f = 1.57
print(f"Optimal Oxidizer/Fuel Ratio: {OFR}")
print(f"Specific Impulse at Sea Level: {Isp:.2f} seconds")
```

```
Optimal Oxidizer/Fuel Ratio: 2.3
Specific Impulse at Sea Level: 289.40 seconds
```

In [115…

```
#Calculating Engine and Nozzle Dimensions (LOX/RP-1)
mass_initial = m_payload + m_structure + m_prop
T = 2 * mass_initial * g0
A_t = T / (C_f * Pc0)
D_t = np.sqrt(4*A_t / np.pi)
A_e = A_t * ((2 / (gamma + 1)) ** (1 / (gamma - 1))) * ((Pc0 / Pe) ** (1 / gamma))
    * np.sqrt( (gamma - 1) / (gamma + 1) * ((1 - ((Pc0 / Pe) ** ((1 - gamma)/gamma)
D_e = np.sqrt(4*A_e / np.pi)

A_c = (MachT * A_t / MachC) * \
np.sqrt(((1 + (((gamma - 1) * MachC ** 2)/2))/(1+ (((gamma - 1) * MachT ** 2)/2)))
D_c = np.sqrt(4*A_c / np.pi)
Lc = Lstar * A_t / A_c
l_nozzle = (D_e - D_t) / (2 * np.tan(15 * np.pi / 180))

print(f"Throat Diameter: {D_t:.3f} Meters")
print(f"Exit Diameter: {D_e:.3f} Meters")
print(f"Chamber Diameter: {D_c:.3f} Meters")
print(f"Minimum chamber Length: {Lc:.3f} Meters")
print(f"Required nozzle length: {l_nozzle:.3f} Meters")
```

```
Throat Diameter: 0.776 Meters
Exit Diameter: 2.434 Meters
Chamber Diameter: 1.899 Meters
Minimum chamber Length: 0.192 Meters
Required nozzle length: 3.094 Meters
```

In [125…

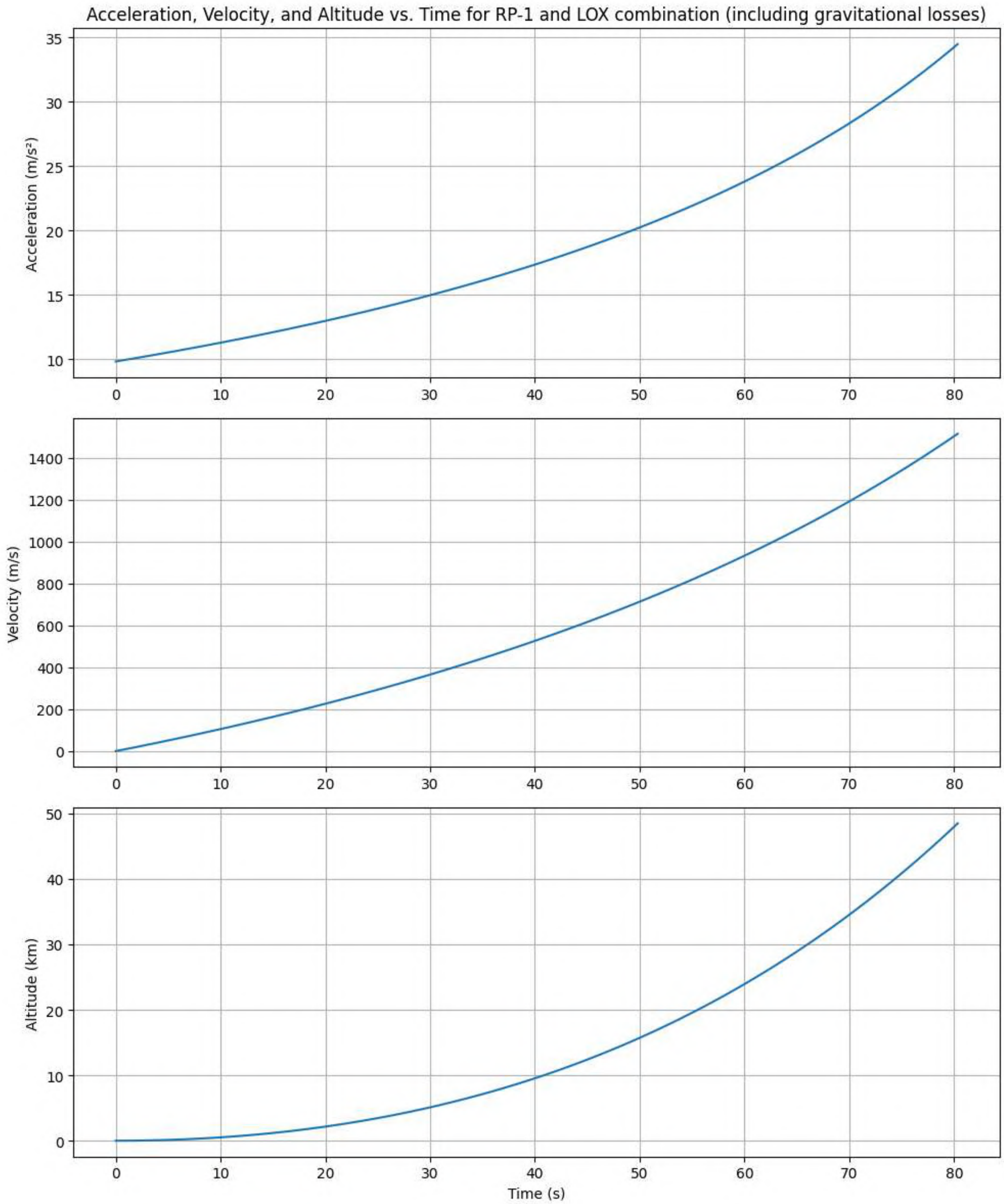
```python
#Calculating Fuel Parameters (LOX/RP-1)


def find_prop_mass(masses):
    m_prop_guess, m_structure_guess = masses
    eq1 = m_prop_guess - (m_payload + m_structure_guess) * (np.exp(deltav / (Isp *
    eq2 = m_structure_guess - (m_prop_guess / 9)
    return (eq1, eq2)

# Provide an initial guess for [m_prop, m_structure]
initial_guess = [100000, 10000]  # You can adjust these as needed

# Solve the system
solution = fsolve(find_prop_mass, initial_guess)
m_prop, m_structure = solution

m_fuel = m_prop / (OFR + 1)
m_ox = m_prop / (1/OFR + 1)
V_lox = m_ox / rho_lox
V_rp1 = m_fuel / rho_rp1
V_tank =  V_lox + V_rp1
h_lox = 4 * V_lox / (np.pi * D_tank ** 2)
h_rp1 = 4 * V_rp1 / (np.pi * D_tank ** 2)
h_tank = h_lox + h_rp1
P_u_lox = T * V_lox / h_lox
P_u_rp1 = T * V_rp1 / h_rp1
P_rp1 = P_u_rp1 + rho_rp1 * (g0+ a_max) * h_rp1 - Pa
P_lox = P_u_lox + rho_lox * (g0+a_max) * h_lox - Pa
t_lox = P_lox * D_tank / (2 * sigma_w)
t_rp1 = P_rp1 * D_tank / (2 * sigma_w)
t_req = max(t_lox,t_rp1)
print(f"Propellant Mass: {m_prop:.1f} Kilograms")
print(f"Structural Mass: {m_structure:.1f} Kilograms")
print(f"Total fuel mass: {m_fuel:.1f} Kilograms")
print(f"Total oxidizer mass: {m_ox:.1f} Kilograms")
print(f"Volume of LOX Section: {V_lox:.1f} Cubic Meters")
print(f"Volume of RP-1 Section: {V_rp1:.2f} Cubic Meters")
print(f"Total Tank Volume: {V_tank:.2f} Cubic Meters")
print(f"Length of LOX Section: {h_lox:.2f} Meters")
print(f"Length of RP-1 Section: {h_rp1:.2f} Meters")
print(f"Total Tank Length: {h_tank:.2f} Meters")
print(f"Required Tank Thickeness: {t_req * 1000:.4f} MilliMeters")
```

```
Propellant Mass: 145018.2 Kilograms
Structural Mass: 16113.1 Kilograms
Total fuel mass: 43944.9 Kilograms
Total oxidizer mass: 101073.3 Kilograms
Volume of LOX Section: 88.6 Cubic Meters
Volume of RP-1 Section: 54.62 Cubic Meters
Total Tank Volume: 143.20 Cubic Meters
Length of LOX Section: 12.53 Meters
Length of RP-1 Section: 7.73 Meters
Total Tank Length: 20.26 Meters
Required Tank Thickeness: 73.4640 MilliMeters
```

In [129…

```
#Plotting Altitude, Velocity, and Acceleration (LOX/RP-1)
mach = np.linspace(0.2, 6, num=200, endpoint=True)
mdot = T / (cstar * C_f)
t_bo = m_prop / mdot
time = np.linspace(0, t_bo, 100)

def vertical_launch(t, y, Isp, m_prop, t_bo):

    v = y[0]
    h = y[1]
    m = y[2]

    gravity = g0 * (radius_earth / (radius_earth + h))**2

    dmdt = -m_prop / t_bo
    dvdt = (-Isp * g0 / m) * dmdt - gravity
    dhdt = v

    return [dvdt, dhdt, dmdt]



sol = solve_ivp(
    vertical_launch, [0, t_bo], [0, 0, mass_initial], method='RK45',
    args=(Isp, m_prop, t_bo),
    dense_output=True
    )

z = sol.sol(time)
acceleration = (-Isp * g0 / z[2, :]) * (-m_prop / t_bo) - \
               g0 * (radius_earth / (radius_earth + z[1, :]))**2



fig, axes = plt.subplots(3, 1, figsize=(10, 12))

# Acceleration vs. Time
axes[0].plot(time, acceleration)
axes[0].set_ylabel('Acceleration (m/s²)')
axes[0].set_title('Acceleration, Velocity, and Altitude vs. Time for RP-1 and LOX c
axes[0].grid(True)

# Velocity vs. Time
axes[1].plot(time, z[0, :])
```

```
axes[1].set_ylabel('Velocity (m/s)')
axes[1].grid(True)

# Altitude vs. Time
axes[2].plot(time, z[1, :] / 1000)
axes[2].set_ylabel('Altitude (km)')
axes[2].set_xlabel('Time (s)')
axes[2].grid(True)

plt.tight_layout()
plt.show()


print(f" total burnout time: {t_bo:.2f} Seconds")
a_max = acceleration[99]
```




```
total burnout time: 80.34 Seconds
```

In [ ]:

Part 1: Liquid Rocket engine.

LO2/LH2

A/B) O/F Ratio and Specific Impulse We can do the same steps for LH2/LO2 as we did for RP-1/LO2 to find the optimal fuel/oxidizer ratio. The CEA tool results in the following values after iterating through to our optimal oxidizer/fuel ratio.

```
          THEORETICAL ROCKET PERFORMANCE ASSUMING FROZEN COMPOSITION
                               AFTER POINT 2

 Pin =  1000.0 PSIA
 CASE = _______________

             REACTANT                    WT FRACTION      ENERGY      TEMP
                                          (SEE NOTE)     KJ/KG-MOL      K
 FUEL        H2(L)                        1.0000000     -9012.000    20.270
 OXIDANT     O2(L)                        1.0000000    -12979.000    90.170

 O/F=    3.30000  %FUEL= 23.255814  R,EQ.RATIO= 2.405055  PHI,EQ.RATIO= 2.405055

                 CHAMBER   THROAT     EXIT
 Pinf/P            1.0000   1.7908   68.050
 P, BAR            68.947   38.501   1.0132
 T, K             2616.67  2355.25  1089.45
 RHO, KG/CU M    2.7409 0 1.7027 0 9.6870-2
 H, KJ/KG        -1350.93 -2741.78 -8548.28
 U, KJ/KG        -3866.45 -5002.93 -9594.21
 G, KJ/KG        -62520.5 -57800.0 -34016.2
 S, KJ/(KG)(K)    23.3769  23.3769  23.3769

 M, (1/n)           8.649    8.661    8.661
 Cp, KJ/(KG)(K)    5.6100   5.2290   4.0941
 GAMMAs            1.2176   1.2302   1.3063
 SON VEL,M/SEC     1750.1   1667.8   1168.9
 MACH NUMBER        0.000    1.000    3.246

 PERFORMANCE PARAMETERS

 Ae/At                      1.0000   7.7269
 CSTAR, M/SEC               2427.9   2427.9
 CF                         0.6870   1.5627
 Ivac, M/SEC                3023.6   4069.7
 Isp, M/SEC                 1667.8   3794.0

 MASS FRACTIONS

 *H              0.00018   *H2             0.13570   H2O             0.86369
 *OH             0.00042

  * THERMODYNAMIC PROPERTIES FITTED TO 20000.K

 NOTE. WEIGHT FRACTION OF FUEL IN TOTAL FUELS AND OF OXIDANT IN TOTAL OXIDANTS
```

$$Optimal\, O/F = 2.6$$

To find the optimal specific impulse, we simply need to divide the exit Isp value given by 9.806 to get into units of seconds. This results in:

$$I_{sp,sl} = 401.4\, seconds$$

Note: For parts C,D,E I will use the same equations as with the previous proj The only change will be the numbers determined by CEA.

C) Engine and Nozzle Dimensions

Using CEA, we were also able to determine our Gamma, Thrust Coefficient, and C*.

Again, let's find the area of the throat and the area of the exit for our liquid engine.

$$A_t = \frac{T}{p_0\, C_F}$$

$$A_e \;=\; A_t \cdot \left(\frac{2}{\gamma+1}\right)^{\frac{1}{\gamma-1}} \left(\frac{p_0}{p_e}\right)^{\frac{1}{\gamma}} \sqrt{\left[\frac{\gamma-1}{\gamma+1}\left(1 \;-\; \left(\frac{p_0}{p_e}\right)^{\frac{1-\gamma}{\gamma}}\right)\right]^{-1}}$$

Assuming the same mach numbers, we can calculate the chamber area:

$$A_c \;=\; A_t \cdot \frac{M_t}{M_c} \sqrt{\left(\frac{1 \;+\; \frac{\gamma-1}{2}\, M_c^2}{1 \;+\; \frac{\gamma-1}{2}\, M_t^2}\right)^{\frac{\gamma+1}{\gamma-1}}}$$

Since we're using a differen fuel, our L* will be a different value as well. In the table given in class, the minimum L* values for LOX/LH2 are between 0.7-1 cm, so we will again take the middle value of 0.85cm as our L*.

$$L_c = \frac{L^*\, A_t}{A_c}$$

Again, we will assume a standard conical nozzle with an opening angle of 15 degrees. We can use the following equation to determine our required nozzle length.

$$L_{nozzle} = \frac{D_e + D_t}{2 \cdot tan(15\backslash\text{degree})}$$

D) Propellant Mass, Propellant Tank Size and Thickness

We'll use the same process to determine propellant and structural mass as with the LO2/RP-1 combination.

$$m_{\text{prop}} = \left(m_{\text{payload}} + m_{\text{structure}}\right)\left(e^{\frac{\Delta V}{I_{\text{sp}}\, g_0}} - 1\right)$$

$$m_{prop} = 9\, m_{structure}$$

$$m_{fuel} = \frac{m_{prop}}{OFR + 1}$$

and

$$m_{ox} = \frac{m_{prop}}{\frac{1}{OFR} + 1}$$

Now, to find propellant tank size. We already know the density of LO2, but we now need to find the density of LH2. Using the following journal:

https://www.cetjournal.it/cet/21/86/038.pdf

We find the density of liquid hydrogen at boiling point is 70.85 kg/m^3

We know that our tank diameter is constrained to 3 meters so we will assume the outer diameter of the tank is 3 meters.

To find the total volume of both tanks, we can do the following:

$$V_{tank} = V_{LH2} + V_{LOX} = m_{ox}\rho_{LOX} + m_{fuel}\rho_{LH2}$$

To find tank length for each, we can use the following equations:

$$h_{LOX} = \frac{4\pi V_{LOX}}{D_{tank}^2}$$

$$h_{LH2} = \frac{4\pi V_{LH2}}{D_{tank}^2}$$

Next, we can use tank thickness. We can use the following equation:

$$t = \frac{P \cdot D_{tank}}{2\,\sigma_w}$$

Where σ_w is the same value as the previous iteration.

Again, we find the pressure in the tanks:

$$P_{LOX} = P_u + \rho_{LOX} \cdot (g_0 + a_{max}) \cdot h_{LOX} - P_a$$

$$P_{LH2} = P_u + \rho_{LH2} \cdot (g_0 + a_{max} \cdot h_{LH2} + P_{LOX} - P_a$$

where $P_u = \frac{T}{A_{tank}}$

finally, plugging in all variables results in:

$$t_{LOX} = \frac{P_{LOX} \cdot D_{tank}}{2\,\sigma_w}$$

$$t_{LH2} = \frac{P_{LH2} \cdot D_{tank}}{2\,\sigma_w}$$

E) Plotting Altitude, Velocity, and Acceleration

Again, we can use all the same equations to determine our altitude, velocity and acceleration.

$$\frac{dv}{dt} = -\frac{I_{\rm sp}\, g_0}{m}\,\frac{dm}{dt} \;-\; g(h)$$

$$\frac{dh}{dt} = v$$

$$\frac{dm}{dt} = -\dot{m} = -\frac{m_p}{t_b}$$

$$g(h) \;=\; G_0\left(\frac{R_{\rm earth}}{R_{\rm earth}+h}\right)^2$$

In [32]:
```
#Initializing variables (LOX/LH2)
import numpy as np
from matplotlib import pyplot as plt
from scipy.interpolate import UnivariateSpline
from scipy.integrate import solve_ivp
from scipy.optimize import fsolve
#Non-Unique Parameters
g0 = 9.806 #m/s^2
deltav = 2300 #m/s
lam = 0.9
Pc0 = 6894760 #Pascals
Pa = 101325 #Pa
Pe = Pa
m_payload = 100000 #kg
D_tank = 3 #m
MachT = 1
MachC = 0.1
radius_earth = 6378388
sigma_w = 750000000 #Pa

#Unique Parameters
Lstar = 0.85 #meters
rho_lox = 1141 #kg/m^3
rho_lh2 = 70.85 #kg/m^3


#from CEA
cstar = 2427.9 #m/s
gamma = 1.23
OFR = 3.3
Isp=3794.0/g0 #s
C_f = 1.5627
```

```python
print(f"Optimal Oxidizer/Fuel Ratio: {OFR}")
print(f"Specific Impulse at Sea Level: {Isp:.2f} seconds")
```

```
Optimal Oxidizer/Fuel Ratio: 3.3
Specific Impulse at Sea Level: 386.91 seconds
```

In [34]:

```python
#Calculating Engine and Nozzle Dimensions (LOX/LH2)
mass_initial = m_payload + m_structure + m_prop
T = 2 * mass_initial * g0
A_t = T / (C_f * Pc0)
D_t = np.sqrt(4*A_t / np.pi)
A_e = A_t * ((2 / (gamma + 1)) ** (1 / (gamma - 1))) * ((Pc0 / Pe) ** (1 / gamma))
    * np.sqrt( (gamma - 1) / (gamma + 1) * ((1 - ((Pc0 / Pe) ** ((1 - gamma)/gamma)
D_e = np.sqrt(4*A_e / np.pi)

A_c = (MachT * A_t / MachC) * \
np.sqrt(((1 + (((gamma - 1) * MachC ** 2)/2))/(1+ (((gamma - 1) * MachT ** 2)/2)))
D_c = np.sqrt(4*A_c / np.pi)
Lc = Lstar * A_t / A_c
l_nozzle = (D_e - D_t) / (2 * np.tan(15 * np.pi / 180) )

print(f"Throat Diameter: {D_t:.3f} Meters")
print(f"Exit Diameter: {D_e:.3f} Meters")
print(f"Chamber Diameter: {D_c:.3f} Meters")
print(f"Minimum chamber Length: {Lc:.3f} Meters")
print(f"Required nozzle length: {l_nozzle:.3f} Meters")
```

```
Throat Diameter: 0.684 Meters
Exit Diameter: 1.980 Meters
Chamber Diameter: 1.667 Meters
Minimum chamber Length: 0.143 Meters
Required nozzle length: 2.418 Meters
```

In [42]:

```python
#Calculating Fuel Parameters (LOX/LH2)


def find_prop_mass(masses):
    m_prop_guess, m_structure_guess = masses
    eq1 = m_prop_guess - (m_payload + m_structure_guess) * (np.exp(deltav / (Isp *
    eq2 = m_structure_guess - (m_prop_guess / 9)
    return (eq1, eq2)

# Provide an initial guess for [m_prop, m_structure]
initial_guess = [100000, 10000]  # You can adjust these as needed

# Solve the system
solution = fsolve(find_prop_mass, initial_guess)
m_prop, m_structure = solution

m_fuel = m_prop / (OFR + 1)
m_ox = m_prop / (1/OFR + 1)
V_lox = m_ox / rho_lox
V_lh2 = m_fuel / rho_lh2
V_tank =  V_lox + V_lh2
h_lox = 4 * V_lox / (np.pi * D_tank ** 2)
h_lh2 = 4 * V_lh2 / (np.pi * D_tank ** 2)
h_tank = h_lox + h_lh2
```

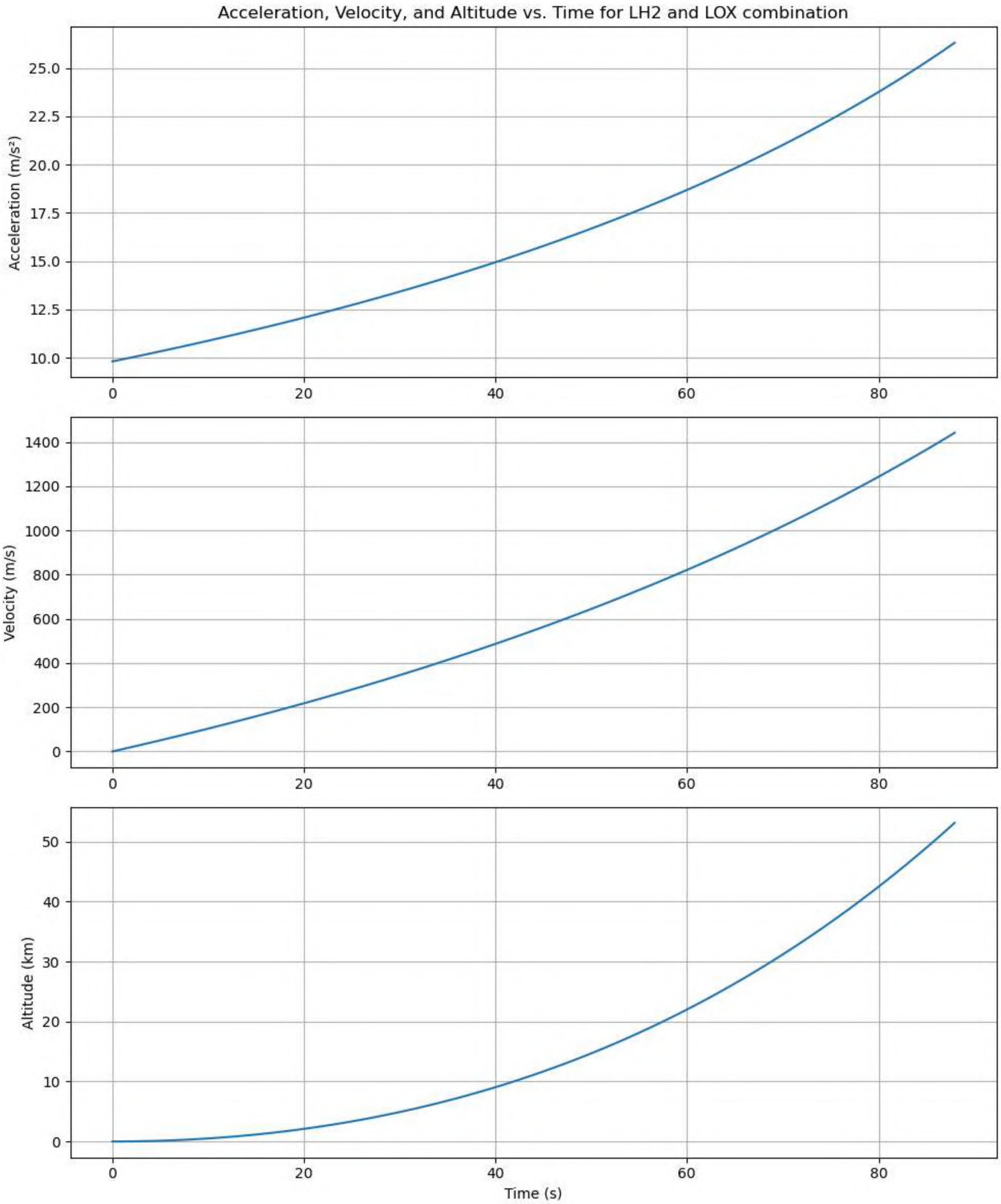
```
P_u_lox = T * V_lox / h_lox
P_u_lh2 = T * V_lh2 / h_lh2
P_lh2 = P_u_lh2 + rho_lh2 * (g0+ a_max) * h_lh2 - Pa
P_lox = P_u_lox + rho_lox * (g0+a_max) * h_lox - Pa
t_lox = P_lox * D_tank / (2 * sigma_w)
t_lh2 = P_lh2 * D_tank / (2 * sigma_w)
t_req = max(t_lox,t_lh2)
print(f"Propellant Mass: {m_prop:.1f} Kilograms")
print(f"Structural Mass: {m_structure:.1f} Kilograms")
print(f"Total fuel mass: {m_fuel:.1f} Kilograms")
print(f"Total oxidizer mass: {m_ox:.1f} Kilograms")
print(f"Volume of LOX Section: {V_lox:.1f} Cubic Meters")
print(f"Volume of LH2 Section: {V_lh2:.1f} Cubic Meters")
print(f"Total Tank Volume: {V_tank:.1f} Cubic Meters")
print(f"Length of LOX Section: {h_lox:.1f} Meters")
print(f"Length of LH2 Section: {h_lh2:.1f} Meters")
print(f"Total Tank Length: {h_tank:.1f} Meters")
print(f"Required Tank Thickeness: {t_req * 1000:.5f} MilliMeters")
```

```
Propellant Mass: 91855.6 Kilograms
Structural Mass: 10206.2 Kilograms
Total fuel mass: 21361.8 Kilograms
Total oxidizer mass: 70493.8 Kilograms
Volume of LOX Section: 61.8 Cubic Meters
Volume of LH2 Section: 301.5 Cubic Meters
Total Tank Volume: 363.3 Cubic Meters
Length of LOX Section: 8.7 Meters
Length of LH2 Section: 42.7 Meters
Total Tank Length: 51.4 Meters
Required Tank Thickeness: 56.54102 MilliMeters
```

In [38]:

```
#Plotting Altitude, Velocity, and Acceleration (LOX/LH2)

mach = np.linspace(0.2, 6, num=200, endpoint=True)
mass_initial = m_payload + m_structure + m_prop
mdot = T / (cstar * C_f)
t_bo = m_prop / mdot

time = np.linspace(0, t_bo, 100)

def vertical_launch(t, y, Isp, m_prop, t_bo):

    v = y[0]
    h = y[1]
    m = y[2]

    gravity = g0 * (radius_earth / (radius_earth + h))**2

    dmdt = -m_prop / t_bo
    dvdt = (-Isp * g0 / m) * dmdt - gravity
    dhdt = v

    return [dvdt, dhdt, dmdt]
```

```python
sol = solve_ivp(
    vertical_launch, [0, t_bo], [0, 0, mass_initial], method='RK45',
    args=(Isp, m_prop, t_bo),
    dense_output=True
    )

z = sol.sol(time)

acceleration = (-Isp * g0 / z[2, :]) * (-m_prop / t_bo) - \
               g0 * (radius_earth / (radius_earth + z[1, :]))**2



fig, axes = plt.subplots(3, 1, figsize=(10, 12))

# Acceleration vs. Time
axes[0].plot(time, acceleration)
axes[0].set_ylabel('Acceleration (m/s²)')
axes[0].set_title('Acceleration, Velocity, and Altitude vs. Time for LH2 and LOX co
axes[0].grid(True)

# Velocity vs. Time
axes[1].plot(time, z[0, :])
axes[1].set_ylabel('Velocity (m/s)')
axes[1].grid(True)

# Altitude vs. Time
axes[2].plot(time, z[1, :] / 1000)
axes[2].set_ylabel('Altitude (km)')
axes[2].set_xlabel('Time (s)')
axes[2].grid(True)

plt.tight_layout()
plt.show()

print(f" total burnout time: {t_bo:.2f} Seconds")
a_max = acceleration[99]
```




```
total burnout time: 87.94 Seconds
```

```
In [ ]:
```

Part 2: Solid Rocket Motor - Aluminum Fuel/Ammonium Perchlorate Oxidi

A/B) O/F Ratio and Specific Impulse

Getting our O/F ratio for the solid rocket motor will be quite a bit different than getting a O/F ratio for a liquid engine.

First, we will define our two Propellant mixtures.

For both, I will use butadiene as my binder.

For my first combination, I will use ammonium perchlorate (NH_4ClO_4) as my oxidizer and Aluminum (Al) as my fuel.

First, I will be finding the densities of the fuel and oxidizer.

I found the density of ammonium perchloarate from the following website: https://pubchem.ncbi.nlm.nih.gov/compound/Ammonium-perchlorate I found the density of Aluminum from the following website: https://www.engineeringtoolbox.com/metal-alloys-densities-d_50.html I found the density of HTPB from the following site: https://pubchem.ncbi.nlm.nih.gov/element/Boron#section=Melting-Point

To determine optimal oxidizer/fuel ratio, I used a fuel mixture of 85% Al and 15% Butadiene. I iterated through oxidizer/fuel ratios from 2-5 at intervals of 0.1. From here, I got a CEA output of the oxidizers and fuels, and found the optimal oxidizer/fuel ratio at 3.0

From here, we need to calculate the true oxidizer/fuel ratio, since butaiene is simply a binder and does not count as the fuel.

We know that at an O/F ratio of 3, our oxidizer will make up

$$Oxidizer = \frac{\frac{O}{F}_0}{1 + \frac{O}{F}_0}$$

and our total "Fuel + Binder" will make up

$$Fuel + Binder = \frac{1}{1 + \frac{O}{F}_0}$$

Since we know our data is taken at 85% Al and 15% Butadiene, we know our total Fuel fraction is:

$$Fuel = Fuel_{percent} \cdot \frac{1}{1 + \frac{O}{F}_0}$$

$$Binder = Binder_{percent} \cdot \frac{1}{1 + \frac{O}{F}_0}$$

So, our actual optimal O/F ratio is:

$$\frac{O}{F}_{true} = \frac{\frac{\frac{O}{F}_0}{1+\frac{O}{F}_0}}{Fuel_{percent} \cdot \frac{1}{1+\frac{O}{F}_0}}$$

THEORETICAL ROCKET PERFORMANCE ASSUMING FROZEN COMPOSITION
AFTER POINT 2

Pin = 1000.0 PSIA
CASE = _______________

	REACTANT	WT FRACTION (SEE NOTE)	ENERGY KJ/KG-MOL	TEMP K
FUEL	AL	0.8500000	0.000	0.000
FUEL	C4H6,butadiene	0.1500000	0.000	0.000
OXIDANT	NH4CLO4(I)	1.0000000	0.000	0.000

O/F= 3.00000 %FUEL= 25.000000 R,EQ.RATIO= 1.121173 PHI,EQ.RATIO= 1.21

	CHAMBER	THROAT	EXIT
Pinf/P	1.0000	1.7162	68.050
P, BAR	68.947	40.174	1.0132
T, K	4181.37	4012.03	2352.69
RHO, KG/CU M	6.5016 0	3.9975 0	1.7192-1
H, KJ/KG	0.00000	-557.64	-3428.85
U, KJ/KG	-1060.47	-1562.60	-4018.17
G, KJ/KG	-37874.2	-36898.0	-24739.1
S, KJ/(KG)(K)	9.0578	9.0578	9.0578
M, (1/n)	32.784	33.193	33.193
MW, MOL WT	29.720	29.905	29.905
Cp, KJ/(KG)(K)	9.2048	8.9869	1.6885
GAMMAs	1.1113	1.1098	1.1742
SON VEL,M/SEC	1085.6	1056.1	831.9
MACH NUMBER	0.000	1.000	3.148

PERFORMANCE PARAMETERS

Ae/At	1.00000	9.3768
CSTAR, M/SEC	1633.2	1633.2
CF	0.6466	1.6034
Ivac, M/SEC	2007.7	2843.8
Isp, M/SEC	1056.1	2618.7

MASS FRACTIONS

*AL	0.00063	ALCL	0.01818	ALCL2	0.00185
ALCL3	0.00033	ALH	0.00007	ALHCL	0.00006
ALHCL2	0.00006	*ALO	0.00601	ALOCL	0.00732
ALOCL2	0.00008	ALOH	0.02019	ALOHCL	0.00450
ALOHCL2	0.00282	ALO2	0.00092	AL(OH)2	0.00209
AL(OH)2CL	0.00163	AL(OH)3	0.00071	AL2O	0.00074
AL2O2	0.00126	AL2O3	0.00005	*CO	0.06594
*CO2	0.01841	*CL	0.06558	CLO	0.00060
CL2	0.00026	*H	0.00277	HALO	0.00005
HALO2	0.00052	HCO	0.00001	HCL	0.14424
HNO	0.00002	HOCL	0.00013	HO2	0.00011
*H2	0.00764	H2O	0.10817	H2O2	0.00001
*N	0.00004	*NH	0.00001	*NO	0.01194
NOCL	0.00001	NO2	0.00001	*N2	0.08377
*O	0.01658	*OH	0.04616	*O2	0.01977
AL2O3(L)	0.33773				

C) Engine and Nozzle Dimensions

Similar to the liquid engine design, using CEA, we were also able to determine our Gamma, Thrust Coefficient, and C*.

Many of the calculations for this section are very similar to the calculations for the liquid engine design, so I'll go over the equations briefly. First, let's find the area of the throat and the area of the exit for our liquid engine.

$$A_t = \frac{T}{p_0 \, C_F}$$

We are given a minimum thrust value, so for our iteration of throat area, we can use that minimum thrust value given, similar to the liquid engine design.

$$A_e \;=\; A_t \cdot \left(\frac{2}{\gamma+1}\right)^{\frac{1}{\gamma-1}} \left(\frac{p_0}{p_e}\right)^{\frac{1}{\gamma}} \sqrt{\left[\frac{\gamma-1}{\gamma+1}\left(1 \;-\; \left(\frac{p_0}{p_e}\right)^{\frac{1-\gamma}{\gamma}}\right)\right]^{-1}}$$

Our Mach number at the throat is by definition 1, and our mach number in the chamber we will again assume as 0.1.

$$A_c \;=\; A_t \cdot \frac{M_t}{M_c} \sqrt{\left(\frac{1 \;+\; \frac{\gamma-1}{2}\, M_c^2}{1 \;+\; \frac{\gamma-1}{2}\, M_t^2}\right)^{\frac{\gamma+1}{\gamma-1}}}$$

This is where the solid engine differs from the liquid engine characteristics substantially, and will be using largely new equations.

To find our nozzle length, we will again assume a standard conical nozzle with an opening angle of 15 degrees.

$$L_{nozzle} = \frac{D_e + D_t}{2 \cdot tan(15\text{\degree})}$$

Next, we can calculate grain length and volume. First, we can calculate the grain volume(assuming grain volume includes binder volume) using values we will find in section D. We will find the total oxidizer and fuel masses, then divide them by their respective densities. Note the masses/densities were found in the following section, and the sources/calculations will be shown in section D.

$$V_{grain} = m_{ox}\, \rho_{ap} + m_{al}\, \rho_{al} + m_{bu}\, \rho_{bu}$$

Next, we can caluclate grain length.

$$L_{grain} = \frac{4 \cdot V_{grain}}{\pi(D_c^2 - D_i^2)}$$

Now, to find our burn areas, we will assume the outer diameter of the grain is equal to chamber diameter, and we will we will assume the inner diameter is equal to $\frac{1}{2}$ of the outer diameter.

First, we will calculate the initial burn area. We can use the following equation:

$$A_{b,0} = \frac{\pi(D_c^2 - D_i^2)}{4} + \pi D_i \, L_{grain}$$

Next, we can caluclate the final burn area:

$$A_{b,f} = \pi D_c (L_{grain} - \frac{D_i}{2})$$

D) Propellant Mass, Propellant Tank Size and Thickness

to calculate the propellant mass(including the binder mass), we can use the rocket equation

$$\Delta V = I_{\mathrm{sp}} \, g_0 \, \ln\!\left(\frac{m_{\mathrm{prop}} + m_{\mathrm{payload}} + m_{\mathrm{structure}}}{m_{\mathrm{payload}} + m_{\mathrm{structure}}} \right)$$

Rearranging gives us:

$$m_{\mathrm{prop}} = \left(m_{\mathrm{payload}} + m_{\mathrm{structure}}\right)\left(e^{\frac{\Delta V}{I_{\mathrm{sp}} \, g_0}} - 1\right)$$

Now, we can use our knowledge about the mass fraction of our booster:

$$\lambda = 0.9 = \frac{m_{prop}}{m_{sctucture} + m_{prop}}$$

$$m_{prop} = 9 \, m_{structure}$$

We can then use a system of linear equations solver in Python to determine our true Propellant and Structural Masses.

Next, we can calculate the fuel and oxidizer mass fractions since we know $m_{ox} = 3m_{fuel}$

Since we know they both add to the total propellant mass, we can say

$$m_{fuel,total} = \frac{m_{prop}}{OFR + 1}$$

Since our fuel is 85 percent aluminum by weight, we simply multiply total fuel mass by the fraction of total fuel mass of each component.

$$m_a l = 0.85 \cdot m_{fuel,total}$$

$$m_b u = 0.15 \cdot m_{fuel,total}$$

and to find our mass of ammonium perchlorate:

$$m_{ox} = \frac{m_{prop}}{\frac{1}{OFR} + 1}$$

To find the required grain volume, we must find the densities of our fuel and our oxidizer. To find the required volume of the grain, we can make a couple quick assumptions. First, we know that the fuel and oxidizer both make up the volume of the grian, and we will also calculate the volume of the binder in this grain volume.

To find the total volume of the propellant grain(including the binder), we can do the following:

$$V_{grain} = m_{ox}\,\rho_{ap} + m_{al}\,\rho_{al} + m_{bu}\,\rho_{bu}$$

Now that we have the volume of the chamber, we want to find the chamber thickness

$$t = \frac{P \cdot D_{chamber}}{2\,\sigma_w}$$

To find ultimate tensile strength, we can assume our material is titanium, and using the following site: https://asm.matweb.com/search/SpecificMaterial.asp?bassnum=MTP641

We find our ultimate tensile strength of titanium to be 950 GPA. Now, we can divide by 1.25 to get our stress to be used for our tank thickness equation.

Next, to find our chamber pressure, we must use the equation for chamber pressure in solid propellant designs, as well as the fact that ambient pressure acts an opposing pressure on the chamber. We know the highest chamber pressure is at the final time, so we will calculate the value at the final time.

Our net chamber pressure will be:

$$P_{c,f} = (\frac{\rho_{prop}\,c^*\,a\,A_{b,f}}{A_throat})^{\frac{1-n}{n}} - P_a$$

We already know all values, and propellant density can be calculated by dividing total propellant mass by total grain volume.

finally, plugging in all variables results in:

$$t_{chamber} = \frac{P_{c,f} \cdot D_{chamber}}{2\,\sigma_w}$$

E) Plotting Altitude, Velocity, and Acceleration

We will use the same process for determining the altitude, velocity, and acceleration as we did in the liquid engine. We will use the following equations to determine these parameters as a function of time.

$$T_{avg} = 2 \cdot m_{booster,0}$$

$$\dot{m_{avg}} = \frac{T_{avg}}{c^{*} C_F}$$

$$t_{bo} = \frac{m_{prop}}{\dot{m_{avg}}}$$

$$g(h) \; = \; G_0 \left(\frac{R_{\text{earth}}}{R_{\text{earth}} + h} \right)^2$$

$$\frac{dv}{dt} = -\frac{I_{\text{sp}}\, g_0}{m} \frac{dm}{dt} \; - \; g(h)$$

$$\frac{dh}{dt} = v$$

$$\frac{dm}{dt} = -\dot{m} = -\frac{m_p}{t_b}$$

In [144…

```
#Initializing variables (Solid Motor: Ammonium Perchlorate/Aluminum with Butadiene

import numpy as np
from matplotlib import pyplot as plt
from scipy.interpolate import UnivariateSpline
from scipy.integrate import solve_ivp
from scipy.optimize import fsolve

#Non-Unique Parameters
g0 = 9.806 #m/s^2
deltav = 2300 #m/s
lam = 0.9
Pc0 = 6894760 #Pascals
Pa = 101325 #Pa
Pe = Pa
m_payload = 100000 #kg
D_tank = 3 #m
MachT = 1
MachC = 0.1
radius_earth = 6378388
sigma_w = 750000000 #Pa


#Unique Parameters
rho_ap = 1950 #kg/m^3
rho_al= 2712 #kg/m^3
rho_butadiene = 614.9 #kg/m^3
Al_Fuel_Fraction = 0.15
n = 0.4
a_0 = 0.04 * 0.0254
a = a_0 / (6894.75 ** n)
rdot_0 = a_0 * ((Pc0 * 0.0001450377 ) ** n) #m/s


#from CEA
```

```
cstar = 1633.2 #m/s
gamma = 1.174
Isp = 2618.7 / g0
OF_cea = 3.0
OFR = (OF_cea / (1 + OF_cea) ) / (Al_Fuel_Fraction / (1 + OF_cea) )
C_f = 1.604
print(f"Optimal Oxidizer/Fuel Ratio: {OFR:.2f}")
print(f"Specific Impulse at Sea Level: {Isp:.2f} seconds")
```

```
Optimal Oxidizer/Fuel Ratio: 20.00
Specific Impulse at Sea Level: 267.05 seconds
```

In [160…

```
#Calculating Engine and Nozzle Dimensions (Ammonium Perchlorate/Aluminum with Butad
mass_initial = m_payload + m_structure + m_prop
T = 2 * mass_initial * g0
A_t = T / (C_f * Pc0)
D_t = np.sqrt(4*A_t / np.pi)
A_e = A_t * ((2 / (gamma + 1)) ** (1 / (gamma - 1))) * ((Pc0 / Pe) ** (1 / gamma))
    * np.sqrt( (gamma - 1) / (gamma + 1) * ((1 - ((Pc0 / Pe) ** ((1 - gamma)/gamma)
D_e = np.sqrt(4*A_e / np.pi)

A_c = (MachT * A_t / MachC) * \
np.sqrt(((1 + (((gamma - 1) * MachC ** 2)/2))/(1+ (((gamma - 1) * MachT ** 2)/2)))
D_c = np.sqrt(4*A_c / np.pi)
l_nozzle = (D_e - D_t) / (2 * np.tan(15 * np.pi / 180) )
D_i = D_c / 2
V_req =  m_ox / rho_ap + m_al / rho_al + m_butadiene / rho_butadiene
l_grain = V_req / (np.pi * ((D_c / 2) ** 2 - (D_i / 2) ** 2))
A_b0 = np.pi * (D_c ** 2  - D_i ** 2) / 4 + np.pi * D_i * l_grain
A_bf = np.pi * D_c * (l_grain - D_i / 2 )



print(f"Throat Diameter: {D_t:.3f} Meters")
print(f"Exit Diameter: {D_e:.3f} Meters")
print(f"Chamber Diameter: {D_c:.3f} Meters")
print(f"Initial Port Radius: {D_i / 2:.3f} Meters")
print(f"Final Port Radius: {D_c / 2:.3f} Meters")
print(f"Required nozzle length: {l_nozzle:.3f} Meters")
print(f"Required Grain Volume: {V_req:.2f} Cubic Meters")
print(f"Required Grain Length: {l_grain:.2f} Meters")
print(f"Initial burn area: {A_b0:.2f} Square Meters")
print(f"Final burn area: {A_bf:.2f} Square Meters")
```

```
Throat Diameter: 0.803 Meters
Exit Diameter: 2.454 Meters
Chamber Diameter: 1.961 Meters
Initial Port Radius: 0.490 Meters
Final Port Radius: 0.981 Meters
Required nozzle length: 3.082 Meters
Required Grain Volume: 87.37 Cubic Meters
Required Grain Length: 38.57 Meters
Initial burn area: 121.07 Square Meters
Final burn area: 234.58 Square Meters
```

In [148…

```
#Calculating Fuel Parameters (Ammonium Perchlorate/Aluminum with Butadiene binder)
```

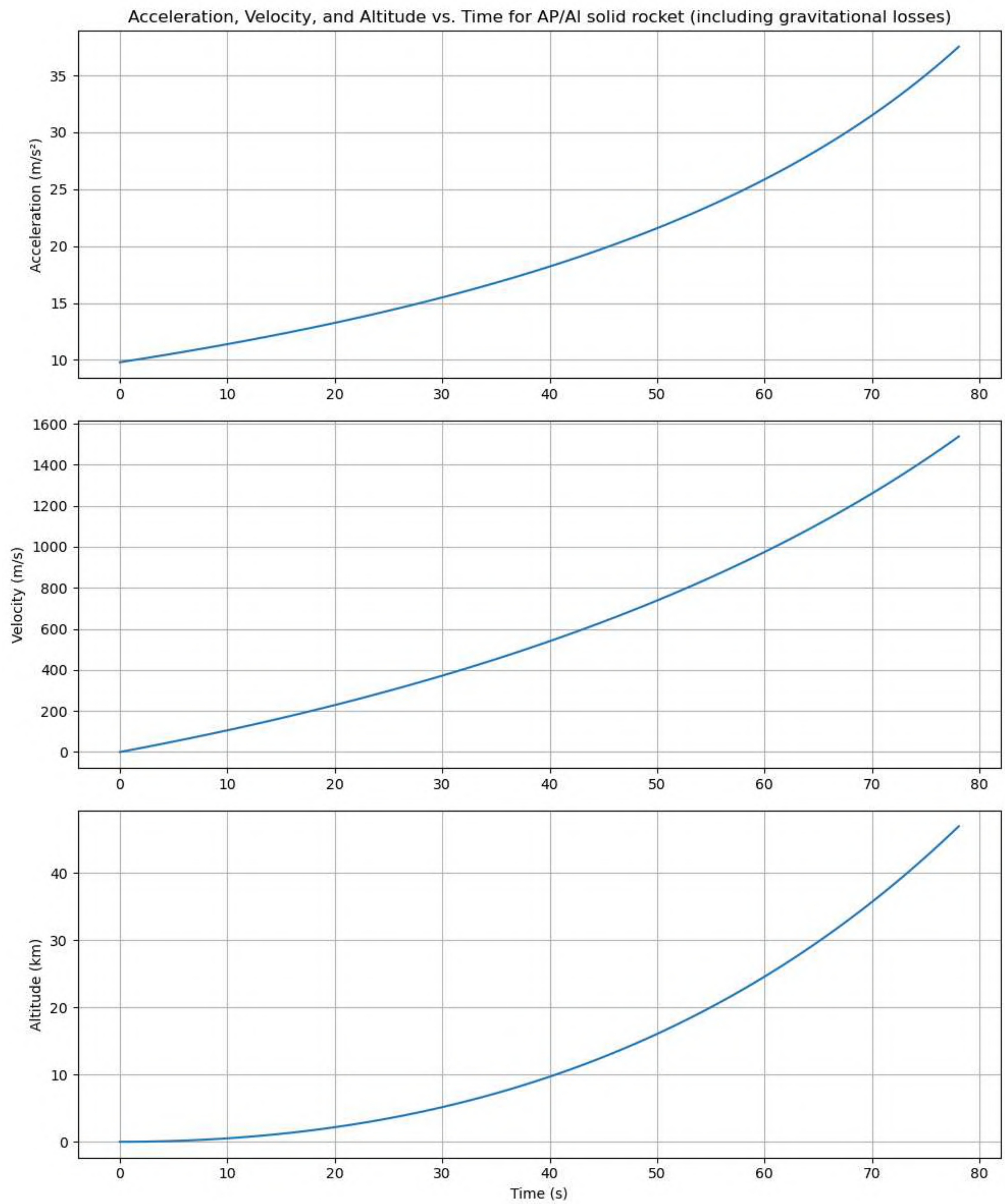
```python
def find_prop_mass(masses):
    m_prop_guess, m_structure_guess = masses
    eq1 = m_prop_guess - (m_payload + m_structure_guess) * (np.exp(deltav / (Isp *
    eq2 = m_structure_guess - (m_prop_guess / 9)
    return (eq1, eq2)

# Provide an initial guess for [m_prop, m_structure]
initial_guess = [100000, 10000]  # You can adjust these as needed

# Solve the system
solution = fsolve(find_prop_mass, initial_guess)
m_prop, m_structure = solution
m_fuel_total = m_prop / (OF_cea + 1)
m_al = 0.85 * m_fuel_total
m_butadiene = 0.15 * m_fuel_total
m_ox = m_prop / (1/OF_cea + 1)
rho_prop = m_prop / V_req
P_cf = (rho_prop * cstar * a * A_bf / A_t) ** (1/(1-n)) - Pa
t_al = P_cf * D_c / (2 * sigma_w)
print(f"Total Propellant Mass(Including Binder): {m_prop:.1f} Kilograms")
print(f"Mass of Ammonium Perchlorate: {m_ox:.1f} Kilograms")
print(f"Mass of Aluminum: {m_al:.1f} Kilograms")
print(f"Mass of Butadiene Binder: {m_butadiene:.1f} Kilograms")
print(f"Structural Mass: {m_structure:.1f} Kilograms")
print(f"Required Chamber Thickeness: {t_al * 1000:.5f} MilliMeters")
```

```
Total Propellant Mass(Including Binder): 166744.0 Kilograms
Mass of Ammonium Perchlorate: 125058.0 Kilograms
Mass of Aluminum: 35433.1 Kilograms
Mass of Butadiene Binder: 6252.9 Kilograms
Structural Mass: 18527.1 Kilograms
Required Thickeness: 43.63420 MilliMeters
```

In [150…

```python
#Plotting Altitude, Velocity, and Acceleration (LOX/RP-1)
mach = np.linspace(0.2, 6, num=200, endpoint=True)
mdot = T / (cstar * C_f)
t_bo = m_prop / mdot
time = np.linspace(0, t_bo, 100)

def vertical_launch(t, y, Isp, m_prop, t_bo):

    v = y[0]
    h = y[1]
    m = y[2]

    gravity = g0 * (radius_earth / (radius_earth + h))**2

    dmdt = -m_prop / t_bo
    dvdt = (-Isp * g0 / m) * dmdt - gravity
    dhdt = v

    return [dvdt, dhdt, dmdt]
```

```
sol = solve_ivp(
    vertical_launch, [0, t_bo], [0, 0, mass_initial], method='RK45',
    args=(Isp, m_prop, t_bo),
    dense_output=True
    )

z = sol.sol(time)
acceleration = (-Isp * g0 / z[2, :]) * (-m_prop / t_bo) - \
               g0 * (radius_earth / (radius_earth + z[1, :]))**2



fig, axes = plt.subplots(3, 1, figsize=(10, 12))

# Acceleration vs. Time
axes[0].plot(time, acceleration)
axes[0].set_ylabel('Acceleration (m/s²)')
axes[0].set_title('Acceleration, Velocity, and Altitude vs. Time for AP/Al solid ro
axes[0].grid(True)

# Velocity vs. Time
axes[1].plot(time, z[0, :])
axes[1].set_ylabel('Velocity (m/s)')
axes[1].grid(True)

# Altitude vs. Time
axes[2].plot(time, z[1, :] / 1000)
axes[2].set_ylabel('Altitude (km)')
axes[2].set_xlabel('Time (s)')
axes[2].grid(True)

plt.tight_layout()
plt.show()


print(f" total burnout time: {t_bo:.2f} Seconds")
a_max = acceleration[99]
```




```
total burnout time: 78.08 Seconds
```

In [ ]:

Part 2: Solid Rocket Motor - Magnesium Fuel/Ammonium Nitrate Oxidizer

A/B) O/F Ratio and Specific Impulse

Again, we will use HTPB for our binder.

I will also be using boron for our fuel, and ammonium nitrate as our oxidizer.

First, I will be finding the densities of the fuel and oxidizer.

I found the density of ammonium nitrate from the following website: https://pubchem.ncbi.nlm.nih.gov/compound/Ammonium-nitrate#section=Density I found the desnity of Magnesium from the following website: https://pubchem.ncbi.nlm.nih.gov/compound/magnesium

This time, I will change up the binder/fuel ratio substantially. I will be using 50% Magnesium and 50% Butadiene by mass. Again, I iterated through oxidizer/fuel ratios from 2-4.5 at intervals of 0.1. From here, I got a CEA output of the oxidizers and fuels, and found the optimal oxidizer/fuel ratio at 2.3.

I used the same process to find the true Oxidizer/Fuel ratio.

$$Oxidizer = \frac{\frac{O}{F}_0}{1 + \frac{O}{F}_0}$$

$$Fuel + Binder = \frac{1}{1 + \frac{O}{F}_0}$$

$$Fuel = Fuel_{percent} \cdot \frac{1}{1 + \frac{O}{F}_0}$$

$$Binder = Binder_{percent} \cdot \frac{1}{1 + \frac{O}{F}_0}$$

So, our actual optimal O/F ratio is:

$$\frac{O}{F}_{true} = \frac{\frac{\frac{O}{F}_0}{1+\frac{O}{F}_0}}{Fuel_{percent} \cdot \frac{1}{1+\frac{O}{F}_0}}$$

```
           THEORETICAL ROCKET PERFORMANCE ASSUMING FROZEN COMPOSITION
                                AFTER POINT 2

Pin =  1000.0 PSIA
CASE = _______________

            REACTANT                    WT FRACTION      ENERGY      TEMP
                                         (SEE NOTE)     KJ/KG-MOL      K
FUEL        C4H6,butadiene               0.5000000        0.000      0.000
FUEL        Mg                           0.5000000        0.000      0.000
OXIDANT     NH4NO3(I)                    1.0000000        0.000      0.000

O/F=    2.30000  %FUEL= 30.303030  R,EQ.RATIO= 2.084866  PHI,EQ.RATIO= 4.254599

                CHAMBER   THROAT     EXIT
Pinf/P            1.0000   1.7509   68.050
P, BAR            68.947   39.377   1.0132
T, K             3208.83  3097.17  1578.75
RHO, KG/CU M    5.4699 0 3.2385 0 1.6347-1
H, KJ/KG        -0.00000  -686.96 -3946.08
U, KJ/KG        -1260.48 -1902.88 -4565.89
G, KJ/KG        -37596.9 -36975.5 -22443.9
S, KJ/(KG)(K)    11.7167  11.7167  11.7167

M, (1/n)          21.166   21.179   21.179
MW, MOL WT        19.168   19.145   19.145
Cp, KJ/(KG)(K)    5.0326   5.9095   2.0085
GAMMAs            1.1434   1.1300   1.2430
SON VEL,M/SEC     1200.5   1172.2    877.7
MACH NUMBER        0.000    1.000    3.201

PERFORMANCE PARAMETERS

Ae/At                      1.0000   8.2658
CSTAR, M/SEC               1816.3   1816.3
CF                         0.6453   1.5467
Ivac, M/SEC                2209.5   3029.9
Isp, M/SEC                 1172.1   2809.3

MASS FRACTIONS

*CO             0.29435  *CO2            0.03057  *H              0.00095
HCN             0.00001  HCO             0.00002  *H2             0.03348
H2O             0.15024  *Mg             0.01830  MgH             0.00017
MgN             0.00003  *MgO            0.00253  MgOH            0.00778
Mg(OH)2         0.01203  NH3             0.00002  *NO             0.00039
*N2             0.24370  *O              0.00009  *OH             0.00311
*O2             0.00004  MgO(cr)         0.20218

 * THERMODYNAMIC PROPERTIES FITTED TO 20000.K

NOTE. WEIGHT FRACTION OF FUEL IN TOTAL FUELS AND OF OXIDANT IN TOTAL OXIDANTS
```

C) Engine and Nozzle Dimensions

Similar to the previous designs, using CEA, we were also able to determine our Gamma, Thrust Coefficient, and C*.

I will use the smae process as the previous solid rocket design to find all engine and nozzle dimensions. The only differences will be the values.

$$A_t = \frac{T}{p_0 C_F}$$

We are given a minimum average thrust, so we will be using that as our required thrust for the design.

$$A_e = A_t \cdot \left(\frac{2}{\gamma+1}\right)^{\frac{1}{\gamma-1}} \left(\frac{p_0}{p_e}\right)^{\frac{1}{\gamma}} \sqrt{\left[\frac{\gamma-1}{\gamma+1}\left(1 - \left(\frac{p_0}{p_e}\right)^{\frac{1-\gamma}{\gamma}}\right)\right]^{-1}}$$

Our Mach number at the throat is by definition 1, and our mach number in the chamber we will again assume as 0.1.

$$A_c = A_t \cdot \frac{M_t}{M_c} \sqrt{\left(\frac{1 + \frac{\gamma-1}{2} M_c^2}{1 + \frac{\gamma-1}{2} M_t^2}\right)^{\frac{\gamma+1}{\gamma-1}}}$$

To find our nozzle length, we will again assume a standard conical nozzle with an opening angle of 15 degrees.

$$L_{nozzle} = \frac{D_e + D_t}{2 \cdot tan(15\backslash\text{degree})}$$

Again, we can calculate grain length and volume. I'm assuming that grain volume includes the binder volume.

$$V_{grain} = m_{ox}\rho_{AP} + m_{fuel}\rho_{Al}$$

$$L_{grain} = \frac{4 \cdot V_{grain}}{\pi(D_c^2 - D_i^2)}$$

To find our burn areas, we will again assume the outer diameter of the grain is equal to chamber diameter, and we will assume the inner diameter is equal to $\frac{1}{2}$ of the outer diameter.

Finally, I will calculate the initial and final burn areas

$$A_{b,0} = \frac{\pi(D_c^2 - D_i^2)}{4} + \pi D_i L_{grain}$$

$$A_{b,f} = \pi D_c(L_{grain} - \frac{D_i}{2})$$

D) Propellant Mass, Propellant Tank Size and Thickness

to calculate the propellant mass(including the binder mass), we will use the same method as with the first solid design.

$$m_{\text{prop}} = (m_{\text{payload}} + m_{\text{structure}})\left(e^{\frac{\Delta V}{I_{\text{sp}}\, g_0}} - 1\right)$$

$$\lambda = 0.9 = \frac{m_{prop}}{m_{sctucture} + m_{prop}}$$

$$m_{prop} = 9\, m_{structure}$$

We can calculate the fuel and oxidizer mass fractions since we know $m_{ox} = 2.3 \cdot m_{fuel}$

Since we know they both add to the total propellant mass, we can say

$$m_{fuel,total} = \frac{m_{prop}}{OFR + 1}$$

Since our fuel is 50 percent magnesiuim by weight, we simply multiply total fuel mass by the fraction of total fuel mass of each component.

$$m_m g = 0.85 \cdot m_{fuel,total}$$

$$m_b u = 0.15 \cdot m_{fuel,total}$$

and to find our mass of ammonium nitrate:

$$m_{ox} = \frac{m_{prop}}{\frac{1}{OFR} + 1}$$

We assume that the fuel and oxidizer both make up the volume of the grian, and we will also calculate the volume of the binder in this grain volume.

To find the total volume of the propellant grain(including the binder), we can do the following:

$$V_{grain} = m_{ox}\, \rho_{ap} + m_{al}\, \rho_{al} + m_{bu}\, \rho_{bu}$$

Now that we have the volume of the chamber, we want to find the chamber thickness

$$t = \frac{P \cdot D_{chamber}}{2\, \sigma_w}$$

Again, we can assume our material is titanium, and using the following site:
https://asm.matweb.com/search/SpecificMaterial.asp?bassnum=MTP641

We find our ultimate tensile strength of titanium to be 950 GPA. Now, we can divide by 1.25 to get our stress to be used for our tank thickness equation.

Our net chamber pressure will be:

$$P_{c,f} = (\frac{\rho_{prop}\, c^*\, a\, A_{b,f}}{A_thsroat})^{\frac{1-n}{n}} - P_a$$

Plugging in all variables results in:

$$t_{chamber} = \frac{P_{c,f} \cdot D_{chamber}}{2\,\sigma_w}$$

E) Plotting Altitude, Velocity, and Acceleration

We will use the same process for determining the altitude, velocity, and acceleration as we did in the liquid engine. We will use the following equations to determine these parameters as a function of time.

$$T_{avg} = 2 \cdot m_{booster,0}$$

$$\dot{m_{avg}} = \frac{T_{avg}}{c^*\, C_F}$$

$$t_{bo} = \frac{m_{prop}}{\dot{m_{avg}}}$$

$$g(h) = G_0 \left(\frac{R_{\text{earth}}}{R_{\text{earth}} + h}\right)^2$$

$$\frac{dv}{dt} = -\frac{I_{\text{sp}}\, g_0}{m}\, \frac{dm}{dt} - g(h)$$

$$\frac{dh}{dt} = v$$

$$\frac{dm}{dt} = -\dot{m} = -\frac{m_p}{t_b}$$

In [60]:

```
#Initializing variables (Solid Motor: Magnesium Fuel/Ammonium Nitrate Oxidizer with

import numpy as np
from matplotlib import pyplot as plt
from scipy.interpolate import UnivariateSpline
from scipy.integrate import solve_ivp
from scipy.optimize import fsolve

#Non-Unique Parameters
g0 = 9.806 #m/s^2
deltav = 2300 #m/s
lam = 0.9
Pc0 = 6894760 #Pascals
Pa = 101325 #Pa
Pe = Pa
m_payload = 100000 #kg
D_tank = 3 #m
MachT = 1
```

```
MachC = 0.1
radius_earth = 6378388
sigma_w = 750000000 #Pa


#Unique Parameters
rho_an = 1720 #kg/m^3
rho_mg = 1738 #kg/m^3
rho_butadiene = 614.9 #kg/m^3
Mg_Fuel_Fraction = 0.5
n = 0.4
a_0 = 0.04 * 0.0254
a = a_0 / (6894.75 ** n)
rdot_0 = a_0 * ((Pc0 * 0.0001450377 ) ** n) #m/s


#from CEA
cstar = 1816.2 #m/s
gamma = 1.243
Isp = 2809.3 / g0
OF_cea = 2.3
OFR = (OF_cea / (1 + OF_cea) ) / (Mg_Fuel_Fraction / (1 + OF_cea) )
C_f = 1.547
print(f"Optimal Oxidizer/Fuel Ratio: {OFR:.2f}")
print(f"Specific Impulse at Sea Level: {Isp:.2f} seconds")
```

```
Optimal Oxidizer/Fuel Ratio: 4.60
Specific Impulse at Sea Level: 286.49 seconds
```

In [64]:

```
#Calculating Fuel Parameters (Ammonium Perchlorate/Aluminum with Butadiene binder)


def find_prop_mass(masses):
    m_prop_guess, m_structure_guess = masses
    eq1 = m_prop_guess - (m_payload + m_structure_guess) * (np.exp(deltav / (Isp *
    eq2 = m_structure_guess - (m_prop_guess / 9)
    return (eq1, eq2)

# Provide an initial guess for [m_prop, m_structure]
initial_guess = [100000, 10000]  # You can adjust these as needed

# Solve the system
solution = fsolve(find_prop_mass, initial_guess)
m_prop, m_structure = solution
m_fuel_total = m_prop / (OF_cea + 1)
m_mg = Mg_Fuel_Fraction * m_fuel_total
m_butadiene = (1 - Mg_Fuel_Fraction) * m_fuel_total
m_ox = m_prop / (1/OF_cea + 1)
rho_prop = m_prop / V_req
P_cf = (rho_prop * cstar * a * A_bf / A_t) ** (1/(1-n)) - Pa
t_al = P_cf * D_c / (2 * sigma_w)
print(f"Total Propellant Mass(Including Binder): {m_prop:.1f} Kilograms")
print(f"Mass of Ammonium Nitrate: {m_ox:.1f} Kilograms")
print(f"Mass of Magnesium: {m_mg:.1f} Kilograms")
print(f"Mass of Butadiene Binder: {m_butadiene:.1f} Kilograms")
```

```python
print(f"Structural Mass: {m_structure:.1f} Kilograms")
print(f"Required Chamber Thickeness: {t_al * 1000:.5f} MilliMeters")
```

```
Total Propellant Mass(Including Binder): 147536.3 Kilograms
Mass of Ammonium Nitrate: 102828.3 Kilograms
Mass of Magnesium: 22354.0 Kilograms
Mass of Butadiene Binder: 22354.0 Kilograms
Structural Mass: 16392.9 Kilograms
Required Chamber Thickeness: 72.89013 MilliMeters
```

In [55]:

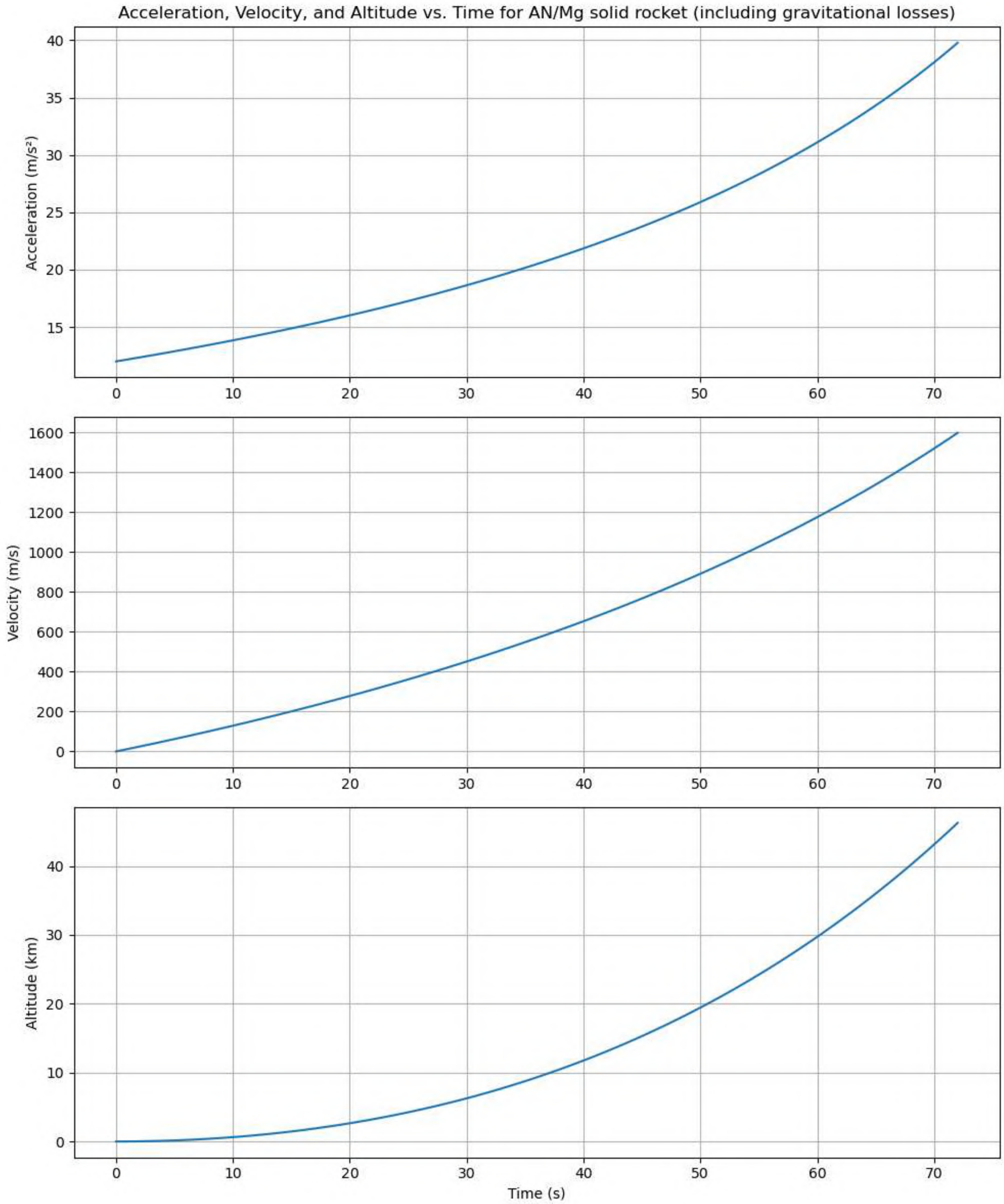
```python
#Calculating Engine and Nozzle Dimensions (Solid Motor: Magnesium Fuel/Ammonium Nit
mass_initial = m_payload + m_structure + m_prop
T = 2 * mass_initial * g0
A_t = T / (C_f * Pc0)
D_t = np.sqrt(4*A_t / np.pi)
A_e = A_t * ((2 / (gamma + 1)) ** (1 / (gamma - 1))) * ((Pc0 / Pe) ** (1 / gamma))
    * np.sqrt( (gamma - 1) / (gamma + 1) * ((1 - ((Pc0 / Pe) ** ((1 - gamma)/gamma)
D_e = np.sqrt(4*A_e / np.pi)

A_c = (MachT * A_t / MachC) * \
np.sqrt(((1 + (((gamma - 1) * MachC ** 2)/2))/(1+ (((gamma - 1) * MachT ** 2)/2)))
D_c = np.sqrt(4*A_c / np.pi)
l_nozzle = (D_e - D_t) / (2 * np.tan(15 * np.pi / 180) )
D_i = D_c / 2
V_req =  m_ox / rho_an + m_mg / rho_mg + m_butadiene / rho_butadiene
l_grain = V_req / (np.pi * ((D_c / 2) ** 2 - (D_i / 2) ** 2))
A_b0 = np.pi * (D_c ** 2  - D_i ** 2) / 4 + np.pi * D_i * l_grain
A_bf = np.pi * D_c * (l_grain - D_i / 2 )



print(f"Throat Diameter: {D_t:.3f} Meters")
print(f"Exit Diameter: {D_e:.3f} Meters")
print(f"Chamber Diameter: {D_c:.3f} Meters")
print(f"Initial Port Radius: {D_i / 2:.3f} Meters")
print(f"Final Port Radius: {D_c / 2:.3f} Meters")
print(f"Required nozzle length: {l_nozzle:.3f} Meters")
print(f"Required Grain Volume: {V_req:.2f} Cubic Meters")
print(f"Required Grain Length: {l_grain:.2f} Meters")
print(f"Initial burn area: {A_b0:.2f} Square Meters")
print(f"Final burn area: {A_bf:.2f} Square Meters")
```

```
Throat Diameter: 0.786 Meters
Exit Diameter: 2.247 Meters
Chamber Diameter: 1.913 Meters
Initial Port Radius: 0.478 Meters
Final Port Radius: 0.957 Meters
Required nozzle length: 2.726 Meters
Required Grain Volume: 109.00 Cubic Meters
Required Grain Length: 50.56 Meters
Initial burn area: 154.08 Square Meters
Final burn area: 300.98 Square Meters
```

In [47]:

```python
#Plotting Altitude, Velocity, and Acceleration (LOX/RP-1)
mach = np.linspace(0.2, 6, num=200, endpoint=True)
mdot = T / (cstar * C_f)
t_bo = m_prop / mdot
```

```python
time = np.linspace(0, t_bo, 100)

def vertical_launch(t, y, Isp, m_prop, t_bo):

    v = y[0]
    h = y[1]
    m = y[2]

    gravity = g0 * (radius_earth / (radius_earth + h))**2

    dmdt = -m_prop / t_bo
    dvdt = (-Isp * g0 / m) * dmdt - gravity
    dhdt = v

    return [dvdt, dhdt, dmdt]



sol = solve_ivp(
    vertical_launch, [0, t_bo], [0, 0, mass_initial], method='RK45',
    args=(Isp, m_prop, t_bo),
    dense_output=True
    )

z = sol.sol(time)
acceleration = (-Isp * g0 / z[2, :]) * (-m_prop / t_bo) - \
               g0 * (radius_earth / (radius_earth + z[1, :]))**2



fig, axes = plt.subplots(3, 1, figsize=(10, 12))

# Acceleration vs. Time
axes[0].plot(time, acceleration)
axes[0].set_ylabel('Acceleration (m/s²)')
axes[0].set_title('Acceleration, Velocity, and Altitude vs. Time for AN/Mg solid ro
axes[0].grid(True)

# Velocity vs. Time
axes[1].plot(time, z[0, :])
axes[1].set_ylabel('Velocity (m/s)')
axes[1].grid(True)

# Altitude vs. Time
axes[2].plot(time, z[1, :] / 1000)
axes[2].set_ylabel('Altitude (km)')
axes[2].set_xlabel('Time (s)')
axes[2].grid(True)

plt.tight_layout()
plt.show()


print(f" total burnout time: {t_bo:.2f} Seconds")
a_max = acceleration[99]
```




```
total burnout time: 72.01 Seconds
```

In [ ]:

Part 1: Hybrid Rocket.

HTPB Fuel/LOX Oxidizer

A/B) Average O/F Ratio and Specific Impulse Since we have an oxidizer/fuel shift, we can only use CEA to determine the average optimal oxidizer/fuel ratio. From there, we will have to calculate the oxidizer/fuel shift as time goes on. So, all values will simply be for the time just after combustion begins. From here, we found an optimal average oxidizer/fuel ratio of 2.1 by iterating from 2-4.5 in intervals of 0.1.

```
          THEORETICAL ROCKET PERFORMANCE ASSUMING FROZEN COMPOSITION
                                 AFTER POINT 2

 Pin =  1000.0 PSIA
 CASE = _______________

             REACTANT                    WT FRACTION      ENERGY      TEMP
                                          (SEE NOTE)     KJ/KG-MOL      K
 FUEL        C4H6,butadiene               1.0000000         0.000     0.000
 OXIDANT     O2(L)                        1.0000000    -12979.000    90.170

 O/F=    2.10000  %FUEL= 32.258065  R,EQ.RATIO= 1.549375  PHI,EQ.RATIO= 1.549375

                 CHAMBER   THROAT     EXIT
 Pinf/P            1.0000   1.7426   68.050
 P, BAR            68.947   39.566   1.0132
 T, K             3658.40  3441.71  1711.39
 RHO, KG/CU M    5.1660 0 3.1904 0 1.6430-1
 H, KJ/KG         -274.77  -989.43 -4277.90
 U, KJ/KG        -1609.41 -2229.61 -4894.58
 G, KJ/KG        -41509.5 -39781.8 -23567.4
 S, KJ/(KG)(K)    11.2712  11.2712  11.2712

 M, (1/n)          22.791   23.074   23.074
 Cp, KJ/(KG)(K)    4.8997   4.4575   1.7919
 GAMMAs            1.1526   1.1525   1.2517
 SON VEL,M/SEC     1240.3   1195.5    878.6
 MACH NUMBER        0.000    1.000    3.221

 PERFORMANCE PARAMETERS

 Ae/At                      1.0000   8.2047
 CSTAR, M/SEC               1807.6   1807.6
 CF                         0.6614   1.5653
 Ivac, M/SEC                2232.9   3047.5
 Isp, M/SEC                 1195.5   2829.5

 MASS FRACTIONS

 *CO             0.52349  *CO2             0.22726  COOH             0.00002
 *H              0.00123  HCO              0.00003  HO2              0.00002
 *H2             0.01027  H2O              0.20765  *O               0.00275
 *OH             0.02246  *O2              0.00481

  * THERMODYNAMIC PROPERTIES FITTED TO 20000.K

 NOTE. WEIGHT FRACTION OF FUEL IN TOTAL FUELS AND OF OXIDANT IN TOTAL OXIDANTS
```

$$Optimal\ Average\ O/F = 2.1$$

To find the optimal specific impulse, we simply need to divide the exit Isp value given by 9.806 to get into units of seconds. This results in:

$$Average\ I_{sp,sl} = 386.9\ seconds$$

Next, we must determine the oxidizer/fuel shift.

To do this, we must first determine the initial masses of the oxidizer and fuels. We will simply use the same process as we did for the liquid rocket engine to determine oxidizer and fuel mass.

Next, we will make the simplifying assumption that our oxidizer mass flow rate will be constant and we will be using a circular grain. Because of this assumption, we can use the following equation for our port radius derived in class:

$$R(t) = \left[a\,t\,(2n+1)\left(\frac{\dot{m}_{ox}}{\pi}\right)^n \ +\ R_i^{2n+1}\right]^{\frac{1}{2n+1}}$$

We can rearrange this to get the equation in terms of our inner port radius:

$$R_i = \left[R(t)^{2n+1} - a\,t\,(2n+1)\left(\frac{\dot{m}_{ox}}{\pi}\right)^n\right]^{\frac{1}{2n+1}}$$

From here, we can plug in the port radius at burnout, which we can assume to be equal to the final diameter of the chamber. However, we still have a few unknowns we need to solve for. Our current unkowns are the oxidizer mass flow rate, the total burn time, the inner port radius, and the final diameter of the chamber. We can estimate the burn time through the following process:

Since we can't get an exact value for our burn time, we must estimate it based on values we currently know. We know our total propellant mass, and we know our required average thrust. Thus, we can say:

$$T = 2 \cdot m_0 \cdot g_0$$

From here, we can say that if all our propellant is used by the end of the burn time, our average mass flow rate would be:

$$\dot{m}_{avg} = \frac{T}{c^*\,C_f}$$

and our burn time would be:

$$t_b = \frac{m_{prop}}{\dot{m}_{avg}}$$

We must calculate propellant mass using the following:

$$m_{\text{prop}} = \left(m_{\text{payload}} + m_{\text{structure}}\right)\left(e^{\frac{\Delta V}{I_{\text{sp}}\, g_0}} - 1\right)$$

$$m_{prop} = 9\, m_{structure}$$

$$m_{fuel} = \frac{m_{prop}}{OFR_0 + 1}$$

and

$$m_{ox} = \frac{m_{prop}}{\frac{1}{OFR_0} + 1}$$

Again, this is an estimate, so our true burn time may vary slightly.

Now, our only unknowns are the inner port radius, and the final chamber diameter. We can again estimate the chamber area(and thus chamber diameter) using the following equation, assuming $M_{chamber} = 0.1$ and $M_{throat} = 1$

$$A_t = \frac{T}{p_0\, C_F}$$

$$A_c \;=\; A_t \cdot \frac{M_t}{M_c} \sqrt{\left(\frac{1 \;+\; \frac{\gamma-1}{2}\, M_c^2}{1 \;+\; \frac{\gamma-1}{2}\, M_t^2}\right)^{\frac{\gamma+1}{\gamma-1}}}$$

Now, our only unknowns are our inner port radius and oxidizer mass flow rate.

Since our oxidizer mass flow rate is constant, we can say our our oxidizer mass flow rate is:

$$\dot{m}_{ox} = \frac{m_{ox}}{t_b}$$

Now, we have the oxidizer mass flow rate.

From here, we can solve for the inner port radius. If we set our final time equal to burnout and our final port radius equal to chamber radius, we get:

$$R_i = \left[R_f^{2n+1} - a\, t_b\, (2n+1)\left(\frac{\dot{m}_{ox}}{\pi}\right)^n\right]^{\frac{1}{2n+1}}$$

Next, we need to calculate the total volume and length of the grain.

$$V_{grain} = m_{lox}\, \rho_{lox} + m_{HTPB}\, \rho_{HTPB}$$

Next, we can caluclate grain length.

$$L_{grain} = \frac{V_{grain}}{\pi(R_f^2 - R_i^2)}$$

Since we have both initial and final port radius, we can finally calculate burn area and port area.

First, we will calculate the burn area. We can use the following equation:

$$A_b = 2\pi r(t)\, L_{grain}$$

Next, we calculate port area:

$$A_p = \pi r(t)^2$$

From here, we can caluclate the fuel mass flow rate as a function of time:

$$\dot{m}_f(t) = \rho_f a \dot{m}_{ox}^n \frac{A_b}{A_p^n}$$

Since we now have both the flow rates for oxidizer and fuels, we can now determine our oxidizer/fuel ratio over time.

$$\frac{O}{F}(t) = \frac{\dot{m}_o}{\dot{m}_f(t)}$$

In the process, we've also found most of the required variables for the rocket. Now, we just need to do calculate nozzle length, and the oxidizer tank dimensions.

$$L_{nozzle} = \frac{D_e + D_t}{2 \cdot tan(15\degree)}$$

$$V_{tank} = V_{LOX} = m_{ox}\rho_{LOX}$$

$$L_{tank} = \frac{4\pi V_{LOX}}{D_{tank}^2}$$

$$t = \frac{P \cdot D_{tank}}{2\,\sigma_w}$$

To find ultimate tensile strength, we can assume our material is titanium, and using the following site: https://asm.matweb.com/search/SpecificMaterial.asp?bassnum=MTP641

We find our ultimate tensile strength of titanium to be 950 GPA. Now, we can divide by 1.25 to get our stress to be used for our tank thickness equation.

Our tank pressure will be:

$$P_{LOX} = P_u + \rho_{LOX} \cdot (g_0 + a_{max}) \cdot h_{LOX} - P_a$$

where $P_u = \frac{Thrust}{A_{tank}}$

finally, plugging in all variables results in:

$$t_{tank} = \frac{P_{LOX} \cdot D_{tank}}{2\,\sigma_w}$$

E) Plotting Altitude, Velocity, and Acceleration

Again, we can use all the same equations to determine our altitude, velocity and acceleration.

$$\frac{dv}{dt} = -\frac{I_{\text{sp}}\, g_0}{m}\,\frac{dm}{dt} \;-\; g(h)$$

$$\frac{dh}{dt} = v$$

$$\frac{dm}{dt} = -\dot{m} = -\frac{m_p}{t_b}$$

$$g(h) \;=\; G_0 \left(\frac{R_{\text{earth}}}{R_{\text{earth}} + h}\right)^2$$

In [302…

```
#Initializing variables (Hybrid: HTPB/LOX)
import numpy as np
from matplotlib import pyplot as plt
from scipy.interpolate import UnivariateSpline
from scipy.integrate import solve_ivp
from scipy.optimize import fsolve
#Non-Unique Parameters
g0 = 9.806 #m/s^2
deltav = 2300 #m/s
lam = 0.9
Pc0 = 6894760 #Pascals
Pa = 101325 #Pa
Pe = Pa
m_payload = 100000 #kg
D_tank = 3 #m
MachT = 1
MachC = 0.1
radius_earth = 6378388
sigma_w = 750000000 #Pa

#Unique Parameters
rho_lox = 1141 #kg/m^3
rho_htpb = 614.9 #kg/m^3
n = 0.68
a_0 = 0.1 * 0.0254
a = a_0 / ((0.45359 / 0.0006451) ** n)


#from CEA
cstar = 1807.6 #m/s
gamma = 1.2517
OFR_0 = 2.1
Isp = 2829.5/g0 #s
```

```
C_f = 1.5653
print(f"Optimal Average Oxidizer/Fuel Ratio: {OFR_0}")
print(f"Average Specific Impulse at Sea Level: {Isp:.2f} seconds")
```

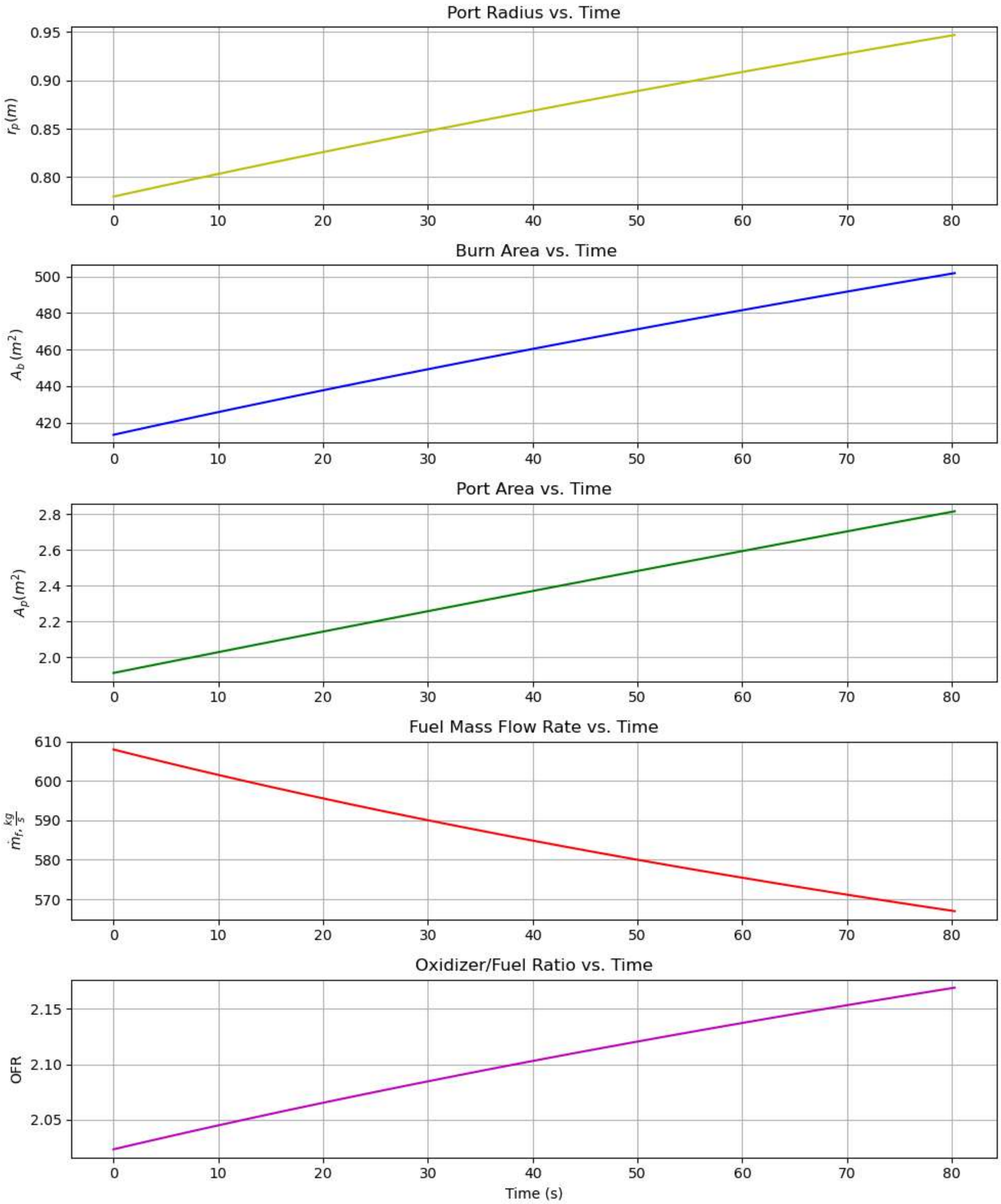
```
Optimal Average Oxidizer/Fuel Ratio: 2.1
Average Specific Impulse at Sea Level: 288.55 seconds
```

In [311…

```
#Determining Oxidizer/Fuel Shift (Hybrid: HTPB/LOX)


def find_prop_mass(masses):
    m_prop_guess, m_structure_guess = masses
    eq1 = m_prop_guess - (m_payload + m_structure_guess) * (np.exp(deltav / (Isp *
    eq2 = m_structure_guess - (m_prop_guess / 9)
    return (eq1, eq2)

# Provide an initial guess for [m_prop, m_structure]
initial_guess = [100000, 10000]  # You can adjust these as needed

# Solve the system
solution = fsolve(find_prop_mass, initial_guess)
m_prop, m_structure = solution

m_fuel = m_prop / (OFR_0 + 1)
m_ox = m_prop / (1/OFR_0 + 1)


mass_initial = m_payload + m_structure + m_prop
Thrust = 2 * mass_initial * g0
A_t = Thrust / (C_f * Pc0)
D_t = np.sqrt(4*A_t / np.pi)
A_e = A_t * ((2 / (gamma + 1)) ** (1 / (gamma - 1))) * ((Pc0 / Pe) ** (1 / gamma))
    * np.sqrt( (gamma - 1) / (gamma + 1) * ((1 - ((Pc0 / Pe) ** ((1 - gamma)/gamma)
D_e = np.sqrt(4*A_e / np.pi)

A_c = (MachT * A_t / MachC) * \
np.sqrt(((1 + (((gamma - 1) * MachC ** 2)/2))/(1+ (((gamma - 1) * MachT ** 2)/2)))
D_c = np.sqrt(4*A_c / np.pi)




mdot = Thrust / (cstar * C_f)
t_bo = m_prop / mdot
mdot_ox = m_ox / t_bo

V_grain = m_fuel / rho_htpb

r_f = D_c/2
r_i = (r_f ** (2*n + 1) - a * t_bo * (2*n + 1) * ((mdot_ox / np.pi ) ** n)) ** (1 /

L_grain = V_grain / (np.pi * (r_f ** 2 - r_i ** 2 ))

time = np.linspace(0, t_bo, 100)

def port_radius(t, a, n, mdot_ox, r_i, rho_htpb):
```

```
    r = (r_i ** (2*n + 1) + a * t * (2*n + 1) * ((mdot_ox / np.pi ) ** n)) ** (1/ (
    A_b = 2 * np.pi * r * L_grain
    A_p = np.pi * r ** 2
    mdot_f = rho_htpb * a * (mdot_ox ** n) * A_b  / (A_p ** n )
    OFR = mdot_ox / mdot_f

    return [r, A_b, A_p, mdot_f, OFR]

r_vals = []
A_b_vals = []
A_p_vals = []
mdot_f_vals = []
OFR_vals = []

# Evaluate the function at each time value
for t_val in time:
    # Ensure you're passing the correct variable names; here mdot_ox is used
    r, A_b, A_p, mdot_f, OFR = port_radius(t_val, a, n, mdot_ox, r_i, rho_htpb)
    r_vals.append(r)
    A_b_vals.append(A_b)
    A_p_vals.append(A_p)
    mdot_f_vals.append(mdot_f)
    OFR_vals.append(OFR)

# Plot each output as a function of time using subplots
fig, axes = plt.subplots(5, 1, figsize=(10, 12))

# Port burn area, A_b vs time
axes[0].plot(time, r_vals, 'y-')
axes[0].set_ylabel(r'$r_p (m)$')
axes[0].set_title('Port Radius vs. Time')
axes[0].grid(True)

axes[1].plot(time, A_b_vals, 'b-')
axes[1].set_ylabel(r'$A_b \, (m^2)$')
axes[1].set_title(' Burn Area vs. Time')
axes[1].grid(True)

# Port area, A_p vs time
axes[2].plot(time, A_p_vals, 'g-')
axes[2].set_ylabel(r'$A_p (m^2)$')
axes[2].set_title('Port Area vs. Time')
axes[2].grid(True)

# Fuel mass flow rate, mdot_f vs time
axes[3].plot(time, mdot_f_vals, 'r-')
axes[3].set_ylabel(r'$\dot{m}_f, \frac{kg}{s}$')
axes[3].set_title('Fuel Mass Flow Rate vs. Time')
axes[3].grid(True)

# Oxidizer/Fuel Ratio, OFR vs time
axes[4].plot(time, OFR_vals, 'm-')
axes[4].set_ylabel('OFR')
axes[4].set_xlabel('Time (s)')
axes[4].set_title('Oxidizer/Fuel Ratio vs. Time')
```

```
axes[4].grid(True)

plt.tight_layout()
plt.show()

OF_shift = OFR_vals[99] - OFR_vals[0]
print(f"Oxidizer/Fuel Shift: {OF_shift:.3f}")
print(f"Throat Diameter: {D_t:.3f} Meters")
print(f"Exit Diameter: {D_e:.3f} Meters")
print(f"Outer Port Radius: {r_f:.3f} Meters")
print(f"Inner Port Radius: {r_i:.3f} Meters")
print(f"Grain length: {L_grain:.3f} Meters")
print(f"Grain mass: {m_fuel:.3f} Kilograms")
print(f"Oxidizer mass: {m_ox:.3f} Kilograms")
print(f"Total Propellant mass: {m_prop:.3f} Kilograms")
print(f"Structural mass: {m_structure:.3f} Kilograms")
```




```
Oxidizer/Fuel Shift: 0.146
Throat Diameter: 0.779 Meters
Exit Diameter: 2.207 Meters
Outer Port Radius: 0.947 Meters
Inner Port Radius: 0.780 Meters
Grain length: 84.361 Meters
Grain mass: 47015.840 Kilograms
Oxidizer mass: 98733.265 Kilograms
Total Propellant mass: 145749.105 Kilograms
Structural mass: 16194.345 Kilograms
```

In [314…

```
## Remaining Parameters
l_nozzle = (D_e - D_t) / (2 * np.tan(15 * np.pi / 180) )
```

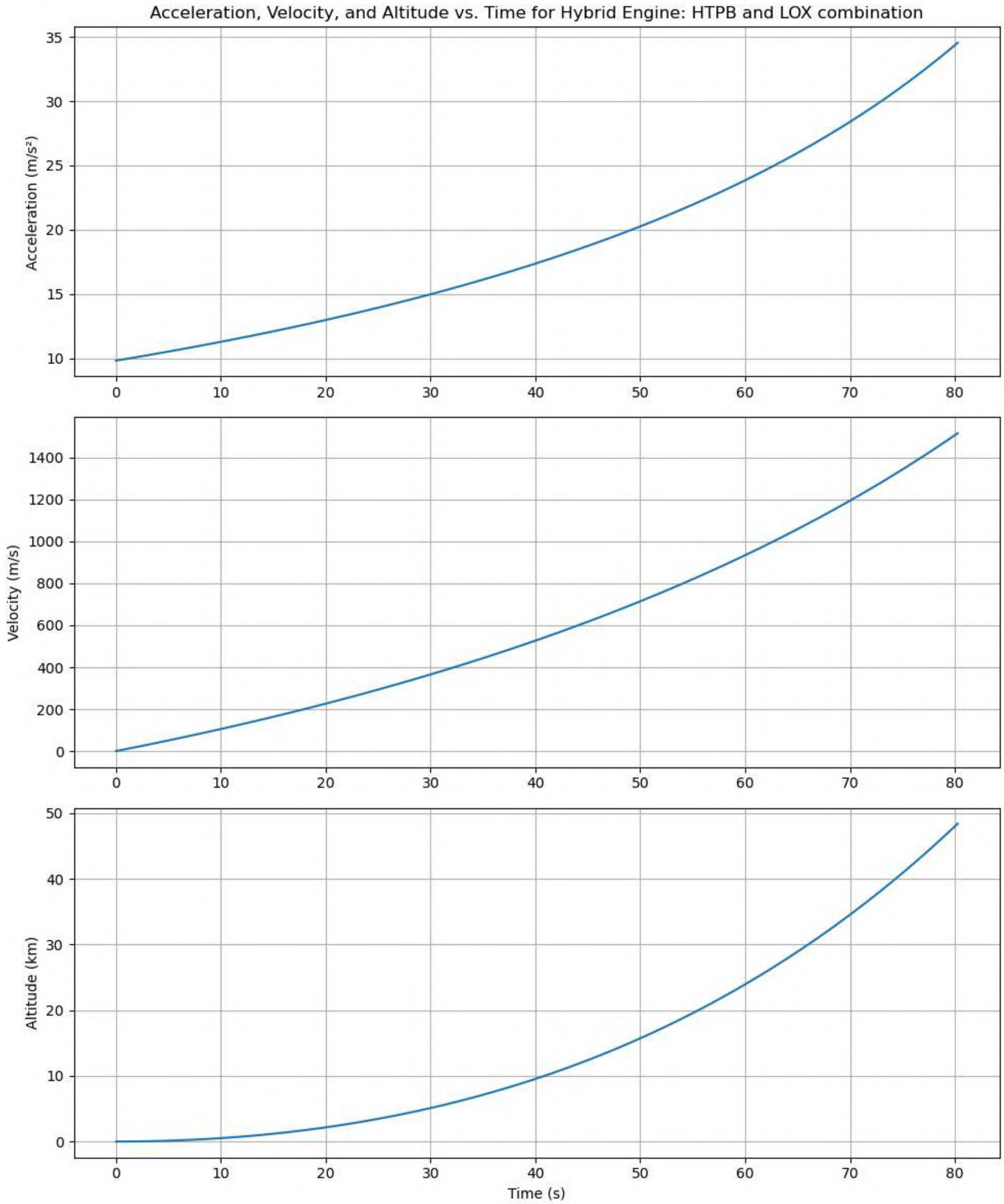
```python
V_tank = m_ox / rho_lox
l_tank = 4 * V_tank / (np.pi * D_tank ** 2)
P_u = Thrust * V_tank / l_tank
P_tot = P_u + rho_lox * (g0 + a_max) * l_tank - Pa
t_req = P_tot * D_tank / (2 * sigma_w)
print(f"Nozzle Length: {l_nozzle:.2f} Meters")
print(f"Tank Volume: {V_tank:.2f} Square Meters")
print(f"Tank Length: {l_tank:.2f} Meters")
print(f"Required Tank Thickeness: {t_req * 1000:.5f} MilliMeters")
```

```
Nozzle Length: 2.67 Meters
Tank Volume: 86.53 Square Meters
Tank Length: 12.24 Meters
Required Tank Thickeness: 73.77425 MilliMeters
```

In [316…

```python
#Plotting Altitude, Velocity, and Acceleration (Hybrid Engine: HTBP/LOX)

time = np.linspace(0, t_bo, 100)

def vertical_launch(t, y, Isp, m_prop, t_bo):

    v = y[0]
    h = y[1]
    m = y[2]

    gravity = g0 * (radius_earth / (radius_earth + h))**2

    dmdt = -m_prop / t_bo
    dvdt = (-Isp * g0 / m) * dmdt - gravity
    dhdt = v

    return [dvdt, dhdt, dmdt]



sol = solve_ivp(
    vertical_launch, [0, t_bo], [0, 0, mass_initial], method='RK45',
    args=(Isp, m_prop, t_bo),
    dense_output=True
    )

z = sol.sol(time)

acceleration = (-Isp * g0 / z[2, :]) * (-m_prop / t_bo) - \
               g0 * (radius_earth / (radius_earth + z[1, :]))**2



fig, axes = plt.subplots(3, 1, figsize=(10, 12))

# Acceleration vs. Time
axes[0].plot(time, acceleration)
axes[0].set_ylabel('Acceleration (m/s²)')
axes[0].set_title('Acceleration, Velocity, and Altitude vs. Time for Hybrid Engine:
axes[0].grid(True)
```

```python

# Velocity vs. Time
axes[1].plot(time, z[0, :])
axes[1].set_ylabel('Velocity (m/s)')
axes[1].grid(True)

# Altitude vs. Time
axes[2].plot(time, z[1, :] / 1000)
axes[2].set_ylabel('Altitude (km)')
axes[2].set_xlabel('Time (s)')
axes[2].grid(True)

plt.tight_layout()
plt.show()

print(f" total burnout time: {t_bo:.2f} Seconds")
a_max = acceleration[99]
```




```
total burnout time: 80.27 Seconds
```

```
In [ ]:
```

Part 1: Hybrid Rocket.

Hybrid: N2O/Paraffin

https://pubchem.ncbi.nlm.nih.gov/compound/nitrous-oxide (N2O density)

https://physics.nist.gov/cgi-bin/Star/compos.pl?ap213 (Paraffin density)

A/B) Average O/F Ratio and Specific Impulse Again, since we have an oxidizer/fuel shift, we can only use CEA to determine the average optimal oxidizer/fuel ratio. From there, we will have to calculate the oxidizer/fuel shift as time goes on. So, all values will simply be for the time just after combustion begins. From here, I iterated the oxidizer/fuel ratio from 2-9 in intervals of 0.5. Then, I found that the optimal oxidizer/fuel was somewhere between 8 and 9 so I iterated from 8 to 9 in intervals of 0.1 and found an 8.4 average optimal oxidizer/fuel ratio.

```
           THEORETICAL ROCKET PERFORMANCE ASSUMING FROZEN COMPOSITION
                                AFTER POINT 2

 Pin =  1000.0 PSIA
 CASE = _______________

             REACTANT                    WT FRACTION      ENERGY      TEMP
                                          (SEE NOTE)     KJ/KG-MOL      K
 FUEL        paraffin                     1.0000000  -1860600.000   298.150
 OXIDANT     N2O                          1.0000000         0.000     0.000

 O/F=    8.40000  %FUEL= 10.638298  R,EQ.RATIO= 1.087920  PHI,EQ.RATIO= 1.087920

                 CHAMBER   THROAT     EXIT
 Pinf/P            1.0000   1.7562   68.050
 P, BAR            68.947   39.260   1.0132
 T, K             2908.85  2701.56  1319.32
 RHO, KG/CU M    8.1160 0 5.0094 0 2.6472-1
 H, KJ/KG         -197.59  -657.39 -2715.72
 U, KJ/KG        -1047.11 -1441.12 -3098.45
 G, KJ/KG        -26208.1 -24814.4 -14512.9
 S, KJ/(KG)(K)     8.9419   8.9419   8.9419

 M, (1/n)          28.470   28.661   28.661
 Cp, KJ/(KG)(K)    2.6872   2.3232   1.3979
 GAMMAs            1.1623   1.1734   1.2619
 SON VEL,M/SEC      993.7    959.0    695.0
 MACH NUMBER        0.000    1.000    3.229

 PERFORMANCE PARAMETERS

 Ae/At                      1.0000   8.0861
 CSTAR, M/SEC               1435.3   1435.3
 CF                         0.6681   1.5636
 Ivac, M/SEC                1776.2   2414.7
 Isp, M/SEC                  959.0   2244.2

 MASS FRACTIONS

 *CO             0.05200   *CO2            0.25946   *H              0.00003
 *H2             0.00058   H2O             0.11156   *NO             0.00313
 *N2             0.56731   *O              0.00016   *OH             0.00296
 *O2             0.00279

  * THERMODYNAMIC PROPERTIES FITTED TO 20000.K

 NOTE. WEIGHT FRACTION OF FUEL IN TOTAL FUELS AND OF OXIDANT IN TOTAL OXIDANTS
```

$$Optimal\ Average\ O/F = 8.4$$

$$Average\ I_{sp,sl} = 386.9\ seconds$$

We will use the same process to determine the oxidizer/fuel shift.

To do this, we must first determine the initial masses of the oxidizer and fuels. We will simply use the same process as we did for the liquid rocket engine to determine oxidizer and fuel mass.

Next, we will make the simplifying assumption that our oxidizer mass flow rate will be constant and we will be using a circular grain. Because of this assumption, we can use the following equation for our port radius derived in class:

$$R(t) = \left[a\,t\,(2n+1)\left(\frac{\dot{m}_{ox}}{\pi}\right)^{n} \;+\; R_i^{2n+1}\right]^{\frac{1}{2n+1}}$$

$$R_i = \left[R(t)^{2n+1} - a\,t\,(2n+1)\left(\frac{\dot{m}_{ox}}{\pi}\right)^{n}\right]^{\frac{1}{2n+1}}$$

We must calculate propellant mass using the following:

$$m_{\text{prop}} = \left(m_{\text{payload}} + m_{\text{structure}}\right)\left(e^{\frac{\Delta V}{I_{\text{sp}}\,g_0}} - 1\right)$$

$$m_{prop} = 9\,m_{structure}$$

$$m_{fuel} = \frac{m_{prop}}{OFR_0 + 1}$$

and

$$m_{ox} = \frac{m_{prop}}{\frac{1}{OFR_0} + 1}$$

We again assume the port radius at burnout to be equal to the final diameter of the chamber. We also assume:

$$T = 2 \cdot m_0 \cdot g_0$$

From here, we can say that if all our propellant is used by the end of the burn time, our average mass flow rate would be:

$$\dot{m}_{avg} = \frac{T}{c^*\,C_f}$$

and our burn time would be:

$$t_b = \frac{m_{prop}}{\dot{m}_{avg}}$$

Again, this is an estimate, so our true burn time may vary slightly.

We can again estimate the chamber area(and thus chamber diameter) using the following equation, assuming $M_{chamber} = 0.1$ and $M_{throat} = 1$

$$A_t = \frac{T}{p_0\,C_F}$$

$$A_c = A_t \cdot \frac{M_t}{M_c} \sqrt{\left(\frac{1 + \frac{\gamma-1}{2} M_c^2}{1 + \frac{\gamma-1}{2} M_t^2}\right)^{\frac{\gamma+1}{\gamma-1}}}$$

Since our oxidizer mass flow rate is constant, we can say our our oxidizer mass flow rate is:

$$\dot{m}_{ox} = \frac{m_{ox}}{t_b}$$

Again, we set our final time equal to burnout and our final port radius equal to chamber radius:

$$R_i = \left[R_f^{2n+1} - a\, t_b\, (2n+1) \left(\frac{\dot{m}_{ox}}{\pi}\right)^n\right]^{\frac{1}{2n+1}}$$

Next, we need to calculate the total volume and length of the grain.

$$V_{grain} = m_{lox}\, \rho_{lox} + m_{HTPB}\, \rho_{HTPB}$$

$$L_{grain} = \frac{V_{grain}}{\pi(R_f^2 - R_i^2)}$$

And to calculate burn area and port area:

$$A_b = 2\pi r(t)\, L_{grain}$$

Next, we calculate port area:

$$A_p = \pi r(t)^2$$

Fuel mass flow rate and OFR as a function of time:

$$\dot{m}_f(t) = \rho_f a \dot{m}_{ox}^n \frac{A_b}{A_p^n}$$

$$\frac{O}{F}(t) = \frac{\dot{m}_o}{\dot{m}_f(t)}$$

Again, we calculate nozzle length and the oxidizer tank dimensions.

$$L_{nozzle} = \frac{D_e + D_t}{2 \cdot tan(15\textcolor{red}{\text{\textbackslash degree}})}$$

$$V_{tank} = V_{N2O} = m_{ox} \rho_{N2O}$$

$$L_{tank} = \frac{4\pi V_{N2O}}{D_{tank}^2}$$

$$t = \frac{P \cdot D_{tank}}{2\, \sigma_w}$$

Again, we assume our material is titanium.

Our tank pressure will be:

$$P_{N2O} = P_u + \rho_{N2O} \cdot (g_0 + a_{max}) \cdot h_{N2O} - P_a$$

where $P_u = \frac{Thrust}{A_{tank}}$

finally, plugging in all variables results in:

$$t_{tank} = \frac{P_{LOX} \cdot D_{tank}}{2\,\sigma_w}$$

E) Plotting Altitude, Velocity, and Acceleration

Again, we can use all the same equations to determine our altitude, velocity and acceleration.

$$\frac{dv}{dt} = -\frac{I_{\text{sp}}\, g_0}{m}\frac{dm}{dt} - g(h)$$

$$\frac{dh}{dt} = v$$

$$\frac{dm}{dt} = -\dot{m} = -\frac{m_p}{t_b}$$

$$g(h) = G_0\left(\frac{R_{\text{earth}}}{R_{\text{earth}} + h}\right)^2$$

In [55]:

```
#Initializing variables (Hybrid: N2O/Paraffin)
import numpy as np
from matplotlib import pyplot as plt
from scipy.interpolate import UnivariateSpline
from scipy.integrate import solve_ivp
from scipy.optimize import fsolve
#Non-Unique Parameters
g0 = 9.806 #m/s^2
deltav = 2300 #m/s
lam = 0.9
Pc0 = 6894760 #Pascals
Pa = 101325 #Pa
Pe = Pa
m_payload = 100000 #kg
D_tank = 3 #m
MachT = 1
MachC = 0.1
radius_earth = 6378388
sigma_w = 750000000 #Pa

#Unique Parameters
rho_n2o = 1220 #kg/m^3
```

```
rho_paraffin = 930 #kg/m^3
n = 0.68
a_0 = 0.1 * 0.0254
a = a_0 / ((0.45359 / 0.0006451) ** n)


#from CEA
cstar = 1435.3 #m/s
gamma = 1.2619
OFR_0 = 8.4
Isp = 2244.2/g0 #s
C_f = 1.5636
print(f"Optimal Average Oxidizer/Fuel Ratio: {OFR_0}")
print(f"Average Specific Impulse at Sea Level: {Isp_0:.2f} seconds")
```

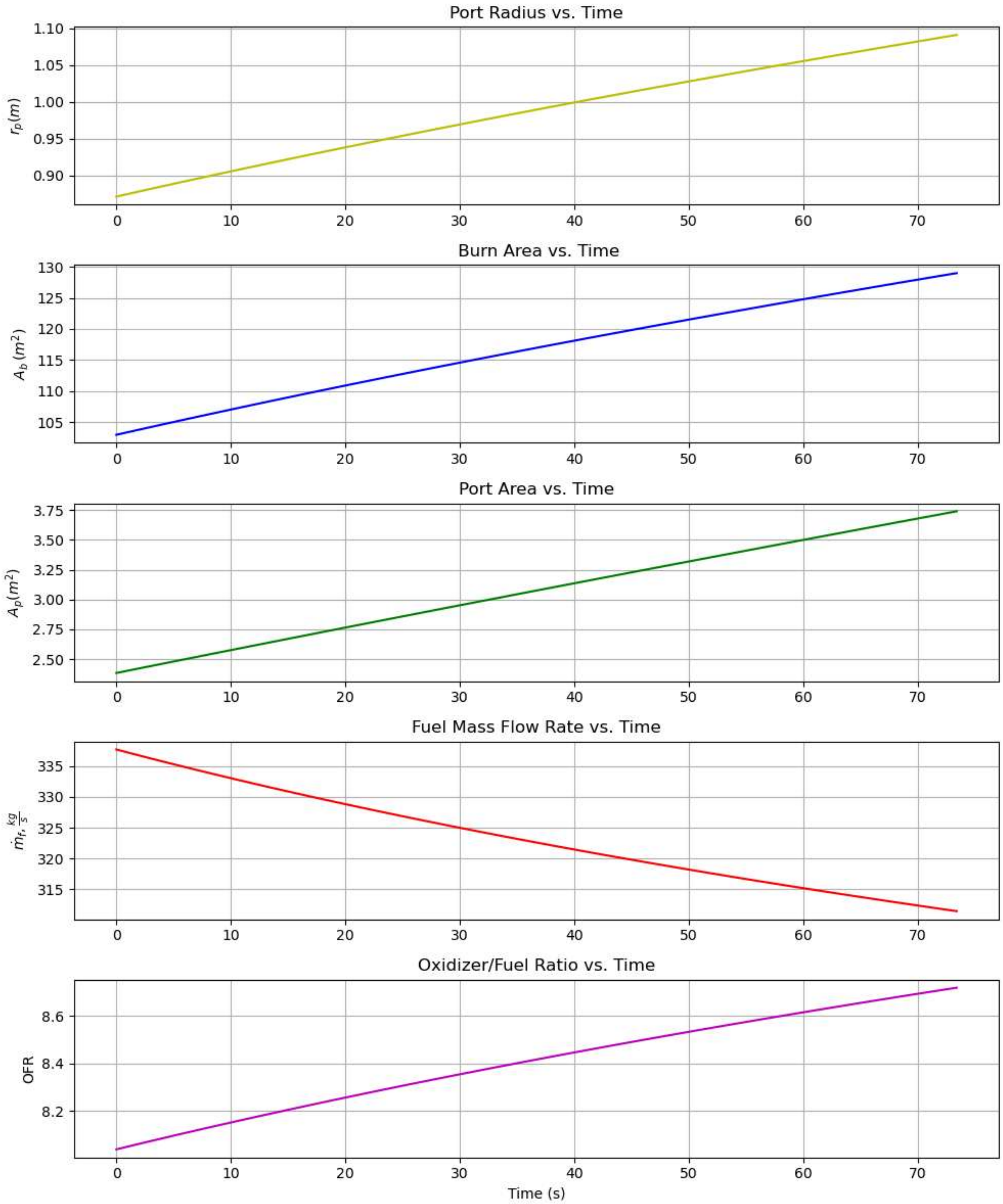
```
Optimal Average Oxidizer/Fuel Ratio: 8.4
Average Specific Impulse at Sea Level: 228.86 seconds
```

In [49]:

```
#Determining Oxidizer/Fuel Shift (Hybrid: N2O/Paraffin)


def find_prop_mass(masses):
    m_prop_guess, m_structure_guess = masses
    eq1 = m_prop_guess - (m_payload + m_structure_guess) * (np.exp(deltav / (Isp *
    eq2 = m_structure_guess - (m_prop_guess / 9)
    return (eq1, eq2)

# Provide an initial guess for [m_prop, m_structure]
initial_guess = [100000, 10000]  # You can adjust these as needed

# Solve the system
solution = fsolve(find_prop_mass, initial_guess)
m_prop, m_structure = solution

m_fuel = m_prop / (OFR_0 + 1)
m_ox = m_prop / (1/OFR_0 + 1)


mass_initial = m_payload + m_structure + m_prop
Thrust = 2 * mass_initial * g0
A_t = Thrust / (C_f * Pc0)
D_t = np.sqrt(4*A_t / np.pi)
A_e = A_t * ((2 / (gamma + 1)) ** (1 / (gamma - 1))) * ((Pc0 / Pe) ** (1 / gamma))
    * np.sqrt( (gamma - 1) / (gamma + 1) * ((1 - ((Pc0 / Pe) ** ((1 - gamma)/gamma)
D_e = np.sqrt(4*A_e / np.pi)

A_c = (MachT * A_t / MachC) * \
np.sqrt(((1 + (((gamma - 1) * MachC ** 2)/2))/(1+ (((gamma - 1) * MachT ** 2)/2)))
D_c = np.sqrt(4*A_c / np.pi)





mdot = Thrust / (cstar * C_f)
t_bo = m_prop / mdot
mdot_ox = m_ox / t_bo
```

```
V_grain = m_fuel / rho_paraffin

r_f = D_c/2
r_i = (r_f ** (2*n + 1) - a * t_bo * (2*n + 1) * ((mdot_ox / np.pi ) ** n)) ** (1 /

L_grain = V_grain / (np.pi * (r_f ** 2 - r_i ** 2 ))

time = np.linspace(0, t_bo, 100)

def port_radius(t, a, n, mdot_ox, r_i, rho_htpb):

    r = (r_i ** (2*n + 1) + a * t * (2*n + 1) * ((mdot_ox / np.pi ) ** n)) ** (1/ (
    A_b = 2 * np.pi * r * L_grain
    A_p = np.pi * r ** 2
    mdot_f = rho_paraffin * a * (mdot_ox ** n) * A_b  / (A_p ** n )
    OFR = mdot_ox / mdot_f

    return [r, A_b, A_p, mdot_f, OFR]

r_vals = []
A_b_vals = []
A_p_vals = []
mdot_f_vals = []
OFR_vals = []

# Evaluate the function at each time value
for t_val in time:
    # Ensure you're passing the correct variable names; here mdot_ox is used
    r, A_b, A_p, mdot_f, OFR = port_radius(t_val, a, n, mdot_ox, r_i, rho_htpb)
    r_vals.append(r)
    A_b_vals.append(A_b)
    A_p_vals.append(A_p)
    mdot_f_vals.append(mdot_f)
    OFR_vals.append(OFR)

# Plot each output as a function of time using subplots
fig, axes = plt.subplots(5, 1, figsize=(10, 12))

# Port burn area, A_b vs time
axes[0].plot(time, r_vals, 'y-')
axes[0].set_ylabel(r'$r_p (m)$')
axes[0].set_title('Port Radius vs. Time')
axes[0].grid(True)

axes[1].plot(time, A_b_vals, 'b-')
axes[1].set_ylabel(r'$A_b \, (m^2)$')
axes[1].set_title(' Burn Area vs. Time')
axes[1].grid(True)

# Port area, A_p vs time
axes[2].plot(time, A_p_vals, 'g-')
axes[2].set_ylabel(r'$A_p (m^2)$')
axes[2].set_title('Port Area vs. Time')
axes[2].grid(True)
```

```python
# Fuel mass flow rate, mdot_f vs time
axes[3].plot(time, mdot_f_vals, 'r-')
axes[3].set_ylabel(r'$\dot{m}_f, \frac{kg}{s}$')
axes[3].set_title('Fuel Mass Flow Rate vs. Time')
axes[3].grid(True)

# Oxidizer/Fuel Ratio, OFR vs time
axes[4].plot(time, OFR_vals, 'm-')
axes[4].set_ylabel('OFR')
axes[4].set_xlabel('Time (s)')
axes[4].set_title('Oxidizer/Fuel Ratio vs. Time')
axes[4].grid(True)

plt.tight_layout()
plt.show()

OF_shift = OFR_vals[99] - OFR_vals[0]
print(f"Oxidizer/Fuel Shift: {OF_shift:.3f}")
print(f"Throat Diameter: {D_t:.3f} Meters")
print(f"Exit Diameter: {D_e:.3f} Meters")
print(f"Outer Port Radius: {r_f:.3f} Meters")
print(f"Inner Port Radius: {r_i:.3f} Meters")
print(f"Grain length: {L_grain:.3f} Meters")
print(f"Grain mass: {m_fuel:.3f} Kilograms")
print(f"Oxidizer mass: {m_ox:.3f} Kilograms")
print(f"Total Propellant mass: {m_prop:.3f} Kilograms")
print(f"Structural mass: {m_structure:.3f} Kilograms")
```




```
Oxidizer/Fuel Shift: 0.679
Throat Diameter: 0.897 Meters
Exit Diameter: 2.521 Meters
Outer Port Radius: 1.091 Meters
Inner Port Radius: 0.871 Meters
Grain length: 18.816 Meters
Grain mass: 23715.777 Kilograms
Oxidizer mass: 199212.530 Kilograms
Total Propellant mass: 222928.307 Kilograms
Structural mass: 24769.812 Kilograms
```

In [50]:

```
#Remaining Parameters (Hybrid: N2O/Paraffin)
l_nozzle = (D_e - D_t) / (2 * np.tan(15 * np.pi / 180) )
```

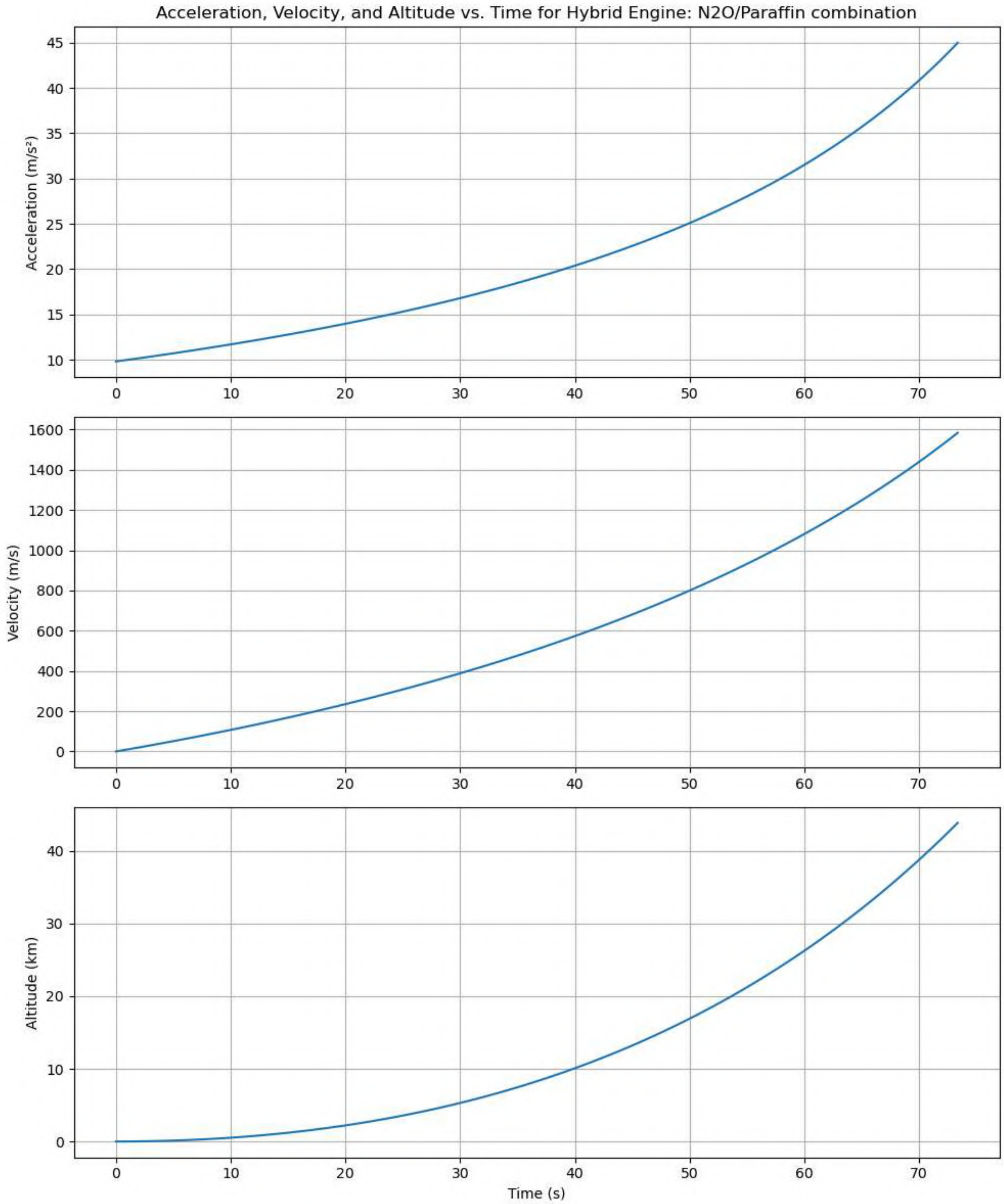
```
V_tank = m_ox / rho_n2o
l_tank = 4 * V_tank / (np.pi * D_tank ** 2)
P_u = Thrust * V_tank / l_tank
P_tot = P_u + rho_n2o * (g0 + a_max) * l_tank - Pa
t_req = P_tot * D_tank / (2 * sigma_w)
print(f"Nozzle Length: {l_nozzle:.2f} Meters")
print(f"Tank Volume: {V_tank:.2f} Square Meters")
print(f"Tank Length: {l_tank:.2f} Meters")
print(f"Required Tank Thickeness: {t_req * 1000:.5f} MilliMeters")
```

```
Nozzle Length: 3.03 Meters
Tank Volume: 163.29 Square Meters
Tank Length: 23.10 Meters
Required Tank Thickeness: 99.28751 MilliMeters
```

In [53]:

```
#Plotting Altitude, Velocity, and Acceleration (Hybrid: N2O/Paraffin)

time = np.linspace(0, t_bo, 100)

def vertical_launch(t, y, Isp, m_prop, t_bo):

    v = y[0]
    h = y[1]
    m = y[2]

    gravity = g0 * (radius_earth / (radius_earth + h))**2

    dmdt = -m_prop / t_bo
    dvdt = (-Isp * g0 / m) * dmdt - gravity
    dhdt = v

    return [dvdt, dhdt, dmdt]



sol = solve_ivp(
    vertical_launch, [0, t_bo], [0, 0, mass_initial], method='RK45',
    args=(Isp, m_prop, t_bo),
    dense_output=True
    )

z = sol.sol(time)

acceleration = (-Isp * g0 / z[2, :]) * (-m_prop / t_bo) - \
               g0 * (radius_earth / (radius_earth + z[1, :]))**2



fig, axes = plt.subplots(3, 1, figsize=(10, 12))

# Acceleration vs. Time
axes[0].plot(time, acceleration)
axes[0].set_ylabel('Acceleration (m/s²)')
axes[0].set_title('Acceleration, Velocity, and Altitude vs. Time for Hybrid Engine:
axes[0].grid(True)
```

```python
# Velocity vs. Time
axes[1].plot(time, z[0, :])
axes[1].set_ylabel('Velocity (m/s)')
axes[1].grid(True)

# Altitude vs. Time
axes[2].plot(time, z[1, :] / 1000)
axes[2].set_ylabel('Altitude (km)')
axes[2].set_xlabel('Time (s)')
axes[2].grid(True)

plt.tight_layout()
plt.show()

print(f" total burnout time: {t_bo:.2f} Seconds")
a_max = acceleration[99]
```




```
total burnout time: 73.37 Seconds
```

In [ ]: