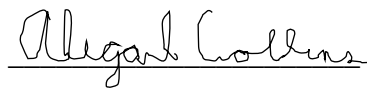


**Design and Analysis of a High-Powered Model Rocket**

A Major Qualifying Project Report  
Submitted to the Faculty of the  
WORCESTER POLYTECHNIC INSTITUTE  
In Partial Fulfillment of the Requirements for the  
Degree of Bachelor of Science  
in Aerospace Engineering  
by



Kirsten Bowers



Abigail Collins



Alex Harrigan



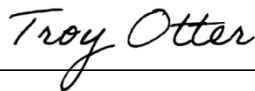
Tyler Hunt




Theresa Larson



Daniel Mattison



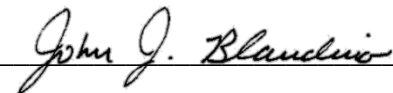
Troy Otter



Connor Walsh

March 24, 2022

Approved by:



John J. Blandino, Advisor

Professor, Aerospace Engineering Department

WPI

*This report represents the work of one or more WPI undergraduate students submitted to the faculty as evidence of completion of a degree requirement. WPI routinely publishes these reports on the web without editorial or peer review.*

## Abstract

This paper describes the design and analysis of a high-powered model rocket designed to use actively controlled canard fins for stabilization, a retromotor to slow its descent, and a compressed CO<sub>2</sub> pressurization system to separate the upper and lower stages, releasing the main parachute. The airframe, custom nose cone, canards, and main fins were modeled using SOLIDWORKS. Aerodynamic loads on the vehicle airframe, canards, and main stabilizing fins were evaluated using computational fluid dynamics (CFD) tools in Ansys Fluent. Results from the CFD analysis were used as inputs to a 6DOF dynamical simulation of the vehicle trajectory and attitude, written in MATLAB using an object-oriented structure. A proportional–integral–derivative controller was designed to control the canard stabilization fins in flight. The control software was tested using MATLAB and Simulink. Ansys Workbench was used for structural analysis of the airframe and main fins. An analysis of the composite motor was completed using Cantera and COMSOL to model the chemical equilibrium reaction and evaluate the temperature distribution in the motor during flight. These results were used to provide chamber conditions in a MATLAB model for ideal rocket performance. Results are presented from these analyses as well as a description of partial prototype construction completed at the subsystem level.

*Certain materials are included under the fair use exemption of the U.S. Copyright Law and have been prepared according to the fair use guidelines and are restricted from further use.*

## Acknowledgements

The MQP team would like to thank and recognize the following individuals listed below for their help and support over the course of the project.

- Project Advisor Professor Blandino for his support and guidance over the course of the project.
- Professor Hera for providing training and one-on-one support in Ansys, Fluent, and COMSOL as well as her boundless support in resolving the issues making Fluent and Ansys Mechanical interface.

# Table of Contents

Abstract .....	i
Acknowledgements .....	ii
Table of Contents .....	iii
List of Figures .....	vi
List of Tables .....	ix
1 Introduction .....	1
1.1 Project Goals .....	1
1.2 Project Design Requirements, Constraints, and Other Considerations .....	2
1.2.1 Design Requirements .....	2
1.2.2 Design Constraints .....	2
1.2.3 Design Considerations .....	2
1.2.4 Safety Considerations .....	2
1.3 Tasks .....	3
1.4 Design Overview and Flight Profile .....	9
1.4.1 Design Overview .....	9
1.4.2 Flight Profile .....	10
1.5 Background and Literature Review .....	11
1.5.1 Airframe and Recovery System .....	11
1.5.2 Flight Dynamics Analysis .....	18
1.5.3 Propulsion, Thermal, and Separation Systems .....	29
2 Airframe and Recovery System Methodology and Results .....	34
2.1 Methodology .....	34
2.1.1 Airframe .....	34
2.1.2 Fins .....	35
2.1.3 Retromotor Nose Cone .....	37
2.1.4 Recovery Parachutes .....	38
2.1.5 Canards Bay .....	40
2.1.6 Airframe Stress Distribution (ARS Analysis Task 1) .....	44
2.1.7 Computational Fluid Dynamics (CFD) .....	54
2.1.8 Canard Aerodynamic Analysis (ARS Analysis Task 2) .....	59
2.1.9 Canard Stress Analysis (ARS Analysis Task 3) .....	65

2.2	Results .....	68
2.2.1	Airframe Stress Distribution (ARS Analysis Task 1).....	68
2.2.2	Canard Aerodynamic Loads (Analysis Task 2).....	70
2.2.3	Canard Stress Distribution (ARS Analysis Task 3).....	72
3	Flight Dynamics Analysis Methodology and Results .....	73
3.1	Methodology .....	73
3.1.1	Avionics .....	73
3.1.2	Electronics Bay .....	75
3.1.3	Dynamic Simulator (FDA Analysis Task 1).....	77
3.1.4	Aerodynamic Loads (FDA Analysis Task 2).....	85
3.1.5	Control System (FDA Analysis Task 3).....	91
3.1.6	Electronics Bay Construction .....	98
3.2	Results .....	99
3.2.1	Vehicle Dynamics and Performance Model (FDA Analysis Task 1).....	99
3.2.2	Drogue Replacement Retrorocket Model (FDA Analysis Task 4).....	106
3.2.3	Vehicle Aerodynamic Loads Simulation (FDA Analysis Task 2).....	108
3.2.4	Canard Control (FDA Analysis Task 3) .....	112
4	Propulsion, Thermal, and Separation Systems Methodology and Results .....	115
4.1	Methodology .....	115
4.1.1	Propulsion and Motors (PTSS Analysis Task 1) .....	115
4.1.2	Ignition System .....	119
4.1.3	Thermal Distribution Model (PTSS Analysis Task 2).....	120
4.1.4	Recovery Bay .....	127
4.1.5	CO <sub>2</sub> System (PTSS Analysis Task 3) .....	128
4.2	Results .....	134
4.2.1	Motor Performance Model (PTSS Analysis Task 1) .....	134
4.2.2	Thermal Model (Analysis Task 2) .....	137
4.2.3	CO <sub>2</sub> Separation Model (PTSS Analysis Task 3).....	139
5	Summary, Recommendations, and Broader Impacts.....	144
5.1	Airframe and Recovery System .....	144
5.2	Flight Dynamics Analysis .....	145
5.3	Propulsion, Thermal, and Separation Systems.....	146

5.4	Project Broader Impacts .....	146
6	References .....	148
7	Appendices .....	153
7.1	Appendix A: Equations in ARS Analysis Task 1 .....	153
7.2	Appendix B: ARS Analysis Task 1 Material Properties .....	153
7.3	Appendix C: FDA Analysis Task 2 Fluent Simulation Data .....	154
7.4	Appendix D: NASA CEA Results .....	157
7.5	Appendix E: Equations in COMSOL Model .....	159
7.6	Appendix F: Equation Derivations used for PTSS Analysis Task 3.....	161

## List of Figures

Figure 1 CAD Model of Innovative Rocket .....	9
Figure 2 CAD Model of Baseline Rocket.....	10
Figure 3 OpenRocket Program and Model .....	10
Figure 4 Flight Profile.....	11
Figure 5 Fiberglass Body Tubes [2]. © Apogee Components Inc. 2021 .....	12
Figure 6 Tangent Ogive Geometric Relationships [3]. © Gary A. Crowell Sr. 1996 .....	13
Figure 7 Common Fin Geometries [1]. © NASA 2015.....	14
Figure 8 Rocket Model with Canards on Upper Airframe .....	14
Figure 9 Plot of Single and Dual Event Recovery Profiles [5]. © Apogee Components Inc. ....	15
Figure 10 Main and Drogue Parachute Locations [1]. © NASA 2015.....	16
Figure 11 DERS Electronics and Separation Device Locations [6]. © Madison West Rocketry	16
Figure 12 Jolly Logic Chute Release [7]. © Jolly Logic 2021 .....	17
Figure 13 Falcon 9 Sample Mission Profile [9]. © SpaceX 2021 .....	18
Figure 14 J2000 Coordinate Frame [11]. © NASA 1998.....	19
Figure 15 ECEF and Geodetic Coordinate Systems [13]. © Chuckage 2021 .....	20
Figure 16 NED Coordinate System [14]. © BasicAirData 2011 .....	20
Figure 17 Body and Wind Frames [14]. © BasicAirData 2011.....	21
Figure 18 Flight Forces on Rocket [15]. © Sampo Niskanen 2011.....	22
Figure 19 6-DOF Simulator Architecture [16]. © Journal of Aerospace Engineering 2010.....	23
Figure 20 Monte-Carlo Simulation Output [16]. © Journal of Aerospace Engineering 2010 ....	23
Figure 21 Simplified Block Diagram of and LQR Controller [32] © IEEE 2019.....	28
Figure 22 Future State and Control Input for MPC [31] © University of Edinburgh 2017 .....	28
Figure 23 Cutaway of Model Rocket Motor [34] © Apogee Components 2019 .....	30
Figure 24 Diagram of Motor Mounting System [40] © George Stine 2004.....	31
Figure 25 Tinder Rocketry Peregrine Exploded View [41] © Tinder Rocketry .....	32
Figure 26 Full Rocket Assembly .....	34
Figure 27 Rocket Airframe .....	34
Figure 28 Fin Side View .....	36
Figure 29 Fin Front View .....	36
Figure 30 Retromotor Nose Cone Half.....	37
Figure 31 Innovative Rocket Parachute Configuration .....	39
Figure 32 Jolly Logic Chute Release [7]. © Jolly Logic 2021 .....	39
Figure 33 Baseline Rocket Parachute Configuration.....	40
Figure 34 Canards Bay SOLIDWORKS Model.....	41
Figure 35 Canard Mechanical Actuation .....	42
Figure 36 Canard Actuation Linkages .....	43
Figure 37 Canard Rotated 5 Degrees Clockwise .....	43
Figure 38 Motor Mount Assembly .....	45
Figure 39 ACP User Interface.....	47
Figure 40 Motor Mount Analysis Workbench.....	47
Figure 41 Motor Mount Mesh.....	48
Figure 42 Motor Mount Boundary Conditions .....	49

Figure 43 Parachute Bay .....	50
Figure 44 Parachute Bay Boundary Conditions.....	50
Figure 45 Parachute Bay Mesh .....	51
Figure 46 Fin Static Pressure Distribution in Static Structural.....	52
Figure 47 Boundary Conditions of Fin .....	53
Figure 48 Fin Mesh.....	54
Figure 49 Overview of Pressure-Based Solution Methods in Fluent [49] © Ansys 2009.....	56
Figure 50 Case 1 Zero Angle of Attack SOLIDWORKS Model .....	60
Figure 51 Case 2 Five Degree Angle of Attack SOLIDWORKS Model .....	61
Figure 52 SpaceClaim Enclosure/Fluid Flow Domain .....	61
Figure 53 Fluent Mesh for Case 2 ISO View .....	63
Figure 54 Fluent Mesh for Case 2 Airframe View .....	63
Figure 55 Fluent Flow Setup.....	64
Figure 56 Ansys Workbench for Fluent and Static Structural.....	65
Figure 57 SpaceClaim Canard Geometry .....	66
Figure 58 Fluent Static Pressure Results - Face Angled into Flow (Case 2).....	67
Figure 59 Imported Static Pressured Distribution - Face Angled into Flow (Case 2).....	67
Figure 60 Motor Mount von Mises Stress .....	68
Figure 61 Recovery Bay von Mises Stress .....	68
Figure 62: Fin von Mises Stress.....	69
Figure 63 Resultant Static Pressure - Symmetric on both Faces (Case 1).....	70
Figure 64 Resultant Static Pressure Distribution - Face angled out of Flow (Case 2) .....	71
Figure 65 Resultant Static Pressure Distribution - Face Angled into Flow (Case 2) .....	71
Figure 66 Canard Stress Distribution – Face Angled into Flow.....	72
Figure 67 HPRC Avionics Stack .....	73
Figure 68 WPI HPRC Avionics Block Diagram of Respective Boards .....	73
Figure 69 Phases of Launch Vehicle Flight.....	75
Figure 70 Electronics Bay Front and Side Views.....	76
Figure 71 Avionics Plate Assembly.....	76
Figure 72 Flat Earth Simulation Coordinate System .....	82
Figure 73 Setup of Ansys Fluent External Flow Simulation [37] .....	86
Figure 74 Model of External Rocket Geometry .....	87
Figure 75 Rocket Geometry Plane of Symmetry .....	87
Figure 76 Fluid Flow Domain.....	88
Figure 77 Mesh of Fluid Domain.....	89
Figure 78 Simulink PID Model.....	92
Figure 79 Roll Forcing Coefficient vs. Mach and Canard Cant .....	94
Figure 80 Roll Forcing Coefficients 2D Plot.....	95
Figure 81 Roll Damping Coefficients vs. Mach and Roll Rate .....	95
Figure 82 Roll Damping Coefficients 2D Plot .....	96
Figure 83 Interior of Electronics Bay .....	99
Figure 84 Test Case 1 Position vs Time .....	100
Figure 85 Test Case 1 Velocity vs Time.....	100



Figure 86 Test Case 2 Euler Angles vs Time .....	101
Figure 87 Test Case 2 Body Rates vs Time .....	102
Figure 88 Full Flight Trajectory .....	103
Figure 89 Full Flight Euler Angles vs Time .....	104
Figure 90 Full Flight Velocity vs Time .....	104
Figure 91 WGS84 Simulation Trajectory .....	105
Figure 92 Position during/after Retromotor Firing .....	106
Figure 93 Retromotor Euler Angles vs Time .....	107
Figure 94 Retromotor Velocity vs Time .....	107
Figure 95 Drag and Lift Coefficients vs. Angle of Attack .....	109
Figure 96 Moment Coefficients vs. Angle of Attack .....	110
Figure 97 Body-Fixed Coordinate System and Force Vectors .....	110
Figure 98 Dynamic Pressure for 180 m/s Flight Speed and 0° Angle of Attack (air flows horizontally from right to left) .....	112
Figure 99 Dynamic Pressure for 180 m/s Flight Speed and 20° Angle of Attack (air flows from top left to bottom right at an angle of 20° from the horizontal) .....	112
Figure 100 Rocket Flight States for Case 1 .....	113
Figure 101 Rocket Flight States for Case 2 .....	114
Figure 102 Geometry and Alignment of Domains in COMSOL Model .....	121
Figure 103 Boundary Conditions Defined in Heat Transfer Model .....	124
Figure 104 Boundary Conditions Defined in Fluid Flow Model .....	126
Figure 105 Isometric View of the Recovery Bay .....	127
Figure 106 The Recovery Bay Sled .....	128
Figure 107 Combustion Product Mole Fractions from CEA and NASA Models .....	135
Figure 108 Motor Temperature Distribution at t = 1.7s .....	137
Figure 109 Velocity Distribution in the Motor at t = 1.7s .....	138
Figure 110 Velocity Streamlines for the Velocity Distribution in the Motor Case at t = 1.7s ...	139
Figure 111 Change in CO <sub>2</sub> Pressure for Transient Choked Flow Phases .....	141
Figure 112 Change in CO <sub>2</sub> Mass for Transient Choked Flow Phases .....	141
Figure 113 Change in CO <sub>2</sub> Temperature for Transient Choked Flow Phases .....	142
Figure 114 Change in Pressure in Recovery Compartment .....	142
Figure 115 Change of Gas Mass in Recovery Compartment .....	143
Figure 116 Drag and Lift Coefficients vs. Angle of Attack for Varying Flight Speed .....	155
Figure 117 Moment Coefficients vs. Angle of Attack for Varying Flight Speed .....	156

## List of Tables

Table 1.1 Airframe and Recovery System (ARS) Subteam Analysis Tasks .....	3
Table 1.2 Propulsion, Thermal and Separation Systems (PTSS) Subteam Analysis Tasks .....	5
Table 1.3 Flight Dynamics Analysis (FDA) Subteam Analysis Tasks.....	6
Table 1.4 Analysis Task Method and Results Sections .....	8
Table 1.5 Flight Stages .....	11
Table 1.6 Comparison of Existing Dynamic Simulators .....	24
Table 1.7 Variables for General Form of Navier-Stokes Equations [23] .....	26
Table 2.1 Motor Mount Assembly Boundary Conditions .....	49
Table 2.2 Parachute Bay Boundary Conditions.....	51
Table 2.3 Fin Boundary Conditions.....	54
Table 2.4 Summary of Relevant Boundary Conditions in Ansys Fluent [48].....	58
Table 2.5 Assumptions used for Fluent Inputs .....	59
Table 2.6 Canard Aerodynamic Load Analysis Cases.....	59
Table 2.7 Fluid Flow Domain Boundary Conditions.....	63
Table 2.8 Fluent Reference Values.....	65
Table 3.1 COTS Included Products HPRC Avionics Stack .....	74
Table 3.2 Flight Dynamics Model Variables.....	77
Table 3.3 WGS84 Parameters [61] .....	79
Table 3.4 Dynamic Simulator Classes .....	83
Table 3.5 Material Properties of Fluid Domain and Rocket Body .....	90
Table 3.6 Boundary Type and Conditions at Domain Surfaces.....	90
Table 3.7 State Space Variables.....	93
Table 3.8 Moments of Inertia.....	93
Table 3.9 Roll Forcing Coefficient .....	94
Table 3.10 Roll Damping Coefficient.....	94
Table 3.11 Variable Definitions for Roll Coefficients.....	96
Table 3.12 Constants and Variables for Control Code .....	97
Table 3.13 Vehicle Properties for Flight Simulation .....	103
Table 3.14 Aerodynamic Simulation Reference Values.....	108
Table 3.15 Initial Conditions for MainLoop.....	112
Table 3.16 PID Coefficient Values.....	113
Table 4.1 Published Cesaroni Propellant Ingredients .....	116
Table 4.2 List of Variables for PTSS Analysis Task 1 Calculations .....	119
Table 4.3 COMSOL Domain Descriptions and Dimensions.....	121
Table 4.4 Heat Source and Mass Flow Rate Calculations .....	122
Table 4.5 Descriptions, Properties, and Initial Conditions of Domains .....	123
Table 4.6 Material Properties for Domains of COMSOL Model .....	124
Table 4.7 Summary of Heat Transfer Boundary Conditions and Inputs .....	125
Table 4.8 Summary of Fluid Flow Boundary Conditions and Inputs.....	127
Table 4.9 Provided and Known Values for Flow Analysis .....	128
Table 4.10 Nomenclature and Units for Flow Analysis .....	129
Table 4.11 Chamber and Exit Property Results.....	136

Table 4.12 Motor Performance Results .....	136
Table 4.13 Summary of CO <sub>2</sub> Canister Initial Conditions .....	139
Table 4.14 Summary of CO <sub>2</sub> Initial and Final Conditions for Phase I Transient Flow .....	140
Table 7.1 Aerodynamic Coefficient Data for Varying Angle of Attack and Flight Speed .....	154

Table of Authorship

<b>Section</b>	<b>Author</b>
Abstract	Team
Acknowledgements	Team
1.1 Project Goals	TO
1.2 Project Design Requirements, Constraints, and Other Considerations	KB
1.3 Tasks	KB
1.4 Design Overview and Flight Profile	KB, CW
Rocket Construction	AH
Canards	AH
Parachutes	AH
Recovery Electronics and Hardware	CW
Powered Descent	CW
Coordinate Systems and Transformations	TO
Vehicle Dynamics	TO
Simulator Architecture	TO
Comparison of Existing Dynamic Simulators	TO
Aerodynamic Simulation	TH
Controls	KB
Propulsion and Motors	DM
Ignition Systems	TL
Motor Mounting System	TL
Separation	AC
2.1.1 Airframe	AH
2.1.2 Fins	AH
2.1.3 Retromotor Nose Cone	AH
2.1.4 Recovery Parachutes	CW
2.1.5 Canards Bay	CW
2.1.6 Airframe Stress Distribution (ARS Analysis Task 1)	AH
2.1.7 Computational Fluid Dynamics (CFD)	TH

<b>Section</b>	<b>Author</b>
2.1.8 Canard Aerodynamic Analysis (ARS Analysis Task 2)	CW
2.1.9 Canard Stress Analysis (ARS Analysis Task 3)	CW
2.2.1 Airframe Stress Distribution (ARS Analysis Task 1)	AH
2.2.2 Canard Aerodynamic Loads (Analysis Task 2)	CW
2.2.3 Canard Stress Distribution (ARS Analysis Task 3)	CW
3.1.1 Avionics	KB
3.1.2 Electronics Bay	KB
3.1.3 Dynamic Simulator (FDA Analysis Task 1)	TO
3.1.4 Aerodynamic Loads (FDA Analysis Task 2)	TH
3.1.5 Control System (FDA Analysis Task 3)	KB
3.2.1 Vehicle Dynamics and Performance Model (FDA Analysis Task 1)	TO
3.2.2 Drogue Replacement Retrorocket Model (FDA Analysis Task 4)	TO
3.2.3 Vehicle Aerodynamic Loads Simulation (FDA Analysis Task 2)	TH
3.2.4 Canard Control (FDA Analysis Task 3)	KB
4.1.1 Propulsion and Motors (PTSS Analysis Task 1)	DM
4.1.2 Ignition System	TL
4.1.3 Thermal Distribution Model	TL
4.1.4 Recovery Bay	TL
4.1.5 CO <sub>2</sub> System (PTSS Analysis Task 3)	AC
4.2.1 Motor Performance Model (PTSS Analysis Task 1)	DM
4.2.2 Thermal Model (Analysis Task 2)	TL
4.2.3 CO <sub>2</sub> Separation Model (PTSS Analysis Task 3)	AC
5.1 Airframe and Recovery System	AH, CW
5.2 Flight Dynamics Analysis	KB, TH, TO
5.3 Propulsion, Thermal, and Separation Systems	AH, TL
5.4 Project Broader Impacts	AH

# 1 Introduction

## 1.1 Project Goals

This project had three overall project goals:

- Design, build, and fly a reusable rocket to an altitude of at least 1500 feet.
- Provide students with the opportunity to work as a team to design, build and test a moderately complex aerospace system in which the overall vehicle performance is critically tied in with the mass and performance of the individual components and assemblies.
- Provide students with specialized training in and opportunity to apply software tools: MATLAB, Ansys – Static Structural Analysis, Ansys - Fluent, Ansys - Dynamic Analysis, Cantera, others

The Airframe and Recovery System, Flight Dynamics Analysis, and Propulsion, Thermal, and Separation Systems subteams outlined specific analysis goals to be complete by the end of the project.

- Airframe and Recovery System (ARS)
  - o Determine Airframe Stress Distribution
    - Create a map of stress distribution
    - Identify critical locations and joints
  - o Determine Canard Aerodynamic Loads
    - Calculate net forces and moments on airframe for each simulation case
    - Evaluate forces and moments produced by different canard fin options
  - o Determine Canard Fin Stress Distribution
    - Create a map of stress distribution
    - Identify critical locations and joints
- Flight Dynamics Analysis (FDA)
  - o Develop Vehicle Dynamics Model
    - Develop numerical simulator for the flight of the vehicle
    - Evaluate forces and moments acting on vehicle during flight
  - o Determine Vehicle Aerodynamic Loads
    - Create pressure contours and plots of forces on vehicle for various flight states
    - Estimate forces on vehicle for various flight states
  - o Develop Canard Control System
    - Create and tune a roll control system using the vehicle's canards
  - o Evaluate Retrorocket Stability Model
    - Confirm that the vehicle will remain stable during and after retrorocket firing
- Propulsion, Thermal, and Separation Systems (PTSS)
  - o Create Motor Performance Model
    - Determine composition of combustion products, chamber pressure and temperature, mass flow rate, and specific impulse

- Determine Temperature Distribution
    - o Temperature distribution through propellant grain, motor casing, and airframe
    - o Model CO2 Ejection System
    - o Create pressure vs. time curve for separation event
    - o Determine required CO2 mass

## 1.2 Project Design Requirements, Constraints, and Other Considerations

All design requirements, constraints, and other considerations for the project were decided by and shared between the three subteams.

### 1.2.1 Design Requirements

- Use a modular and adaptable design to create a baseline and innovative rocket.
- Use a camera as a payload to record video during flight.
- Use traditional dual deploy parachutes to recover the baseline rocket and a nose cone retromotor to recover the innovative rocket.
- Use a CO2 separation system.
- Use a Level-2 high impulse motor for the baseline rocket to reach an apogee near 4,000 feet.
- Use a Level-1 motor for the retromotor.
- Use a custom avionics board to record data to be analyzed after flight.
- Use an actuated canard system to control the roll of the rocket.

### 1.2.2 Design Constraints

- The rocket must leave the launch rails at a velocity of at least 52 feet per second so that the rocket is stable.
- The rocket will have a stability between 2 and 3 cal in order to be not under or over stable.
- The internal components must be non-interfering with other components in close proximity and be easily accessible.

### 1.2.3 Design Considerations

- All innovative systems must have a replacement system that adheres to the standards of the NAR code [1] for the ability to launch at a National Association of Rocketry (NAR) site.
- Rocket must have a redundant recovery system to help ensure safe descent of launch vehicle.
- The rocket body will be predominately made of fiberglass in order to satisfy a combination of reducing weight and strength.
- The rocket will predominately consist of commercially available materials and components.

### 1.2.4 Safety Considerations

- Rocket must comply with safety guidelines from the NAR [1], such as but not limited to:

- o The rocket must be launched by a person who holds the certification level of the motor being used.
- o The rocket must be built with lightweight materials.
- o Commercially certified made rockets motors are the only launching motors permitted.
- o The rocket must be launched with an electrical launch system and electric motor igniters which are installed after the rocket is on the launch pad or in a designated preparation area.
- o The launch system must have a launch switch that has a safety interlock and is installed and tested before launch.
- o Before the launch, a five second countdown is required and everyone in the vicinity is to be notified.
- o If the rocket does not launch, the launcher will watch 60 seconds after initial attempt to diagnose the rocket.
- o The launch location will be adequate for the motor selected and have enough open space for the required safety distances.
- o The recovery system should be a system like a parachute so that that all parts of the rocket return to the ground safely, undamaged, and ready to be flown again.
- Simulations of rocket performance will occur before launch to help ensure stable flight.
- Operators will observe safe handling of black powder and motor used for launch.

### 1.3 Tasks

In order to meet the project goals specified in Section 1.1, each subteam created their own analysis tasks. Each of these analysis tasks are described in Table 1.1, Table 1.2, and Table 1.3. Table 1.4 lists the section of the report where each task is discussed more in depth, including the method taken and results of the task being described.

Table 1.1 Airframe and Recovery System (ARS) Subteam Analysis Tasks

<b>ARS Analysis Task 1: Airframe Stress Distribution</b>
<b>Problem Statement:</b>
Identify critical locations of high stress throughout the airframe and internal structure during peak acceleration loads.
<b>Solution Methodology:</b>
<ul style="list-style-type: none"> <li>• Tool(s): Ansys Mechanical Workbench / Dynamic Analysis, SOLIDWORKS Stress Analysis</li> <li>• Inputs: Airframe solid model including information on materials, joint/bond point models, acceleration due to primary motor, aerodynamic loads (forces and moments), impulse from separation events (black powder or CO2 pressurization actuated)</li> </ul>
<b>Analysis Products:</b>
<ul style="list-style-type: none"> <li>• Map of stress distribution</li> <li>• Identification of critical locations/joints</li> </ul>



<b>Use of Results:</b>
<ul style="list-style-type: none"> <li>• Stress data will be used to determine weak points within airframe design during 1) maximum acceleration and 2) separation events</li> <li>• Determine method(s) to improve structural integrity of the airframe and internal structures e.g., electronics bay and motor bay</li> </ul>

<b>ARS Analysis Task 2: Canard Aerodynamic Loads</b>
<b>Problem Statement:</b>
Evaluate the aerodynamic forces and moments on 1) a single canard fin mounted on airframe and 2) a set of fins arranged on the airframe
<b>Solution Methodology:</b>
<ul style="list-style-type: none"> <li>• Tool: Ansys Fluent</li> <li>• Required Inputs: canard fin solid model, initial and boundary conditions (flight velocity, etc.), fluid properties, mesh characteristics, number, and placement of fins on the airframe</li> <li>• Evaluate forces and moments produced by canard fin options</li> </ul>
<b>Analysis Products:</b>
<ul style="list-style-type: none"> <li>• Net forces and moments on airframe for each “case” considered. A “case” consists of a combination of fin type, placement, number, and flight conditions.</li> <li>• Estimate of loads on single fin</li> </ul>
<b>Use of Results:</b>
<ul style="list-style-type: none"> <li>• Forces and moments on vehicle (with multiple fins) will be used as an input in the vehicle dynamics model</li> <li>• Loads on single fin will be used as input in canard fin stress analysis.</li> </ul>

<b>ARS Analysis Task 3: Canard Fin Stress Distribution</b>
<b>Problem Statement:</b>
Identify critical locations of high stress throughout the canard fin during peak acceleration loads.
<b>Solution Methodology:</b>
<ul style="list-style-type: none"> <li>• Tool(s): Ansys Mechanical Workbench / Dynamic Analysis, SOLIDWORKS Stress Analysis</li> <li>• Required Inputs: Fin solid model including information on materials, flight speed aerodynamic loads on the fin</li> <li>• Estimate stresses on the fin as a function of speed and vehicle orientation</li> </ul>
<b>Analysis Products:</b>
<ul style="list-style-type: none"> <li>• Map of stress distribution</li> <li>• Identification of critical locations/joints</li> </ul>

<b>Use of Results:</b>
<ul style="list-style-type: none"> <li>• Identify maximum allowable flight speed (as a function of vehicle orientation)</li> <li>• Determine method(s) to improve structural integrity of fin and fin-airframe attachment method</li> <li>• If fins are servo-controlled, conform that servo has sufficient torque to position fin under load</li> </ul>

Table 1.2 Propulsion, Thermal and Separation Systems (PTSS) Subteam Analysis Tasks

<b>PTSS Analysis Task 1: Motor Performance Model</b>
<b>Problem Statement:</b>
Create a simplified model for the selected motors, which can be used to estimate performance (thrust and Isp) that can be compared with published data
<b>Solution Methodology:</b>
<ul style="list-style-type: none"> <li>• Tools: Cantera, MATLAB</li> <li>• Major Assumptions: chemical equilibrium in chamber, frozen flow, steady state, isentropic flow in nozzle</li> <li>• Required Inputs: Propellant composition and properties, chamber and nozzle geometry, ambient conditions (p, T)</li> <li>• Formulate model that couples equilibrium chemistry with flow through nozzle</li> </ul>
<b>Analysis Products:</b>
<ul style="list-style-type: none"> <li>• Composition of combustion products, chamber pressure and temperature, mass flow rate, thrust, specific impulse</li> </ul>
<b>Use of Results:</b>
<ul style="list-style-type: none"> <li>• Compare predicted performance with manufacturer data</li> <li>• Estimate heat flux for thermal analysis</li> </ul>

<b>PTSS Analysis Task 2: Temperature Distribution</b>
<b>Problem Statement:</b>
Estimate the temperature distribution through the propellant grain, motor casing, and rocket body subject to heat flux from the motor
<b>Solution Methodology:</b>
<ul style="list-style-type: none"> <li>• Tool: COMSOL</li> <li>• Required Inputs: Material and property data for motor, casing, and rocket body. External, flight boundary conditions (velocity, ambient pressure, temperature, air properties)</li> </ul>
<b>Analysis Products:</b>
<ul style="list-style-type: none"> <li>• Temperature and heat flux distribution through motor and airframe structure</li> </ul>

<b>Use of Results:</b>
<ul style="list-style-type: none"> <li>• Conduct thermal analysis on the motor body tubes to see if there are any points of risk for structural degradation due to overheating (adhesives used for joints, etc.)</li> </ul>

<b>PTSS Analysis Task 3: CO2 Separation System Model</b>
<b>Problem Statement:</b>
Create a simplified model for the CO2-based stage separation system
<b>Solution Methodology:</b>
<ul style="list-style-type: none"> <li>• Tools: MATLAB, Excel, force balance (scale)</li> <li>• Experimentally measure force required for separation</li> <li>• Formulate transient gas pressurization model</li> <li>• Required Inputs: Force required to jettison the nose cone (or other stage), CO2 system parameters (volume, pressure, orifice diameter, etc.)</li> </ul>
<b>Analysis Products:</b>
<ul style="list-style-type: none"> <li>• Pressure vs. time curve for separation event, required CO2 mass as a function of airframe design</li> </ul>
<b>Use of Results:</b>
<ul style="list-style-type: none"> <li>• Evaluate design options, integrate calculated separation time into flight event model</li> </ul>

Table 1.3 Flight Dynamics Analysis (FDA) Subteam Analysis Tasks

<b>FDA Analysis Task 1: Vehicle Dynamics and Performance Model</b>
<b>Problem Statement:</b>
Create an integrated model that can be used to estimate the vehicle attitude dynamics (angles and rates) as a function of time from launch to impact, as well as the rocket trajectory (including max altitude and range)
<b>Solution Methodology:</b>
<ul style="list-style-type: none"> <li>• Tool: MATLAB</li> <li>• Required Inputs: rocket geometry and inertia properties; center of pressure, center of mass; thrust, simplified drag and moment models/data from related analysis tasks, wind profile in given topography, avg. wind speed across the rocket's altitude range</li> <li>• Formulate model consisting of <i>two, coupled systems</i> of nonlinear ODEs, one for attitude dynamics (Euler solver) and one for the equations of motion (Newton's 2<sup>nd</sup> law) describing the vehicle trajectory. Euler equations are solved at each time step as the equations of motion are solved for the trajectory</li> </ul>
<b>Analysis Products:</b>
<ul style="list-style-type: none"> <li>• Simulation of rocket trajectories, capturing statistically randomized variation in wind speed and direction</li> </ul>

<ul style="list-style-type: none"> <li>• Evaluation of forces and moments acting on vehicle of given design and flight state (attitude and velocity) when subject to wind disturbances and their effect on trajectory</li> </ul>
<b>Use of Results:</b>
<ul style="list-style-type: none"> <li>• Landing probability distribution plot (safety plot)</li> <li>• Data that can be used to compare stability and performance of innovative fin design with baseline</li> <li>• Provide estimates of perturbation on vehicle attitude (angles and rates) when subject to transient wind disturbances.</li> <li>• Estimate upper limit on wind disturbance to maintain stable flight</li> <li>• Evaluation of effectiveness of innovative fin design, i.e., compare vehicle's ability to maintain stable flight with baseline design</li> </ul>

<b>FDA Analysis Task 2: Vehicle Aerodynamic Loads – Simulation</b>
<b>Problem Statement:</b>
Estimate the aerodynamic forces and moments on the vehicle as a function of velocity and vehicle attitude
<b>Solution Methodology:</b>
<ul style="list-style-type: none"> <li>• Tools: SimScale, Ansys Fluent</li> <li>• Required Inputs: rocket geometry and inertia properties, center of pressure, center of mass, drag and moment coefficient data for similar vehicles (from literature), initial and boundary conditions, fluid properties, mesh characteristics, wind profile in given topography, avg. wind speed across the rocket's altitude range</li> </ul>
<b>Analysis Products:</b>
<ul style="list-style-type: none"> <li>• Pressure contours, plots of forces and moments acting on vehicle of given design and flight state (attitude and velocity)</li> <li>• Estimate of forces and moments acting on vehicle of given design and flight state (attitude and velocity) when subject to wind disturbances</li> </ul>
<b>Use of Results:</b>
<ul style="list-style-type: none"> <li>• Provide load estimates to be used in structural stress analysis</li> <li>• Provide forces and moment data that can be used in “table-lookup” for simulation of vehicle trajectory and attitude</li> <li>• Provide estimates of perturbation on vehicle attitude (angles and rates) when subject to transient wind disturbances.</li> </ul>

<b>FDA Analysis Task 3: Canard Control System</b>
<b>Problem Statement:</b>
Develop a control system for actuation of canard fins during flight to control the roll of the vehicle.
<b>Solution Methodology:</b>
<ul style="list-style-type: none"> <li>• Tool: MATLAB</li> </ul>

<ul style="list-style-type: none"> <li>• Create a 1-DOF Simulink model of the roll of the vehicle with canard control</li> <li>• Linearize model, select control law, and tune gains</li> <li>• Test control system (full-scale vehicle on car or small-scale wind tunnel test)</li> </ul>
<b>Analysis Products:</b>
<ul style="list-style-type: none"> <li>• A tuned control system.</li> </ul>
<b>Use of Results:</b>
<ul style="list-style-type: none"> <li>• Will be used during ascent to control canard system.</li> </ul>

<b>FDA Analysis Task 4: Drogue Replacement Retrorocket Model</b>
<b>Problem Statement:</b>
Evaluate vehicle stability just before, during, and after retrorocket firing.
<b>Solution Methodology:</b>
<ul style="list-style-type: none"> <li>• Tools: MATLAB, OpenRocket</li> <li>• Required Inputs: motor performance data, vehicle configuration and inertia properties</li> <li>• Evaluate dynamic stability of vehicle under different flight conditions</li> </ul>
<b>Analysis Products:</b>
<ul style="list-style-type: none"> <li>• Confirmation that vehicle will (or will not) remain dynamically stable during and after retrorocket firing.</li> </ul>
<b>Use of Results:</b>
<ul style="list-style-type: none"> <li>• Will establish feasibility of retrorocket and data will be used to select proper motor size.</li> </ul>

Table 1.4 Analysis Task Method and Results Sections

	<b>Methodology Section</b>	<b>Results Section</b>
<b>Airframe and Recovery System Subteam</b>		
ARS Analysis Task 1: Airframe Stress Distribution	2.1.6	2.2.1
ARS Analysis Task 2: Canard Aerodynamic Loads	2.1.8	2.2.2
ARS Analysis Task 3: Canard Fin Stress Distribution	2.1.9	2.2.3
<b>Propulsion, Thermal, and Separation System Subteam</b>		
PTSS Analysis Task 1: Motor Performance Model	4.1.1	4.2.1
PTSS Analysis Task 2: Temperature Distribution	4.1.3	4.2.2
PTSS Analysis Task 3: CO2 Separation System Model	4.1.5	4.2.3
<b>Flight Dynamics Analysis Subteam</b>		
FDA Analysis Task 1: Vehicle Dynamics and Performance Model	3.1.3	3.2.1
FDA Analysis Task 2: Vehicle Aerodynamic Loads – Simulation	3.1.4	3.2.3
FDA Analysis Task 3: Canard Control System	3.1.5	3.2.4
FDA Analysis Task 4: Drogue Replacement Retrorocket Model	3.1.3	3.2.2

## 1.4 Design Overview and Flight Profile

### 1.4.1 Design Overview

To complete the tasks and goals above, the team designed an innovative rocket, seen in Figure 1. A custom nose cone, including centering rings, was designed for the retromotor and its motor retention. Below that, a canard bay includes the canards as well as the hardware, such as a servo, designed to actuate the canards. The avionics bay houses the payload, a camera, as well as a battery and avionics board stack. The avionics bay also serves as an attachment point for the shock cord connected to the parachute. The recovery bay has a bulkhead at the top which acts as the other attachment point for the parachute and its shock cord. The recovery bay also houses the recovery electronics, such as the primary and back up altimeters as well as their batteries and respective CO2 ejection systems. Below that, the motor tube will be epoxied to three centering rings to secure the motor within the airframe. Towards the aft of the rocket, there are four trapezoidal fins that ensure stability throughout flight. At the aft of the rocket, there is a motor retention flange used to prevent the motor from ejecting from the rocket.

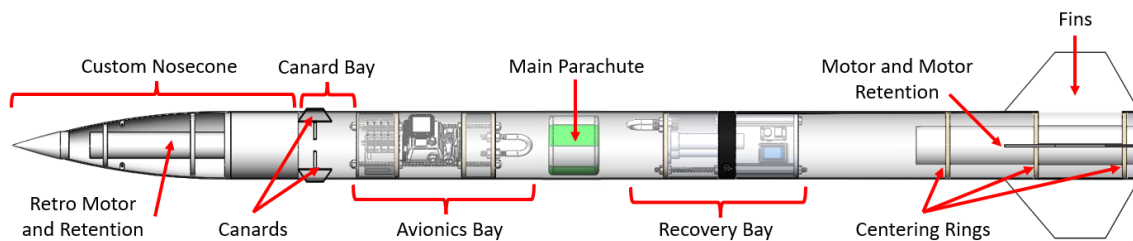


Figure 1 CAD Model of Innovative Rocket

Due to safety concerns at the launch site, the active canard system could not be flown. Since the team desired to launch their rocket, a baseline version was planned to be constructed as seen in Figure 2. This requires the replacement of the custom nose cone with a standard COTS nose cone in conjunction with the addition of a main parachute. The drogue and main parachute would both exit the vehicle at apogee, but the main would be held closed by a Jolly Logic Chute Release. The Chute Release would disengage at 700 feet and allow the main parachute to unfurl. The canard system would have remained locked during flight.

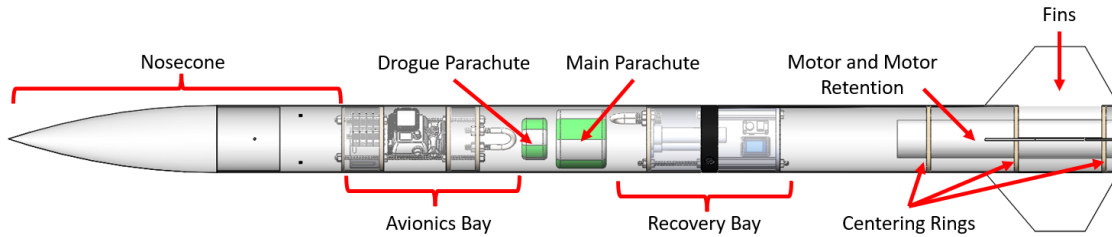


Figure 2 CAD Model of Baseline Rocket

In conjunction with a CAD model, a software tool called OpenRocket was used to maintain an additional model of the rocket, seen in Figure 3. This software was able to allow us to perform critical flight simulations as well as provide the team with important flight data.

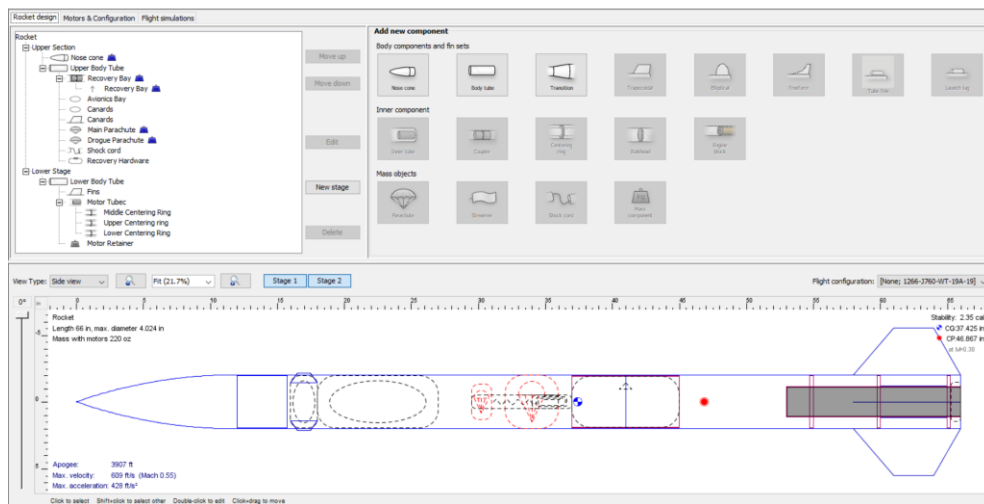


Figure 3 OpenRocket Program and Model

#### 1.4.2 Flight Profile

Below in Figure 4 is the flight profile for our innovative rocket. This flight profile depicts the 8 main mission stages starting from launch to landing and gives descriptions of the key stages powered ascent, apogee, retromotor burn, separation, and main parachute deployment.

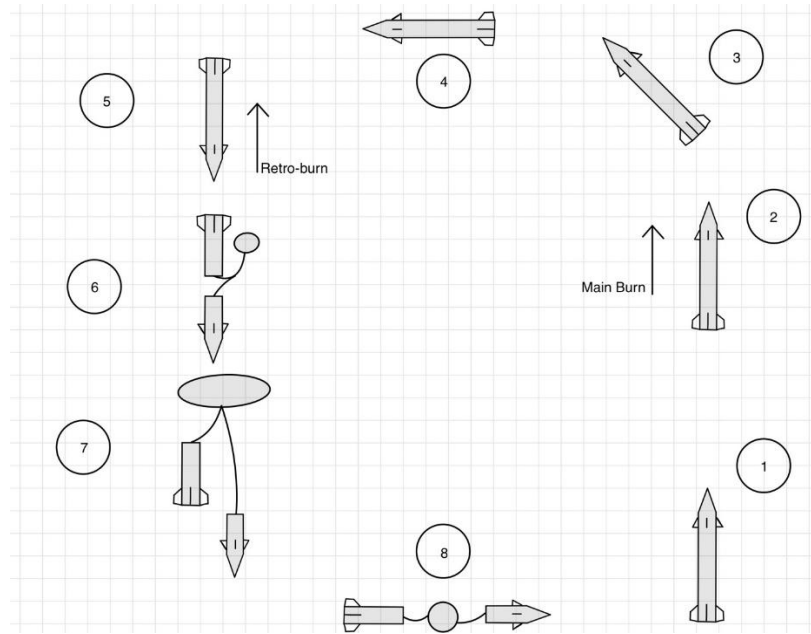


Figure 4 Flight Profile

Table 1.5 Flight Stages

<b>1</b>	Rocket on the launch pad.
<b>2</b>	Powered ascent of the rocket.
<b>3</b>	Unpowered ascent of the rocket and start of nose over.
<b>4</b>	Rocket has reached apogee and completed nose over beginning decent.
<b>5</b>	Retromotor burns when rocket meets ignition criteria.
<b>6</b>	Separation occurs separating the upper and lower airframe deploying the main parachute.
<b>7</b>	Parachute releases allow the main parachute to open. Then the upper and lower airframes descend under the main parachute.
<b>8</b>	The upper and lower airframes land with the main parachute.

Due to unforeseen delays, construction of the baseline rocket was unable to be completed. As a result, some designs may be incomplete or in a prototyping phase and the only section that discusses construction of the rocket is the construction of the electronics bay in Section 3.1.6.

## 1.5 Background and Literature Review

### 1.5.1 Airframe and Recovery System

#### *Rocket Construction*

High-powered rockets have three basic external components: the airframe, the nose cone, and the fins, as well as various other internal components discussed in other sections [1]. The airframe is the main body of the rocket that serves as the structure all other components mount to.



The airframe can be made from a wide variety of materials, ranging from cardboard on small model rockets to complex composites such as carbon fiber on larger rockets that experience greater forces. The airframe material chosen must have a high strength-to-weight ratio to minimize the mass of the rocket while retaining structural stability [1]. An example of a rocket airframe body tubes can be seen in Figure 5.



Figure 5 Fiberglass Body Tubes [2]. © Apogee Components Inc. 2021

The nose cone is the most forward portion of a rocket and serves primarily to reduce aerodynamic drag. The nose cone can also be used as payload space if the design of the rocket allows. Many different geometric forms exist for the profile of a nose cone such as conical, tangent ogive, and elliptical. The geometry of an ogive nose cone, one of the most common types, is described by a segment of a circle whose center is in line with the base of the nose cone, as shown in Figure 6. In this diagram, the tip of the nose is the bottom-left most corner of the gray shaded area and the base of the nose is length  $R$ . Length  $L$  is the centerline axis of the nose cone. For subsonic velocities (below Mach 0.8), shorter, blunted, elliptical shapes are the most effective options; however, once transonic or supersonic speeds are reached, other geometries such as Von Karman Ogive or Power Series geometries are typically the most efficient [1,3].

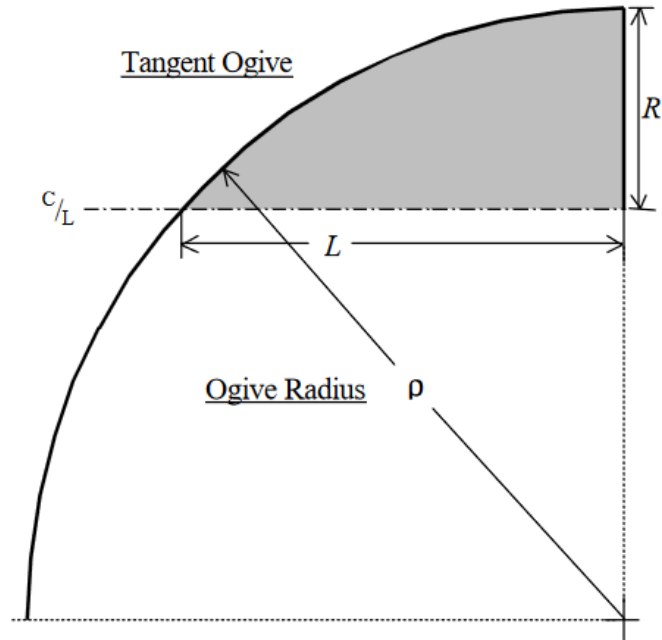


Figure 6 Tangent Ogive Geometric Relationships [3]. © Gary A. Crowell Sr. 1996

The fins are the third major external component of the rocket. They serve to stabilize the flight of the rocket so that it maintains a vertical orientation during ascent and does not spin out of control. Fins stabilize a rocket by moving the center of pressure (the mean point of all aerodynamic forces) towards the rear of the rocket. With the center of mass forward of the center of pressure the rocket is referred to as stable and will maintain a low angle of attack. Static stability of a model rocket is calculated as the ratio of the distance between the aerodynamic center of pressure of the rocket and its center of mass versus the diameter of the rocket. This ratio, often referred to as stability margin, provides a simple means of estimating the dynamic characteristics of a typical model rocket. A negative stability margin indicates the rocket is unstable and should not be flown. Rockets with a stability margin from 0-2 are considered under stable and will fly at large angles of attack. Rockets with a stability margin greater than 4 are considered over stable and will aggressively correct their trajectory in response to a disturbance. A stability margin between 2-3 is typically considered to be desirable. Fins are typically flat and fixed forms that are mounted at the rear of the airframe, although they can also be designed as airfoils or with geometric angles of attack depending on the needs of the rocket. The geometry of the fins can vary, but commonly fin geometry is either tapered or delta. Examples of these and other common fin geometries can be seen in Figure 7 [1].

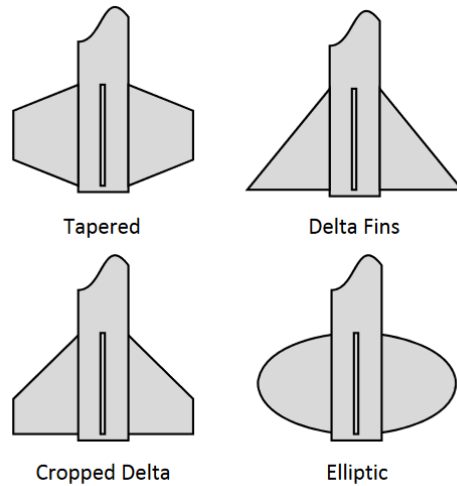


Figure 7 Common Fin Geometries [1]. © NASA 2015

### *Canards*

Canards are secondary fins that are mounted towards the nose of the airframe, typically on the upper section. While canards are used in many areas of aeronautics, such as on some designs of delta wing aircraft, their purpose in high powered rocketry is to enable the active control of a rocket, rather than provide the passive stability that fins provide. More detailed information on the active control of rockets using canards can be found in Section 3.1.5. It is important for canards to be as small as possible to prevent the center of pressure being shifted too far forward by the canards and disrupting the stability of the rocket, but still large enough to accomplish their control goals [4]. The size difference between the fins on the lower airframe and the canards on the upper airframe can be seen in Figure 8. The geometry of canards otherwise follows the same design principles (e.g., low drag) as the larger fins, as described in Section 2.1.2.



Figure 8 Rocket Model with Canards on Upper Airframe

An important distinction between fins and canards beyond location on the rocket is canard's ability to mechanically actuate. Mechanical actuation of the canards allows for aerodynamic forces to be varied, creating the torques required to change the attitude of the rocket. Control is achieved by adjusting the angle of attack of the canard. When coupled with actuation of

the other canards, this creates yawing, pitching, or rolling moments that control the attitude of the rocket.

### Parachutes

A parachute recovery system functions by deploying one or more parachutes at apogee or during the descent to slow the rocket down through aerodynamic drag to a speed that is safe for both the rocket and individuals on the ground. Outside of model rocketry, parachutes are a common means of recovery for space vehicles or other hardware in the realm of commercial or government rocketry.

A typical parachute recovery system for high powered rocketry consists of two parachute deployments, a drogue and a main, also called a Dual Event Recovery System (DERS) [1]. The drogue parachute controls the initial descent speed of the rocket and limits how far the rocket can drift. Once the rocket has descended to about 700 feet Above Ground Level (AGL), the larger main parachute is deployed to arrest the rapid descent even further to a speed that is safe for impact with the ground [1]. Figure 9 depicts the advantages of a DERS system over a single parachute system and Figure 10 depicts typical parachute location within the airframe to maintain an effective center of mass. In Figure 9, the DERS system is pictured on the right. The diameter of the drogue and main parachutes is governed by the mass of the rocket and desired descent velocities.

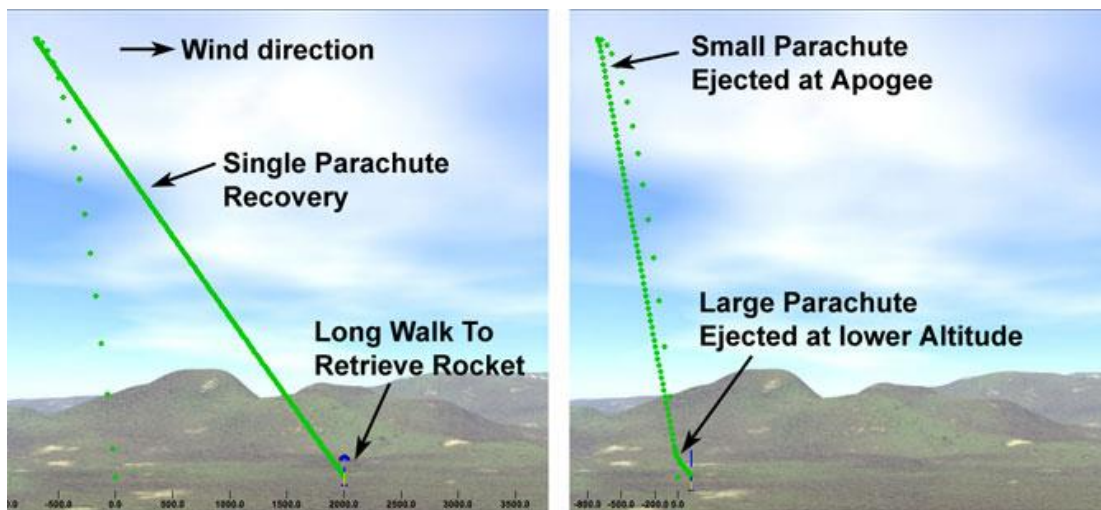


Figure 9 Plot of Single and Dual Event Recovery Profiles [5]. © Apogee Components Inc.

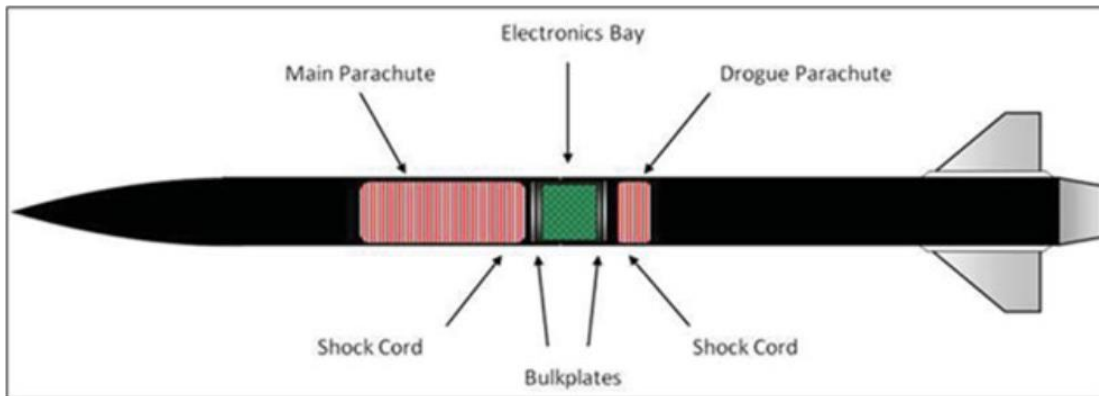


Figure 10 Main and Drogue Parachute Locations [1]. © NASA 2015

As shown in Figure 10, the main parachute is usually attached to the electronics bay bulkhead via shock cord. Shock cord is a material that is used to connect the drogue parachute, main parachute, and rocket airframe pieces together during the recovery stage of a flight. It helps absorb some of the force of parachute deployment while ensuring that the parachutes and rocket airframe stay together.

#### *Recovery Electronics and Hardware*

A typical high-power rocket dual deployment recovery system includes redundant altimeters. These altimeters are housed in an avionics bay in the airframe where the separation device is located, as shown previously in Figure 10. The inclusion of altimeters allows for the control device to initiate airframe separation and parachute release. As a result of having multiple parachutes, DERS rockets require several separation devices to deploy all parachutes, as seen in Figure 11.

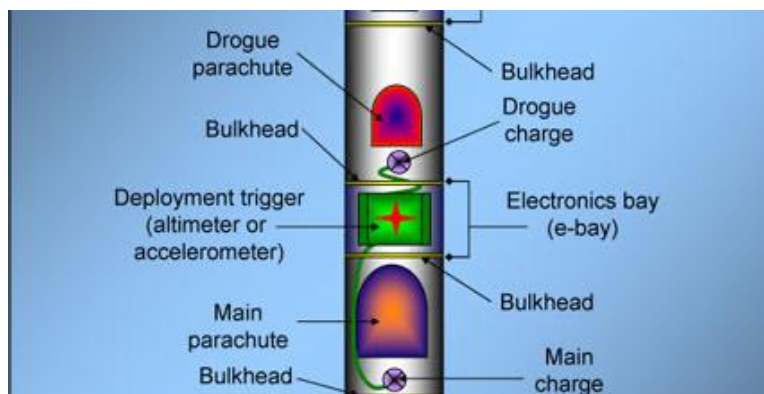


Figure 11 DERS Electronics and Separation Device Locations [6]. © Madison West Rocketry

An alternative method is to utilize a single separation event combined with a parachute release device. This method allows for there to be only one separation at apogee deploying the drogue parachute along with the main parachute still folded and retained by a parachute release device as shown in Figure 12. Parachute release devices allow for a parachute to be deployed from any altitude while still allowing for the parachute to be opened at a target attitude.



Figure 12 Jolly Logic Chute Release [7]. © Jolly Logic 2021

The device shown in Figure 12 is a Chute Release by Jolly Logic. These parachute releases work by being wrapped around the main parachute with elastic bands that are pinned into a spring-loaded mechanism that is actuated by a built-in altimeter. The built-in altimeter has a range of deployment altitudes from 100 feet to 1000 feet in 100-foot increments allowing for a desired altitude to be selected. At the targeted altitude, the parachute release devices actuate the spring-loaded mechanism releasing the pins retaining the elastic bands allowing the main parachute to open.

### *Powered Descent*

Powered descent is an alternative form of recovery system that utilizes retro-firing rocket motors or engines to decelerate a spacecraft or rocket booster before landing. Powered descent has most notably been used to initiate landing for the lunar module during the Apollo missions and is currently being used on reusable launch systems like SpaceX's Falcon boosters [8]. Reusable launch systems utilize powered descent during recovery of their rocket boosters to decelerate the booster upon reentry and initiate landing at a specific target location as seen in Figure 13.

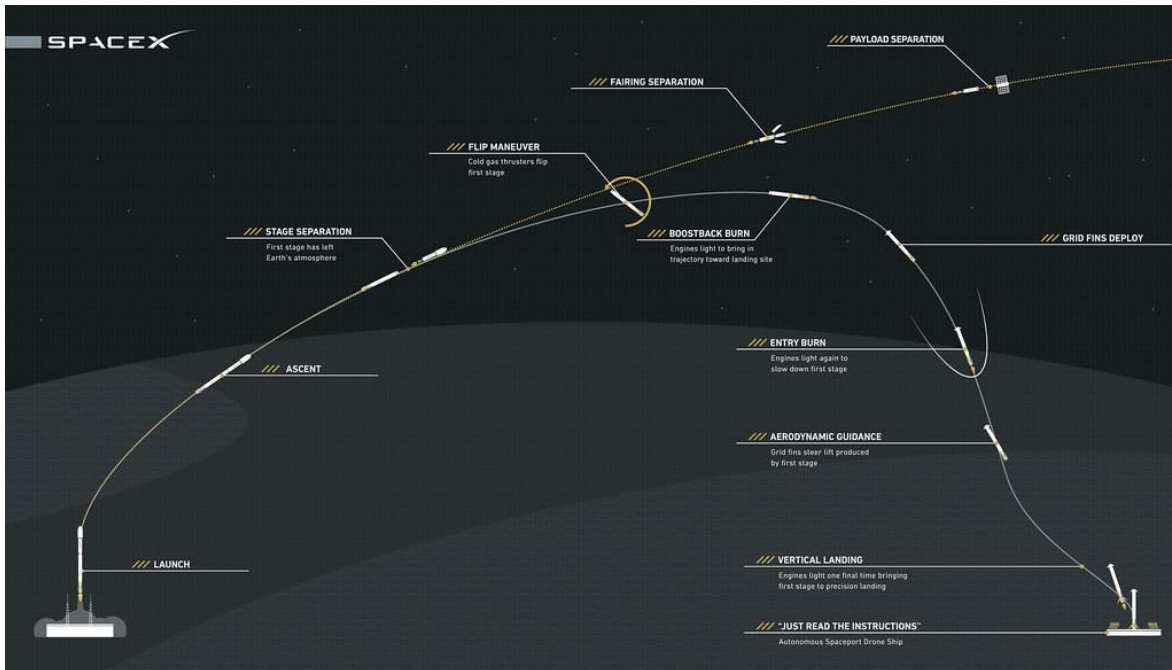


Figure 13 Falcon 9 Sample Mission Profile [9]. © SpaceX 2021

Utilization of powered descent in high-power rocketry is extremely uncommon due to the complexity and cost of a system needed to properly implement powered descent. These constraints make a “partial” powered descent system far more feasible in high-power rocketry. An example of a partial powered descent system would be a secondary motor or “retromotor” that replaces a drogue parachute. Like a drogue parachute, a retromotor provides an opposing force to that of the force of gravity on the rocket decelerating the rocket before main parachute deployment. This use of partial powered descent is like the entry burn seen in Figure 13.

### 1.5.2 Flight Dynamics Analysis

#### *Coordinate Systems and Transformations*

Aerospace simulations utilize a variety of coordinate frames to describe the position and orientation of a vehicle. These could be as simple as a flat plane, or as complex as modelling the motion of the solar system, depending on the fidelity of the simulation. Whatever form they take, the coordinate systems form a foundation for describing the forces and moments on, and movement of the vehicle. For vehicles operating from ground level up to orbit, it is common to utilize an Earth centered coordinate system as the inertial reference frame. The team has chosen to ignore the non-inertial effect caused by the Earth’s orbit around the sun due to the relatively short timescale for the flight of a model rocket or other earth-bound ballistic projectile.

To simulate flight on a rotating Earth the inertial frame is chosen with an origin at Earth’s center of mass and a fixed orientation relative to the stars. A common frame chosen for this reference is J2000, defined with its Z-axis towards the Earth’s north pole, it’s X-axis towards the



mean equinox at 12:00 Terrestrial Time on January 1<sup>st</sup>, 2000, and its Y-axis completing a right-handed coordinate frame along the equatorial plane as shown in Figure 14 [10].

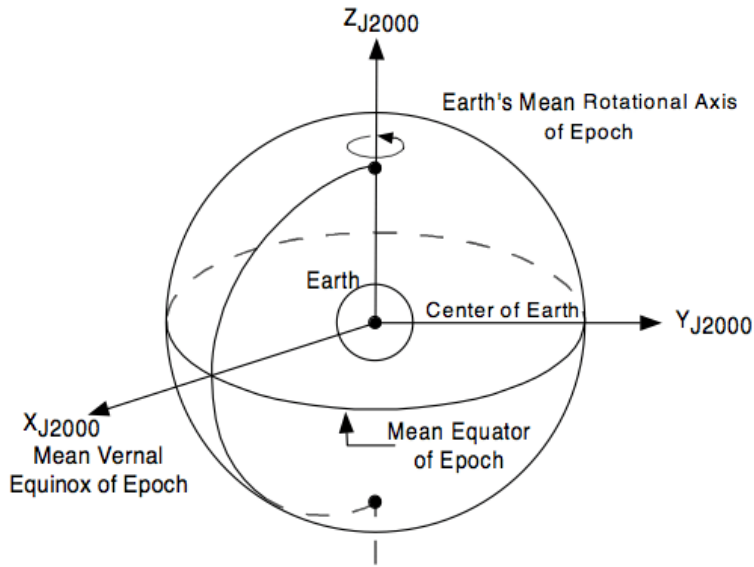


Figure 14 J2000 Coordinate Frame [11]. © NASA 1998

Earth-fixed coordinate frames rotate with the Earth as it spins and are useful for tracking the movement of an object over the Earth through time since a particular point on Earth will always maintain the same coordinates. An Earth-Centered-Earth-Fixed (ECEF) cartesian coordinate system has its origin at the center of mass of the earth, its Z-axis pointed to the north pole, and X-axis pointing to the Greenwich Meridian as shown in Figure 15 [12]. Another coordinate system with a similar purpose is the geodetic coordinate frame, consisting of a latitude, longitude, and altitude, also shown in Figure 15. Geodetic systems are used more often than ECEF coordinates to identify the locations of objects on or around earth, and the coordinates are also used to in gravity models and for weather forecasts. The most common geodetic system in use currently is the 1984 World Geodetic System (WGS84), which is the same used by GPS satellites.



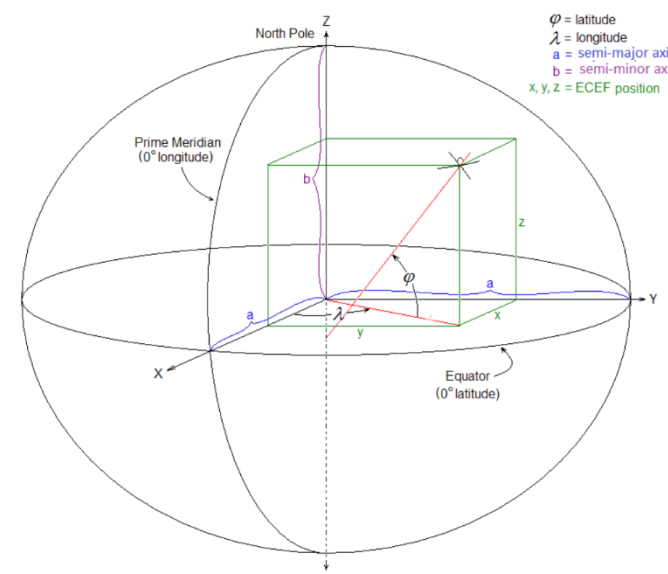


Figure 15 ECEF and Geodetic Coordinate Systems [13]. © Chuckage 2021

To describe motion nearby to a point relatively close to the Earth, a local tangent coordinate frame can be defined, representing the Earth's surface as a flat plane tangent to the surface of the ellipsoid. This flat-plane assumption is useful for easily understanding range and possible landing locations and is valid so long as the vehicle remains close to its starting location. The most common local tangent frame in aerospace applications is the North-East-Down frame, a right-handed Cartesian frame with its Z-axis pointed towards the center of the Earth, its X-axis pointed north, and its Y-axis pointed east as shown in Figure 16. This system can be defined on a point tangent to the ellipsoid or attached to a vehicle to describe relative positions of objects from the vehicle's perspective.

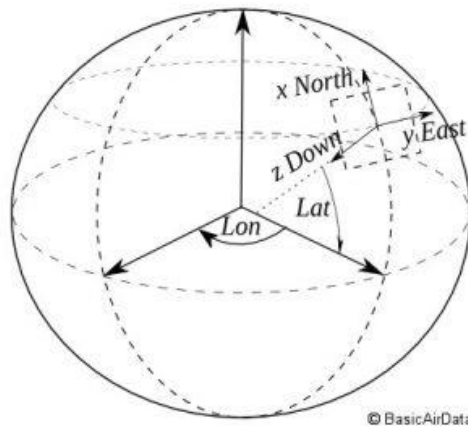


Figure 16 NED Coordinate System [14]. © BasicAirData 2011

Aside from earth-based coordinate systems, vehicle centered coordinate systems provide references for thrust and aerodynamic forces acting on the vehicle. It is common to define a body frame located at the center of gravity of the vehicle with its X-axis pointed towards the nose, Z-axis down relative to the vehicle's geometry, and Y-axis to the side [14]. Thrust forces can be defined in this frame as they generally do not vary with angle of attack. A wind frame can also be defined with an X-axis that points into the relative wind. The rotation of this frame compared to the body frame can be used to determine angle of attack and angle of sideslip, though the latter is mainly applicable to aircraft rather than longitudinally symmetric launch vehicles. Both frames are shown in Figure 17.

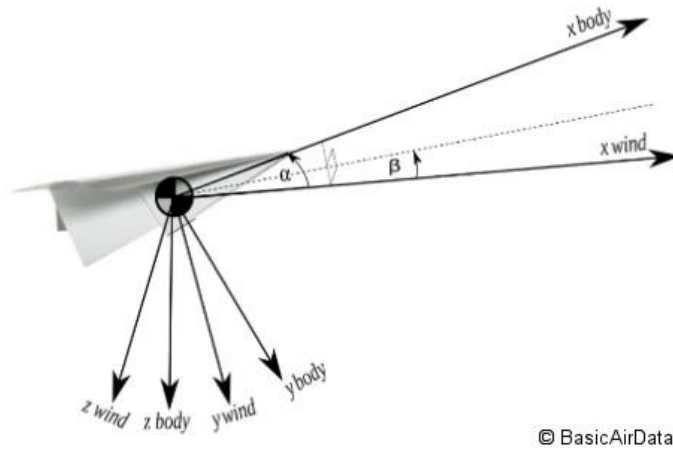


Figure 17 Body and Wind Frames [14]. © BasicAirData 2011

Rotations between each of the Cartesian frames can be accomplished via direction cosine matrices [10]. Transformations between coordinates in each Cartesian frame are similarly trivial, though transformations to geodetic coordinates are more difficult. A closed form solution exists to convert ECEF coordinates to geodetic coordinates, though a solution to convert the opposite direction generally requires iterative methods to compute accurately.

### *Vehicle Dynamics*

Three forces are generally considered to act upon a rocket in flight – aerodynamic loads, thrust, and gravity, shown in Figure 18 [15]. Thrust and gravity generally produce no moments since the thrust force is aligned with the center of gravity of the vehicle, and gravity acts on all parts of the vehicle equally. Aerodynamic loads on the other hand act at the center of pressure, so produce both forces on the vehicle along with moments which act on the vehicle differently.

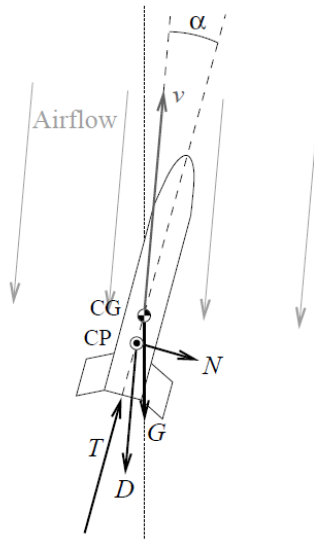


Figure 18 Flight Forces on Rocket [15]. © Sampo Niskanen 2011

The dynamics of a vehicle can be broken up into forces on the vehicle that change its position and moments that cause the vehicle to rotate. These problems are interlinked, because the attitude of the vehicle will influence the forces acting upon it, and these forces lead directly generate the moments.

Thrust and gravitational forces can be determined from manufacturer supplied motor thrust curves and component masses, respectively. Aerodynamic forces can be computed by determining aerodynamic coefficients based on the geometry of the vehicle. A typical set of coefficients that can be used to simulate 6-DOF flight for a rocket are the coefficients of drag and normal force and roll moment [16]. Pitch and yaw moment coefficients are not needed in a simple model because these moments arise from the normal force on the vehicle. More coefficients can be defined depending on the fidelity of the model, such as damping coefficients that account for the current angular velocity of the vehicle [15]. Forces and moments can then be calculated using the current dynamic pressure and a reference area.

### *Simulator Architecture*

Accurate prediction of the stability and flight characteristics of model rockets prior to a launch allows for detailed analyses of a vehicle's flight prior to reaching the launch pad. To generate this prediction, a numerical simulator can be created to determine a vehicle's trajectory based on the coordinate frames, forces, and moments described above. Forces and moments can be combined to create a state space representation of the vehicle, which is solved over small timesteps using numerical solvers, primarily Runge-Kutta methods or variations [16]. A simplified block diagram of a typical simulator architecture is shown in Figure 19.

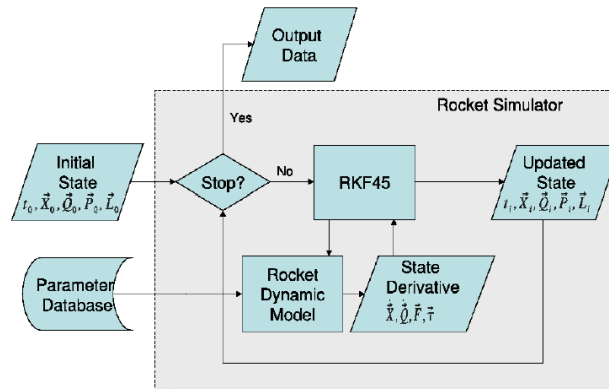


Figure 19 6-DOF Simulator Architecture [16]. © Journal of Aerospace Engineering 2010

The vehicle is given an initial state based on launch location and conditions. The simulator then proceeds using parameters such as the motor’s thrust curve, atmospheric and wind conditions, and location as inputs to the dynamic model [16]. The simulator proceeds by solving the equations of motion using numerical methods until an end condition is reached, generally either apogee or landing. Other end conditions could be modeled such as the start of a second stage motor, when the mass and aerodynamic properties of the vehicle would change, or at parachute deployment [15,16].

In addition to a single simulation, multiple simulations can be run using varying initial conditions, environmental conditions, and vehicle properties through Monte-Carlo simulations. The Monte-Carlo method allows uncertainty parameters to be assigned to different values such as the drag or thrust of the rocket, running multiple iterations of the simulation with these randomly varied values [16]. With enough simulations, probabilistic models of the vehicles flight can be determined which are particularly useful for estimating the vehicles landing site as shown in Figure 20.

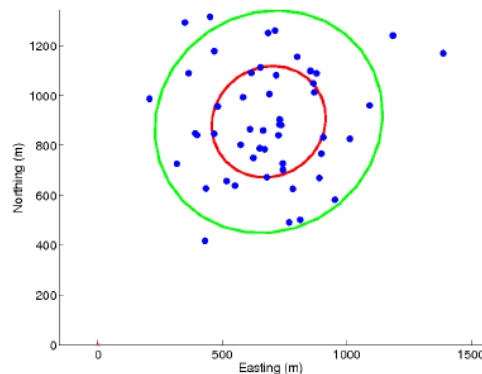


Figure 20 Monte-Carlo Simulation Output [16]. © Journal of Aerospace Engineering 2010

### Comparison of Existing Dynamic Simulators

Numerous dynamic simulators are currently available to the researchers and hobbyists alike that can provide this key information, including OpenRocket, RockSim, RockSim Pro, and RASAero II. Each varies in specific implementation, but all allow the ability to design a launch vehicle and determine static stability, input information such as the motor and recovery system, and then simulate the flight from launch to landing. The capabilities of some of the most common simulators are given in Table 1.6.

Table 1.6 Comparison of Existing Dynamic Simulators

	<b>OpenRocket</b> [15]	<b>RockSim</b> [17]	<b>RockSim Pro</b> [17]	<b>RASAero II</b> [18]
<b>Simulation Type</b>	6-DOF	3-DOF (x, z, $\theta$ )	6-DOF	3-DOF (x, z, $\theta$ )
<b>Range Model</b>	Rotating Ellipsoid Earth	Flat Plane	Rotating Ellipsoid Earth	Flat Plane
<b>Stability Model</b>	Extended Barrowman	Proprietary Extended Barrowman	Proprietary Extended Barrowman	Extended Barrowman
<b>Maximum Altitude</b>	80 km	86 km	632 km	Not Specified
<b>Maximum Velocity</b>	Mach 5	Mach 2	Mach 10	Mach 10

Additionally, previous rocket MQPs have produced simulators with varying capabilities. The Design, Analysis, and Test of a High-Powered Model Rocket-2 MQP in 2021 [19] produced the most capable simulator to date, successfully implementing a 6-DOF model, though they did encounter errors with determination of the aerodynamic coefficients. This simulator used a flat plane range model, reducing accuracy for rockets that fly higher and for longer.

### Aerodynamic Simulation

The aerodynamic characteristics of any rocket carry immense importance for both its design and performance. The way a rocket acts while moving through a fluid will determine its overall flight performance, which includes properties such as flight stability and behavior at apogee. While in flight, a rocket experiences various forces exerted on it by the atmosphere. As discussed in previous sections, the most basic forces acting on a rocket are thrust, drag, lift, and weight, which are referred to as governing forces [20]. Additionally, a rocket in flight is also subject to surface, pressure, viscous, unsteady, body, friction, centripetal, tangential, and Coriolis forces [21]. When considering aerodynamic analysis, many of these forces can be neglected due

to various assumptions made regarding the fluid flow around the rocket. For example, if flow is assumed to be steady and inviscid, then the unsteady and viscous forces will not be considered in the analysis. In terms of real-world application, however, these assumptions are not always true to the physical condition of the flow and neglecting these forces may reduce the accuracy of the analysis [22]. The validity of these assumptions is a key factor in selecting a method for performing aerodynamic analysis.

Many methods exist for carrying out this analysis, and they each come with their own benefits and drawbacks. One method would be to compute results analytically through simplified use of the Navier-Stokes equations, shown in Equations (1)-(5). However, to carry out a closed-form analytical calculation, multiple assumptions and simplifications need to be made. This inherently reduces the accuracy of the results by providing idealized solutions that do not serve as a good representation of the real-world system. A significantly more accurate method of aerodynamic analysis is to perform wind tunnel testing. The stability and flight behavior of the rocket can be predicted by analyzing the behavior of a scale model under various conditions in the wind tunnel. As an empirical method, this would provide results accurate to the real-world system at the expense of being significantly more time consuming than other methods. This issue was encountered by the WARRIORS I project, which had planned to conduct wind tunnel testing as part of its aerodynamic analysis but was unable to collect data due to time constraints [24].

Continuity:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} + \frac{\partial(\rho w)}{\partial z} = 0 \quad (1)$$

X-Momentum:

$$\frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u^2)}{\partial x} + \frac{\partial(\rho uv)}{\partial y} + \frac{\partial(\rho uw)}{\partial z} = -\frac{\partial P}{\partial x} + \frac{1}{Re} \left[ \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \tau_{xz}}{\partial z} \right] \quad (2)$$

Y-Momentum:

$$\frac{\partial(\rho v)}{\partial t} + \frac{\partial(\rho uv)}{\partial x} + \frac{\partial(\rho v^2)}{\partial y} + \frac{\partial(\rho vw)}{\partial z} = -\frac{\partial P}{\partial y} + \frac{1}{Re} \left[ \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \tau_{yy}}{\partial y} + \frac{\partial \tau_{yz}}{\partial z} \right] \quad (3)$$

Z-Momentum:

$$\frac{\partial(\rho w)}{\partial t} + \frac{\partial(\rho uw)}{\partial x} + \frac{\partial(\rho vw)}{\partial y} + \frac{\partial(\rho w^2)}{\partial z} = -\frac{\partial P}{\partial z} + \frac{1}{Re} \left[ \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \tau_{zz}}{\partial z} \right] \quad (4)$$

Energy:

$$\begin{aligned}
 & \frac{\partial(E_t)}{\partial t} + \frac{\partial(uE_t)}{\partial x} + \frac{\partial(vE_t)}{\partial y} + \frac{\partial(wE_t)}{\partial z} \\
 &= -\frac{\partial(uP)}{\partial x} - \frac{\partial(vP)}{\partial y} - \frac{\partial(wP)}{\partial z} - \frac{1}{RePr} \left[ \frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} + \frac{\partial q_z}{\partial z} \right] \\
 &+ \frac{1}{Re} \left[ \frac{\partial}{\partial x} (u\tau_{xx} + v\tau_{xy} + w\tau_{xz}) + \frac{\partial}{\partial y} (u\tau_{xy} + v\tau_{yy} + w\tau_{yz}) \right. \\
 &\left. + \frac{\partial}{\partial z} (u\tau_{xz} + v\tau_{yz} + w\tau_{zz}) \right]
 \end{aligned} \tag{5}$$

Table 1.7 Variables for General Form of Navier-Stokes Equations [23]

Position	Velocity	Time	Density	Total Energy	Heat Flux	Pressure	Stress	Reynolds Number	Prandtl Number
$x, y, z$	$u, v, w$	$t$	$\rho$	$E_t$	$q$	$P$	$\tau$	$Re$	$Pr$
m	m/s	s	kg/m <sup>3</sup>	J	W/m <sup>2</sup>	Pa	Pa		

A more efficient method of aerodynamic analysis is computational fluid dynamics (CFD), which uses numerical methods to analyze fluid flows. The Navier-Stokes equations, which stem from mass, momentum, and energy conservation principles, serve as the foundation for CFD [25]. CFD requires various inputs for the analysis to be successful. Some inputs, such as the model geometry and the fluid/modeling domain, can be obtained through a computer aided design (CAD) model of the object. Additionally, boundary conditions of the flow must also be provided [26]. Once these conditions have been provided, a mesh is created to discretize the model prior to running the analysis. Meshing is a crucial part of the CFD process, as it defines elements within the domain that will be analyzed linearly [27]. After the mesh is created, the program will run iteratively until the solutions converge, providing the results to the aerodynamic analysis. A poor mesh will result in less accurate solutions, or it may prevent the solutions from converging at all. In that case, a new mesh must be created, and the iterative process is run again.

While different methods of aerodynamic analysis have been used by past MQP teams, CFD simulations have proven to be the most effective [19]. CFD provides similar accuracy to wind tunnel testing while requiring significantly less time and setup to complete. In previous years, MQP teams have used various software tools, such as SimScale and Ansys Fluent, to perform CFD analysis. When comparing these two programs, SimScale appears to be a more simplified general simulation software that is easier to learn and operate while Ansys Fluent provides more in-depth analysis through its ability to model three-dimensional fluid flows [28].

## Controls

A control system, whether passive or active, is required to design a stable rocket and maintain this stability throughout the duration of its flight. Passive controls are fixed devices that provide stability due to their presence on the rocket's exterior, the most popular being fins, as discussed in Section 1.5.1 [29]. Active controls can be moved during the flight of the rocket not only to provide stability, but also the ability to steer the launch vehicle. Active control systems can include movable fins, canards, vanes, gimbaled nozzles, vernier rockets, fuel injection, and attitude-control rockets. Canards are a type of actuated fin, the main difference being their location on the rocket, with canards being mounted on the forward end, and fins at the aft. Canards inherently cause instability due to their placement, moving the center of pressure closer to the center of gravity. Despite this, they are commonly used on missiles and airplanes to increase maneuverability and are a source of overall active stability [30]. During flight, the canards are tilted to deflect air flow, causing the rocket to change direction or adapt to undesired directional changes.

Active control systems, such as canards, require the development of control laws to provide the desired movement for guidance. The Proportional, Integral, and Derivative (PID) controller is one of the more common forms. PID controllers compute the proportional, integral, and derivative of the differences, or what is referred to as the error, between the output and the reference input. This controller then outputs a control signal dependent on the PID coefficients,  $K_p$ , the proportional coefficient,  $K_i$ , the integral coefficient, and  $K_d$ , the derivative coefficient. The input-output relationship is defined below in Equation (6) relating the coefficients to the error  $e(t)$  [29].

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (6)$$

The proportional term,  $K_p$ , focuses on the present or instantaneous error. Essentially, it is trying to steer the controlled vehicle back in the direction it deviated from. The integral term,  $K_i$ , allows for the elimination of steady state errors, essentially all the previous errors that have occurred prior to the present. The derivative term,  $K_d$ , is the best estimate of the future error. This is also known as the damping term, since as the more rapid change the term predicts, the greater damping effect. In principle, a sole derivative controller cannot be implemented because it only anticipates future response needs. PID controllers function as a Single-Input Single-Output (SISO) system, where the system uses a single variable and is limited to one output and one input. This is not always desired for complex problems, such as controlling rocket attitude, which typically become Multiple-Input Multiple-Output (MIMO) problems [31].

The Linear Quadratic Regulator (LQR) controller is a control system that functions as a MIMO, which requires a linear system. LQR is based off two matrices, Q and R, representing the difference, or cost, given to the distance between the state and target and the penalty paid to execute



the actions, respectively. For example, the cost of the controller is typically a deviation of a key measurements, such as altitude. The price paid, for example the time it takes, to execute this small deviation from the desired altitude would be the penalty. These are then used to compute the control matrix,  $K$ , which is then multiplied by the current state output vector to produce the optimal control input vector as seen in Figure 21.

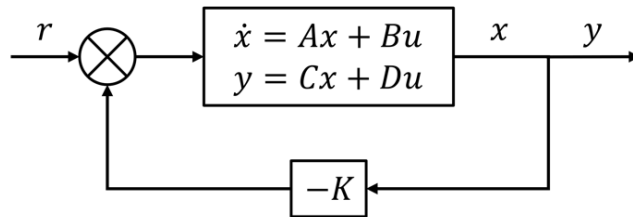


Figure 21 Simplified Block Diagram of an LQR Controller [32] © IEEE 2019

Another type of control system is Model Predictive Control (MPC). This control system is like LQR, but instead includes not only the current timeslot in its optimization, but also accounts for future timeslots. This allows the user to find an optimal solution for both the present and future state at the same time, as seen in Figure 22 [31].

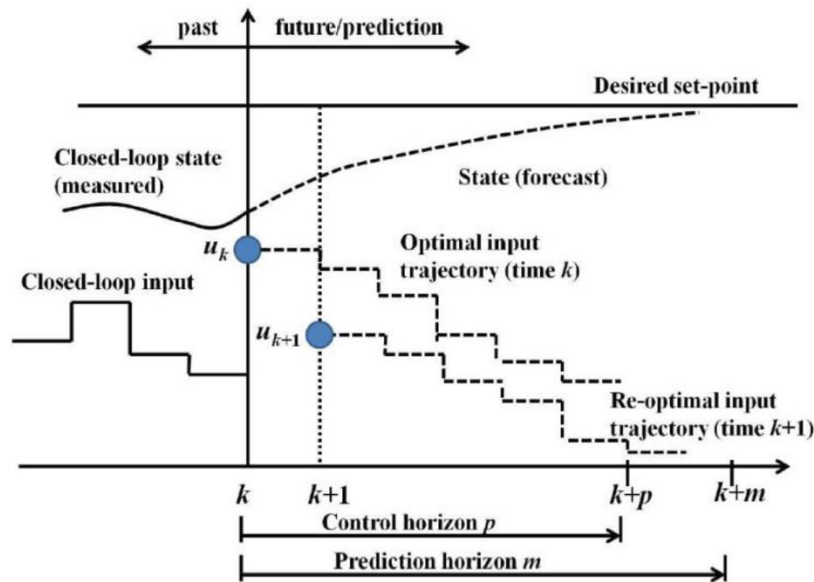


Figure 22 Future State and Control Input for MPC [31] © University of Edinburgh 2017

Active control has been attempted by a few previous MQPs. Most recently in 2021, the Design, Analysis, and Test of a High-Powered Model Rocket-2 MQP [19] designed an actively controlled set of grid fins for stabilization. The control theory used was 3 PID's, one for each rotational axis, the goal being to stabilize roll followed by yaw and then pitch at same time. Prior to this in 2019, the Design and Integration of a High-Powered Model Rocket-II MPQ [33] designed actuated flap fins. They attempted several different methods for control law including LQR and a non-linear estimation of the K matrix. Unfortunately, neither team was able to develop a control law that met their design needs or standards. For the team's purposes, a non-linear PID control system was pursued.

### 1.5.3 Propulsion, Thermal, and Separation Systems

#### *Propulsion and Motors*

The motor is the component of a rocket that is responsible for propulsion and a change in the rocket's momentum. The purpose of a rocket motor is to exert a thrust force downward out of the aft end of the rocket. Combustion in the motor occurs because of the interaction between the fuel and the oxidizer. Rocket motors are usually classified into two main categories, liquid (generally referred to as an engine) and solid, although other types of motors do exist. Liquid engines keep the oxidizer and fuel stored separately as liquids. They are then pumped into the combustion chamber at the correct ratio to be ignited and exhausted [1]. To accomplish combustion, liquid engines are far more mechanically complex than solid motors. Although more complicated, these engines allow for a throttle change, and may even be stopped and restarted during flight. Most model rockets, however, use a solid rocket motor. Within a solid rocket motor, a solid propellant is packed into the motor casing, such as black powder or a composite [1]. This propellant is composed of a mixture of the fuel and oxidizer so that all the fuel is stored within the motor casing [1]. Black powder motors are easy to obtain, relatively inexpensive, and ignite with ease [34]. Although more expensive, composite motors have a higher energy density than black powder, therefore, to obtain the same amount of power, a composite motor would be smaller than a black powder motor. Thus, a smaller sized rocket may still use a more powerful motor.

At launch, the solid rocket motor is ignited, and the propellant's surface within the motor casing starts to burn until all the propellant is used [35]. As the propellant burns, hot exhaust gas is ejected out of the motor's nozzle and produces thrust. Composite motors may have multiple grains of propellant stored within them. These grains can have different patterns, such as formed, cored or sawn. These patterns allow for various amounts of area that the flame surface acts over while the motor is firing. This is useful because the propellant surface in which the flame burns affect the thrust produced by the motor. For example, as a motor with a simple hole down the center burns, the propellants surface area would increase over the time of burn. This would result in an increase of thrust over the duration of its burn, where as a C-slot has a slot cut down one side of the grains and results in a lower thrust over a longer period for the same amount of propellant.

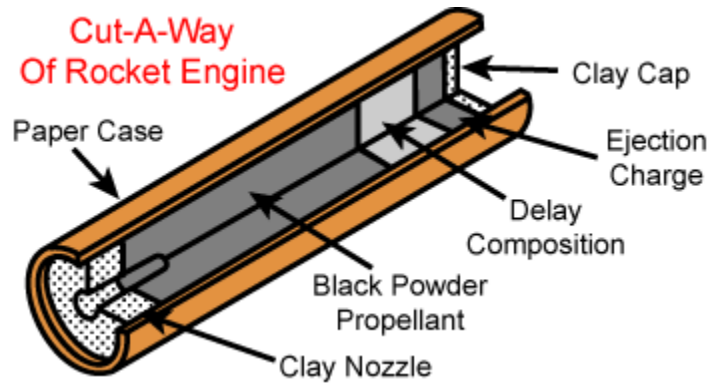


Figure 23 Cutaway of Model Rocket Motor [34] © Apogee Components 2019

Model rocket motors are commercially sold and are regulated by the National Association of Rocketry (NAR). These motors are designated by a 3-part code. First, a letter is given to specify the total impulse range of the motor, which is a measure of its total energy. Then, a number is used to designate the average thrust of the motor in units of Newtons, which is how quickly its energy is delivered. Finally, the last number is used to represent the time delay of the rocket, which is how many seconds it takes for the ejection charge to activate after motor burnout [35]. For motors without ejection charges, the delay is represented with a ‘P’ for plugged.

### *Ignition Systems*

Motor operation for high-powered rocketry requires an ignitor to generate the heat necessary to ignite the propellant inside the rocket motor [36]. For a rocket to launch, a battery provides current that heats up an ignitor, and once the ignitor reaches the ignition temperature of the motor, the combustion reaction starts. This ignition period is typically short if the system is built correctly. Different types of ignitors used in model rocketry, such as pyrogens, low-current systems, ignition powders, and other compounds for composite propellants [37,38].

The set-up for ignition systems changes very little depending on the type of ignitor used. Pyrogens are extremely flammable substances, and when this system is used, pyrogen coated wires are connected to lead wires. Then, current flows from the battery through these lead wires, heating the pyrogen. As the pyrogen heats up, pieces of burning pyrogen are sent flying into the rocket motor in order to ignite it [37]. Another type of ignition system is an electric ignition system, also referred to as an electric match, or e-match. In this system, the battery is connected to the electric match through two switches [38]. The electric match system is often used instead of the pyrogen, due to it being a safer system [37]. Although the electric match still requires the use of pyrogen, only the ends are dipped rather than the wires being coated. Similar to the process for the pyrogen-based ignition system, once the wires become hot enough, energy dissipates between the battery and the ignitors, and the burning process begins [39]. The electric match results in more localized heating making it easier for a separation system rocket motor to function accurately.

The wires of an ignition system can be made from different materials, depending on the requirements, but nichrome is often used because it has a high electric resistance [36,37]. When choosing the size of wire to be used, wire diameter and length must be able to provide enough electrical resistance to then produce enough heat for ignition. For a pyrogen system to assure accurate operation, the wires' temperature of ignition should be much greater than that of the pyrogen ignition point. Per NAR guidelines, all people must be at least 30 feet away from the launch pad, leading to ignition systems being designed to be controlled by switches which are located near the people launching the rocket. In this system, the battery is close to the ignitor and does not require the current to travel far from the battery. Although long wires are still required to transport the signal from the switch to the launch system, they do not need to be extra durable because it is not the current from the battery that is traveling through them [37].

### *Motor Mounting System*

The motor mounting system allows for the motor to be secured during flight, and it also aligns the motor with the rocket's airframe [37]. Assuring the motor is properly secured helps ensure its movement will not damage the inside of the rocket which could affect the deployment of the recovery system, as well as help ensure its stability [19]. The four main components of a motor mounting system are the motor mounting tube, centering rings, thrust mount, and motor retainer as seen in Figure 24. The motor, with a diameter slightly smaller than the motor mount tube, is inserted inside of the tube. The thrust mount prevents the motor from moving forward in the tube when ignited and transfers the motor thrust into the motor mounting system and from there the rocket. The motor mounting tube has a smaller diameter than the tube of the airframe allowing for it to fit inside. The outer diameter of the centering ring is equal to the inner diameter of the airframe tube as seen in Figure 24. The inner diameters of the rings are equal to the diameter of the motor to ensure they stay in place, as well as ensuring the motor remains aligned with the central axis of the main airframe.

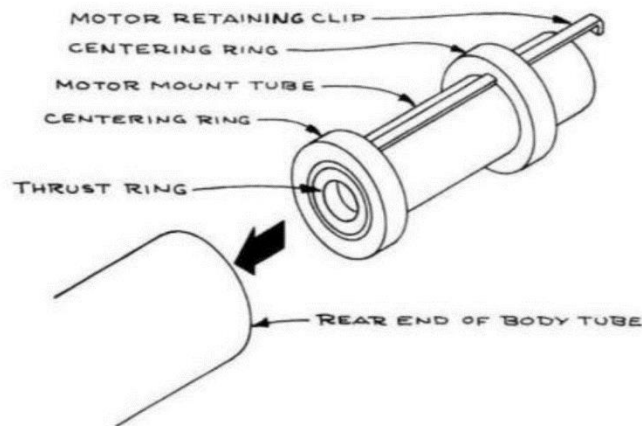


Figure 24 Diagram of Motor Mounting System [40] © George Stine 2004

## Separation

The separation of high-powered rockets, also known as ejection, is the event in which the recovery system is deployed. The most common method for separation in model rocketry is to use black powder charges that generate hot gases that pressurize the internal volume of the rocket's airframe, exerting a net force that will eject the nose cone, shock cord, and parachute from the body of the rocket [1]. Using black powder does have some drawbacks, such as creating residue on the rocket, risk of burning/melting the recovery system, decreased reliability at altitudes above 20,000 feet and unsustainable burns over 50,000 feet, along with the byproducts being corrosive and acidic [1,41]. Rockets that have additional flammable or energetic components on board can pose concerns related to how close the black powder charges are to these components. This could possibly create a dangerous event upon igniting the black powder for separation and is not a risk that should be taken lightly.

An alternative method is using CO<sub>2</sub> for separation. Compressed CO<sub>2</sub> is released from a canister to pressurize the body of the rocket with cold gas, rather than hot gas, and in similar fashion creates a net force that can eject the nose cone and release the recovery system [39]. CO<sub>2</sub> kits can be commercially bought, such as the Peregrine Exhaustless CO<sub>2</sub> Ejection System by Tinder Rocketry, which includes 8- and 12-gram CO<sub>2</sub> cartridges, cartridge housings, an opening pin, alignment collars, charge cups, O-rings, housing caps, lift pistons, return springs, pyrotechnic measuring vials, and mounting hardware [41]. An exploded view of their CO<sub>2</sub> system can be seen below in Figure 25.

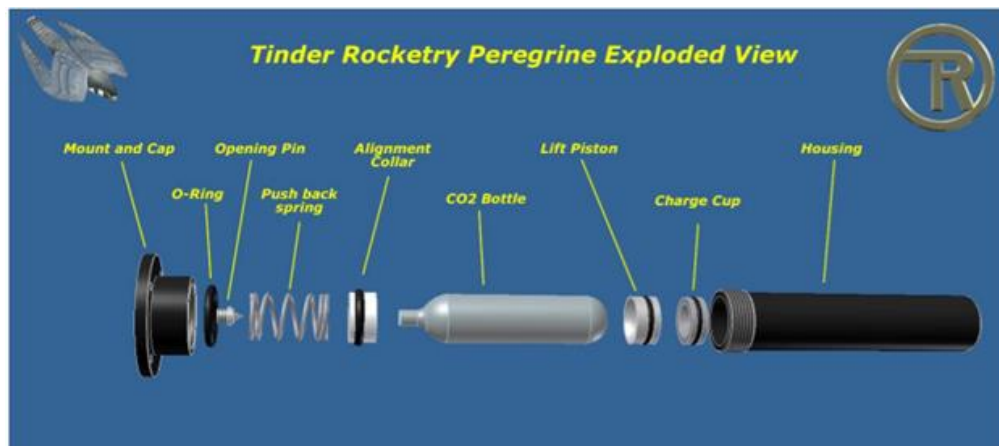


Figure 25 Tinder Rocketry Peregrine Exploded View [41] © Tinder Rocketry

Using a CO<sub>2</sub> system for separation has several advantages over black powder. One of these is that CO<sub>2</sub> is exhaustless and clean, which means that there is no heat or residue that can affect the rocket and flame retardant cloths are not needed to protect the parachute from melting [41]. In addition, Tinder Rocketry has tested their Exhaustless Peregrine System in a simulation that

showed reliability in being used up 80,000 feet in altitude. CO<sub>2</sub> systems are reusable, like black powder systems, and CO<sub>2</sub> canisters can be easily replaced after being discharged and leave the rest of the setup intact.

## 2 Airframe and Recovery System Methodology and Results

### 2.1 Methodology

#### 2.1.1 Airframe

The airframe of the rocket contains all other component systems. Figure 26 depicts the rocket in its entirety. The lower airframe is 24 inches long and contains the motor mount, fins, and is where the recovery bay in the middle of the rocket is adhered to. The upper airframe is 29 inches long and contains the canards, parachute bay, electronics/payload bay, and the canard actuation system. The electronics bay is detailed in Section 3.1.1 and the motor mount is detailed in Section 2.1.6.

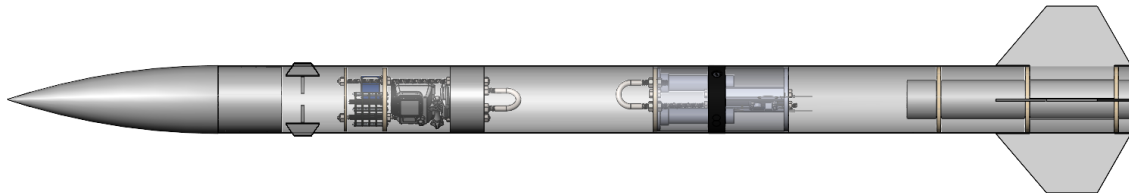


Figure 26 Full Rocket Assembly

The team designed the airframe itself around 4-inch diameter G12 fiberglass tube. The team chose this size to keep the airframe manageable for the team and budget while also allowing the airframe to contain larger components such as a camera payload. The upper and lower airframes are separated by a 1-inch section of the recovery bay, resulting in a total airframe length of 54 inches. With the nose cone included, the total length is 65.75 inches. The exterior of the airframe without the nose cone can be seen in Figure 27. The recovery bay is detailed in Section 4.1.4. The upper airframe is fixed to the recovery bay by way of four shear pins that are designed to break at a certain applied load. This allows the recovery system to separate the upper and lower airframes when it is necessary to deploy the parachute while maintaining a rigid body throughout the rest of the flight.

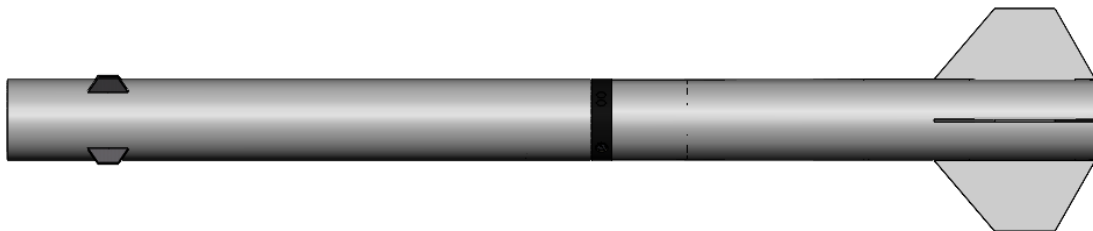


Figure 27 Rocket Airframe

The team chose 4-inch G12 fiberglass tubing due to its high strength-to-weight ratio as well as the fact that it is readily available. The specific tubing chosen was purchased from Madcow Rocketry and has an outside diameter of 4.024 inches and an inside diameter of 3.900 inches. It weighs 12.80 ounces per linear foot and has both 30° and 45° wind angles. The wind angle of the tube describes the angle relative to the longitudinal axis that the fiberglass fiber is wound at. As an example, if the fiber was straight up the length of the tube it would have a wind angle of 0°. The airframe couplers were made from the same material as the airframe and purchased from Madcow Rocketry. They have an outer diameter of 3.899 inches and an inner diameter of 3.755 inches. The airframe tubing cost \$125 for a 60-inch segment and the coupler cost \$48 for two 9-inch segments [42]. Assembling the outer airframe consists of cutting the airframe and coupler fiberglass tubing to length as well as cutting slots in the lower for mounting the fins and shear pins, a camera window into the upper and drilling holes in the upper to secure the nose cone with bolts.

### 2.1.2 Fins

The team designed the rocket with a set of four trapezoidal fins. The fins are not shaped as airfoils due to the active control system of canards and are instead flat plates that are not cambered in any way. Because the fins are only 0.125" thick, they are not rounded on the leading edge as the extra drag produced by a square edge is minimal. Normally, shaping fins as asymmetric airfoils is done to induce spin; however, a stable platform for capturing video of the launch from the onboard camera was desired. The fins were designed with a root chord of 8 inches, a tip chord of 3 inches, a height of 3.5 inches, a sweep length of 3 inches, and a sweep angle of 40.6°. The fins, shown in Figure 28, were manufactured from 3-mm fiberglass sheets to simplify the fabrication. This material can be easily milled on a CNC machine and has a high strength-to-weight ratio. These fins were fabricated by the company SendCutSend, as milling fiberglass sheets is not offered at WPI facilities due to the dust produced in the process [43]. The fin dimensions were chosen to produce a center of pressure that was towards the motor and behind the center of mass by at least a factor of twice the diameter of the rocket. This ratio is generally considered a safe margin for stability within the rocketry community. With the rocket's mass and fin size, it had a stability factor of approximately 2.4 times the diameter of the rocket. Stability factor is described in Section 1.5.1. The geometry of the fins was taken from the CAD model of these parts is shown in Figure 28 and Figure 29. This fin geometry was created in OpenRocket because that software had the ability to quickly find the stability factor provided by a given fin geometry.



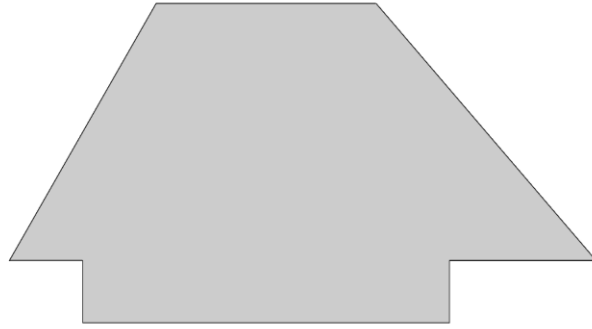


Figure 28 Fin Side View



Figure 29 Fin Front View

The fins were mounted to the airframe by slotting them through and then epoxying them to the motor mount assembly at the rear of the rocket. Slotting the fins through the airframe involved cutting a slot into the airframe for each fin that is large enough for the tab on the base of the fin to fit through. The fin tabs were epoxyed to the motor tube as well as the airframe itself. Originally, the fins were intended to slot up through the rear of the rocket to make them easily removable, but due to the nature of the filament-wound fiberglass, cutting perpendicular to and through the edges would compromise the structural integrity of the material. Running the fins through the airframe and mounting them inside the rocket also meant that the fins had an extra point of contact with the airframe and were therefore mounted more reliably and had an increased moment arm over which the forces could be reduced.

### 2.1.3 Retromotor Nose Cone

The team designed the retromotor system to take the place of a conventional drogue parachute during the recovery of the rocket. As discussed in Section 1.5.1 the retromotor works by imparting a thrust opposite the motion of the rocket at a point during the descent, slowing the rocket to a speed that is safe for the deployment of the main parachute.

The ARS subteam decided to house the retromotor within the nose cone of the rocket and to fire out through the tip of the nose cone. The PTSS subteam selected a Cesaroni P-38-4G I470 motor as the retromotor. This motor has a 38-milimeter casing and fit well within the geometry of the 3:1 Ogive nose cone. The retromotor nose cone assembly was custom designed and built rather than modifying an off-the-shelf nose cone as the cuts that would be necessary to wire the igniter to the motor would compromise the structural integrity of the nose. It also was built as two clamshell halves to simplify assembly, as seen in Figure 30.

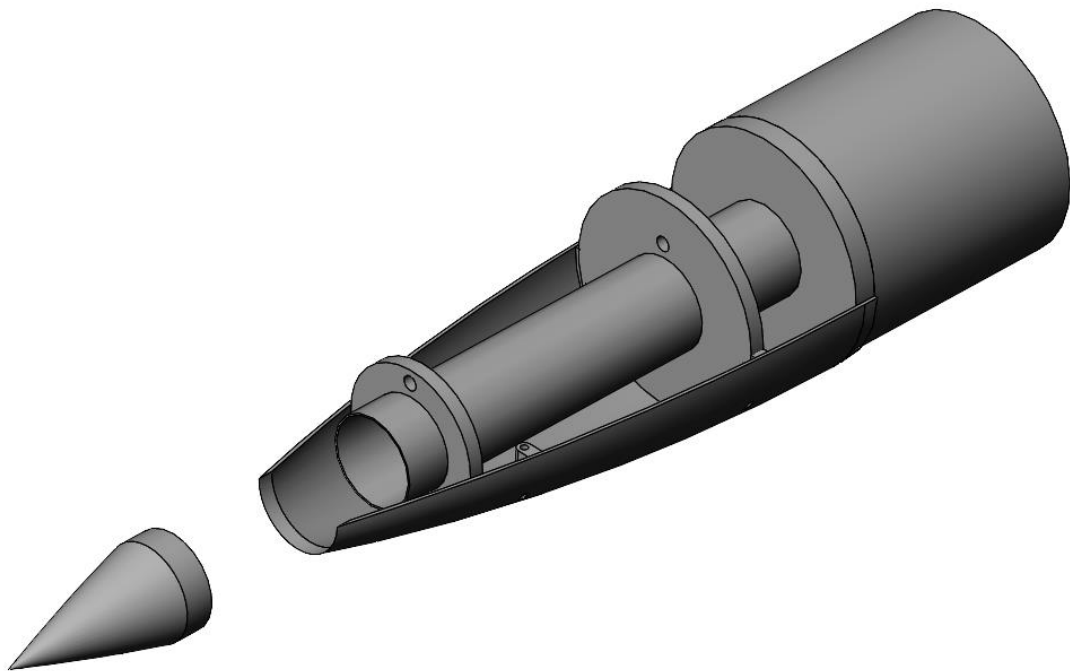


Figure 30 Retromotor Nose Cone Half

The nose cone tip is a separate piece that was designed to be blown off when the motor ignites. During the ascent phase of the flight, the tip shields the motor and igniters from disruption by the airflow as well as keeping the nose cone more aerodynamic. This nose cone tip is friction

locked in place by a short shoulder, so the pressure and thrust generated by motor ignition would easily eject it off the nose cone and allow the retromotor to function as intended.

The two clamshell halves of the nose cone are secured with through-bolts that are run through reinforced channels, as can be seen in Figure 30. The holes these bolts run through are then plugged from the outside to maintain a uniform exterior geometry and minimize the drag of the nose cone.

The motor itself rests within a solid fiberglass tube that is mounted within one of the two halves. This break in symmetry was done to remove potential points of failure. The applied thrust could potentially break a poorly made thrust ring and applying the thrust all around the edge of a uniform fiberglass tube, as with the ascent motor, solves this issue. The thrust ring is the part that transfer the thrust of the motor to the body of the rocket.

Due to safety concerns, the team decided not to flight test the retromotor nose cone assembly. Failure of this system would mean that the rocket would present a severe hazard to those on the launch range and such a recovery system does not have the decades of demonstrated reliability that traditional parachute recovery does. To that end, the rocket will be flown with a 3:1 ogive nose cone and use a conventional drogue/main parachute setup for the actual rocket launch. The drogue parachute is discussed in Section 2.1.4. The mass of the flight nose cone will be adjusted with pieces of scrap steel to match that of the retromotor nose cone to maintain the same weight and balance that had gone into the design consideration. The nose cone that the rocket will be flown with is a Madcow Rocketry 3:1 Ogive Filament Wound Fiberglass Nose Cone with a composite tip.

#### 2.1.4 Recovery Parachutes

Our rocket is designed to fly with one of two different parachute configurations depending which recovery system is being utilized. Parachute sizing was determined using simulations performed with the rocket's OpenRocket model where a ground hit velocity under 15 feet per second was selected as a constraint.

The first parachute configuration is for use with the innovative recovery system, or "retromotor" as described in Section 1.5.1. With the retromotor serving the same function as a drogue parachute the rocket will fly with only the main parachute installed. The main parachute will be housed in the parachute bay seen in Figure 31 and attached to U-bolts on both the avionics bay bulkhead and the recovery bay bulkhead using 1/4-inch nylon shock cord and hardware. The main parachute will be a 60-inch diameter hemispherical parachute from Spherachutes.

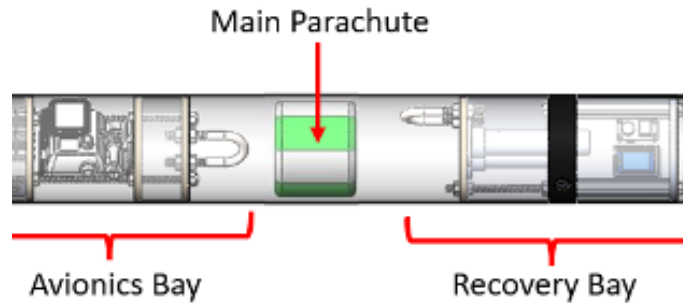


Figure 31 Innovative Rocket Parachute Configuration

In the innovative rocket configuration airframe separation will occur shortly after retromotor burn out. During this separation the main parachute will then be deployed from the rocket. After deployment of the main parachute two Jolly Logic chute releases which are installed on the main parachute for redundancy as pictured in Figure 32 will release allowing the main parachute to open at the intended altitude of 600 ft. A full flight profile can be seen in Figure 4 depicting this deployment. Also, for more information on the Jolly Logic chute release refer to Section 1.5.1.



Figure 32 Jolly Logic Chute Release [7]. © Jolly Logic 2021

The second parachute configuration will be for the baseline rocket and features a drogue parachute in place of the retromotor. The baseline configuration can be seen below in Figure 33.

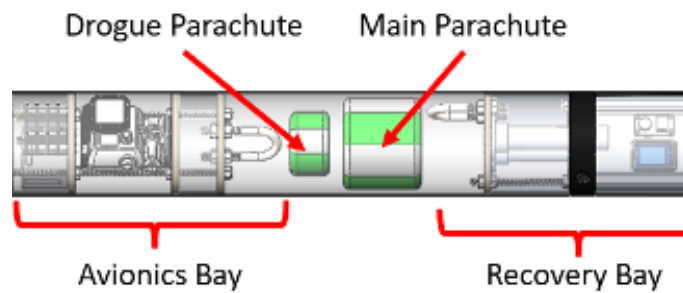


Figure 33 Baseline Rocket Parachute Configuration

The drogue parachute is a 24-inch diameter hemispherical parachute from Spherachutes. The main parachute is the same as the previous configuration and both parachutes are housed in the same parachute bay. In the baseline configuration of the rocket airframe separation will occur at apogee. During this separation the drogue parachute will be deployed from the rocket along with the main parachute. The drogue parachute will begin opening immediately while the main will be held close by two Jolly Logic chute release. Once the targeted main parachute deployment altitude is reached the chute releases will release allowing for the main parachute to open and the rocket to descend under the main parachute until landing. An example of this dual deployment system can be seen in Figure 9.

#### 2.1.5 Canards Bay

Located in the upper airframe just below the nose cone and integrated with the avionics bay the canards bay houses the hardware designed to actuate the four canards. The full canards bay design can be seen in Figure 34 below. This design meets the goal of providing the canards control system described in Section 3.1.5 to control the roll of the rocket body allowing the payload to capture clear images. A single servo actuation system was designed to accommodate the need for all four canards to be actuated in the same direction and at the same angle at any time. This helps to keep the controls design simple compared to that of a four-servo system where each servo actuates a canard. A four-servo system was considered for this project but was deemed not necessary for the goal of controlling roll only.

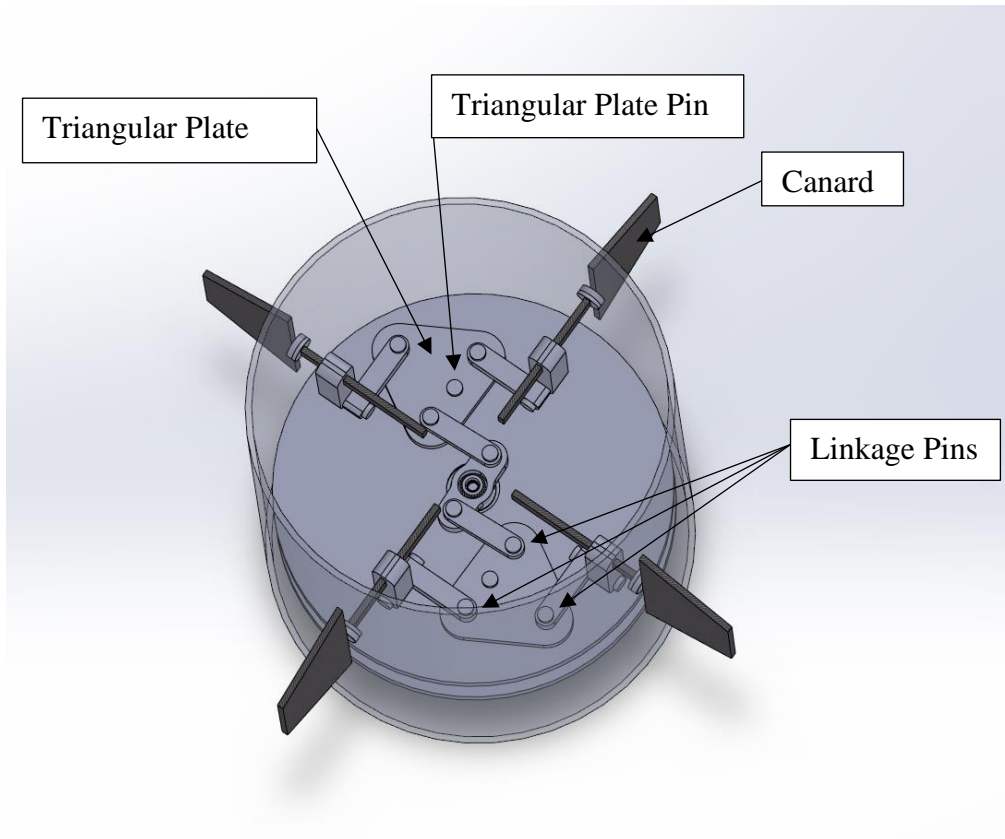


Figure 34 Canards Bay SOLIDWORKS Model

The mechanical actuation system seen in Figure 34 is designed to transfer the rotation of the servo equally to all four canards simultaneously. This is done by creating two symmetrical systems attached to opposite ends of the servo head. Each system features a triangular plate that sits atop a pin mounted to the upper canards bay plate. This pin allows for the plate to rotate with the actuation of the servo. The plate has 3 linkage pin holes equally distanced allowing for equal distribution of the servo rotation.

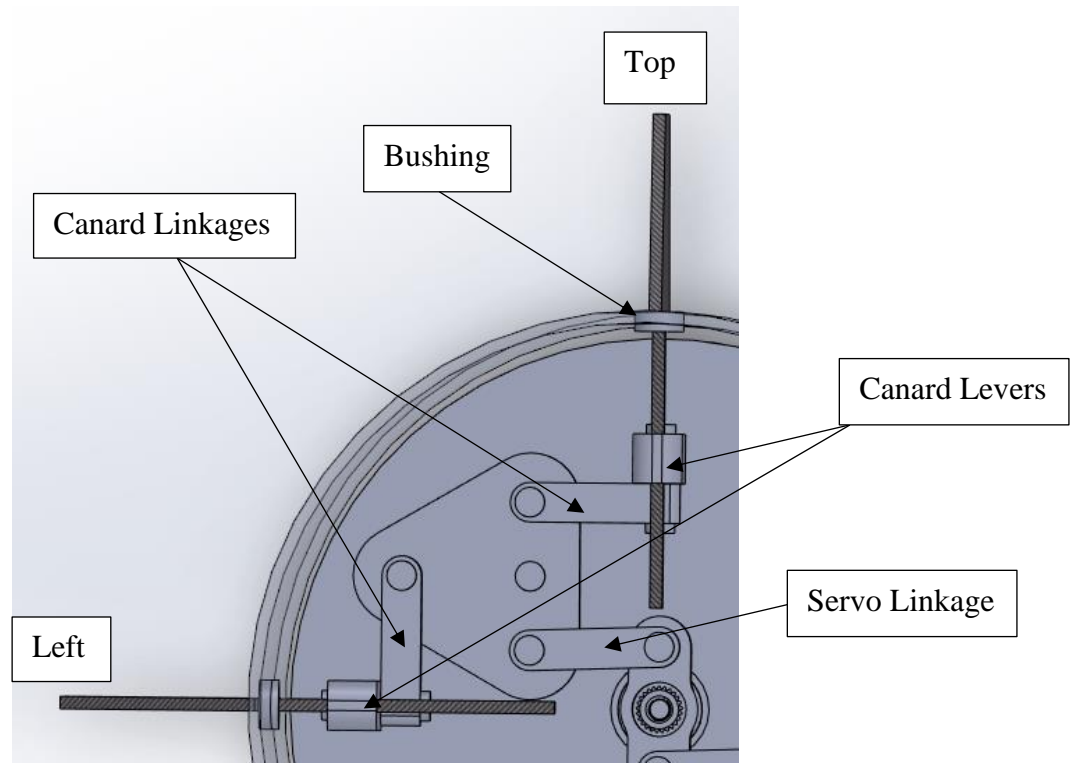


Figure 35 Canard Mechanical Actuation

The servo linkage connecting the servo head to the triangular plate is designed to rotate the triangular plate clockwise (when viewing the actuation system in Figure 35). This rotation of the triangular plate is due to the servo linkage and causes the two equal length canard linkages to be actuated. With both linkages being of equal length one linkage pulls on the lever system attached to the left canard and the other linkage pushed on the lever system attached to the top canard. The canard levers work by rotating the canard tabs with the bushings due to the motion of the canard linkages acting on the bottom of the canard lever. The connection between the canard linkages, canard levers and canards can be seen in Figure 36. The motion of these components can also be seen as denoted by the black arrows in Figure 36.

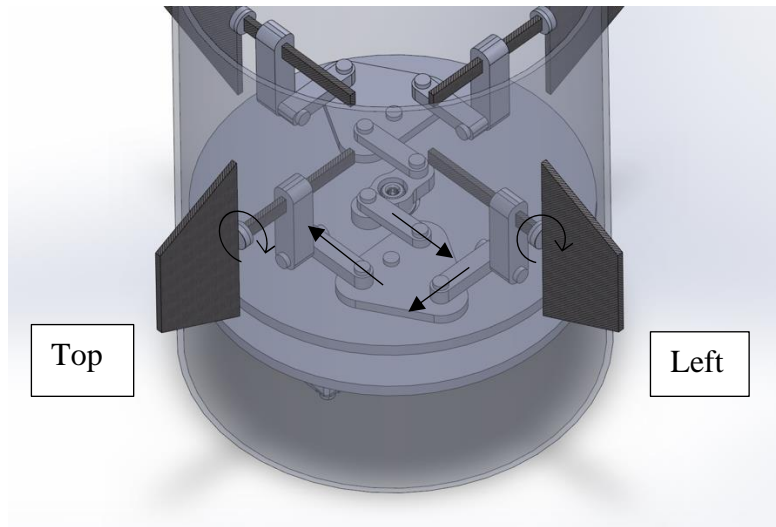


Figure 36 Canard Actuation Linkages

These motions of the canard levers cause the canards to both rotate clockwise relative to the rocket body as seen in Figure 37. The canards themselves are fixed to the airframe its by way of circular bushings chosen to fit the airframe holes seen in Figure 35. These bushings allow the canards to rotate about their tabs while preventing the canards from detaching from the canards bay or airframe.

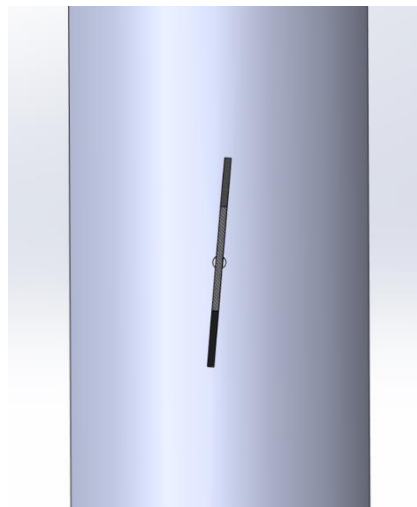


Figure 37 Canard Rotated 5 Degrees Clockwise

It is important to note that to comply with NAR guidelines, the canards bay will not be flown on the baseline rocket for launch. During launch a simulated weight will be installed in its place and the canards themselves will be fixed at an angle of attack of zero degrees on the airframe. For this reason, the canards bay is only in a prototype stage and is not a tested final design. It is



also important to note that this canards bay design was based on the roll control system designed by the Portland State Aerospace Society at Portland State University [62].

### 2.1.6 Airframe Stress Distribution (ARS Analysis Task 1)

This task aimed to determine the stresses on various critical systems of the rocket during peak-stress events, such as the motor mount at peak thrust, the parachute bay during the separation event, and the exterior airframe at max aerodynamic load. Failure at any one of these points would result in catastrophic part failure and the total loss of the rocket.

For these different analyses, both Ansys Mechanical Workbench and SOLIDWORKS FEA were used. Both applications use finite element analysis (FEA) to perform structural analyses on complex geometries. In the case of this analysis, the stress and deformation of the parts when under load were the areas of interest. To this end, the von Mises stress of the parts under load were calculated. Estimating the von Mises stress is a type of failure analysis used to analyze materials. It works by converting the stress tensor as defined by Hooke's Law into a singular scalar value and then comparing that to the yield strength of the material in question. Hooke's law can be seen in Equation (7) and the final von Mises equation can be seen in Equation (8) [44,45].

$$\epsilon_{ij} = \frac{1}{E} \{ (1 + \nu)\sigma_{ij} - \nu\delta_{ij}\sigma_{kk} \} \quad (7)$$

$$\sigma_v = \sqrt{\frac{1}{2} [(\sigma_{11} - \sigma_{22})^2 + (\sigma_{22} - \sigma_{33})^2 + (\sigma_{33} - \sigma_{11})^2] + 3(\sigma_{12}^2 + \sigma_{23}^2 + \sigma_{31}^2)} \quad (8)$$

Where:

$\epsilon_{ij}$  = Strain Tensor

$E$  = Young's Modulus

$\nu$  = Poisson's Ratio

$\sigma_{ij}$  = Stress Tensor

$\delta_{ij}$  = Identity Matrix

$\sigma_{kk}$  = Principal Stresses

$\sigma_v$  = von Mises Stress

$\sigma_{11}, \sigma_{22},$  and  $\sigma_{33}$  = Principal Stresses

$\sigma_{12}, \sigma_{23},$  and  $\sigma_{13}$  = Off - Axis Stresses

The Young's Modulus also governs the resultant von Mises stresses. This value represents the relationship between the stress applied to a material and the strain induced in it. This relationship can be seen in Equation (9), where  $E$  is the Young's Modulus,  $\sigma$  is the applied stress, and  $\epsilon$  is the induced strain [46]. The Young's Modulus is easily defined for isotropic materials,

like steel or aluminum, but the fiberglass and plywood the team worked with were more complex when it came to defining a Young's Modulus. How the difference between the isotropic and anisotropic materials are handled is discussed in the *Motor Mount Assembly Analysis* subsection below.

$$E = \frac{\sigma}{\epsilon} \quad (9)$$

Where:

$E$  = Young's Modulus

$\sigma$  = Stress

$\epsilon$  = Strain

### *Motor Mount Assembly Analysis*

Analysis of the motor mount assembly was the first completed. Failure of this part would result in the motor failing to impart its thrust to the rocket as a whole and instead sliding forward inside the airframe and colliding with the recovery bay, resulting in asymmetric thrust and the destabilization of the rocket. The part consists of three plywood centering rings around a fiberglass motor tube capped with a plywood stopper plate, as can be seen in Figure 38, depicting the assembly in SOLIDWORKS before it has been loaded into Ansys and meshed.

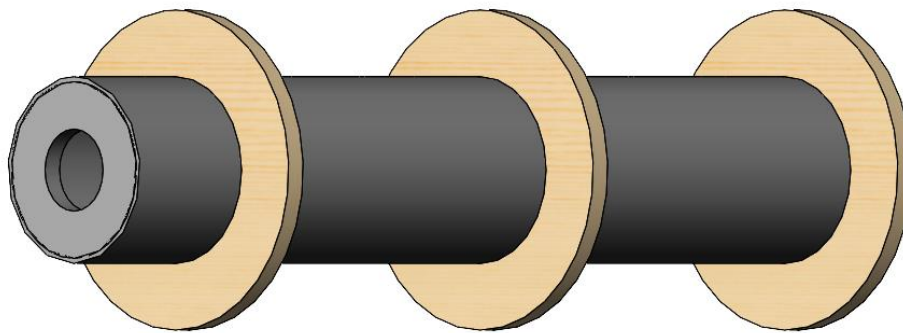


Figure 38 Motor Mount Assembly

Both fiberglass and plywood are anisotropic materials, meaning that their material properties can vary depending on the direction of the load and which “layer” of the material is

being analyzed due to their fibrous nature. Fiberglass is a composite of glass fibers at various angles relative to each other with an epoxy matrix binding them together. The tube is a wound fiberglass tube with glass fibers wound around at 30° and 45° angles [42]. Plywood is a composite wood product that is made from layers of wood veneer adhered together with each layer having its grain rotated at some angle relative to the previous [47]. The fabrication techniques used in plywood make it quasi-isotropic so the assumption that the material has isotropic properties was valid. A quasi-isotropic material is one that is anisotropic but is assembled in a way that it has isotropic behavior. For the fiberglass, however, an isotropic assumption was not as valid. The Ansys Composite PrepPost (ACP) tool was used within Ansys workbench to model the more complex behavior of fiberglass. This tool allows the user to model the individual plies of composite material. While the two different wind angles were known, assumptions were made about other unknown properties. These were the specific axial Young's Moduli, Poisson's Ratios, number of plies, and epoxy matrix used. The default unidirectional E-glass from the Ansys engineering data material set was used to fill in these gaps. Shown in Appendix B: ARS Analysis Task 1 Material Properties are the properties of the materials assumed in this analysis.

Using ACP required modeling the motor tube first as a shell geometry, which represents a zero-volume surface. This was done by importing the original motor tube into Ansys SpaceClaim and then isolating the outer face of the tube. The shell geometry was imported into ACP. In addition, the fiberglass-epoxy composite was assumed to be unidirectional (UD) E-Glass. Various types of fiberglass are already available within the engineering materials database in Ansys and could be used to model each unidirectional ply. A series of videos by LEAP Australia was used to help work through using ACP [48].

The first step within ACP was setting up the rosettes. A rosette is a coordinate system on the surface of the geometry that is used to define the principal axis of the composite fibers, the transverse direction, and the direction normal to the surface. In this case it was also used to define the vector from which the wind angles were defined. Then, the team created an "oriented selection set" that merged the rosette with the surface of interest. Next, the team created a modeling group so that the alternating 30° and 45° plies could be inserted in groups of two in order to cut down on the work required. Each modeling group contained two plies, one of each wind angle. These groups were applied to create an alternating wind angle stacking sequence. The full layout of the ACP UI can be seen in Figure 39. The green lines in that figure indicate the filament wind direction for the ply currently selected.

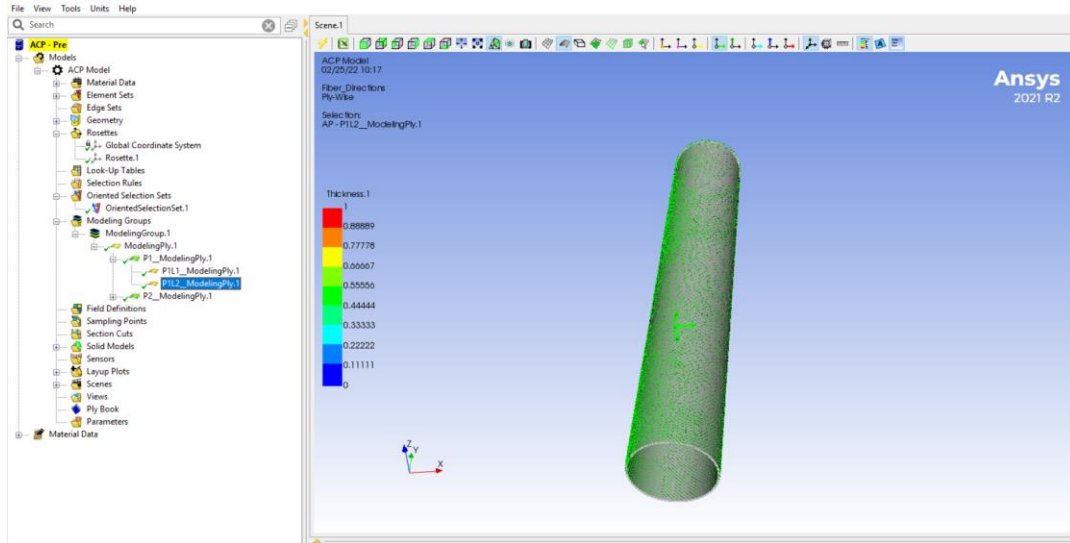


Figure 39 ACP User Interface

With the ACP model created, the team imported it back into Static Structural so that it could have boundary conditions applied and the simulation run. The plywood centering rings were imported as well from their own group in order to merge the ACP model of the motor tube and the traditional model of the centering rings. The workflow on Ansys Workbench can be seen in Figure 40. The lines between the blocks in this figure show how each section imports data into the next. Because the “Setup” box in the ACP block was selected at the time the screen capture was taken, there are some connections that are bolded. This has no specific meaning other than to highlight what connections were relevant to what was selected at the time. The workflow goes from left to right, starting with the geometry being imported into ACP, the results and material data from ACP being exported into the final model and the centering rings model, respectively, and finally the centering ring model being exported into the final model as well.

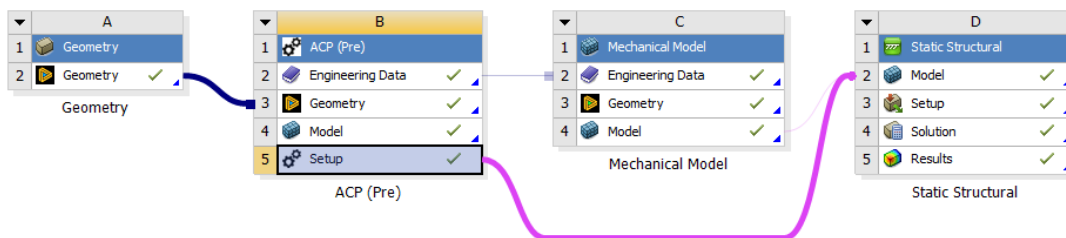


Figure 40 Motor Mount Analysis Workbench

Meshing the part is a critical aspect of any FEA task. It breaks the complex geometry of a part into small, definable regions that the software can analyze in discrete chunks. The quality of the mesh directly drives the accuracy of the result, so having a mesh without oddly or under defined regions is crucial. Both the centering rings and the motor tube were meshed within their respective sections before being merged. The combined mesh can be seen in Figure 41.

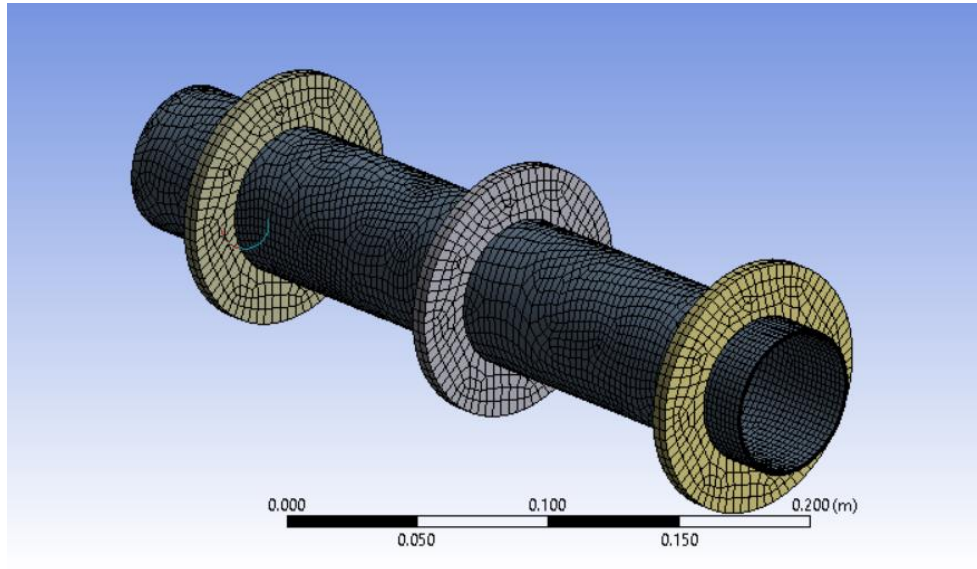


Figure 41 Motor Mount Mesh

The team fixed this meshed body along the outer face of each centering ring to simulate their bond to the rigid airframe wall, and the centering rings were fixed to the motor tube with a fixed boundary. Then, the maximum thrust determined by the PTSS subteam of 937 Newtons was applied to the bottom face of the motor tube. This simulates how the thrust is transferred from the motor casing itself to the tube it is mounted within. Figure 42 depicts the boundary conditions used in this analysis and Table 2.1 describes each element.

Table 2.1 Motor Mount Assembly Boundary Conditions

Boundary	Location on Geometry	Description
Fixed Support	Outer face of centering rings	Rigid fixture of the outside of each centering ring models how they would be epoxied to the airframe
Contact Bond	Contact area of the inner face of each centering ring and motor tube	Rigid contact bond between each centering ring and the motor tube models how they would be epoxied together
Force	Bottom face of motor tube	939 N motor thrust force is the load applied by the motor

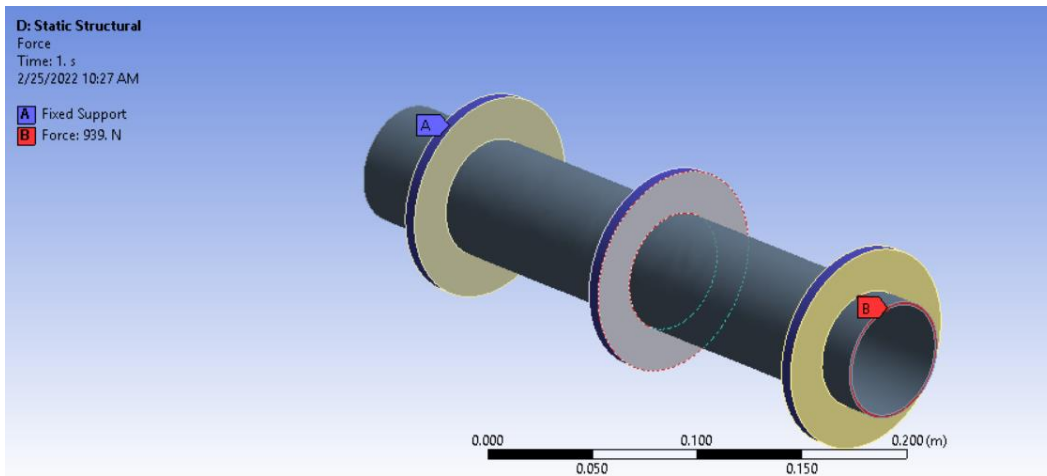


Figure 42 Motor Mount Boundary Conditions

### Parachute Bay Analysis

The second major area to be analyzed was the parachute bay, depicted in Figure 43. This part experiences its maximum load when the CO<sub>2</sub> ejection system is fired, pressurizing the parachute bay. The maximum pressure reached within the chamber is 21.84 psi before the shear pins that hold the upper airframe to the recovery bay and lower airframe fail and the airframe separates. This value was determined by PTSS Analysis Task 3. This bay consists of the upper bulkhead of the recovery bay, the section of airframe that the parachute is stored within, and the lower bulkhead of the avionics bay. Both bulkheads are epoxied to the interior of the airframe.

Figure 26 depicts the full assembly and the area of interest here is just forward of the midpoint of the rocket. Analysis of this area was necessary to ensure that the separation event did not catastrophically blow out the airframe of the rocket or break the bulkheads on the avionics or recovery bays. Any of these failures would at best result in the failure of the parachute to deploy and at worst cause the breakup of the rocket in the air.

Analysis of this system started with simplifying the parts involved. The bulkheads depicted in Figure 44 are simpler versions of those found on the avionics bay and recovery bay stripped down to basic plywood disks. Doing this removed the complex geometries on their surfaces from the nuts, threaded rods, and CO<sub>2</sub> system hardware that penetrate the parachute bay. Also not depicted are the parachutes and shock cord that are stored in this bay. Like the hardware on the avionics bay and recovery bay bulkheads, these are extraneous to the analysis being done on this section of the airframe. The bulkheads are also made from 0.25-inch plywood and the airframe is made from the same wound fiberglass as the motor tube except with a 4-inch radius.



Figure 43 Parachute Bay

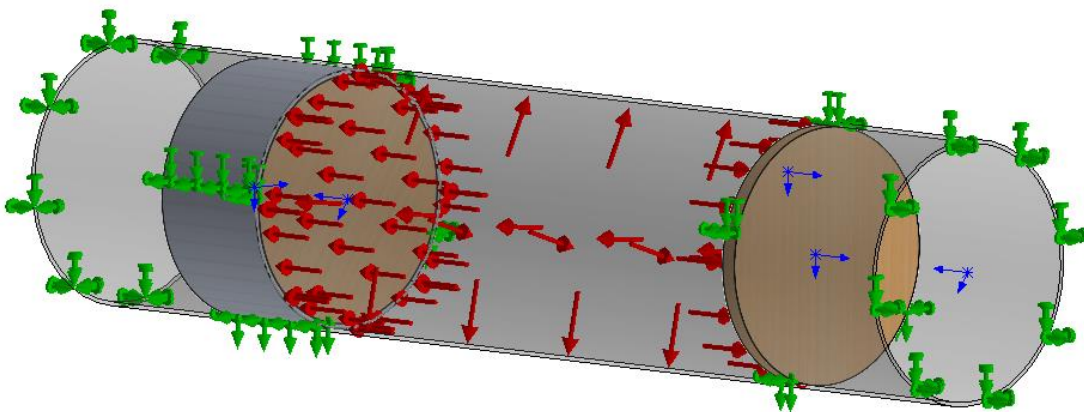


Figure 44 Parachute Bay Boundary Conditions

The analysis was performed on a part bound by fixed supports on the top and bottom faces of the airframe cutaway to fix the position of the part to simulate the rigidity of the airframe, and the bulkheads were contact bound to the interior of the airframe. “Contact bound” is a term used in FEA to describe two separate parts or faces that are in contact with each other and considered rigidly attached. Then, the team applied a 21.84 psi load to the interior of the bay, around both the interior of the airframe tube, and the inner faces of both bulkheads. These boundary conditions can be seen in Figure 44. The meshed part is depicted in Figure 45 and was of satisfactory quality to progress with the analysis. The green arrows in the figure are not point fixtures but rather represent the fact that the part is fixed along the whole face. The same applies to the red arrows that represent the pressure load. The boundary conditions are described in Table 2.2.

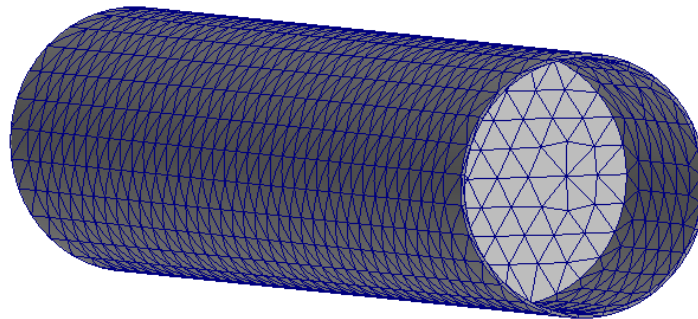


Figure 45 Parachute Bay Mesh

Table 2.2 Parachute Bay Boundary Conditions

<b>Boundary</b>	<b>Location on Geometry</b>	<b>Description</b>
Fixed Support	Top and bottom faces of airframe	Rigid fixture of the upper and lower simulates how the airframe exists beyond where the model cuts of
Contact Bond	Contact area of internal bulkhead with airframe	Rigid contact bond between the bulkheads and the airframe models how they would be epoxied in place
Pressure	Inner faces of bulkheads and inner face of airframe.	21.84 psi load applied to the interior of the parachute bay is the load on the system



## Fin Analysis

The third major area analyzed was the load on the fins created by the velocity of the rocket. The maximum velocity achieved by the rocket was calculated to be 607 feet/second by the PTSS subteam. While the loads on the fins would almost certainly not be enough to shear them off their mounts, it was nevertheless important to ensure that that was the case. Failure of the fins would result in the total destabilization of the rocket and its subsequent total loss as well as unsafe conditions for individuals on the ground.

This analysis was also performed in Ansys Static Structural. The team imported the fin geometry from the Fluent workspace of FDA Analysis Task 2, detailed in Section 3.2.3. This was done so that the position of the fin in the global coordinate system was identical between the Fluent data and the Static Structural model. The team encountered significant challenges in correctly importing the pressure data from Fluent to Static Structural. The correctly imported data can be seen in Figure 46. Originally, the data was imported directly from fluent to static structural in much the same way as was done in the motor mount analysis, linking ACP to Static Structural. This was not successful in correctly importing data due to different mesh types used by the team's Fluent analysis and Static Structural. Static Structural uses tetrahedral mesh elements while Fluent can be set up to use hexagonal mesh elements, as was done in the team's analysis. This means that Ansys was unable to map the static pressure data based on the mesh elements. More details on Fluent meshing can be found in Section 2.1.7.

To resolve this, the team instead exported the data from Fluent as an ASCII text file which maps the pressure data in terms of the X, Y, and Z position of each point rather than trying to directly map between elements. Then, this text file was imported into Static Structural as an external data file and mapped onto the fin from there. This is why importing the fin geometry from Fluent was necessary as that preserved the X, Y, and Z positioning of the part.

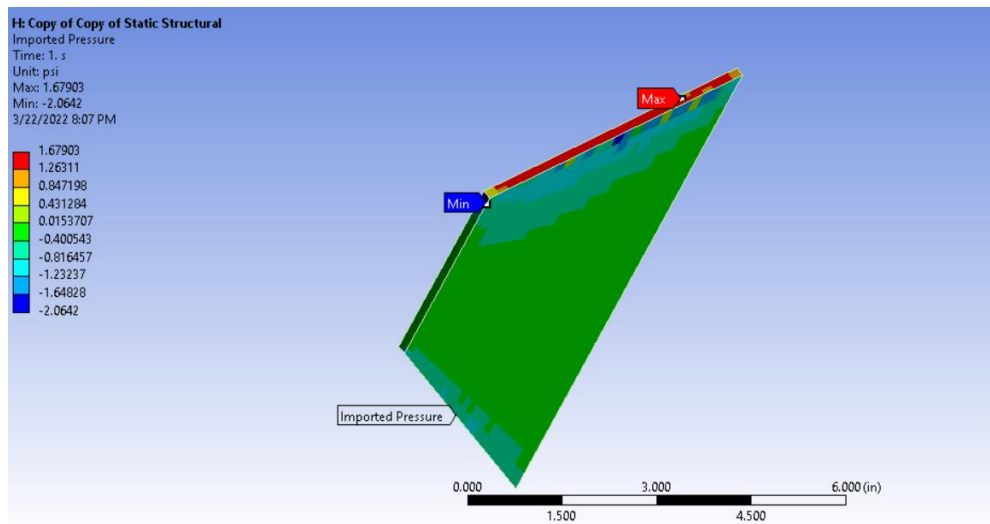


Figure 46 Fin Static Pressure Distribution in Static Structural

The team designed the fin to be made from fiberglass. Unlike the fiberglass tube described in the motor mount analysis, however, this fiberglass is made in such a way as to act as a quasi-isotropic material, much like the plywood in the motor mount analysis. This meant that the ACP tool did not need to be used to model this material and instead could have the default E-Glass material from Ansys applied to it.

With the pressure data accurately imported, the team assigned part fixtures, depicted in Figure 47. A fixed support was used along the face of the fin mounted to the airframe to simulate the rigid bond that the epoxy adhesive would create. With the boundary conditions assigned the part was meshed. The meshed part can be seen in Figure 48 and the boundary conditions are described in Table 2.3.

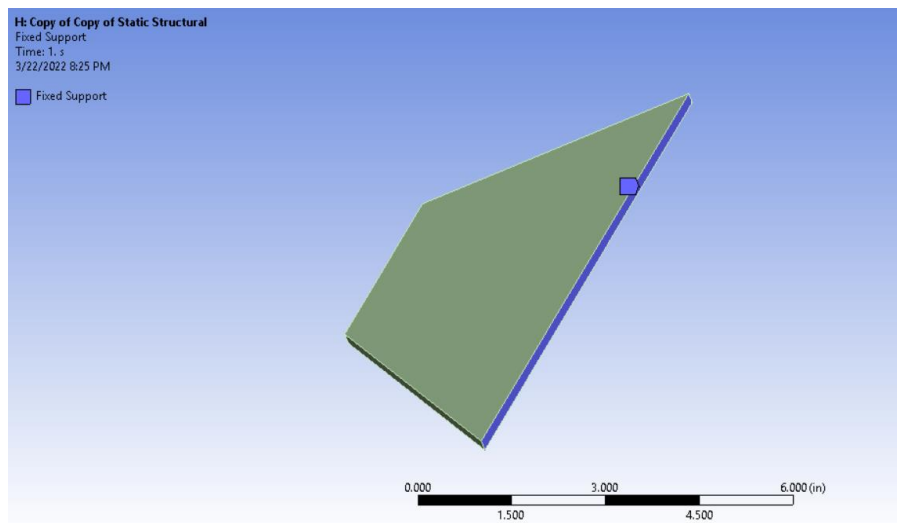


Figure 47 Boundary Conditions of Fin

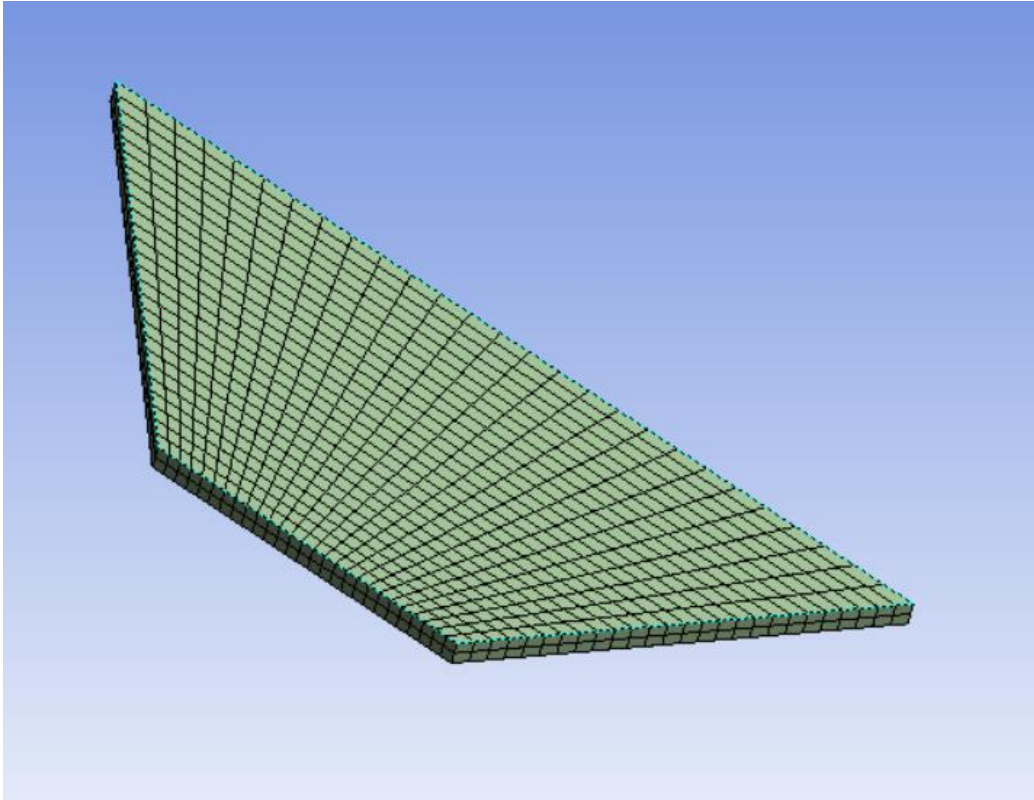


Figure 48 Fin Mesh

Table 2.3 Fin Boundary Conditions

<b>Boundary</b>	<b>Location on Geometry</b>	<b>Description</b>
Fixed Support	Long face of fin side	Rigid fixture on the long face of the fin edge models where the fin would be attached to the airframe.
Pressure	Across all faces besides fixed support face	Varied pressure distribution is imported from Fluent and is the load on the system.

### 2.1.7 Computational Fluid Dynamics (CFD)

Due to the complex geometry and characteristics of the physical system, the team determined that computational fluid dynamics (CFD) would be the most effective choice for the aerodynamic simulations. Ansys Fluent was chosen as the primary CFD simulation software used for ARS Analysis Task 2 and FDA Analysis Task 2, discussed in Sections 2.1.8 and 3.1.4

respectively. Fluent provides users with near complete control over fluid parameters, boundary parameters, and mesh design. This allows for the simulation results to be more accurate to the physical system, assuming the user inputs are valid. Due to the increased control that Fluent provides, it is also relatively complex and requires more time to learn than other CFD simulation software, such as SimScale. However, the team determined that the increased control and accuracy of results was worth the additional time investment to properly learn how to use Fluent.

To carry out CFD simulations, the team must first select a solver. Within Ansys Fluent, two main solvers are available to numerically model fluid flow: a density-based solver and a pressure-based solver [49]. The density-based solver is primarily used for high-speed compressible flows while the pressure-based solver is designed for incompressible and low-speed compressible flows. Air flows that have a Mach number of 0.5 or less demonstrate very little change in density and can therefore be considered incompressible for the purposes of aerodynamic simulation. According to simulations run in OpenRocket, the maximum speed of the rocket during flight was estimated to be about 185 m/s, or a Mach of 0.54. Although this estimated maximum flight speed was slightly above the typical threshold of what is considered low-speed compressible flow, most of the rocket's flight was expected to have a flow speed at or below Mach 0.5. Therefore, the air flow around the rocket could be treated as incompressible and Fluent's pressure-based solver was preferred. Should there exist any anomalies or inaccuracies at higher flight speeds ( $M = 0.5$  to 0.54), then they could be addressed on a case-by-case basis.

The pressure-based solver, which uses a pressure correction to iteratively determine a velocity field that satisfies the conservation equations and boundary conditions that describe the system, follows either a coupled or segregated algorithm to solve the governing equations of the fluid flow, as shown in Figure 49 [50]. The governing equations in question are the Navier-Stokes Equations, discussed in Section 1.5.2. In the segregated algorithm, the individual governing equations are decoupled from each other and solved one by one. This method is more memory-efficient, as only one equation is stored at a time. However, solving the governing equations sequentially results in slower solution convergence. In contrast, the coupled algorithm solves the governing momentum and continuity equations simultaneously as a system. Because of this, the speed of solution convergence greatly increases, but the memory requirement of the coupled algorithm can be nearly twice that of the segregated algorithm [50]. Since system memory was not an immediate concern, the team selected the coupled algorithm for our analysis.

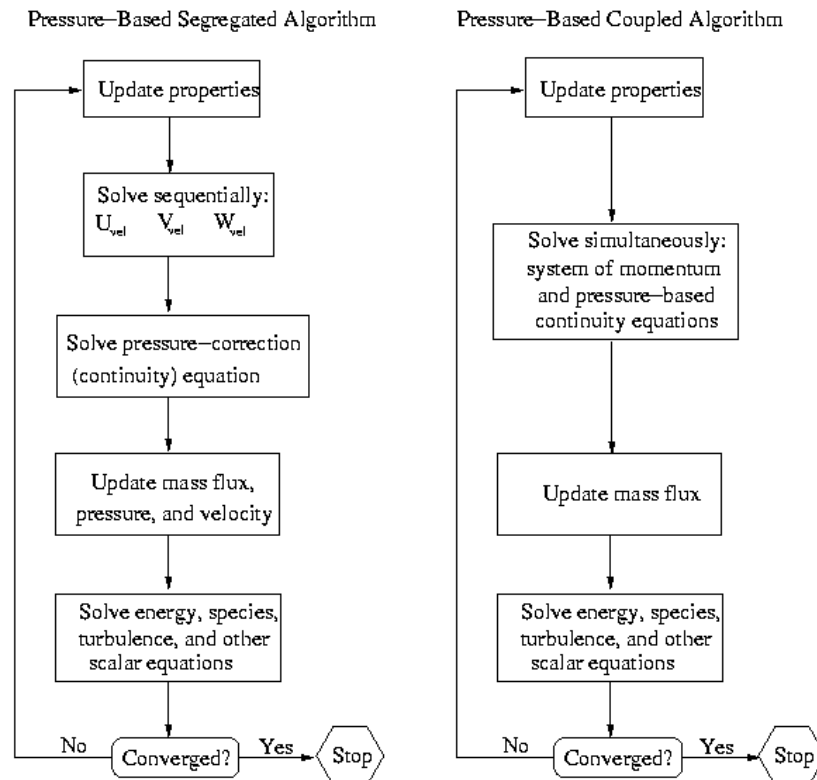


Figure 49 Overview of Pressure-Based Solution Methods in Fluent [49] © Ansys 2009

To determine solution convergence, Fluent examines the residual values of the solver variables. The residuals are defined as the error between the results of consecutive iterations divided by the value of the current iteration. If the value of the  $n^{th}$  iteration is  $m(n)$  and the value of the previous iteration is  $m(n - 1)$ , then the residual value  $R$  can be calculated using Equation (10).

$$R = (m(n) - m(n - 1))/m(n) \quad (10)$$

The cutoff for residual convergence can be manually set by the user. The default value for residual convergence of momentum, energy, and continuity variables in Fluent is 0.001 in most models [49]. The number of iterations must also be set manually to prevent the simulation from running continuously if the system does not converge. If convergence is not reached, the system must be updated by the user by increasing the maximum number of iterations, editing the mesh characteristics, or editing the initialization settings of the simulation. The design of the mesh for a simulation is very dependent on the geometry being examined and is discussed in more detail in Sections 2.1.8 and 3.1.4 for the specific cases.

Within Fluent’s pressure-based solver algorithm, the team must also select a solver model. There are numerous pressure-based models available in Fluent, each with their own assumptions regarding the flow. For aerodynamic simulation, viscous models are typically used because they take fluid viscosity and wall roughness into account. This increases their accuracy when examining turbulent flow. Turbulent flow corresponds to a flow with a Reynold’s number greater than 3500 [51]. The Reynold’s number  $R_e$  can be determined through Equation (11), with Table 2.4 providing an outline of the related values.

$$R_e = \rho u L / \mu \quad (11)$$

Table 2.4 Values Needed to Determine Reynold's Number

Value	Variable	Units
Reynold’s number	$R_e$	-
Fluid density	$\rho$	kg/m <sup>3</sup>
Fluid dynamic viscosity	$\mu$	kg/(m·s)
Characteristic length	L	m
Relative flow speed	u	m/s

For the project, the team identified the density and viscosity values as the density and viscosity of air at 0°C and 1 atm, which are 1.275 kg/m<sup>3</sup> and 1.715·10<sup>-5</sup> kg/(m·s), respectively. The temperature was selected to be 0°C rather than standard temperature because it better matches the expected conditions at our intended launch site, which was at Lake Winnepesaukee in mid-February. Although these parameters do vary with altitude, our rocket was not expected to exceed 4000 feet at apogee, meaning there would not be any significant change in the air density or viscosity when compared to sea level. Additionally, the characteristic length was taken to be equal to the length of our rocket at 1.7 meters. The length of the rocket was used for this characteristic dimension rather than the cross-sectional diameter due to the nature of the flow over the rocket. The airflow ran from the nose to the tail of the rocket, so this length was used as the characteristic dimension similar to how the characteristic length used for an airfoil would be measured from the leading edge to the trailing edge. Given these parameters, the geometry of our rocket, and the expected flight speeds of up to 185 m/s, the Reynold’s number for our flow ranges from 10<sup>5</sup> to 10<sup>7</sup>, all of which is comfortably within the range of turbulent flow. This means that a viscous solver model must be used.

Viscous models vary greatly in complexity, with classifications ranging from zero-equation to four-equation models. The designation of the number of equations refers to the number of transport equations solved as part of the given method. Zero-equation models, as the most

simplified option, focus purely on local flow effects and provide little information regarding the characteristics of the overall flow domain. These models, as the name suggests, do not include any transport equations and instead solve for viscous effects algebraically. One-equation models introduce the transport equation for turbulent kinetic energy  $k$ , referred to as the  $k$ -equation. Although one-equation models do provide a more complete description than zero-equation models, they are typically inaccurate when examining transition flows from wall-bounded to free-shear flow, meaning they would not be ideal for examining boundary layer separation in our aerodynamic simulations [49]. Two-equation models, however, build upon one-equation methods by introducing a second transport equation. When considering two-equation models, the  $k$ - $\varepsilon$  and  $k$ - $\omega$  models are commonly used. In addition to the standard  $k$ -equation, these methods each introduce transport equations for turbulence dissipation rate  $\varepsilon$  and the inverse of the turbulent time scale  $\omega$ , respectively. The  $k$ - $\varepsilon$  model is designed to accurately model heat transfer problems and free-stream flow not closely bound by walls, meaning it is not ideal for systems involving significant boundary layer separation. The  $k$ - $\omega$  model, however, was designed to more accurately model boundary layer separation in wall-bounded flow, and was therefore used for our aerodynamic simulations, as it more accurately simulates flow around aerodynamic bodies [49]. Although more complex three- and four-equation models do exist, they were not considered for this project since two-equation models provide sufficient accuracy for the expected flow conditions.

The boundary conditions of the flow are also critical when simulating aerodynamic loads. Boundary conditions help determine the characteristics of the fluid as it enters the flow domain, interacts with the geometry of the vehicle or object being examined, and exits the flow domain. Fluent offers numerous boundary condition types for a wide variety of applications. Table 2.4 provides a summary of the boundary types relevant to our aerodynamic simulations.

Table 2.4 Summary of Relevant Boundary Conditions in Ansys Fluent [48]

Boundary Type	Description
Velocity Inlet	Define flow velocity and other flow properties at flow inlet
Pressure Outlet	Define static pressure of flow at flow outlet
Wall	Define boundary between fluid and solid regions. In viscous flow scenarios, wall roughness and slip conditions must be defined
Symmetry	Mirrors geometry and flow pattern

Each boundary used is assigned to a face or surface of a user-defined flow domain. The design of this domain is dependent on the geometry of the object being examined, and the boundary conditions assigned to different regions of the domain depend on the characteristics of the flow. Additionally, each boundary condition requires certain flow properties to be defined. Table 2.5 provides an overview of the assumptions made regarding these properties used in simulations for both ARS Analysis Task 2 and FDA Analysis Task 2.

Table 2.5 Assumptions used for Fluent Inputs

Assumption	Affected Input(s)
Simple/standard walls	Wall Boundary Condition: walls defined as stationary, no-slip, standard roughness
Pressure outlet displays zero static pressure	Pressure Outlet Boundary Condition: gauge (static) pressure set to 0 atm
Air flow is incompressible (constant density) at sea level and 0°C	Fluid material set to air; density set to 1.275 kg/m <sup>3</sup> (0.0796 lbm/ft <sup>3</sup> )

#### 2.1.8 Canard Aerodynamic Analysis (ARS Analysis Task 2)

To determine the aerodynamic loads on the canards during flight, CFD analysis was performed using Ansys Workbench and Ansys Fluent as the primary software. For a detailed explanation of CFD and an overview of Ansys Fluent refer to Section 2.1.7. Finding the aerodynamic loads on the canard's during flight is not only crucial for the control system to properly control the roll of the rocket but also for the mechanical design of the canards and their actuation system. Comparison of the loads to the specifications of the servo ensures the servo can operate under the maximum loads during flight as well as to ensure structural integrity of the system. For this analysis, the process in Figure 49 was followed to simulate the flow over the canards during flight.

Table 2.6 Canard Aerodynamic Load Analysis Cases

Case Number	Rocket Velocity (m/s)	Canard Angle of Attack (deg)	Rocket angle of attack (deg)
1	185	0	0
2	185	5	0



This analysis focuses on a single canard and a 90-degree cut of the upper portion of the rocket for two primary cases assuming symmetry of the rocket body at zero rocket angle of attack. In addition, all four canards will be actuated the same direction and angle of attack. The two cases are listed in Table 2.6 with the primary difference between the cases being the angle of attack of the canard. Both cases assume a fluid velocity of 185 m/s representing the maximum flight velocity taken from the OpenRocket simulation described in Section 1.4. For Case 1 the canard is at zero angle of attack representing the canards system not being actuated. The zero angle of attack model is also used later in the analysis to confirm the accuracy of the Fluent results with the results from Section 3.2.3. For Case 2 the model is identical except the canard is at an angle of attack of five degrees representing the maximum angle of attack the canards would be controlled to during flight. Five degrees was chosen as the maximum angle of attack based off member experience with similar systems on model rockets when designing the control system.

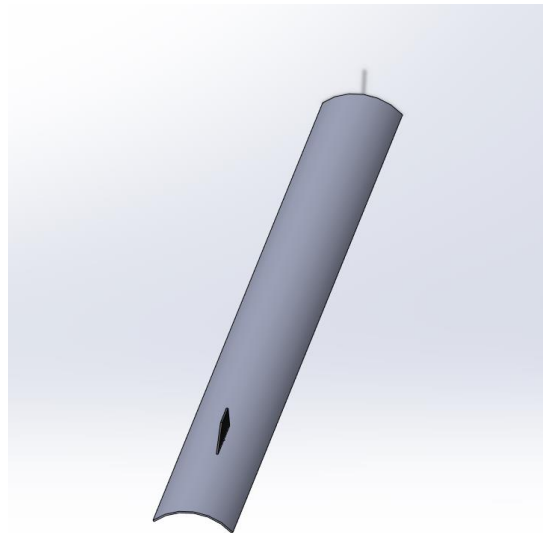


Figure 50 Case 1 Zero Angle of Attack SOLIDWORKS Model

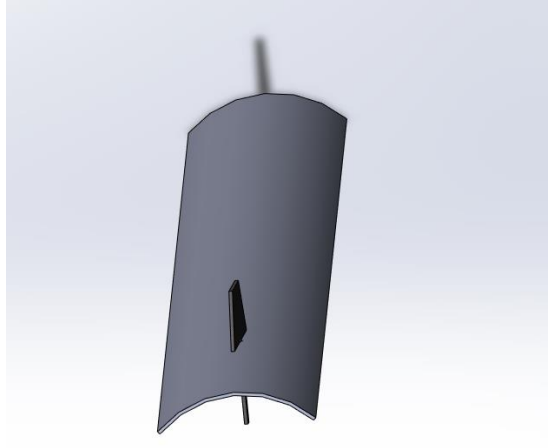


Figure 51 Case 2 Five Degree Angle of Attack SOLIDWORKS Model

The CAD models created in SOLIDWORKS for both cases can be seen in Figure 50 and Figure 51 above. These simplified models only feature a singular canard allowing for a much more in-depth analysis of the forces created and acting on the canards. Due to the incompressible and turbulent flow the pressure-based solver in Ansys Fluent with a viscous model was used for this analysis. For an explanation of this decision see Section 1.5.2. It is also important to note that due to the symmetry of the four canards, negative angles of attack for the canards do not need to be considered because they are symmetrical over the plane parallel to the velocity of the rocket.

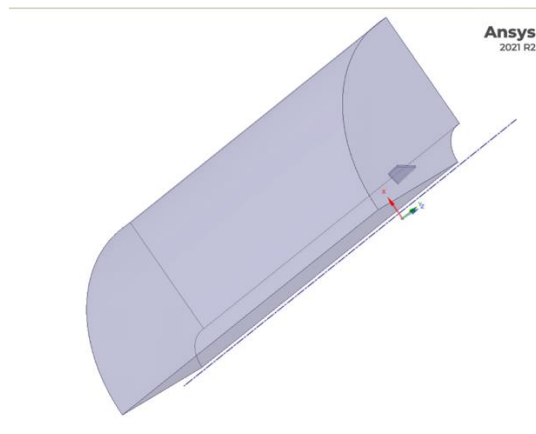


Figure 52 SpaceClaim Enclosure/Fluid Flow Domain

These models were then imported into the Ansys Workbench in SpaceClaim where an enclosure was modeled around the geometry to serve as the fluid flow domain as seen in Figure 52. An enclosure is required while using Ansys Fluent pressure-based solver to avoid infinite

bounds. Due to geometry of the canard and cut of the airframe the enclosure tool was not used in SpaceClaim to avoid unwanted airflow under the airframe cut. For this specific analysis a custom enclosure was created to ensure airflow started and ended along the outer surface of the airframe. The enclosure shape can be seen in Figure 52. This enclosure geometry is a modified quarter cylinder to match the airframe cutout geometry with the inlet and outlet faces located at the top and bottom edges of the airframe respectively to ensure airflow does not go below the airframe. The side walls of the enclosure also are located at the edges of the airframe cut to prevent unwanted airflow while the top of the enclosure was put at an arbitrary distance from the airframe. The distance between the airframe and the outer wall of the enclosure does not matter because it will be later defined as an inlet to avoid any interference of the flow over the canard. The inner curved wall of the enclosure will act as the airframe during this analysis. This will allow for airflow to be properly modeled over the airframe. Once the custom geometry was created the custom enclosure tool was used to make the geometry an enclosure and cut the canard geometry out of the enclosure. This creates the final fluid flow domain where the geometry of the canard is treated as an inside wall of the domain. It is important to remember to suppress all bodies other than the enclosure including the canard itself to avoid errors when creating the mesh later. Also labeling the walls of the enclosures with the “named sections” tool in SpaceClaim will make the set-up process easier.

For both cases after the model of the flow domain was created, it was then transferred to Fluent Mesh to create the mesh. To design the mesh, a default mesh was created in “Fluent Meshing”. A boundary layer of cells was then created around the geometry of the canard and the upper portion of the rocket to increase the accuracy of the flow field and simulate any boundary layer effects that may occur. This mesh was then refined until the quality was high enough for the accuracy required by this simulation. Refining the mesh is important because the quality of the mesh created directly impacts the accuracy of the results. The final mesh for both cases is very similar, the mesh for case 2 can be seen in Figure 53 and Figure 54. The highly condensed region towards the top of Figure 54 is the canard.

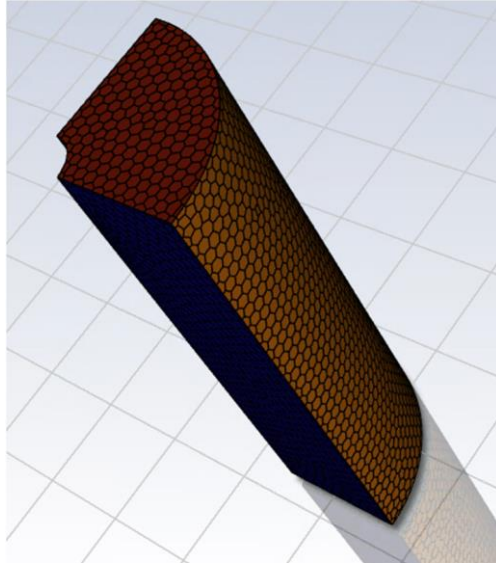


Figure 53 Fluent Mesh for Case 2 ISO View

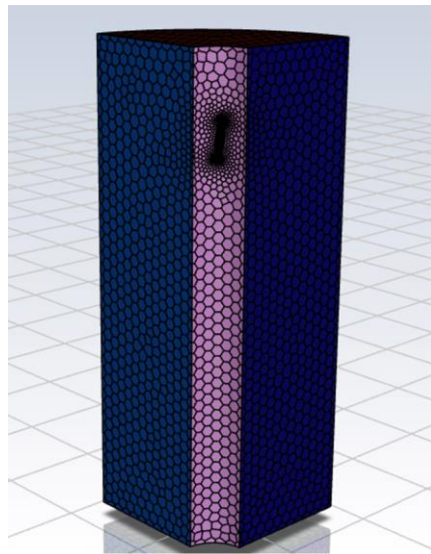


Figure 54 Fluent Mesh for Case 2 Airframe View

For both cases once the mesh was designed it was then imported into the Fluent set up where the conditions in Table 2.6 were defined and then boundary conditions were assigned to each face of the fluid flow domain. These conditions can be found in Table 2.7.

Table 2.7 Fluid Flow Domain Boundary Conditions

Face	Boundary Type	Velocity (X, Y, Z) (m/s)
Airframe	Wall	N/A
Top	Velocity Inlet	(0, -185,0)
Bottom	Pressure Outlet	N/A
Side 1	Velocity Inlet	(0, -185,0)
Side 2	Velocity Inlet	(0, -185,0)
Outside	Velocity Inlet	(0, -185,0)

The faces listed in Table 2.7 can be seen in Figure 55. The orange top face towards the nose cone of the rocket was defined as the velocity inlet to simulate the rocket properly moving through the air at maximum velocity while the bottom surface was defined as the pressure outlet to ensure accuracy of flow through the domain. The green curved outside face and both pink side faces of the domain were defined as velocity inlets with airflow in the negative Y direction of 185 m/s to prevent the surface from interference with the flow near the canard while still properly modeling the flow at a rocket angle of attack of zero degrees. The face acting as the airframe was defined as a wall to properly simulate flow over the airframe before and after the canard. The fluid domain properties can be seen in Table 2.8 and the definition of each boundary condition can be seen in Table 2.7.

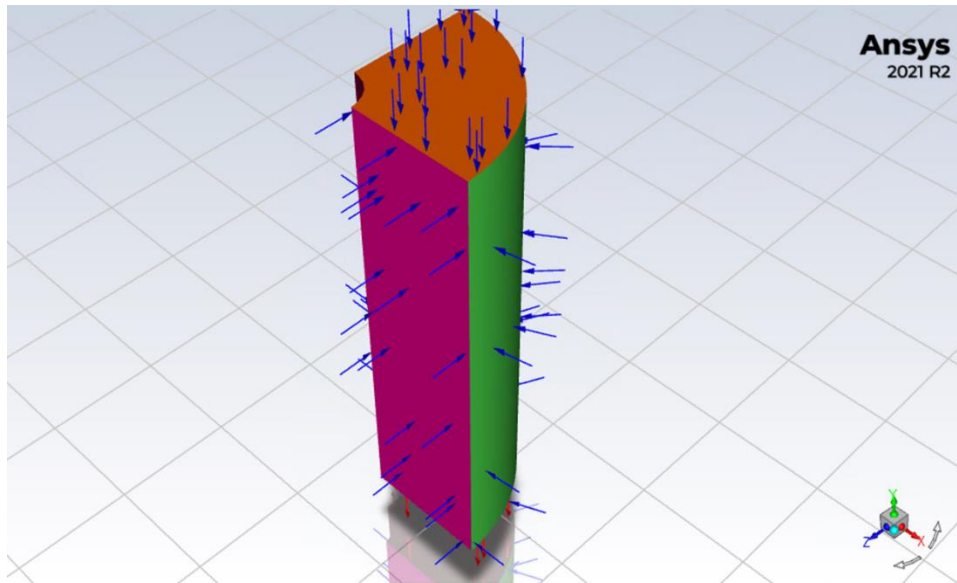


Figure 55 Fluent Flow Setup

Fluent also requires reference values for some results. For both cases in this analysis the reference values of area, length and flow density were used and can be found in Table 2.8. The

area is the project frontal area and was estimated to be equal to the area of the front edge of the canard, the length was set as the root cord of the canard since the airframe is just a wall and not part of the domain and the flow density was set as the density of air at 0°C.

Table 2.8 Fluent Reference Values

Reference Value	Value
Area	$4.06 \times 10^{-5} \text{ m}^2$
Length	0.0508 m
Flow Density	$1.275 \text{ kg/m}^3$

Once these conditions were set the Fluent simulations were both initialized utilizing the hybrid initialization tool and then run to produce the desired results found in Section 2.2.2. It is important to note that the simulation must be run with enough iterations, so convergence is reached. In the case of these simulations, this number was 150.

### 2.1.9 Canard Stress Analysis (ARS Analysis Task 3)

To ensure mechanical integrity of the canards and their actuation system and that the chosen servo can function under the aerodynamic loads due to flight, an FEA structural analysis was performed on the canards. For this analysis, the maximum aerodynamic loads found during Case 2 of the Fluent analysis in Sections 2.1.8 was imported into the Ansys Mechanical Workbench and Ansys Static Structural. For more information on FEA and setting up Ansys Mechanical Workbench see Section 2.1.6.

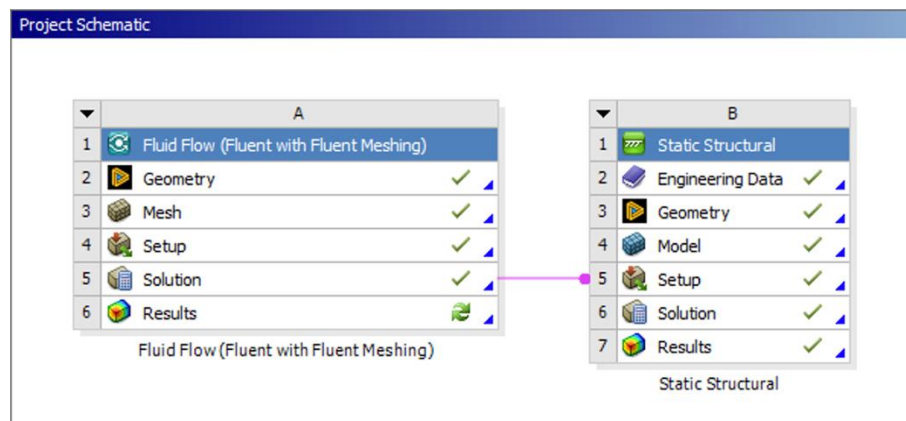


Figure 56 Ansys Workbench for Fluent and Static Structural

To import Fluent results into Static Structural a static structural module must be added to the Workbench with the solution from the Fluent module connected to the setup for Static

Structural as seen in Figure 56. Once the Workbench was set up the canard geometry from the Fluent simulation was imported in to Static Structural. This geometry was then edited in SpaceClaim to properly set up the analysis. First the canard tab was removed and then the position of the canard was confirmed to match that of the fluent analysis. This ensures the pressure data imported in the analysis is applied to the proper coordinates on the canard.

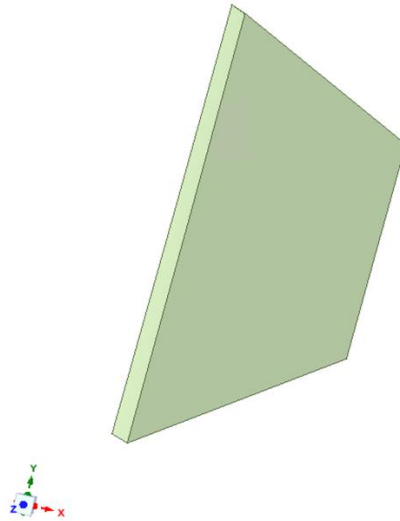


Figure 57 SpaceClaim Canard Geometry

Once the geometry definition was finished, the setup of the Static Structural analysis was completed. First a mesh was created on the geometry and refined to a resolution factor of 5 in the “mesh options” to gain accurate results. Mesh resolution decides the size of the nodes in the mesh with a higher resolution factor meaning smaller nodes. The higher the resolution the more accurate the model. The higher the resolution of a mesh the more complex the model becomes increasing run times and system requirements. A mesh resolution of 5 was determined to provide adequately accurate data while keeping run times reasonable for this analysis.

Following the setup of the mesh the material properties for the canard’s fiberglass were then set. Utilizing the “fixed support” option the bottom surface of the canard was set as a fixed support representing the canard fixed to the airframe during flight. The static pressure distribution from the Fluent analysis for Case 2 in section 2.1.8 was then imported under the imported loads section of the Static Structural setup. Case 2 was selected as it represents the highest possible aerodynamic loads on a canard assuming zero rocket angle of attack and maximum velocity. The results from Case 2 can be seen in Figure 58 and the loads imported from that same analysis can be seen in Figure 59 below.

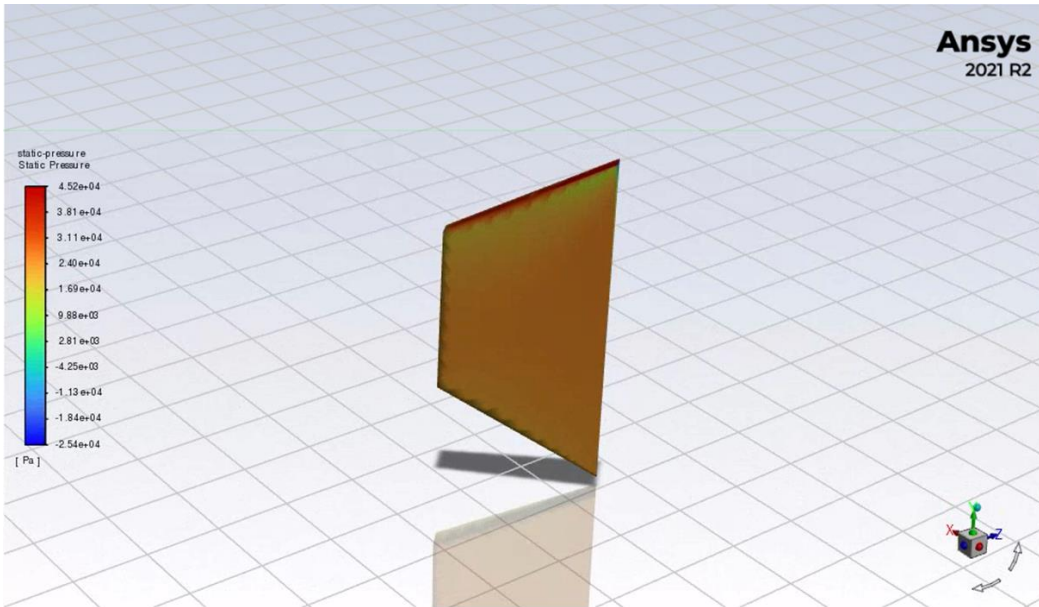


Figure 58 Fluent Static Pressure Results - Face Angled into Flow (Case 2)

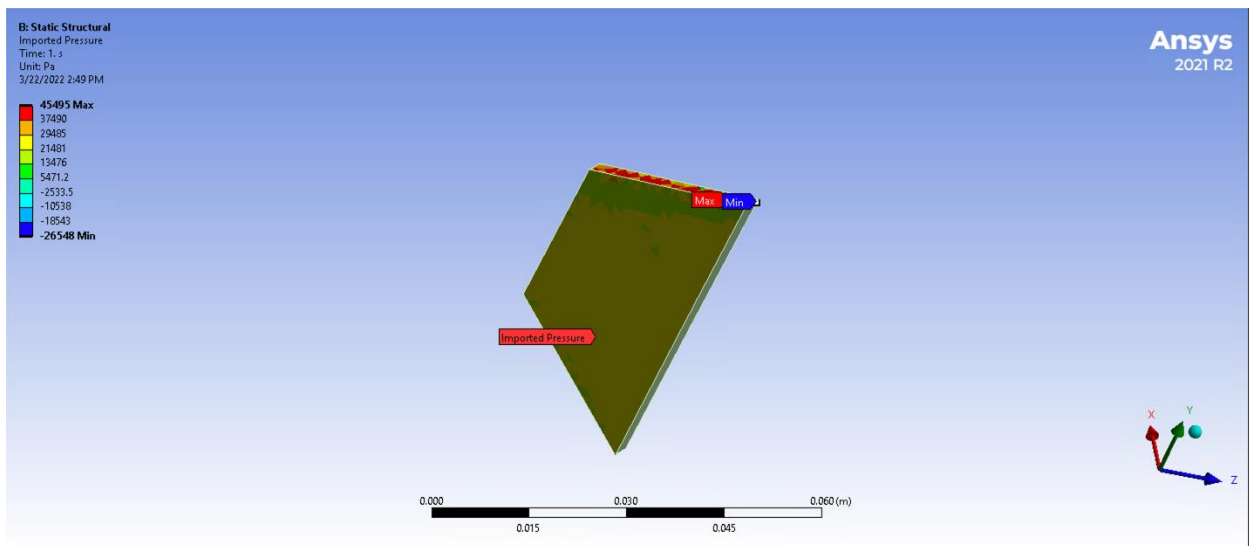


Figure 59 Imported Static Pressured Distribution - Face Angled into Flow (Case 2)

The resultant static pressure distribution from section 2.2.2 and the imported pressure distribution were compared to ensure the data was properly and accurately imported and applied to the canard. Figure 58 and Figure 59 show the pressure distributions to match when analyzing the canard face angled into the airflow while a discrepancy of about 200 Pa can be seen in the maximum pressure values. This discrepancy was determined to be reasonable, and the simulation was then run to produce the results in Section 2.2.3.



## 2.2 Results

### 2.2.1 Airframe Stress Distribution (ARS Analysis Task 1)

The motor mount analysis showed a maximum von Mises stress of 10.11 MPa. This is below the yield strength of the materials in question so failure of the part will not occur. A visualization of the stress distribution can be seen in Figure 60. The stress distribution here is in line with expectations. The most stress is carried by the joint between the lower centering ring and the motor tube with a decreasing load on the other two ring joints.

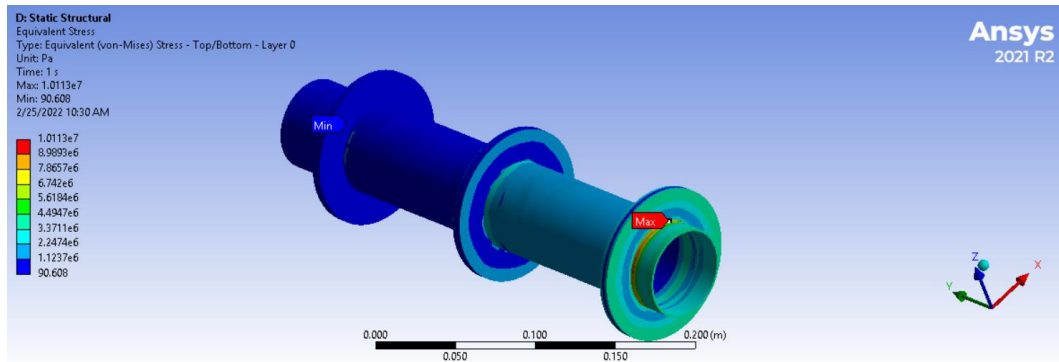


Figure 60 Motor Mount von Mises Stress

The recovery bay analysis showed a maximum von Mises stress of 6.078 MPa. This is below the yield strength of the materials in question and the part will not fail. The von Mises stress distribution can be seen in Figure 61. Figure 61 shows a cross section of the parachute bay in order to better visualize the stresses on the inner walls. The stress distribution is in line with expectations of how the pressure vessel would perform, with higher stresses along the cylinder walls than on the faces.

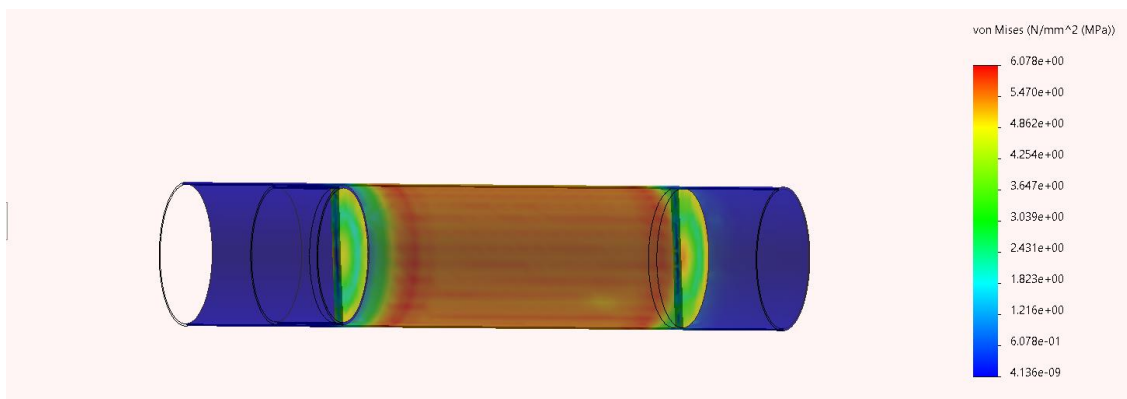


Figure 61 Recovery Bay von Mises Stress

The fin load analysis gave a maximum von Mises stress of 63474 Pa. This is of a far smaller magnitude than the previous two mechanical analyses but is to be expected as the load is also of a far smaller magnitude. The von Mises stress distribution is shown in Figure 62. This maximum load is far below the yield strength of the fiberglass so the part will not fail.

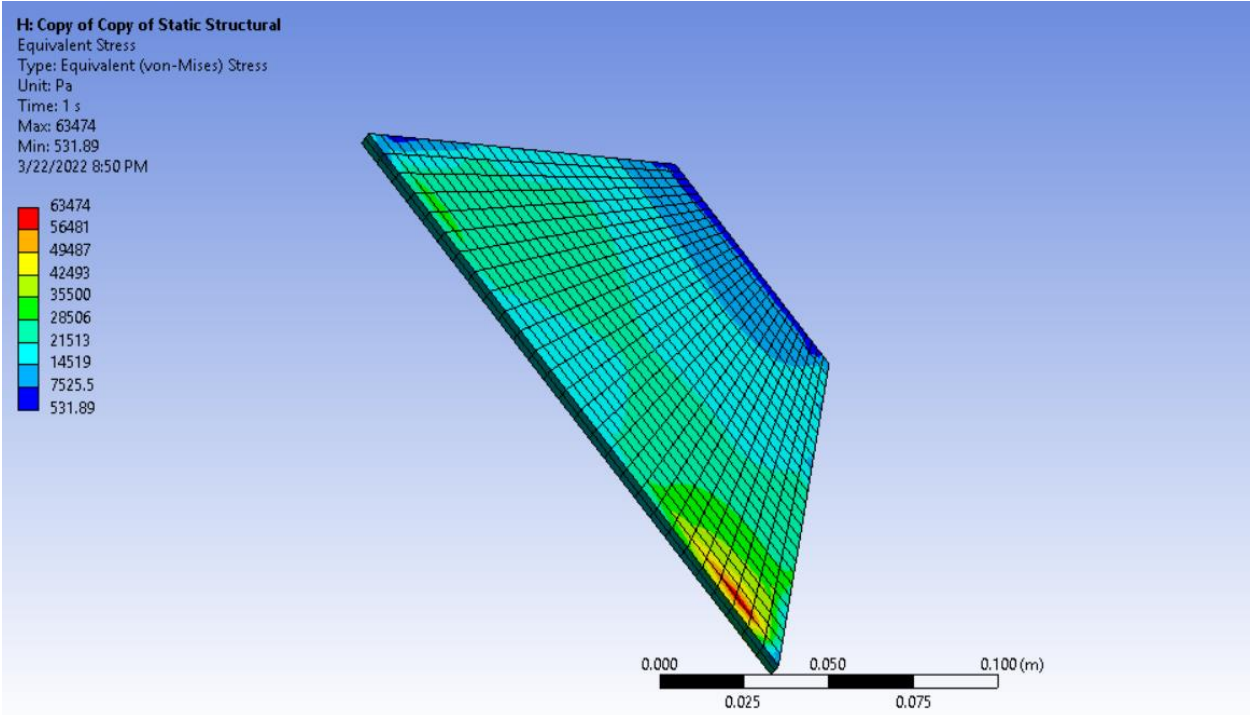


Figure 62: Fin von Mises Stress

### 2.2.2 Canard Aerodynamic Loads (Analysis Task 2)

Following the completion of the Fluent analysis of the aerodynamic loads on a single canard the team analyzed the results for the 2 given cases listed in Table 2.6. For both cases the results consisted of static pressure distribution on a single canard with the set conditions in Table 2.7 and respective case conditions. First, the results for Case 1 shown in Figure 63 were analyzed. As seen in Figure 63 the maximum stress on the canard for Case 1 is 19900 Pa along the leading edge with the maximum pressure on either canard face being 1550 Pa. Both canard faces have symmetrical pressure distributions due to the symmetry of the flow.

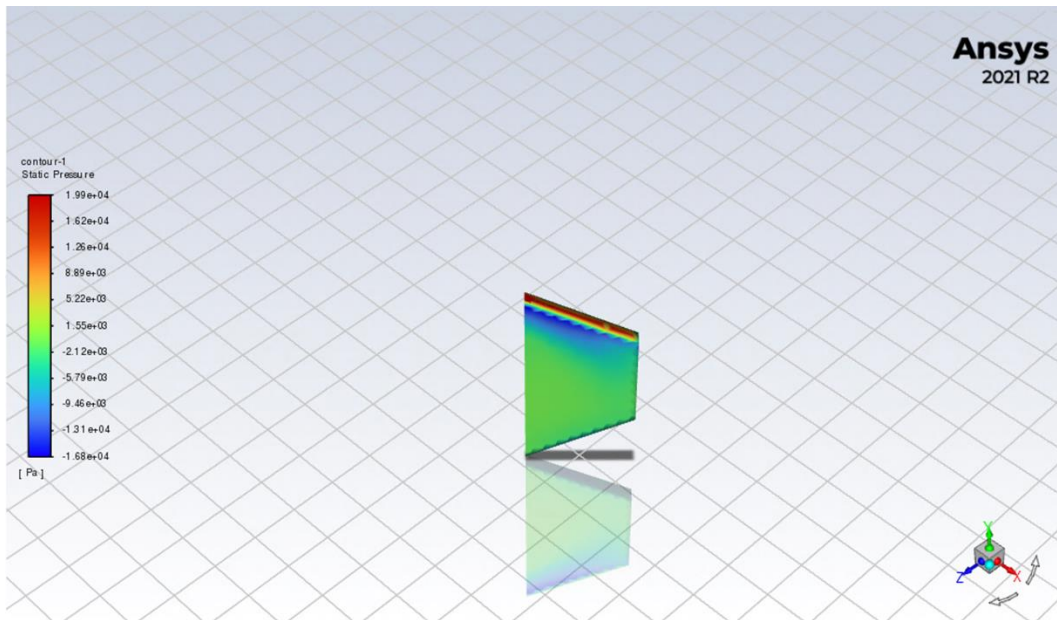


Figure 63 Resultant Static Pressure - Symmetric on both Faces (Case 1)

After completing the analysis of the results for Case 1, the results of Case 2 were analyzed to determine the maximum aerodynamic loads on a canard during flight. Case 2 was specifically completed at the maximum velocity of the rocket taken from the open Rocket model shown in Section 1.4 and at a canard angle of attack five degrees which would be the maximum angle of attack of a canard during ascent assuming zero rocket angle of attack. This assumption of zero angle of attack was made due to various factors, with the major factor being the asymmetry of the model. In Section 2.1.8 outlining the setup of this analysis the assumption of symmetry around the vertical axis of the rocket was made to create the simplified geometry of a canard. This assumed a rocket angle of attack of zero to ensure a symmetrical flow over the rocket body and canards. However, once a non-zero rocket angle of attack is introduced, the fluent model must be altered to allow for a different direction of airflow depending on the rocket's angle of attack. This type of flow symmetry was found to be too complicated to complete with a simplified geometry like used for case 2, requiring a full rocket body geometry in a more complex fluent model to gain accurate results. This type of analysis was deemed to be too similar to that of the analysis done in Section

3.1.4 and not necessary to complete the maximum canard stress distribution analysis in Section 2.1.9.

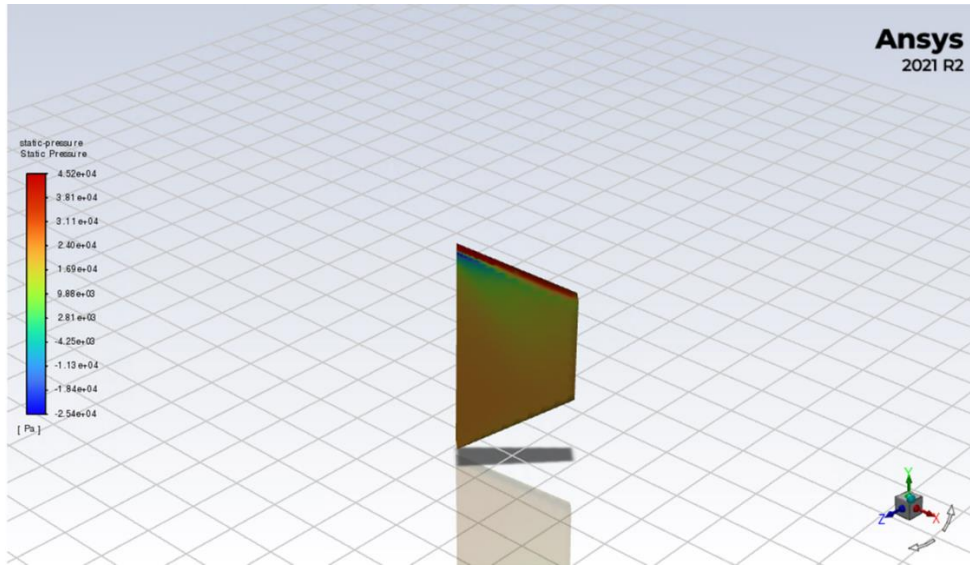


Figure 64 Resultant Static Pressure Distribution - Face angled out of Flow (Case 2)

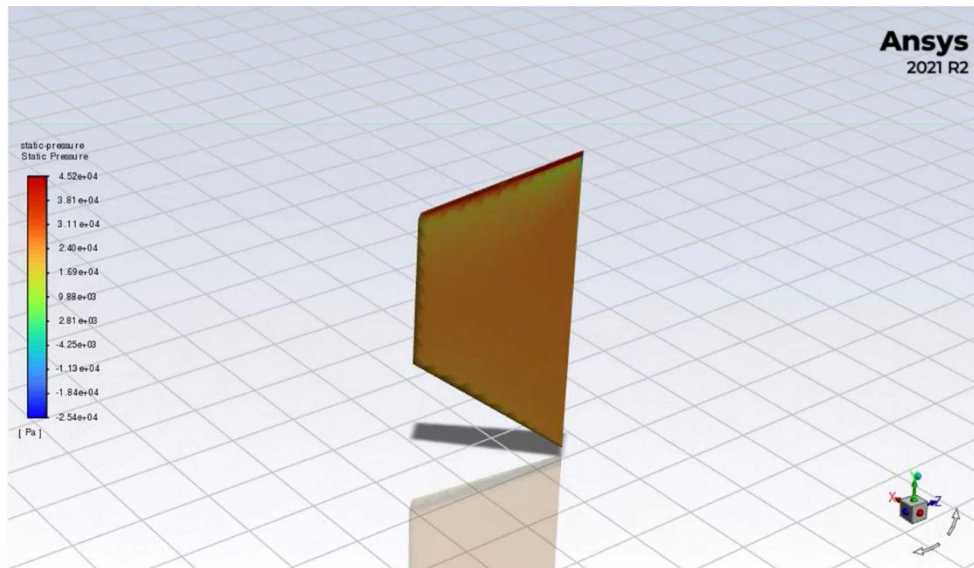


Figure 65 Resultant Static Pressure Distribution - Face Angled into Flow (Case 2)

The goal of the analysis of case 2 was to produce the maximum aerodynamic loads on a canard during flight to complete the analysis in Section 2.1.9. As seen in Figure 64 and Figure 65 the resultant maximum static pressure for a canard at the conditions in case 2 was about 45200 Pa

on the leading edge of the canard with the face of the canard angled into the flow experiencing a maximum pressure of about 38100 Pa. This resultant pressure distribution is as expected with the highest pressure being on the face exposed to the flow and the lowest pressure being on the face away from the flow.

### 2.2.3 Canard Stress Distribution (ARS Analysis Task 3)

The goal for the analysis in Section 2.1.9 was to find the stress distribution on a canard with maximum aerodynamic loads to then confirm the structural integrity of the canards and to understand the maximum stress on the canards fixture to the rocket during baseline flight. The stress distribution on the canard was found as the von Mises stress distribution, depicted in Figure 66. The maximum stress was found to 0.733 MPa and can be seen along the upper portion of the bottom face of the canard. This is far below the reported yield strength of 262 MPa for the fiberglass used to create the canards. This confirms the integrity of the canards at maximum angle of attack. The maximum stress was also considered when planning the fixture of the canards to the airframe for baseline flight. Failure of the canards fixture could cause unwanted roll or throw off the rockets planned flight path completely.

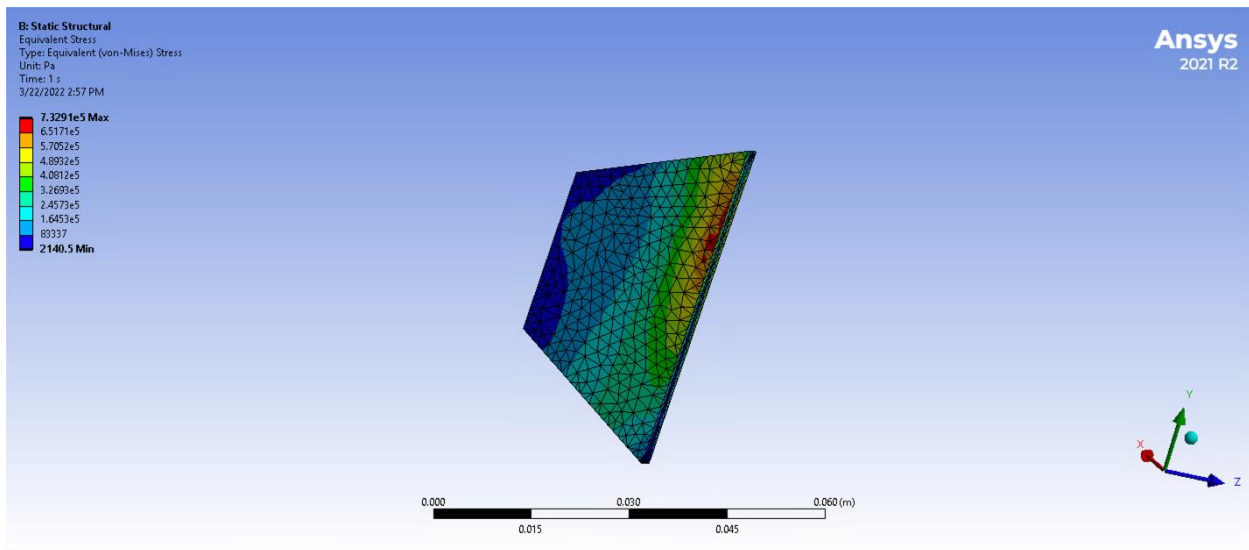


Figure 66 Canard Stress Distribution – Face Angled into Flow

### 3 Flight Dynamics Analysis Methodology and Results

#### 3.1 Methodology

##### 3.1.1 Avionics

To actively control the roll of our rocket as well as collect data, multiple electronic components are needed. Since there are a multitude of components being utilized to achieve this, our rocket uses a stackable board system, created by WPI's High Power Rocketry Club (HPRC). The stackable board system consists of several printed circuit boards (PCBs) stacked vertically with standoffs in-between, as depicted in Figure 67. This allows boards to be easily replaced if there is an issue, as well as save space. The avionics stack consists of four boards as outlined below in Figure 68.

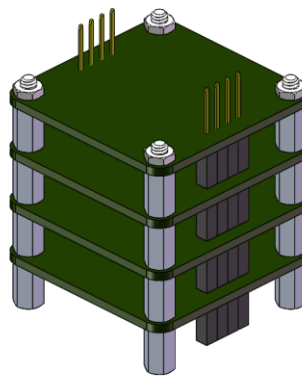


Figure 67 HPRC Avionics Stack

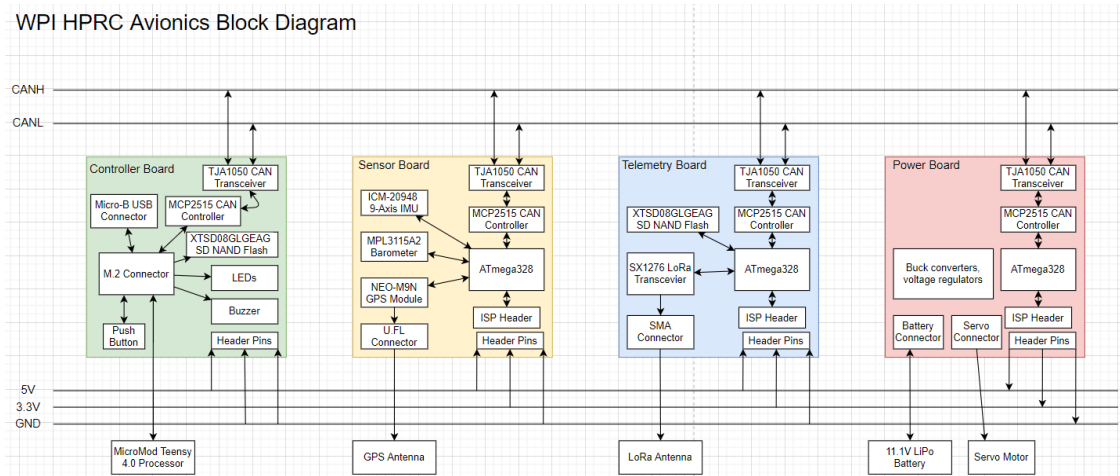


Figure 68 WPI HPRC Avionics Block Diagram of Respective Boards

The controller board's purpose is to perform computationally intensive tasks, such as state estimation and controls. It also receives data and sends commands to the other boards as well as

logs data onboard the rocket to the SD Nand Flash memory. Other major components consist of the Controller Area Network (CAN) Controller and Transceiver and the Teensy 4.0 microcontroller. This last component is the computer, which is programmed using the Arduino IDE, which uses a derivative of C++.

The sensor board provides sensor information for the rocket, as well as gathers this data to send to the ground station. Major components include a GPS module, a barometer, and a 9-axis Inertial Measurement Unit (IMU) which has an accelerometer, rate gyro, and magnetometer.

The telemetry board's primary purpose is to store and send a telemetry data string. It's second function is to assist in debugging system performance. It consists primarily of a LoRa transceiver module and another SD Nand Flash memory. The power board provides bus power to all other boards as well as controls the servo attached to the canard actuation system. Each of those tasks are controlled by their own separate power regulation system. A list of COTS items for the stack can be found in Table 3.1.

Table 3.1 COTS Included Products HPRC Avionics Stack

<b>Name</b>	<b>Function</b>	<b>Manufacturer</b>
MicroMod Teensy 4.0 [52]	Microcontroller	SparkFun
MCP2515 [53]	CAN Controller	Microchip
TJA1050 [54]	CAN Transceiver	Philips Semiconductors
ICM-20948 [55]	9-Axis IMU	TDK InvenSense
MPL3115A2 [56]	Barometer	NXP Semiconductors
NEO-M9N [57]	GPS Module	u-blox
ATmega328p [58]	Microcontroller	Atmel
XTSD08GLGEAG [59]	SD NAND Flash	XTX Technology Limited
SX1276 [60]	LoRa Transceiver	Semtech



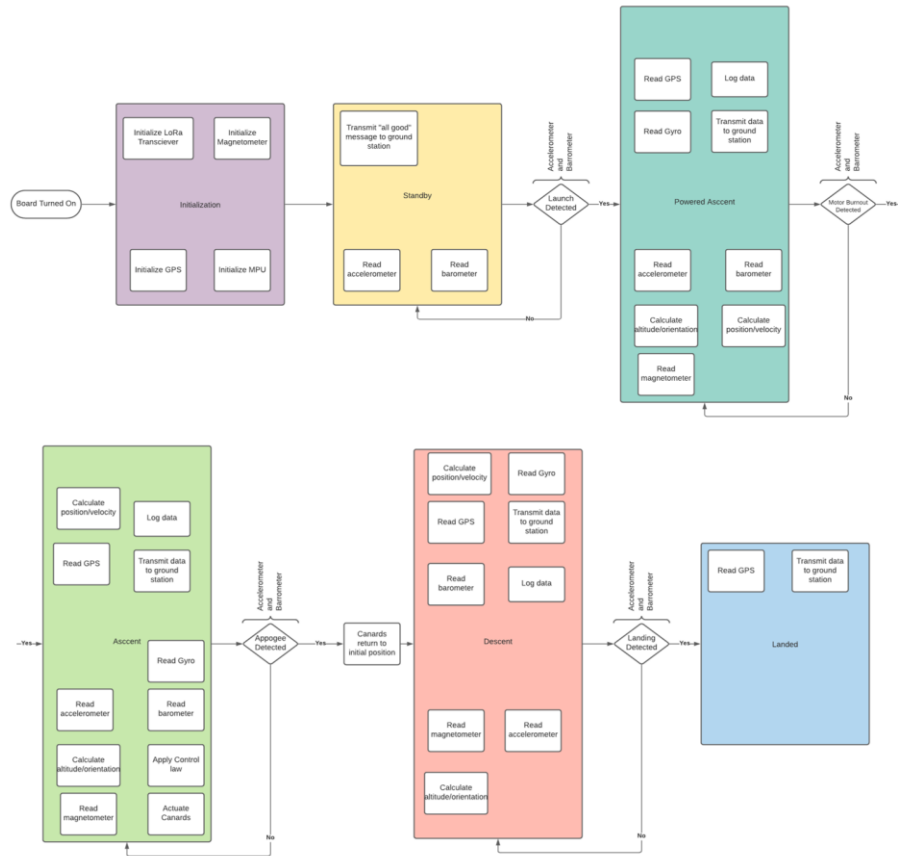


Figure 69 Phases of Launch Vehicle Flight

An outline of the logic that the avionics stack will follow during each phase of launch vehicle flight is shown in Figure 69. Most notably for our purposes, the canards would be allowed to actuate according to their control logic after motor burnout is detected and during ascent. The canards will be allowed to actuate only after motor burnout to help ensure the rocket is stable during motor burn. This happens until the avionics stack senses that the rocket has reached apogee. The canard actuation is then stopped and the canards themselves will be brought to their nominal stationary position parallel to the rocket body before the rocket enters the descent phase of flight.

### 3.1.2 Electronics Bay

The electronics bay is designed to house the payload, avionics board stack and battery, as well as serve as an attachment point for the main parachute. There are four bulk plates in total, each 0.25-inch thickness, except for the top plate. The top plate was designed with an 0.125-inch thickness but could be easily switched to a 0.25-inch plate. This is used to provide a stronger foundation for the canard system. All bulk plates are laser cut from plywood and have unique hole patterns dependent on what components mount to them.



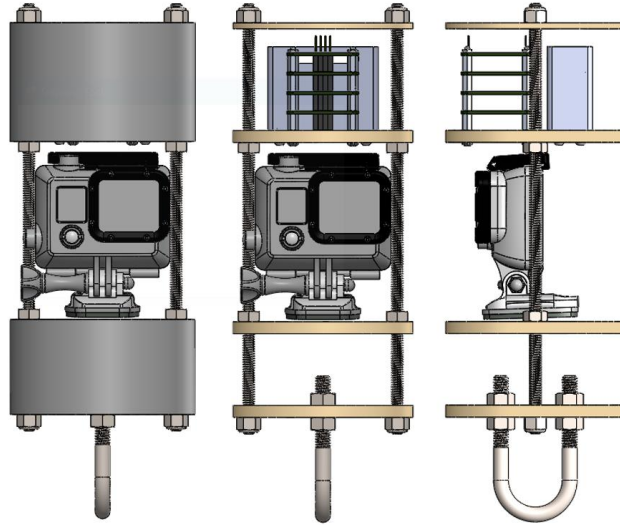


Figure 70 Electronics Bay Front and Side Views

Two 0.25-inch threaded rods were chosen as the attachment mode for these plates, as well as to provide the structure of the electronics bay and provide a way to transfer the recovery load between the plates. Plates slide onto the rods and attach to them with the use of nuts. Locknuts are used on the top and bottom for extra security. Two fiberglass coupler tubes create a more rigid structure, provide protection for the avionics, and allow the electronics bay to friction fit into the rocket. The main parachute shock cord attaches to the U-bolt at the bottom of the bay via a quick link. The payload, a camera, comes with a custom holder and mount that is attached to a plate through the use of a screw.

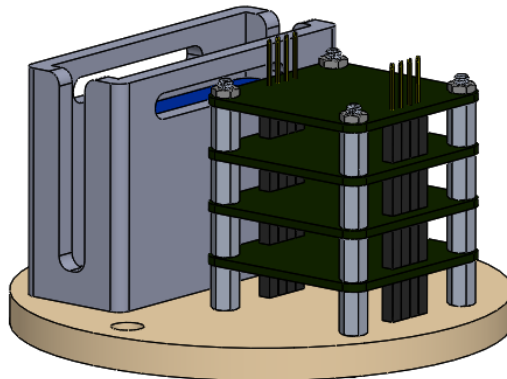


Figure 71 Avionics Plate Assembly

The custom avionics board stack and battery sit atop their own plate. The custom housing for the Turnigy nano-tech 450mAh 3S 65C LiPo battery is 3D printed out of Polylactic Acid

(PLA), since it is a non-load bearing component placed away from the motor. Four heat set inserts are placed at the bottom of the battery housing in order to mount to the wood plate. A slit at the top of the battery housing was made so that velcro could secure the battery in place during flight. An additional slit was made along each side in order for the battery wires to be easily accessible while the battery is in the housing.

### 3.1.3 Dynamic Simulator (FDA Analysis Task 1)

The variables used in the dynamic simulator are defined in Table 3.2.

Table 3.2 Flight Dynamics Model Variables

<b>Description</b>	<b>Variables</b>	<b>Units</b>
Position Vector	$\mathbf{p}$	m
Orientation Quaternion	$\mathbf{q}$	ul
Velocity Vector	$\mathbf{v}$	m/s
Angular Velocity	$\boldsymbol{\omega}^b$	rad/s
Force Vector	$\mathbf{F}$	N
Vehicle Mass	m	kg
Moment Vector	$\mathbf{M}$	Nm
Principal Moments of Inertia	$I_x^b, I_y^b, I_z^b$	kg m <sup>2</sup>
Gravitational Acceleration	$g$	m/s <sup>2</sup>
Surface Gravitational Potential	$\gamma$	m/s <sup>2</sup>
Thrust Vector	$\mathbf{T}$	N
Motor Misalignment Angle	$\theta$	rad
Atmospheric Density	$\rho$	kg/m <sup>3</sup>
Drag Coefficient	$C_d$	ul
Normal Force Coefficient	$C_n$	ul
Roll Moment Coefficient	$C_r$	ul
Reference Area	$A$	m <sup>2</sup>
Reference Length	$L$	m
Launch Heading	$\psi$	rad
Launch Angle (from horizontal)	$\theta$	rad
Launch Latitude	$\phi$	deg
Angular Velocity of Earth	$\boldsymbol{\omega}_e$	rad/s

### *Flight Dynamics Model*

The vehicle is represented with a state space model consisting of 13 state variables:

$$x = \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \\ \mathbf{v} \\ \boldsymbol{\omega}^b \end{bmatrix}$$

With  $\mathbf{p}$  as the 3-component position vector,  $\mathbf{q}$  as the 4-component quaternion orientation,  $\mathbf{v}$  as the 3-component velocity vector, and  $\boldsymbol{\omega}^b$  as the 3-component angular velocity vector. The position, quaternion, and velocity vectors are all defined relative to the inertial frame, while the angular velocity vector is defined relative to the body frame of the vehicle. Since the quaternion describes the orientation of the body in the inertial system, any inertial system can be defined, and the kinematic and dynamic equations can use the quaternion vector for rotations.

$$\mathbf{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \quad \mathbf{q} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad \mathbf{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \quad \boldsymbol{\omega}^b = \begin{bmatrix} \omega_x^b \\ \omega_y^b \\ \omega_z^b \end{bmatrix}$$

To propagate the variables forward in time, the team must define expressions for the kinematics and dynamics for the state variables. The translational and attitude kinematics of the vehicle can simply be defined by Equations (12) and (13) respectively.<sup>1</sup>

$$\dot{\mathbf{p}} = \mathbf{v} \tag{12}$$

$$\dot{\mathbf{q}} = \frac{1}{2} \mathbf{q} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}^b \end{bmatrix} \tag{13}$$

The translational dynamics equation requires support for both forces defined in the inertial frame ( $\mathbf{F}$ ), and those defined in the body frame ( $\mathbf{F}^b$ ). Translation dynamics are defined by Equation (14), with  $m$  taken as the instantaneous mass of the vehicle.<sup>2</sup>

$$\begin{bmatrix} 0 \\ \dot{\mathbf{v}} \end{bmatrix} = \frac{\mathbf{q} \otimes \begin{bmatrix} 0 \\ \mathbf{F}^b \end{bmatrix} \otimes \mathbf{q}^* + \begin{bmatrix} 0 \\ \mathbf{F} \end{bmatrix}}{m} \tag{14}$$

Finally, attitude dynamics can be defined by Euler's equations as shown in Equation (15). This formulation requires that the axes of rotation be aligned with the principal axes of inertia.

---

<sup>1</sup> $(\mathbf{u} \otimes \mathbf{v})_{ij} = u_i v_j$

<sup>2</sup>  $\mathbf{q}^* = -\frac{1}{2}(\mathbf{q} + \mathbf{i}q\mathbf{i} + \mathbf{j}q\mathbf{j} + \mathbf{k}q\mathbf{k})$

Applied moments and the vehicle's moment of inertia are defined in the body frame, which for a symmetric rocket meets this assumption.

$$\dot{\boldsymbol{\omega}}^b = \begin{bmatrix} \frac{M_x^b - (I_z^b - I_y^b)\omega_y^b\omega_z^b}{I_x} \\ \frac{M_y^b - (I_x^b - I_z^b)\omega_x^b\omega_z^b}{I_y} \\ \frac{M_z^b - (I_y^b - I_x^b)\omega_y^b\omega_x^b}{I_z} \end{bmatrix} \quad (15)$$

### *Forces acting on the vehicle*

Three primary forces act on the vehicle during flight – gravity, thrust, and aerodynamic forces. The gravitational force is calculated in the inertial reference frame and acts at the center of mass of the vehicle. For the flat earth model, gravitational acceleration is defined using Equation (16).

$$\mathbf{g} = \begin{bmatrix} 0 \\ 0 \\ 9.80665 \end{bmatrix} \left( \frac{m}{s^2} \right) \quad (16)$$

For simulations using the WGS84 ellipsoid, the surface gravitational potential ( $\gamma$ ) is calculated by the closed formula developed by Somigliana and adjusted for altitude using a Taylor series approximation [61]. The approximation is generally only meant for near-surface geodetic heights, though error from the exact formulation is on the order of  $10^{-4}$  m/s<sup>2</sup> up to 100 km, which is more than accurate enough for even simulations of space shot vehicles. The surface potential formula is given in Equation (17) and the altitude adjustment expansion is given in Equation (18).

$$\gamma = \gamma_e \frac{1 + k \sin^2(\phi)}{\sqrt{1 - e^2 \sin^2(\phi)}} \quad (17)$$

$$\gamma_h = \gamma \left[ 1 - \frac{2}{a} (1 + f + m - 2f \sin^2(\phi)) h + \frac{3}{a^2} h^2 \right] \quad (18)$$

Table 3.3 WGS84 Parameters [61]

Description	Variable	Value
Ellipsoidal Flattening Reciprocal	$1/f$	298.257223563
Ellipsoid Semi-major Axis	$a$	6378137.0 m
Ellipsoid Semi-minor Axis	$b$	6356752.3142 m
Theoretical Gravity at Pole	$\gamma_p$	9.8321849378 m/s <sup>2</sup>
Theoretical Gravity at Equator	$\gamma_e$	9.7803253359 m/s <sup>2</sup>
First Ellipsoidal Eccentricity Squared	$e^2$	$6.6943799901 \cdot 10^{-3}$
$\frac{b\gamma_p}{a\gamma_e} - 1$	$k$	0.00193185265241
$\frac{\omega^2 a^2 b}{GM}$	$m$	0.00344978650684

Variables and values for variables in the surface potential equations of the WGS84 coordinate system can be found in Table 3.3 [61]. The gravitational potential acts along the geodetic normal, or along the Z-axis of a North-East-Down (NED) tangent coordinate system, so it must be rotated to the inertial J2000 frame.

Thrust force is generated by the vehicle's motors, and its magnitude is defined by thrust curves available online from motor manufacturers, or from independent testing. The magnitude of the thrust is defined along the X-axis of the motor's coordinate system ( $\mathbf{T}^m$ ), which is in turn defined by a misalignment angle  $\theta$  about the Y-axis of the body fixed coordinate system. Motor misalignment generally arises due to manufacturing tolerances (e.g., misalignment of the motor mount tube) and can impact the flight dynamics of the vehicle significantly causing a constant pitching moment. The formula to determine thrust in the body fixed axes system is given by Equation (19).

$$\mathbf{T}^b = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} \mathbf{T}^m \\ 0 \\ 0 \end{bmatrix} \quad (19)$$

Aerodynamic forces are defined by a set of aerodynamic coefficients, determined using empirical formulae, CFD, or wind tunnel testing. The drag coefficient  $C_d$  creates a drag force along the X-axis of the body frame, while the normal force coefficient  $C_n$  creates a normal force acting in the Y-Z plane, with its direction dependent on the current roll angle of the vehicle relative to the wind velocity. The magnitudes of these forces are computed by the standard form given in Equation (20).

$$F = \frac{1}{2} \rho \|\mathbf{v}\|^2 C_{d,n} A \quad (20)$$

These coefficients are assumed to vary with velocity and angle of attack. Angle of attack is calculated by rotating the current air relative velocity vector of the vehicle into the body frame, then computing the unsigned angle between this vector and the body frame X axis. The angle between two 3D vectors  $a$  and  $b$  can be computed using Equation (21).

$$\theta = \text{atan2}(|a \times b|, a \cdot b) \quad (21)$$

### *Moments acting on the vehicle*

The moments applied to the vehicle are a result of both thrust and aerodynamic forces. The thrust forces were explained in the previous section, and the moment can be calculated by multiplying the forces by the moment arm to the C.G. for each vector component. Similarly, the normal force vector produces a moment about the C.G. due to the moment arm between the center of gravity and center of pressure.

This leaves the only the roll moment coefficient, which is further broken down into a roll forcing coefficient, and a roll damping coefficient. The roll forcing coefficient is taken to be a function of angle of attack and Mach number and generally is a result of fins or canards angled relative to the vehicles vertical axis. The roll damping coefficient is taken to be a function of roll rate and Mach number and is a result of the induced angle of attack seen at the fins due to the angular velocity of the vehicle. Both coefficients for a given vehicle can be determined again using empirical formulae, CFD, or wind tunnel testing. The roll moment coefficient is found by adding these two coefficients. The rolling moment is then found through a modified version of Equation (20), multiplying by the reference length of the vehicle as shown in Equation (22).

$$M = \frac{1}{2} \rho \|\mathbf{v}\|^2 C_r A L \quad (22)$$

### *Coordinate frames*

For a simulation over a flat earth, the simulator uses the NED frame as the inertial frame as shown in Figure 72.

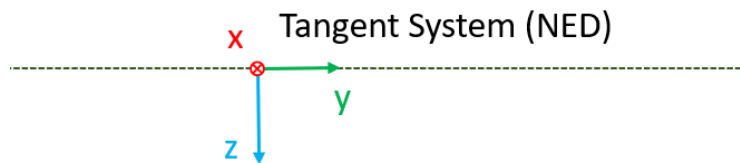
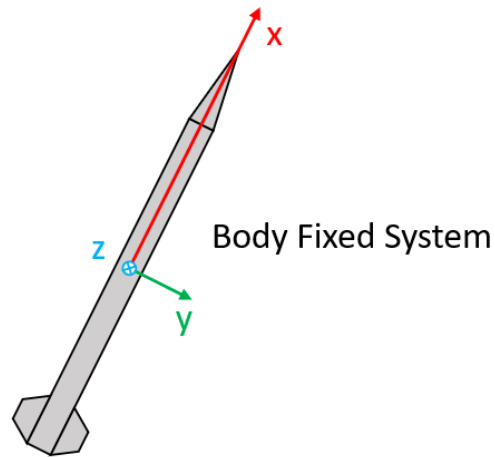


Figure 72 Flat Earth Simulation Coordinate System

The position of the center of mass is defined in the tangent frame, and the body fixed system is used to represent the orientation of the vehicle. The X-Axis is taken as the longitudinal or roll axis of the vehicle, with the Y and Z axes corresponding to the pitch and yaw of the vehicle respectively. When simulating using the ECI frame, the body axes remain the same, but the inertial reference frame is replaced with the right-handed ECI frame located at the center of the Earth.

### *Simulator Architecture*

The simulator is written in MATLAB and uses an object-oriented structure to simplify development and future expansion. Object-oriented programming is built upon classes, which are user-defined data types that contain properties and functions, referred to as “methods”. When using the program, the user can create multiple instances of a given class with different properties, which are known as objects. Properties in turn are variables linked to a specific object. Each object will also have a set of methods, which are functions defined within the class that can access the properties of the object directly.

As a simple example, consider the case of a piece of software designed to track students at a school. The *Student* class would store properties and methods. Such properties might include

name, birth date, and grade. A basic method could be a function that calculates the current age of the student, using the birth date property. Each student at the school would be assigned a unique object with all their information, which could be accessed or changed as needed. Such a structure increases modularity by making independent objects self-contained and can easily be scaled by simply creating more instances.

The dynamic simulator contains 5 classes, defined with their key properties and methods in Table 3.4.

Table 3.4 Dynamic Simulator Classes

Class	Properties	Methods
Rocket	Dry Mass Properties Aerodynamics Object Motor Objects	Compute Total Thrust Compute Total Mass Properties
Motor	Mass and Thrust Curves Motor Statistics	Load Motor Data Compute Thrust Compute Mass Properties
Aerodynamics	C.P. Position Aerodynamic Coefficient Functions	Load Aerodynamic Data Compute Aerodynamic Coefficients
Environment	Launch Time Launch Location Geoid Model Atmosphere Model Weather Model	Compute Gravitational Acceleration Set Launch Site Elevation
Simulator	Launch Angle and Heading Simulation Type Timestep / Maximum Time Initial State Vector	Set Initial Conditions Run Simulation Compute Forces and Moments Plot Results

*Rocket* contains vehicle properties such as dry mass and moments of inertia, and methods to calculate wet masses and moments of inertia and thrust as they vary during motor burn. The *Rocket* class takes inputs from both the *Motor* and *Aerodynamics* classes.

*Motor* contains motor properties such as wet and dry mass, the thrust and mass curves of the motor, and methods to calculate the thrust and mass of the individual motor as it burns.

*Aerodynamics* contains aerodynamic properties of the vehicle and methods to calculate aerodynamic coefficients given Mach number, angle of attack, and roll rate.

*Environment* contains properties of the simulation environment such as launch position, elevation, and time. Also stores the geoid, atmosphere, and weather models to be used in the simulation. Contains methods to calculate gravitational acceleration given a time and state vector.

*Simulator* contains simulation properties such as launch angle and heading, rail length, and the desired simulation type. Contains the main simulation method which runs a given case, as well



as methods to calculate initial conditions, state derivative methods for the different simulation types, and methods to rotate vectors between coordinate systems. This class also contains methods to plot results. Takes inputs from both the *Rocket* and *Environment* classes.

### *Running a simulation*

To run a simulation, the user first creates an instance of the rocket class. The *Aerodynamics* class can then be instantiated (the process of creating an instance of a class) and passed to the *Rocket* class, as can one or more instances of the *Motor* class. With each motor added to the vehicle, the user can define different start conditions to support multiple stages or motor clusters. The user then creates an instance of the *Environment* class and selects the models to be used. Finally, the user passes these classes into the *Simulator* class and runs the simulation.

The simulator supports multiple simulation models the user can choose from. The simulation type can be selected from either 3DOF translation with no rotation, 3DOF rotation, or a full 6DOF model. When either 3DOF model is used the derivatives of the unused states are set to zero.

In the *Environment* class, the geoid model can be selected from either *none*, where no gravitational forces are defined, *flat* where the flat earth gravity model described previously is used, or *wgs84* where the WGS84 gravity model is used. The weather model can be selected from either *none* where no atmosphere is modeled, *isa* where the atmosphere model is based on the International Standard Atmosphere or *forecast* where forecast or historic atmospheric data for the launch date and time will be retrieved from NOAA's servers. Similarly, the weather model can be selected as *none* where no wind will be modeled, *constant* where the user can select a desired constant wind speed or *forecast* where forecast or historic winds aloft data is used.

With initial conditions defined in various frames, the simulator must compute the initial state vector. For a flat or no geoid model, initial position, velocity, and angular velocity are all set to 0. The initial orientation relative to a NED system is calculated based on the defined launch angle and heading by the Euler angle to quaternion conversion shown in Equation (23).

$$\mathbf{q} = \begin{bmatrix} \cos(\psi/2) \\ 0 \\ 0 \\ \sin(\psi/2) \end{bmatrix} \begin{bmatrix} \cos(\theta/2) \\ 0 \\ \sin(\theta/2) \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (23)$$

For the WGS84 model, the position is found by converting the launch latitude, longitude, and elevation to ECEF, then to J2000 coordinates. The conversion from geodetic to ECEF coordinates is given by Equation (24).

$$\mathbf{p} = \begin{bmatrix} (N(\phi) + h) \cos(\phi) \cos(\lambda) \\ (N(\phi) + h) \cos(\phi) \sin(\lambda) \\ \left(\frac{b^2}{a^2} N(\phi) + h\right) \sin(\phi) \end{bmatrix} \quad (24)$$

where:

$$N(\phi) = \frac{a}{\sqrt{1-e^2 \sin^2(\phi)}} = \text{prime vertical radius of curvature}$$

The initial velocity of the vehicle is a result of the rotation of Earth about its axis. The velocity in the J2000 frame is therefore given by Equation (25).

$$\mathbf{v} = \boldsymbol{\omega}_e \times \mathbf{p} \quad (25)$$

The initial orientation of the vehicle is described by first computing the orientation in the local tangent frame, then rotating this orientation to the J2000 frame. Finally, the initial angular velocity is set to the angular velocity of the earth rotated to the body coordinate system.

With initial conditions defined, a custom implementation of the 4<sup>th</sup> order Runge-Kutta method is used to numerically propagate the vehicles state over time. A custom solver was chosen over MATLAB's built-in solver to pass parameters about the vehicle into the function more easily without the need for global variables, and to enable a detection of flight events with more complex triggers. At each timestep, forces and moments applied to the vehicle are calculated as described previously based on the current time and state of the vehicle and passed into the translation and attitude dynamics functions to compute state derivatives. The simulator can be configured with different stop points, including rail exit, burnout, apogee, or ground impact.

#### 3.1.4 Aerodynamic Loads (FDA Analysis Task 2)

As discussed previously, the team selected Ansys Fluent as the primary software used to simulate the aerodynamic loads on the rocket. A more detailed description of the general use of Fluent, including solver types, general assumptions, and boundary types, can be found in Section 2.1.7. For FDA Analysis Task 2, the simulation focused on the aerodynamic loads on the entire rocket body during ascent. An overview of the process of setting up an external flow in Fluent can be found in Figure 73.

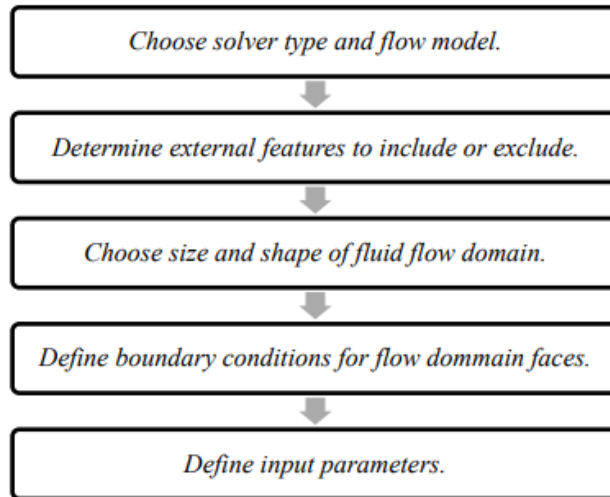


Figure 73 Setup of Ansys Fluent External Flow Simulation [37]

Given the speed of the air flow around the rocket during its ascent, the flow was considered incompressible. As a result, the team used the pressure-based solver in Fluent for this simulation. Additionally, because the flow is considered turbulent, a viscous model must be used. The  $k-\omega$  model was selected to simulate the aerodynamic loads due to its ability to examine boundary layer separation more accurately.

Once the solver type and flow model were selected, the external geometry of the rocket was modeled in Fluent. Since both the baseline and innovative designs of our rocket had identical external geometries, only one model needed to be created. To obtain an accurate model of the external geometry, the team modelled the rocket using SOLIDWORKS. The CAD model used for the aerodynamic simulations was simplified such that the fins and canards were flush with the main tube of the rocket body. All other minor external features were neglected, as they would have very little impact on the simulation results while immensely increasing mesh complexity and simulation run time. This simplified model was saved as a single continuous part, as shown in Figure 74, and imported to Ansys SpaceClaim, a modeling software tool attached to Ansys Workbench.



Figure 74 Model of External Rocket Geometry

In SpaceClaim, an enclosure was modeled around the simplified rocket geometry to serve as the fluid flow domain for the simulation. This enclosure is necessary to provide a finite restriction on the flow domain, as Fluent's pressure-based solver was not designed to simulate flows with infinite bounds. The rocket model itself was then suppressed, leaving a "mold" of the rocket geometry in the enclosure. This was done because the Fluent simulation examines the interaction between the air flow and the rocket geometry, not the rocket structure.

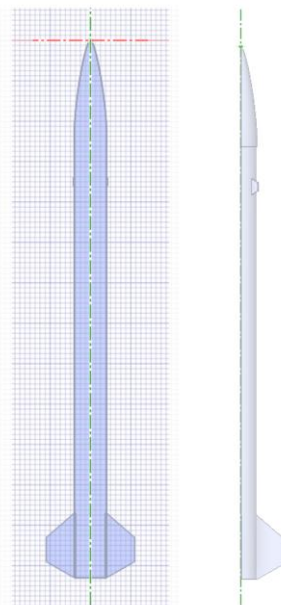


Figure 75 Rocket Geometry Plane of Symmetry

Additionally, the rocket geometry displays symmetry across a plane through its vertical axis when the fins and canards are oriented accordingly, as shown in Figure 75. This allows the cylindrical enclosure to be bisected along this plane of symmetry, reducing the flow domain size by half. This reduces the complexity of the mesh and the simulation overall. With this simplification, the aerodynamic loads were only simulated on one half of the rocket. The resulting loads were then mirrored to the other half of the rocket to produce a full view of the aerodynamic loads on the entire rocket body. This assumption of symmetry holds for non-zero angles of attack provided the inlet air flow was defined properly. The component of the inlet velocity normal to the plane of symmetry must be zero, resulting in the velocity vector of the inlet airflow being parallel to the plane of symmetry of the rocket. The fluid domain can be seen in Figure 76. The dimensions of the fluid domain were selected based on where relevant flow disturbances would occur close to the rocket. Through running several preliminary test cases it was determined that the domain needed to extend 20 meters below the bottom of the rocket, and it needed to enclose a radius of 0.2 meters around the sides of the rocket. During the test cases, the dynamic and static pressure contours were examined to ensure that any disturbances or deviations from the free stream pressure were enclosed by the flow domain. If the domain were made to be too narrow or too short, then the resulting aerodynamic coefficients and pressure data would be much less accurate.

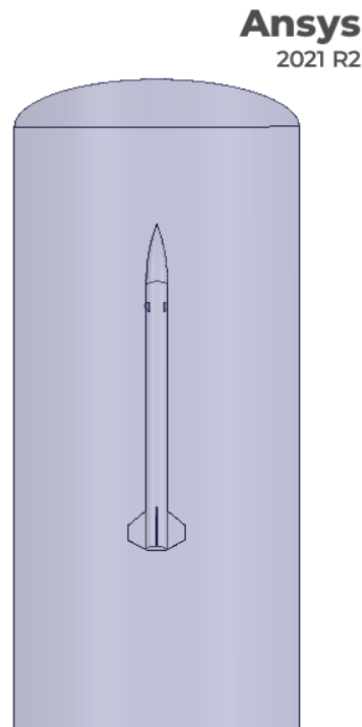


Figure 76 Fluid Flow Domain

Once modeled, the flow domain was transferred to Fluent to design a mesh. This was a critical step in the simulation process, as the quality of the mesh has a direct effect on the accuracy

and validity of the results. The design of the fluid domain mesh was performed iteratively, starting with a crude automatic mesh generated by Fluent. After each iteration, updates were made to the mesh until it was determined to be of high enough quality, meaning it had a maximum aspect ratio on the order of 10:1. In this context, the aspect ratio refers to the “stretching” of the mesh cells. In an ideal system, the aspect ratio would be roughly equal to 1, meaning the cells are generally uniform shapes such as equilateral triangles or squares. However, in more realistic systems, the cell stretching should be minimized. One key update to the mesh was the addition of boundary layer cells around the walls of the rocket geometry. These cells served to more finely discretize the flow field immediately next to the rocket body, providing a better simulation of any boundary layer effects on the aerodynamic loads. The final mesh of the fluid domain can be seen in Figure 77.

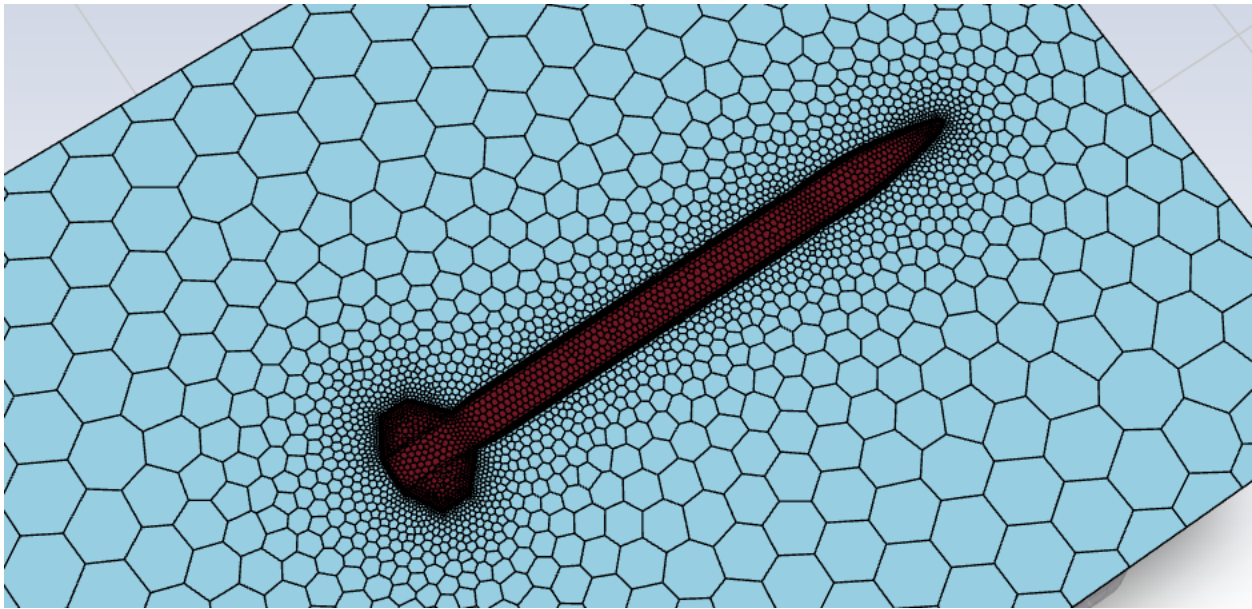


Figure 77 Mesh of Fluid Domain

Following the mesh design, boundary conditions were assigned to each face of the fluid domain. The top surface and bottom surface of the domain were defined as a velocity inlet and a pressure outlet, respectively, while the flat face of the half-cylinder was defined as a symmetry boundary. The curved face was defined as a velocity inlet with the velocity vector of the airflow identical to the vector defined for the velocity inlet on the top surface. The curved face was considered an inlet to accurately model non-zero angles of attack. When the velocity vector of the flow field was angled with respect to the rocket body, air needed to flow through this surface of the domain and interact with the rocket as a uniform stream. The velocity inlet boundaries used were defined based on the rocket orientation, meaning the magnitude and direction of the flow with respect to the rocket was updated for each case. The velocity inlets were not considered normal flow inlets, as this would conflict with the physical characteristics of the system.

Table 3.5 presents the material properties assigned to the fluid domain as well as the rocket body. Fluent provides very limited choices when assigning material properties to solids, so aluminum was selected to model the walls of the rocket body to present smooth walls. The physical characteristics of the solid material have no effect on the simulation, since Fluent only examines the fluid flow around the solid body. Table 3.6 presents the boundary type and boundary conditions assigned to each surface of the domain.

Table 3.5 Material Properties of Fluid Domain and Rocket Body

<b>Domain</b>	<b>Material Assigned to Domain in Fluent</b>	<b>Material Properties</b>	<b>Description</b>
Fluid Flow	Air	$\rho = 1.275 \text{ kg/m}^3$ $\mu = 1.715 \cdot 10^{-5} \text{ kg/(m}\cdot\text{s)}$	Density and viscosity of air at sea level and 0°C were chosen to more accurately reflect flight conditions
Rocket Body	Aluminum	$\rho = 2719 \text{ kg/m}^3$ $K_s = 0$	Aluminum density plays no role in CFD simulation, $K_s$ is the roughness height of the material with 0 corresponding to smooth walls

Table 3.6 Boundary Type and Conditions at Domain Surfaces

Boundary Condition	Input Values/Equations	Domain Affected	Description
Velocity Inlet	$V = 20 \text{ to } 180 \text{ m/s}$ $\rho = 1.275 \text{ kg/m}^3$ $\alpha = 0 \text{ to } 20^\circ$	Fluid	Flow velocity magnitude and direction (relative to origin) and air density defined at inlet at top face of fluid domain
Velocity Inlet	$V = 20 \text{ to } 180 \text{ m/s}$ $\rho = 1.275 \text{ kg/m}^3$ $\alpha = 0 \text{ to } 20^\circ$	Fluid	Flow velocity magnitude and direction (relative to origin) and air density defined at inlet at curved cylindrical face of fluid domain
Pressure Outlet	$\partial p / \partial r = (\rho v \theta^2) / r$ $\rho = 1.275 \text{ kg/m}^3$ $P_{\text{gauge}} = 0 \text{ atm}$	Fluid	Expected outlet static (gauge) pressure defined at outlet on bottom face of fluid domain in order to compute pressure gradient
Wall	$\tau = \mu(\partial v / \partial n)$ $\mu = 1.715 \cdot 10^{-5} \text{ kg/(m}\cdot\text{s)}$	Fluid, Rocket Body	Flow viscosity defined at wall boundary between fluid and rocket body in order to predict shear stresses
Symmetry	N/A	Fluid, Rocket Body	A symmetry boundary requires no inputs, it simply mirrors the simulated geometry and flow patterns

The range of inlet flow velocities is based on the expected range of flight speed, with the estimated maximum speed of the rocket being about 185 m/s, as discussed in Section 2.1.7. The team ran simulations from 20 m/s to 180 m/s in intervals of 40 m/s. Similarly, multiple cases were examined for varying angle of attack, with simulations being run from 0° to 20° in intervals of 2°. Although the rocket would be angled with respect to the air flow, the opposite was modeled in Fluent. The process of rotating the rocket geometry within the flow domain would have been significantly more difficult and time consuming. Because of this, the flow velocity vector was rotated with respect to the body-fixed frame of the rocket.

### 3.1.5 Control System (FDA Analysis Task 3)

The process for the creation of the active control system chosen for the rocket began with deciding which element of the rocket needed to be controlled. The roll of the rocket was desired



to be limited to zero in order for the payload to have a better opportunity for a higher quality video throughout the rocket’s flight. Research was conducted on the subject of roll control for model rockets since past MQPs have relied mostly on passive control systems. A similar project was undertaken by the Portland State Aerospace Society (PSAS) [62]. The control logic was chosen to be a PID control system, as mentioned in Section 1.5.2. MATLAB and Simulink were used to test the controls system before implementation in the rocket.

*Simulink Model*

The Simulink model was based off of MathWorks’ own example of a PID Simulink model [63] and can be seen in Figure 78.

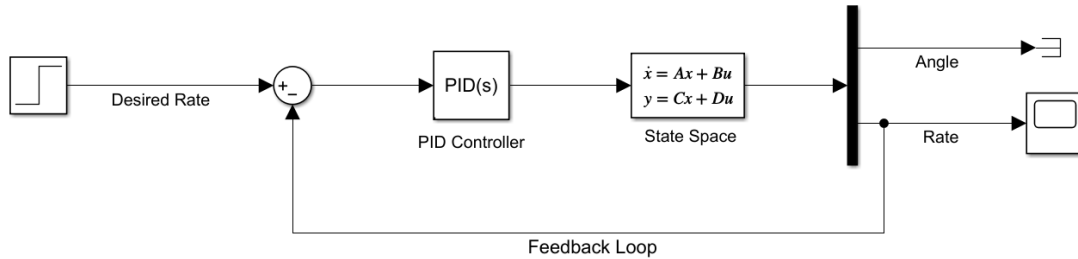


Figure 78 Simulink PID Model

The desired roll rate of zero is represented by the step box, the leftmost component. Moving right, this is where the feedback loop is defined. The PID component box is what our team used Simulink for. Once the state space is included along with other data, the PID box can automatically tune, or calculate, each of the P, I, and D coefficients to obtain the desired outcome using Equation (1) in Section 1.5.2. The user can also decide how aggressive to make the controller in the tune setting. The state space is modeled as  $x_{n+1}$  and  $y_n$ . The matrix definitions are given by:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \\ I_{yy} \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad D = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \text{Initial Conditions} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

The state space above is a modified version of the linearized state space model given in Equation (26). This assumes that all canards are being actuated at the same time and being moved to the same angle of attack. For our purposes, the sum of the force was multiplied by the distance from the center of mass the force was acting over and replaced by the moment it creates as seen in Equation (27). Variable definitions for both can be seen in Table 3.7.

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \left\{ \begin{array}{c} 0 \\ \frac{l_f}{I_{yy}} \end{array} \right\} u \quad (26)$$

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \left\{ \begin{array}{c} 0 \\ \frac{l_f}{I_{yy}} \end{array} \right\} mom \quad (27)$$

Table 3.7 State Space Variables

Variable	Definition
$u$	Sum force of canards acting to rotate the rocket
$l_f$	The distance at which the force acts
$I_{yy}$	Moment of inertia about the y-axis
$mom$	Moment created by canards

The team's SOLIDWORKS model of the rocket was used to acquire the moments of inertia for this state space, as listed in Table 3.8. The moments of inertia were calculated relative to the center of mass and aligned with the output coordinate system using the positive tensor notation.

Table 3.8 Moments of Inertia

Moments of Inertia (kg m <sup>2</sup> )		
$I_{xx} = 2126455.04$	$I_{xy} = 1617.06$	$I_{xz} = 42.18$
$I_{yx} = 1617.06$	$I_{yy} = 16522.14$	$I_{yz} = -731.19$
$I_{zx} = 42.18$	$I_{zy} = -731.19$	$I_{zz} = 2126799.54$

### Obtaining Coefficients

Using the OpenRocket model of our rocket, the software was adjusted manually to acquire roll forcing coefficients (RFC) and roll damping coefficients (RDC), as described in Section 3.1.3 for various canard cants (angles of attack), Mach numbers, and roll rates, as seen in Table 3.9 and Table 3.10 respectively.

Table 3.9 Roll Forcing Coefficient

Roll Forcing Coefficient								
Canard Cant (deg)	Mach Number							
	0	0.1	0.2	0.3	0.4	0.5	0.55	
0	0	0	0	0	0	0	0	
1	0.016	0.016	0.016	0.016	0.016	0.016	0.016	
2	0.032	0.032	0.032	0.032	0.032	0.032	0.033	
3	0.048	0.048	0.048	0.048	0.048	0.049	0.049	
4	0.064	0.064	0.064	0.064	0.065	0.065	0.065	
5	0.079	0.080	0.080	0.080	0.081	0.081	0.082	

Table 3.10 Roll Damping Coefficient

Roll Damping Coefficient								
Roll Rate (r/s)	Mach Number							
	0	0.1	0.2	0.3	0.4	0.5	0.55	
0	0	0	0	0	0	0	0	
1	9.0	0.61	0.31	0.21	0.17	0.14	0.13	
2	9.0	1.2	0.62	0.43	0.33	0.28	0.27	
3	9.0	1.8	0.94	0.64	0.50	0.42	0.40	
4	9.0	2.5	1.2	0.85	0.67	0.56	0.53	
5	9.0	3.1	1.6	1.1	0.83	0.71	0.66	

These tables of data were then plotted to produce Figure 79 through Figure 82. Using MATLAB, a poly fit curve command was applied to each graph to obtain an equation that would help define each coefficient as it relates to our rocket. Equations (28) and (29) were found to be the results of the fit curve with an  $r^2$  value of 0.9999 and 0.9495, respectively. The variables associated with these equations can be found in Table 3.11.

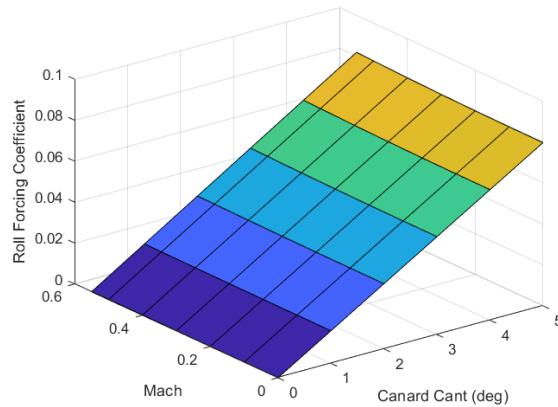


Figure 79 Roll Forcing Coefficient vs. Mach and Canard Cant

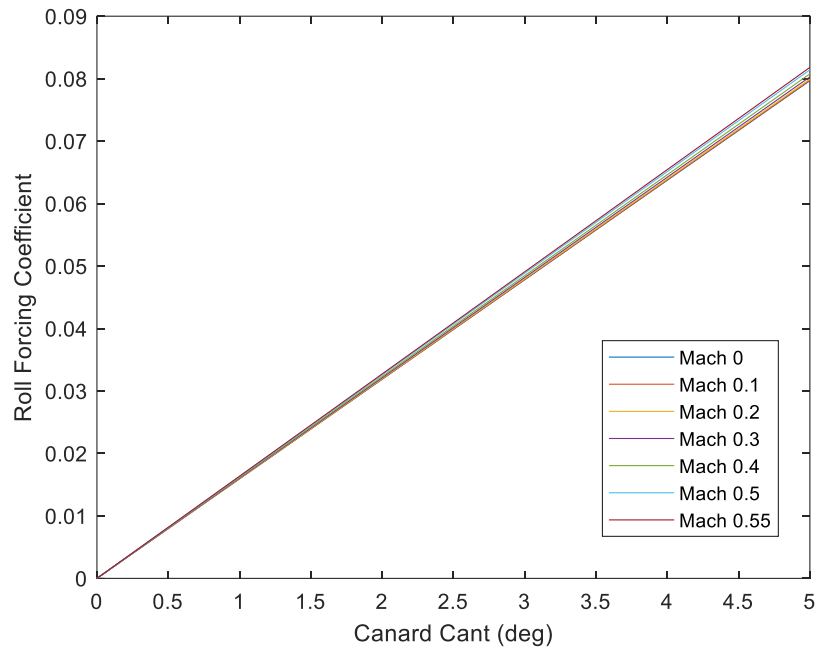


Figure 80 Roll Forcing Coefficients 2D Plot

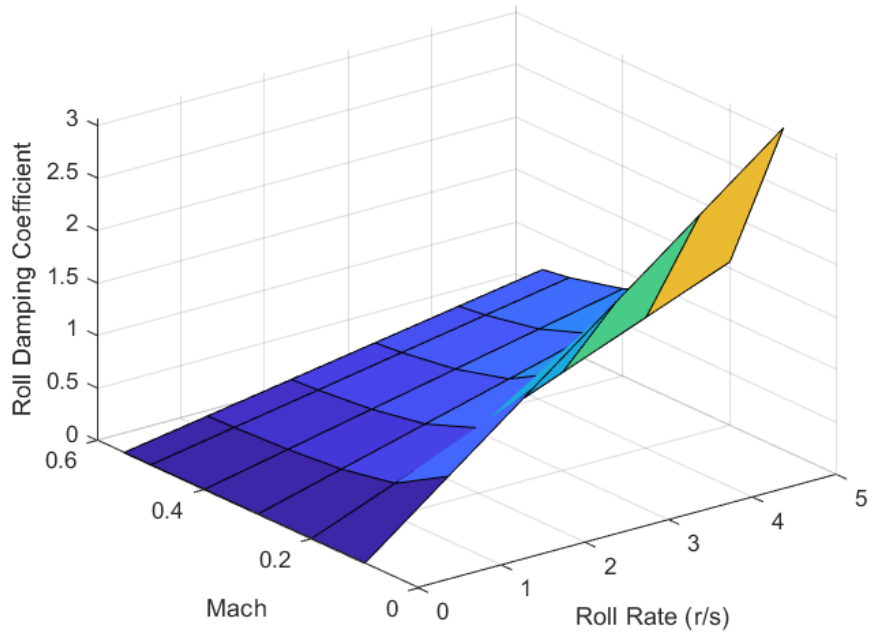


Figure 81 Roll Damping Coefficients vs. Mach and Roll Rate

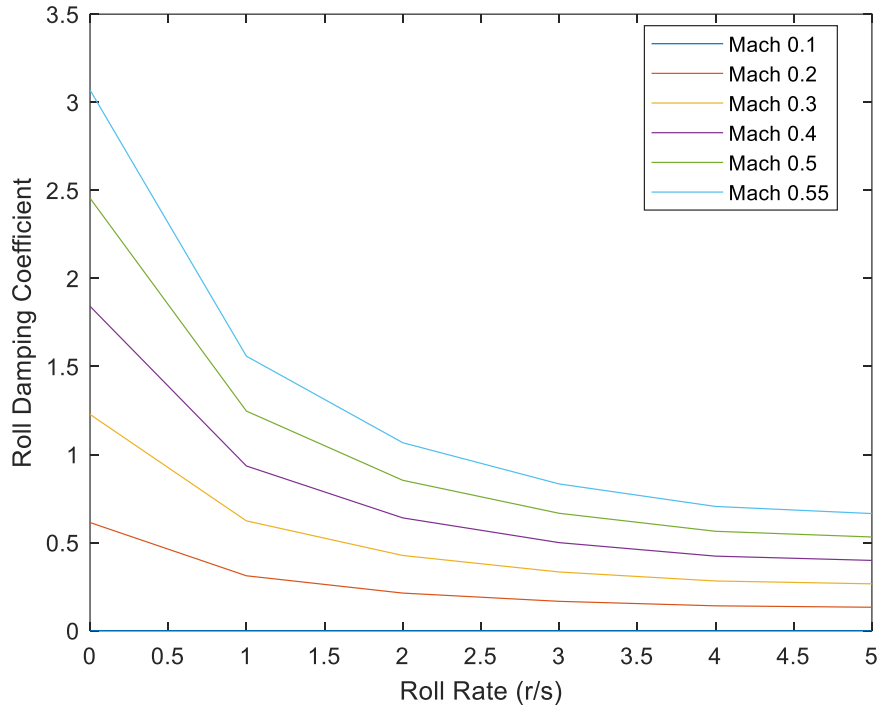


Figure 82 Roll Damping Coefficients 2D Plot

$$RFC = -0.0005765 + 0.0161 \cdot c + 0.001969 \cdot M \quad (28)$$

$$RDC = 0.7585 + 0.5815 \cdot r - 6.029 \cdot M - 0.9314 \cdot r \cdot M + 9.16 \cdot M^2 \quad (29)$$

Table 3.11 Variable Definitions for Roll Coefficients

Variable	Definition
$c$	Canard Cant Angle
$M$	Mach Number
$r$	Roll Rate

### State Space

Although a linearized state space was used to obtain the proportional, integral, and derivative coefficients for the PID controller, a non-linearized state space was needed for the simulation of the canards. The state space, denoted as  $x$ , and the derivative of the state space, denoted as  $\dot{x}$ , are defined as:

$$x = \begin{bmatrix} \textit{Vertical Position} \\ \textit{Velocity} \\ \textit{Angular Position} \\ \textit{Angular Velocity} \end{bmatrix}, \dot{x} = \begin{bmatrix} \textit{Velocity} \\ \textit{Acceleration} \\ \textit{Angular Velocity} \\ \textit{Angular Acceleration} \end{bmatrix}$$

### Forces and Other Variables

Other parameters need to be taken into account for the simulation of the control logic. Constants used for the simulation can be found in Table 3.12. All values in the table were taken from the OpenRocket model, aside from the moment of inertia and gravitational acceleration.

Table 3.12 Constants and Variables for Control Code

Definition	Variable	Value	Unit
Mass of Rocket	m	5.1596	Kg
Moment of Inertia	I	1.6522E+04	$Kg/m^2$
Total Drag Coefficient	Cd	.57	-
OpenRocket Reference Area	A	.0082	$m^2$
OpenRocket Reference Length	L	.1022	m
Gravitational Acceleration	g	9.80665	$m/s^2$
Velocity	v	-	m/s
Density of Air	rho	-	$Kg/m^3$
Speed of Sound	s	-	m/s

Using the MATLAB function *atmosisa*, the density of the air and speed of sound were able to be obtained dependent on the position of the rocket in its flight. The speed of sound and velocity were then used to determine the Mach of the vehicle using Equation (30). Two forces were taken

into account: gravitational and drag as seen in Equations (31) and (32) respectively. The combination of the two amounted to the total force.

$$Mach = v/s \quad (30)$$

$$Gravitational\ Force = -g \cdot m \quad (31)$$

$$Drag\ Force = -0.5 \cdot rho \cdot C_d \cdot A \cdot v^2 \quad (32)$$

### *Moments*

The total moment can be broken into the roll damping and roll forcing moments. Rearranging Equation (29), the roll damping moment can be found using Equation (33). Since the roll forcing moment is the moment dependent on the canard cant, this is the moment that the control system uses as the PID control variable. This means that the roll forcing moment is calculated from the PID equation, Equation (6) in Section 1.5.2 and is re-represented in Equation (34).

$$Roll\ Damping\ Moment = 0.5 \cdot rho \cdot RDC \cdot L \cdot A \cdot v^2 \quad (33)$$

$$Roll\ Forcing\ Moment = - (K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt}) \quad (34)$$

### *Nonlinear Model*

The model created to simulate the control of the canards consisted of three main parts. The first function, *xdot2*, contained the constants, forces, moments, PID values, and coefficients. This was where the roll forcing moment was used to identify what canard angle was desired through the use of Equation (28). A Runge–Kutta method was utilized in a function title *rk4* to approximate the states using numerical integration throughout the iterative process. The body of code that utilized these two functions was called *MainLoop* and was responsible for the *rk4* loop and plotting, as well as the input of constants and initial conditions.

#### 3.1.6 Electronics Bay Construction

Construction was begun on the electronics bay. In Figure 83, the interior of the bay can be seen with one minor alteration compared to the design in Section 3.1.2. In the physical version, nuts were used on both the top and bottom of every plate in order to ensure the plates would remain where desired.



Figure 83 Interior of Electronics Bay

## 3.2 Results

### 3.2.1 Vehicle Dynamics and Performance Model (FDA Analysis Task 1)

To verify the performance and accuracy of the dynamic simulator, the team examines a series of test cases run using the simulator.

#### *Test Case 1: 3DOF Translation*

This test case models a point mass subject to two arbitrarily defined forces. The first force acts from  $t = 1$  seconds to  $t = 2$  seconds and is defined as a vector  $[2 \ 2 \ -3]$  newtons. The second force then acts from  $t = 3$  seconds to  $t = 4$  seconds and is defined as a vector  $[0 \ 0 \ 3]$  newtons. The point mass itself has a constant mass of 1 kilogram. The position and velocity over time of the point mass under these conditions are presented in Figure 84 and Figure 85.



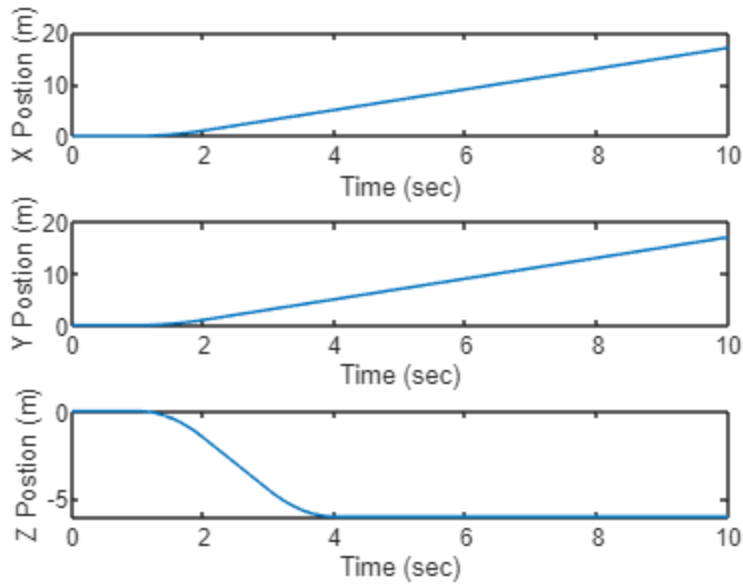


Figure 84 Test Case 1 Position vs Time

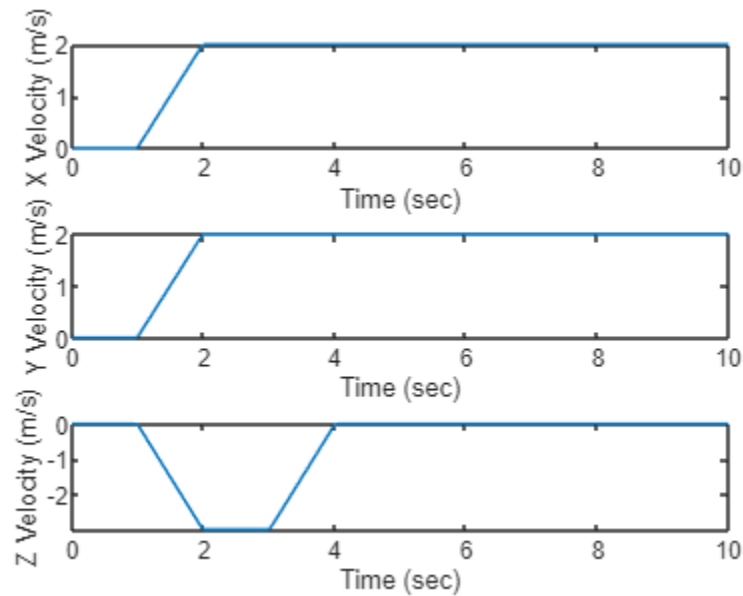


Figure 85 Test Case 1 Velocity vs Time

Examining the simulator output, the case of 3DOF translation is handled appropriately. As expected under constant forces, velocity changes linearly while the forces act, causing the particle's position to vary quadratically.

The total momentum change delivered by the two forces is  $[2 \ 2 \ 0]$  kg/(m/s). Again, examining the final velocity of the system with a mass of 1 kilogram, the expected and actual momentum of the system after experiencing these forces is the same.

*Test Case 2: 3DOF Rotation*

This test case models a rigid body subject to two moments. These moments are produced by the same force vectors as used in Test Case 1 but applied at a distance of 1 meter from the center of rotation of the rigid body. The body has principal inertias of  $[1 \ 1 \ 1]$  kg·m<sup>2</sup>. The Euler angles and body rates are presented in Figure 86 and Figure 87.

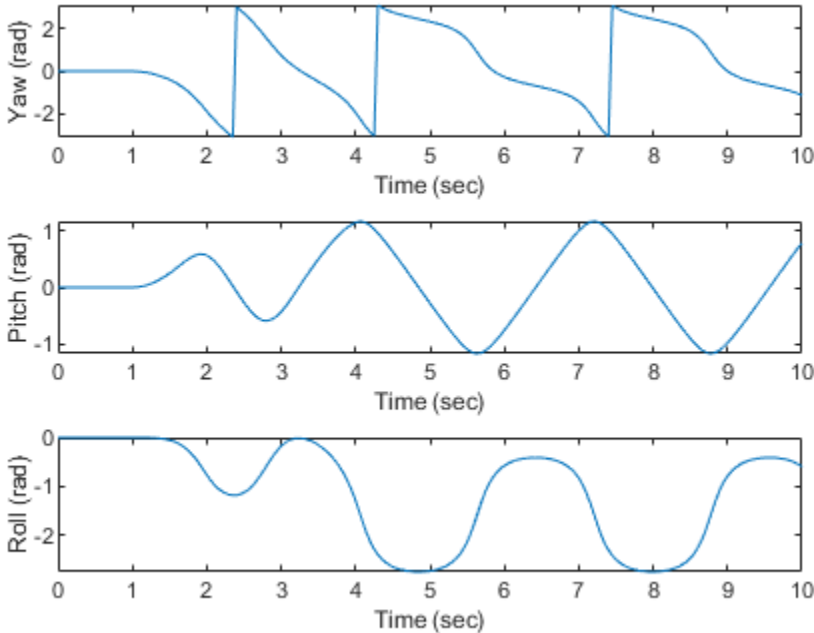


Figure 86 Test Case 2 Euler Angles vs Time

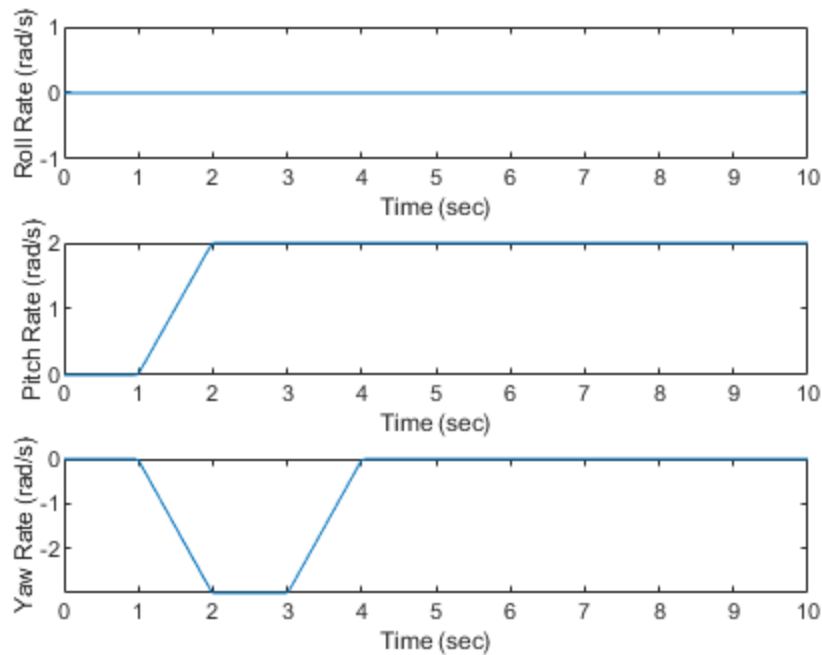


Figure 87 Test Case 2 Body Rates vs Time

The pitch and yaw angular rates match their expected values from the applied moments. There is zero roll rate because the simulated forces act at a point along the centerline of the vehicle, and therefore cannot exert any rolling moment. The Euler angles appear reasonable for this test case. Note that the Euler angles do initially seem to indicate that the vehicle is rolling, and that the integral of the body rates would not match the Euler angles. This is because the rates are expressed in the body frame, not the inertial frame as the Euler angles are, so the simulator is operating as expected.

*Full Flight Simulation – Flat Earth*

The flight of the vehicle from launch until impact with no recovery system is simulated using the dynamic simulator. The vehicle properties are defined in, with aerodynamic properties taken from Section 3.2.3, excluding roll coefficient which was set to zero. This was chosen since the geometry of the vehicle should not have produced any rolling moment due to its symmetry.

Table 3.13 Vehicle Properties for Flight Simulation

Property	Value
Dry Mass	5.168 kg
Dry C.G. Location	0.834 m
Reference Length	0.102 m
Dry Principal Moments of Inertia	[0.0123 1.432 1.432] kg m <sup>2</sup>
C.P. Location	1.19 m

The environment for the simulation is defined using the flat earth model, with a constant northern wind speed of 2 m/s. The vehicle launches on a northern heading with an 85-degree launch rod angle from horizontal. The trajectory is shown in Figure 88, with Euler angles and velocity shown in Figure 89 and Figure 90. Increasing altitude is defined as the negative Z-Axis per the definition of the NED coordinate system.

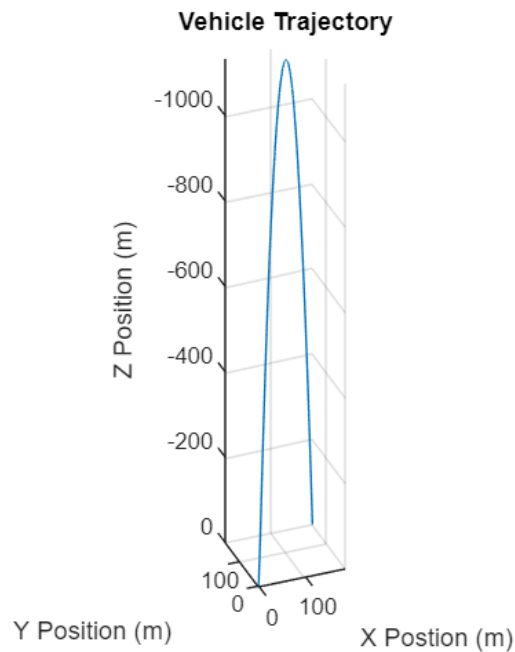


Figure 88 Full Flight Trajectory

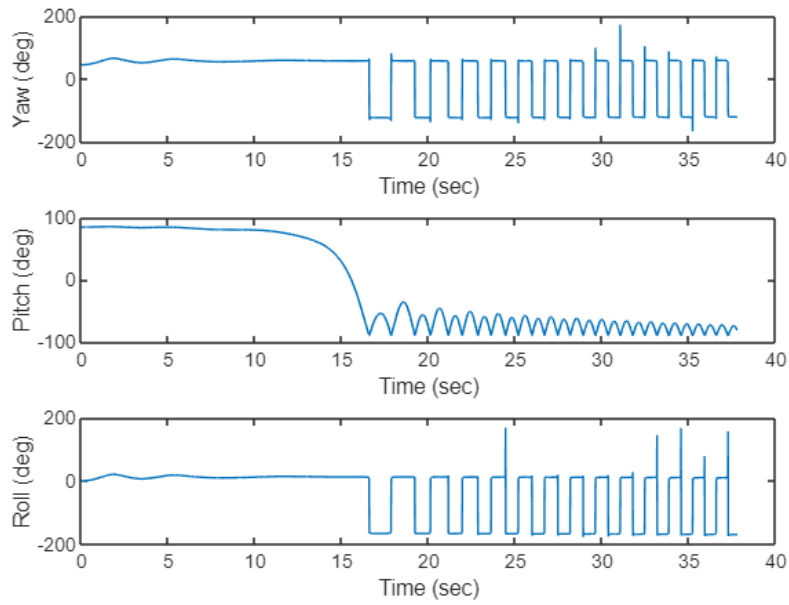


Figure 89 Full Flight Euler Angles vs Time

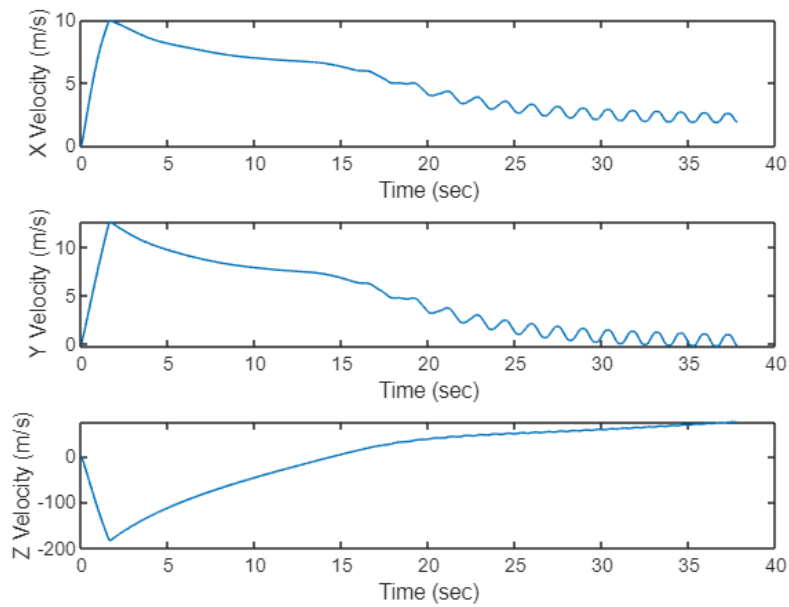


Figure 90 Full Flight Velocity vs Time

The apogee from the dynamic simulator is 1279 meters, compared to the apogee predicted by OpenRocket of 1134 meters. This difference is expected as OpenRocket and this

simulator use different aerodynamic models, but the results are reasonably close, suggesting the accuracy of this simulator.

As expected, the vehicle shows a small pitch/yaw oscillation during ascent due to a relatively large angle of attack as the vehicle lifts off. Since no launch rail is modelled, the vehicle experiences as high as a 90-degree angle of attack at liftoff, but this spike is so transient it does not have a major impact on the vehicle's flight. As the vehicle reaches apogee it pitches over fully to follow the flipping velocity vector, then oscillates more dramatically. This deviates from the results generated by OpenRocket, which has a much smaller oscillation after apogee, again due to differences in the aerodynamic coefficients used. The velocity of the vehicle increases when the motor is firing, then remains relatively constant in x axis as the vehicle oscillates and decreases due to drag and gravity in the z axis. It is therefore reasonable to say that the simulator operates as expected based on the results of this simulation and their similarity to those produced by OpenRocket.

#### *Full Flight Simulation – WGS84 Ellipsoid*

To demonstrate the ability of the simulator to simulate flight over the WGS84 ellipsoid, the same flight as for the flat earth case is simulated, with the only change being the geoid model used. In this case, the simulated vehicle is launched from the center of WPI's campus, with the trajectory of the vehicle shown in Figure 91.

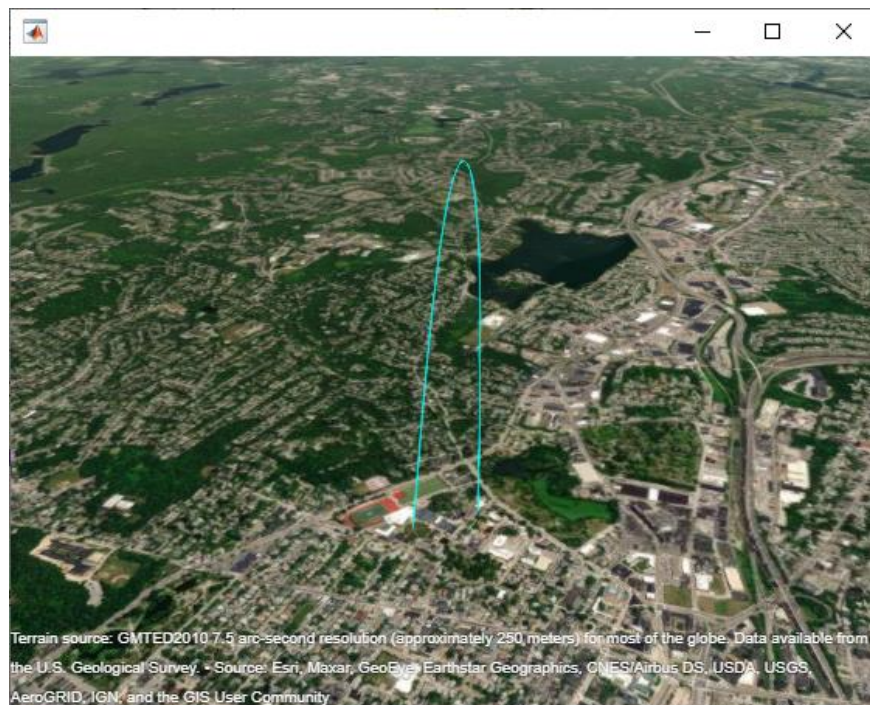


Figure 91 WGS84 Simulation Trajectory

As expected, the simulation results are nearly identical to those produced by the flat earth case, since the flight is too short and low to notice any significant difference due to Coriolis acceleration or change in gravity with altitude.

### 3.2.2 Drogue Replacement Retrorocket Model (FDA Analysis Task 4)

The dynamic simulator is used to verify that the rocket does not become unstable during descent and the firing of the retromotor. The vehicle is defined as for the full flight simulation, with the additional retromotor defined with a nozzle exit at the nose cone tip, and with an angle of 180 degrees to point upwards from the nose cone.

The vehicle's initial conditions are determined from the OpenRocket simulation of the flight due to the oscillation inconsistencies as the vehicle moves past apogee mentioned previously. For this case, the altitude of the vehicle is set to 215 meters, and the velocity vector is set as [0 8.33 115] m/s. Figure 92 shows the position of the vehicle during and after the retro motor firing. Figure 93 and Figure 94 show the Euler angles and the vehicle's velocity respectively.

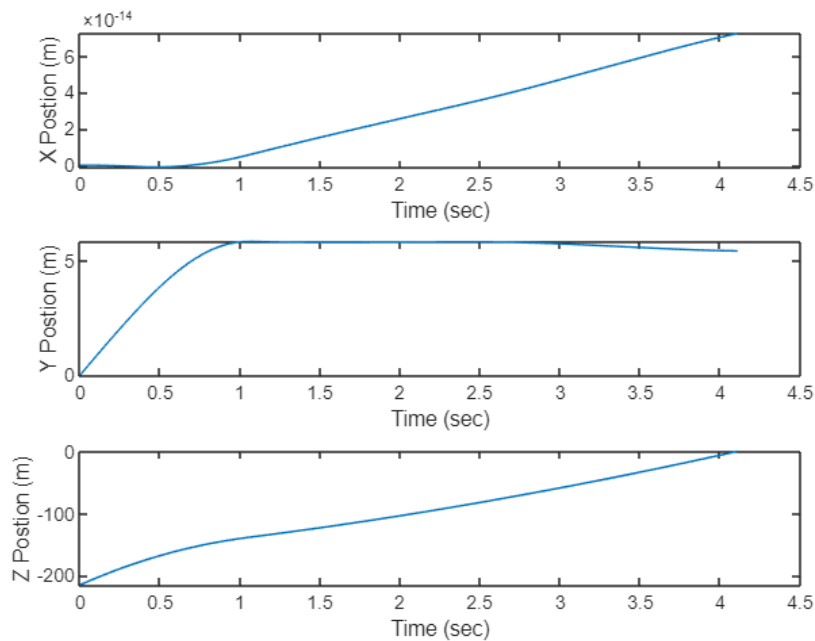


Figure 92 Position during/after Retromotor Firing

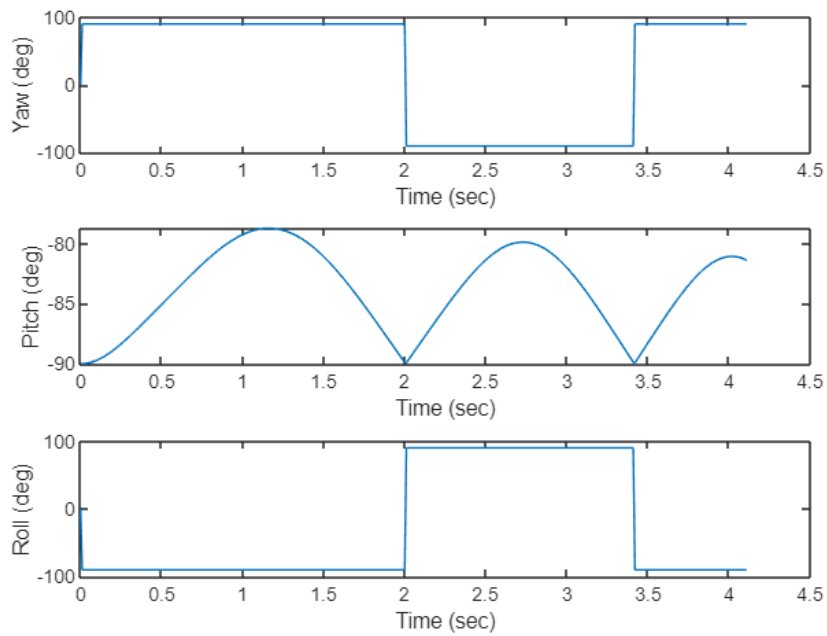


Figure 93 Retromotor Euler Angles vs Time

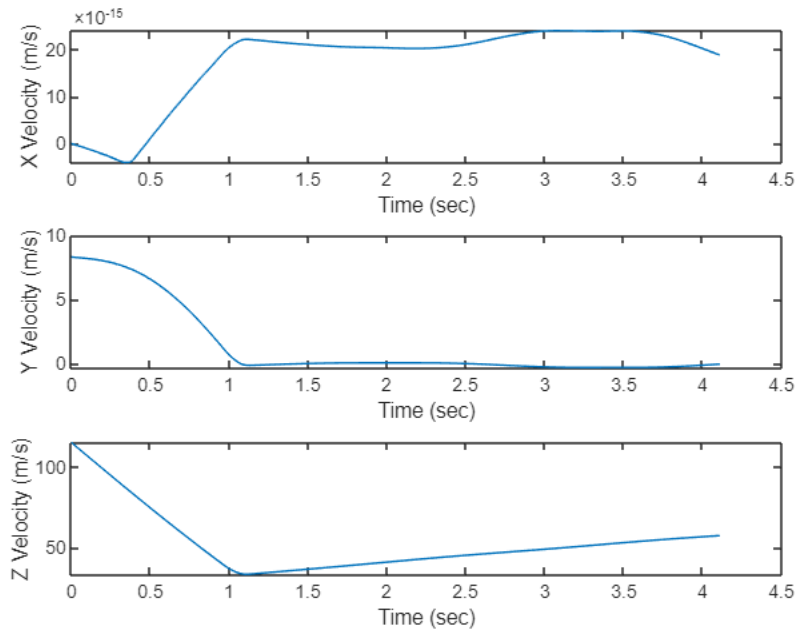


Figure 94 Retromotor Velocity vs Time

The simulation shows that the vehicle maintains stability while the motor is firing, with the vehicle oscillating within +/- 10 degrees of zero angle of attack flight. These angles of attack, while high compared to typical angles of attack seen during launch, are not enough to destabilize



the rocket. The jumps in yaw and roll angles are due to the singularities present in attitude description using Euler angles and have no bearing on the actual orientation of the vehicle. The vertical velocity of the vehicle decreases to just under 40 m/s at motor burnout, which is a safe velocity to deploy the main parachute. This result is expected as vehicles are generally expected to maintain stability at speeds greater than ~52 ft/s (~16 m/s) [1]. Based on the 75 m/s change in velocity due to motor firing, the team would expect the minimum safe descent speed to maintain stability prior to motor ignition to be 91 m/s.

### 3.2.3 Vehicle Aerodynamic Loads Simulation (FDA Analysis Task 2)

Following the setup and preparation discussed in Section 3.1.4, the team ran simulations in Ansys Fluent in order to determine the aerodynamic coefficients for our rocket geometry and flight conditions. These coefficients consisted of the drag coefficient, the lift coefficient, and the coefficients associated with each of the three principal moments acting on the rocket during flight: roll, pitch, and yaw. These coefficients were determined at the varying flight speeds and angles of attack mentioned previously. The team ran simulations at intervals of 40 m/s in flight speed, from 20 to 180 m/s, and 2° in angle of attack, from 0° to 20°, resulting in 55 total cases covering each combination of speed and angle. The table containing the Fluent outputs for each of these cases can be found in Appendix C: FDA Analysis Task 2 Fluent Simulation Data.

Various reference values needed to be defined in order for Fluent to provide the aerodynamic coefficients. For our analysis, these reference values were the reference area and the characteristic length of the rocket, as well as the density of the air. These values, as well as their descriptions, are found in Table 3.14. Due to the relatively small angles of attack considered, the reference area was held to half of the circular cross-sectional area of our rocket’s body tube. This simplification assumes that the fins and canards provided negligible forward-facing area and that there would be negligible change to the forward-facing area of the rocket as the angle of attack varied. The value used was half of the cross-sectional area to account for the simulation only using half of the rocket geometry in the simulation domain, as discussed with the symmetry considerations in Section 3.1.4.

Table 3.14 Aerodynamic Simulation Reference Values

Reference Value	Symbol	Value	Description
Area	$A$	0.0040535 m <sup>2</sup>	The projected frontal area of the rocket
Length	$L$	1.7 m	The length, measured front to back, of the rocket
Flow Density	$\rho$	1.275 kg/m <sup>3</sup>	The density of the fluid flowing around the rocket (taken to be air at 0°C)

Using the data obtained from the various cases, the team was able to determine relationships between the aerodynamic coefficients and angle of attack and plot these relationships for each simulated flight speed. This coefficient data was then implemented into the Dynamic Simulator described in Section 3.1.3. The plots of the lift, drag, roll moment, yaw moment, and pitch moment coefficients versus angle of attack can be found in Appendix C: FDA Analysis Task 2 Fluent Simulation Data. After comparing the data for varying flight speeds, the team determined that the aerodynamic coefficients showed no significant variation with the flight speed. This is to be expected, as the aerodynamic coefficients are characteristic of the rocket geometry at a specific flight orientation, not a specific flight speed. As a result, the data for each coefficient at a given angle of attack was averaged across each flight speed to provide a more general relationship. Plots of the averaged relationships between the aerodynamic coefficients and angle of attack are shown in Figure 95 and Figure 96.

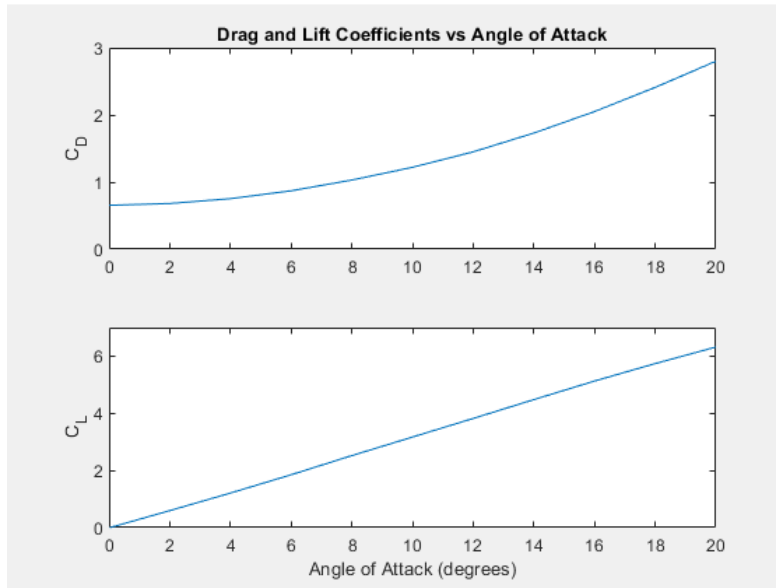


Figure 95 Drag and Life Coefficients vs. Angle of Attack

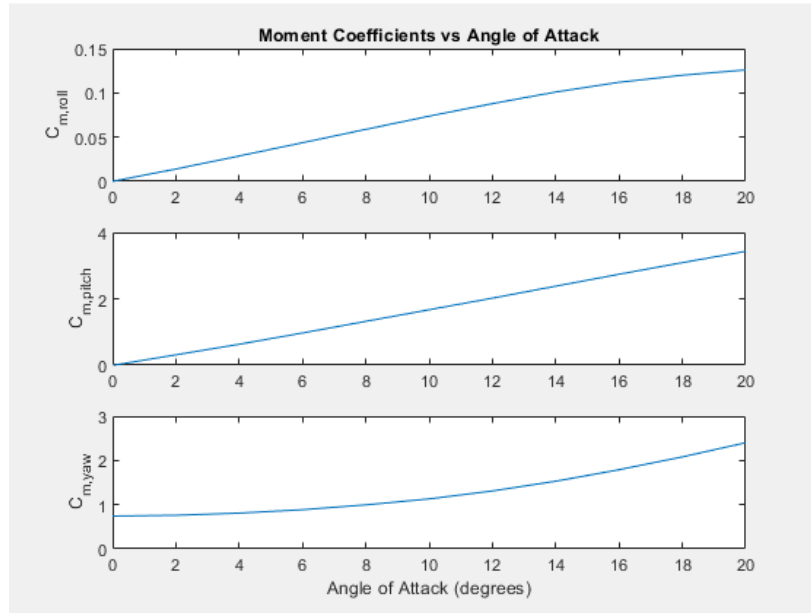


Figure 96 Moment Coefficients vs. Angle of Attack

The force and moment coefficient data were determined using a body-fixed coordinate system with the origin at the center of mass of the rocket as estimated by our OpenRocket model. The axes of this system were oriented such that the z-axis ran through the central axis of the rocket and the y-axis was normal to the plane of symmetry of the model. The origin of this coordinate system, as well as the lift and drag force vectors, are shown in Figure 97. With this orientation, roll moment was measured about the z-axis, pitch was measured about the y-axis, and yaw about the x-axis.

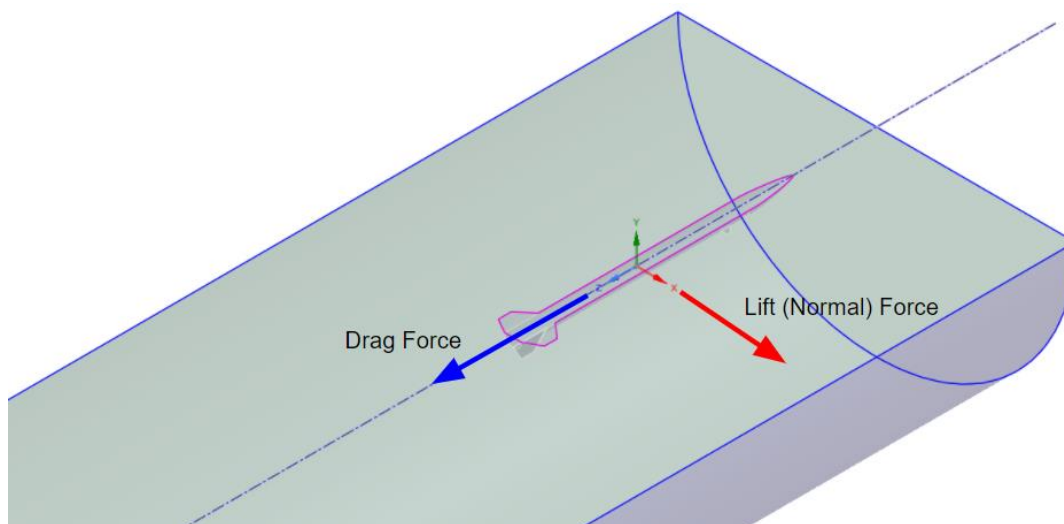


Figure 97 Body-Fixed Coordinate System and Force Vectors

The resulting simulated relationships between aerodynamic coefficients and angle of attack were determined to be consistent with the physical system. Both the drag and lift coefficients displayed behavior that was expected when examining this kind of vehicle [64]. The drag coefficient was found to be approximately 0.65 for a zero angle of attack, with an exponential increase as the angle of attack increased through 20°. Typical drag coefficient values for similar rockets follow this same trend, with zero angle of attack values between 0.6 and 0.75 and an increasing trend as angle of attack approaches 90° [64]. Additionally, the lift coefficient for similar vehicles typically begins at 0 for a zero angle of attack and increases almost linearly until a critical angle of attack is reached. Our lift coefficient results followed this same trend, with the zero-degree angle of attack lift coefficient found to be -0.0076. This value was not exactly 0 due to the nature of numerical models such as Fluent, but for the purposes of our project it was certainly within a reasonable error. Further, the moment coefficient results also displayed reasonable trends when compared to anticipated values. A minor exception to this would be the roll moment coefficient. For the given fin and canard orientation used in the aerodynamic simulations, the roll moment coefficient was expected to remain close to 0 for each angle of attack. However, an increasing trend was observed as angle of attack increased. It is believed that this trend was a result of the combination of the turbulence model and geometry used in the Fluent simulations. Although the magnitude of the roll moment coefficient did remain relatively small, being at least one order of magnitude smaller than the pitch and yaw moment coefficients, this error was taken into consideration when implementing the aerodynamic data into the Dynamic Simulator.

In addition to the aerodynamic coefficient data, our Fluent simulations were also used to provide static and dynamic pressure data on the surface of the rocket during flight. The pressure data was obtained for the same angle of attack and flight speed conditions as previously mentioned as well as two additional cases at a flight speed of 185 m/s and angles of attack of 0° and 10°. These cases were simulated in order to provide the static pressure data used for the analysis of the aerodynamic stresses on the fins, described in Section 2.1.2. The additional simulations were included in order to address the highest anticipated aerodynamic stresses on the rocket fins, which would occur at the maximum flight speed. The dynamic pressure contours for a flight speed of 180 m/s and angles of attack of 0° and 20° are shown in Figure 98 and Figure 99, respectively.

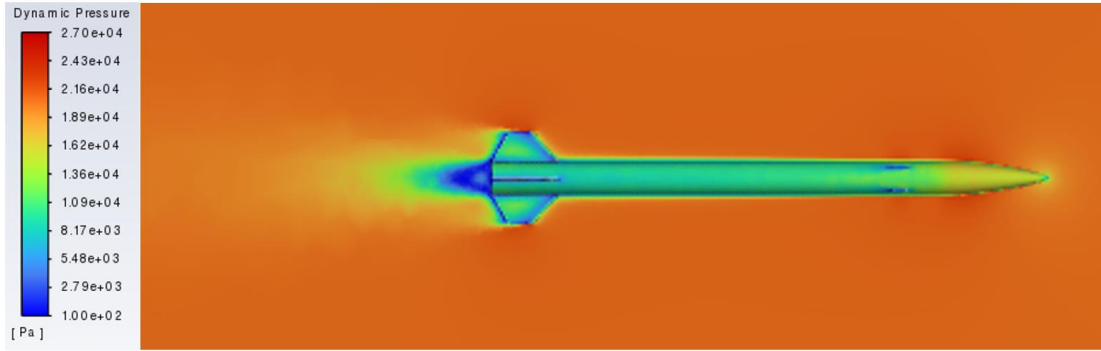


Figure 98 Dynamic Pressure for 180 m/s Flight Speed and 0° Angle of Attack (air flows horizontally from right to left)

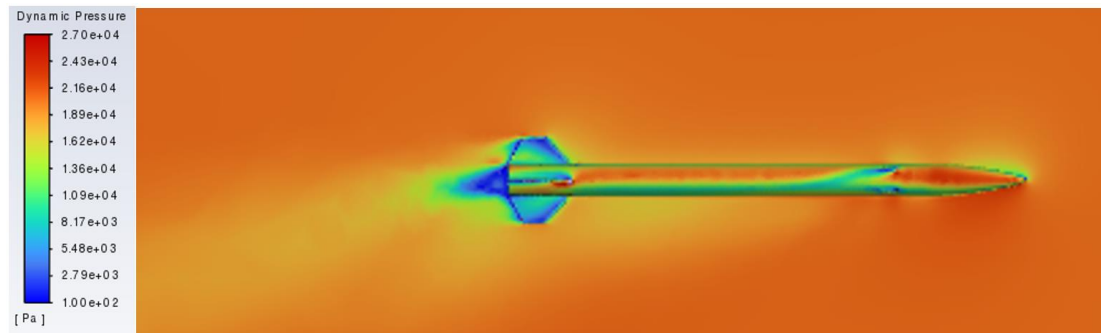


Figure 99 Dynamic Pressure for 180 m/s Flight Speed and 20° Angle of Attack (air flows from top left to bottom right at an angle of 20° from the horizontal)

### 3.2.4 Canard Control (FDA Analysis Task 3)

The results of the canard control system that were developed in MATLAB to be translated to a flight controller are presented in this section. Since no physical model of the canard system was completed, a software test was conducted to apply simulated disturbances and confirm that the PID controller design controls the roll rate of the launch vehicle as desired. In order to do so, the initial conditions, taken from the OpenRocket model and seen in Table 3.15, were implemented in the *Mainloop* code, as described in section 3.1.5.

Table 3.15 Initial Conditions for MainLoop

Initial Conditions		
	Case 1	Case 2
Position (m)	274.32	274.32
Velocity (m/s)	165.506	165.506
Angular Position (deg)	0	5
Angular Velocity (deg/s)	3	10

Two cases were evaluated. The position and velocity of the rocket remained the same for both and were what OpenRocket reported as such at the time of motor burnout. Due to varying launch conditions, it is not easy to estimate the angular position or velocity at the time of motor burnout, so two reasonable values for each were chosen for each.

Table 3.16 PID Coefficient Values

Proportional Coefficient, $K_p$	34603.7880218722
Integral Coefficient, $K_I$	13167.9530071796
Derivative Coefficient, $K_D$	-3103.50303505272

Using the *Mainloop* code, along with its functions *rk4* and *xdot2*, the flight of the rocket with canards, starting after motor burnout and ending at apogee, was simulated utilizing the created PID controller. The tuned values for the PID coefficients can be seen in Table 3.16. The position, velocity, angular rotation, and angular velocity of the rocket for Case 1 can be seen in Figure 100, and Case 2 in Figure 101.

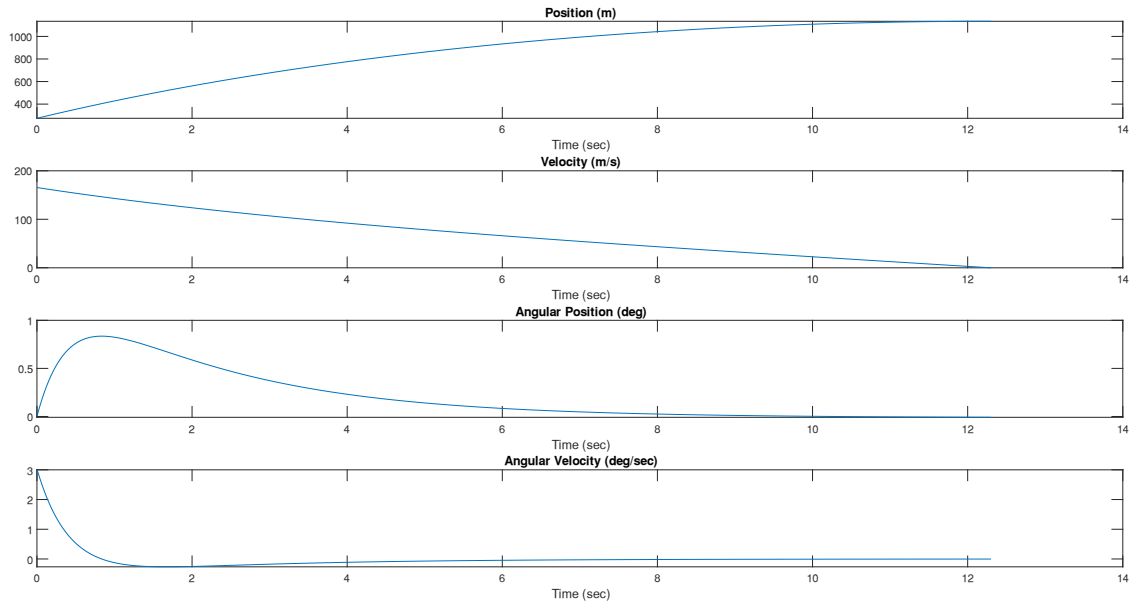


Figure 100 Rocket Flight States for Case 1

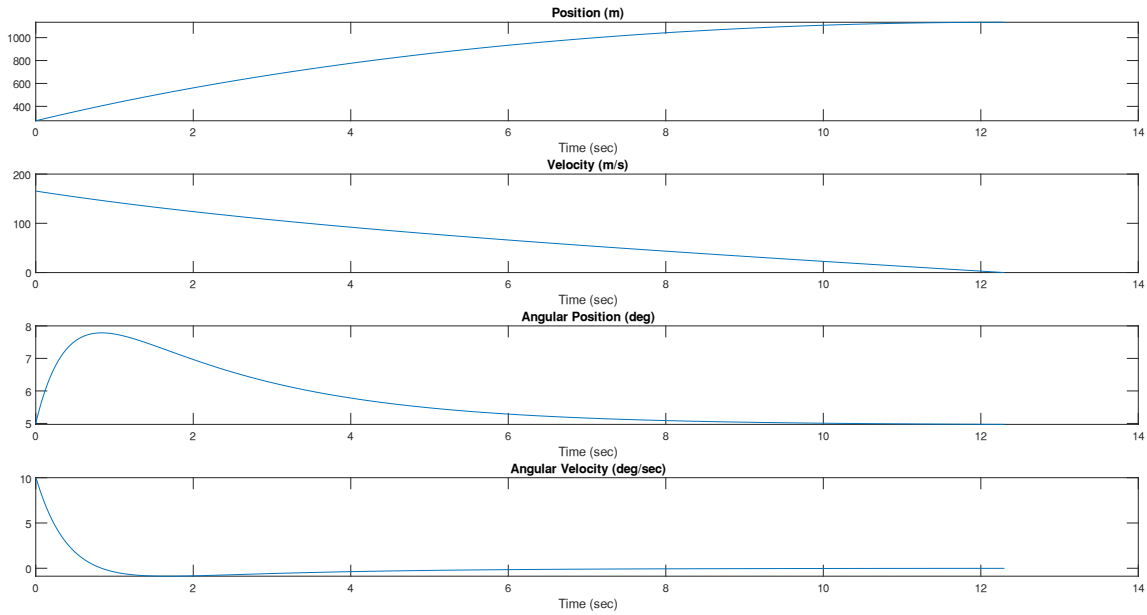


Figure 101 Rocket Flight States for Case 2

The simulator correctly shows the velocity reducing from the initial in a smooth fashion to zero, indicating apogee. The apogee from the non-linear simulator is just under 1200 meters, as opposed to the apogee of 1235 meters given by OpenRocket. This small error is expected since OpenRocket and the code created for this work use different aerodynamic models. Additionally, the simulation runs for a little over 12 seconds suggesting that the time between motor burnout and apogee is just over 12 seconds. Our OpenRocket model predicted the time of motor burnout to be 2.89 seconds after motor ignition and the time to apogee as 15.1 seconds, leaving 12.21 seconds between motor burnout and apogee. Since both the apogee and time results are fairly close, this suggests that this simulation is accurate.

As seen in Figure 100 and Figure 101, independent of the angular position (roll) or angular velocity (roll rate) at the time of motor burnout, the PID controller is able to bring the roll rate of the rocket to zero. This first happens at about a second after burnout and fully within five seconds after the canards would be activated during flight, allowing the rocket to reach zero roll rate before apogee.

## 4 Propulsion, Thermal, and Separation Systems Methodology and Results

### 4.1 Methodology

#### 4.1.1 Propulsion and Motors (PTSS Analysis Task 1)

##### *Motor Selection*

The selection of a high-powered rocket motor is important for several reasons. The motor directly correlates to how high the rocket will travel to reach its apogee, and how fast the rocket is traveling as it leaves the launch rail. The team decided an apogee goal above 3000 feet would be sufficient for the payload and size of our rocket. The launch rail is important because it is a rigid structure that guides the rocket upward until it is traveling fast enough that the aerodynamic forces provided by the fins keep it stable. OpenRocket allowed different rocket motor choices and configurations to be easily modeled and simulated. This provided estimates on the rockets mass, velocity off the launch rail, and height of apogee for different specified motors. A velocity off the launch rail greater than 52 feet per second was sought after to ensure the rockets flight would remain stable as it leaves the launch rail [1]. To get off the launch rail with this velocity, a Cesaroni J760 White Thunder motor was selected. According to its manufacturer data, this motor provides a total impulse of 1,265.7 Ns and an average thrust of 757.7 N [65]. Our OpenRocket simulation showed this motor would allow an off-rail exit velocity of about 74 feet per second and would also provide an apogee of about 3800 feet. Cesaroni was chosen because they produce high quality rocket motors that have previously been used in other WPI projects.

##### *Cantera Equilibrium Model*

A model of the selected motor was simulated using Cantera to estimate rocket performance data that could be compared to published rocket data. Cantera is an open-source software that can be used through multiple different interfaces [66]. For the purposes of this task, MATLAB was chosen to be used in conjunction with the tools provided by Cantera. Once MATLAB Cantera is downloaded, MATLAB code can reference and use all the tools provided by Cantera. These tools are stored in libraries that includes object-oriented methods and functions. They allow users to solve problems that involve species transport processes, thermodynamic calculations, and chemical kinetic calculations [66]. Cantera uses an input document as a mechanism to load transport information about specified compounds in a certain ideal gas. This analysis utilizes the input document that was developed by the previous Design, Analysis, and Test of a High-Powered Model Rocket-2 MQP in 2021 [19]. This document includes thermodynamic data provided by the National Institute of Standards and Technology (NIST) as well as data located within the libraries downloaded with Cantera [67]. The input document contains two polynomials for each elemental species considered in the reaction, one specifies specific heat at constant pressure and the other specifies entropy. In addition to this data, the document sets initial conditions for temperature, pressure, and the mole fraction of the reactant species. When the input document is called it creates a gas object with these initial conditions. These conditions can then be changed using Cantera's *set* method on the gas object.



To create this model, the propellant composition had to be determined. Listed in Table 4.1 are the values published by Cesaroni concerning the composition of the motor propellant ingredients.

Table 4.1 Published Cesaroni Propellant Ingredients

<b>Ingredient</b>	<b>Percentage</b>
Ammonium Perchlorate	40-85%
Metal Powders	1-40%
Synthetic Rubber	10-30%

Based on these vague published values, the analysis assumes a propellant composition of 80% ammonium perchlorate and 20% aluminum. It is also assumed that the propellant is already in the gas phase rather than the usual solid rocket propellant. These assumptions were chosen to simplify the complex reaction of rocket fuel, by reducing the fuel and oxidizer each to a single ideal gas species. This would help limit the amount of trace species given off by the reaction. Equation (35) shows this reaction written in its chemical equation form.



Each of the letters paired to a product species represent the mole fraction of that species to be determined by the equilibrium model. Generally, the coefficients in Equation (35) represent the mole numbers and not necessarily the mole fractions. In the case of Equation (35) the products coefficients add to one, and so the reactants side will also add to one, representing the equilibrium composition on a molar basis. The *set* method may be used on the gas object to specify the mole fraction of the reactant species. If the sum of the reactant's mole fractions were to add up to greater than one, Cantera would convert it to a ratio so the reactants would still be set on a molar basis. It is also worth noting that the *set* method may be used to set mass fractions as well as mole fractions.

To obtain the equilibrium properties of the gas object, the *equilibrate* method is used with what conditions wish to be fixed. UV designates fixed specific internal energy and specific volume, SV designates fixed specific entropy and specific volume, and SP designates fixed specific entropy and pressure [66]. When a rocket motor burns, heat is lost to the surrounding environment. To complete this equilibrium model, this analysis assumes that the heat lost to the surroundings is negligible, and so a fixed specific internal energy was designated. This also assumes that the propellant is already in the gas phase, and that its specific volume is fixed. These assumptions allow Cantera to compute the equilibrium model to find values of temperature, pressure, enthalpy, and entropy of the gas. For the propellant to ignite it must reach a temperature of 288 Celsius, so this was used for the initial temperature of the propellant [68].

The initial pressure was assumed to be standard atmospheric pressure at sea level. With initial temperature, initial pressure, and the molar composition of the reactants set, MATLAB can run the equilibrium model and store the data of the exit gas and its molar composition.

To find the heat of reaction for PTSS analysis task two in section 4.1.3, the methods *enthalpy\_mole* and *enthalpy\_mass* are utilized before the *equilibrate* method to record the initial enthalpy of the gas object after it is declared, and the initial conditions are set. Both methods are used to determine enthalpy of the propellant in terms of J/kg and J/kmol. The *equilibrium* method is then run assuming fixed specific internal energy and specific volume. Then the *enthalpy\_mole* and *enthalpy\_mass* methods are used again to store the final enthalpy of the equilibrium model. The heat of reaction for this analysis is then found by taking the difference of the initial and final enthalpy in terms of J/kg of propellant.

### *CEA Equilibrium Model*

NASA's CEA (Chemical Equilibrium with Applications) website was used to perform a separate equilibrium model. This equilibrium model would be used in conjunction with the Cantera analysis to compare motor performance data and to confirm the major product species of the reaction between the fuel (aluminum) and the oxidizer (ammonium perchlorate) that was assumed in the Cantera model. The CEA analysis requires the chamber pressure as an input, and so the chamber pressure from the Cantera model was used for this. Next aluminum and ammonium perchlorate were designated as the fuel and oxidizer starting at a temperature of 300 K. Both the fuel and oxidizer are assumed to be in a gaseous phase for this calculation similar to the Cantera calculation. The last argument required of this equilibrium analysis was the oxidizer to fuel weight ratio. This weight ratio was determined by finding 80% the molecular weight of ammonium perchlorate (93.992 g/mol) and dividing it by 20% of the molecular weight of aluminum (5.396) to obtain an oxidizer to fuel weight ratio of 17.41. The output of the NASA model is shown in Appendix D: NASA CEA Results. The dominant species produced were oxygen, water, hydroxyl, nitrogen, hydrogen chloride, chlorine, and alumina, which confirmed the trace species specified in the Cantera input document. Lastly CEA was able to use the nozzles geometry to determine exit properties of the motor. The nozzle geometry was assumed to be that of a commercially available 54-millimeter model rocket nozzle. Other assumptions of this analysis include isentropic flow, no nozzle throat erosion, and no boundary layer losses. These properties are presented and discussed in Section 4.2.1.

### *Rocket Flow Equations*

The Cantera equilibrium model provided the conditions within the motor chamber needed to calculate the thrust and specific impulse of the rocket. These conditions included temperature, pressure, density, specific heat at constant volume, specific heat at constant pressure, and the mean molecular weight. The nozzle geometry was assumed to be that of a commercially available 54-millimeter model rocket nozzle. This analysis assumes isentropic flow, no nozzle throat erosion, and no boundary layer losses. The nozzle's area ratio was then found using the throat radius and

the exit radius. Equation [31] was used to calculate the area ratio. MATLAB's aerospace toolbox includes a function called *flowisentropic* that uses the area ratio and the specific heat ratio which is also known as gamma. The function uses a residual method to solve for the Mach number in the supersonic region because a closed form solution of the area ratio equation for Mach number cannot be found. This residual method involves guessing a Mach number. It then solves for the area ratio in terms of the guessed Mach number and compares it to the user determined area ratio. This process iterates changing the Mach number each time until the resulting area ratio is equal to the input area ratio. At a given Mach number the stagnation property can be related to the static property. These stagnation properties include temperature, pressure, and density. After solving for the Mach number, the *flowisentropic* method uses Equations [32] [33] and [34] to find the stagnation ratios which are scalar values of exit temperature, exit pressure, and exit density.

$$\frac{A_e}{A_t} = \frac{r_e^2}{r_t^2} \quad (36)$$

$$\frac{T_e}{T_{02}} = \left(1 + \frac{\gamma - 1}{2} M_e^2\right)^{-1} \quad (37)$$

$$\frac{P_e}{P_{02}} = \left(1 + \frac{\gamma - 1}{2} M_e^2\right)^{-\frac{\gamma}{\gamma - 1}} \quad (38)$$

$$\frac{\rho_e}{\rho_0} = \left(1 + \frac{\gamma - 1}{2} M_e^2\right)^{-\frac{1}{\gamma - 1}} \quad (39)$$

The scalar value of each gas property was then multiplied to the corresponding combustion chamber property found using the equilibrium model. This allowed the exit conditions of the motor to be obtained. The formulas in Equations (40) - (44) were then used to compute the thrust and specific impulse so that they could be compared to data provided by the manufacturer. The variables used in these equations are described in Table 4.2.

$$\gamma = \frac{C_p}{C_v} \quad (40)$$

$$u_e = \sqrt{\frac{2\gamma\mathfrak{R}T_{02}}{(\gamma - 1)M} \left(1 - \left(\frac{P_e}{P_{02}}\right)^{\frac{\gamma-1}{\gamma}}\right)} \quad (41)$$

$$\dot{m} = \rho_e u_e A_e \quad (42)$$

$$T = \dot{m}u_e + (P_e - P_a)A_e \quad (43)$$

$$I_{sp} = \frac{T}{\dot{m}g} \quad (44)$$

Table 4.2 List of Variables for PTSS Analysis Task 1 Calculations

Description	Variables	Units
Nozzle Radius at Exit	$r_e$	m
Nozzle Radius at Throat	$r_t$	m
Gas Constant	$\mathfrak{R}$	J/mol K
Specific Heat Constant Pressure	$C_p$	J/kg K
Specific Heat Constant Volume	$C_v$	J/kg K
Molecular Weight	$M$	g/mol
Pressure Chamber	$P_{02}$	MPa
Pressure Exit	$P_e$	MPa
Standard Atmospheric Pressure at Sea Level	$P_a$	MPa
Temperature Chamber	$T_{02}$	K
Temperature Exit	$T_e$	K
Gas Density Chamber	$\rho_e$	Pa
Gas Density Exit	$\rho_0$	Pa
Mach Number	$M_e$	-
Specific Heat Ratio	$\gamma$	-
Mass Flow Rate	$\dot{m}$	Kg/s
Thrust	$T$	N
Specific Impulse	$I_{sp}$	s

#### 4.1.2 Ignition System

The J760 motor that was chosen for the rocket design required the use of an ignitor as described in Section 1.5.3. This ignitor is used for ignition of the main rocket motor. The ignitor that was chosen was an e-match because it is a commonly used and reliable ignitor, as well as it is what came with the motor that was ordered. For test launch, the ignitor is remotely powered and controlled by the controller that is provided at the launch site.

#### 4.1.3 Thermal Distribution Model (PTSS Analysis Task 2)

A thermal and fluid flow model was developed for the specific J760 motor that was used. This was important to be able to estimate the external temperature of the motor case to assure that during operation it did not rise too high and create points that had a risk of structural degradation. COMSOL, a Multiphysics simulation software, was used to create a 2D axisymmetric model using the heat transfer in both solids and fluids, as well as the laminar flow physics interfaces. It was time-dependent to assure the thermal distribution in the motor was accurately represented. Design, Analysis, and Test of a High-Powered Model Rocket – 2 [19], a previous MQP report, was used to design the very similar methodology for this section. Their process for this analysis task was followed very closely for this project.

COMSOL is a Multiphysics simulation software that can be used when heat transfer needs to be modeled. It allows the user to choose whether to work with a solid, fluid, or both. If working with a fluid, there are options to model either turbulent or laminar flow. It also allows for the tracking of the velocity of the fluid when there is fluid flow. The underlying equations for these models are listed in Appendix E: Equations in COMSOL Model. The user has the ability to input the boundary conditions as well as initial conditions of a model, as well as to choose whether heat transfer and fluid flow should be coupled together or not. There are many built in materials so when a model, like the one created for this project, is made up of different materials it is possible to distinguish between them. If a necessary material is not part of the library, then assuming the user has enough characteristics of the material they are able to specify it themselves. COMSOL can be used by people of all skill levels because it has built in models to help a user get started, or one can start with a blank slate. The typical steps followed when creating a model in COMSOL are as follows. The type of model must be specified – for this project a 2D axisymmetric model was used to model the motor. After that, the shape and size of the model has to be specified as well as materials selected. Once materials are chosen, the boundary conditions and initial conditions must be input by the user. Then, the time for the model to run needs to be chosen as well as a mesh. The mesh is chosen from pre-determined shapes in COMSOL, as well as specifying a number of points for it to include. Typically, the smallest number of points possible to adequately do the model should be chosen so that the model doesn't take a long time to run. The remainder of this section describes the model for PTSS Analysis Task 2 in greater detail.

The first step of designing the model in COMSOL was to define four domains of interest, which allowed for the analysis to be simplified. These are schematically represented in Figure 102. The domains chosen were the combustion gas, heat source, propellant grain, and aluminum motor case. Domains in this case are the different sections of the model. They each have their own initial conditions as well as size and material. A simple geometry was chosen as seen in Figure 102. The domains were all the same length of 252.22 millimeters (9.93 inches) based on available dimensions of the motor grain length for the J760 motor. The outer diameter of this motor's propellant grain was 54 millimeters (2.126 inches). This information was used to estimate the radial extent of each of the four domains. These radial extents, otherwise referred to as radii, are defined as in Figure 102. The following estimates of the radii of each domain were found by taking

the outer radius of the propellant grain and subtracting two millimeters for domains 2 and 3. The other two widths were estimated by using the same, approximate ratio of domain 1 to domain 4 in Design, Analysis, and Test of a High-Powered Model Rocket – 2 [19], and using that ratio of about 6.8:1 for this model.

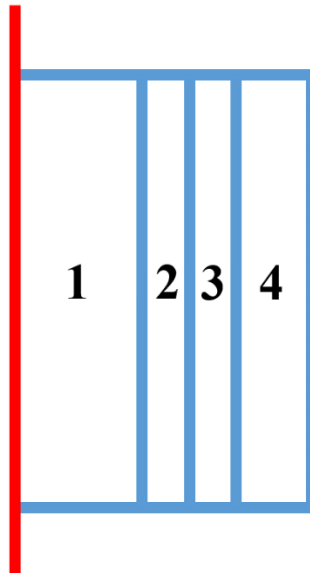


Figure 102 Geometry and Alignment of Domains in COMSOL Model

Moving from left to right, the domains as well as their description and dimensions are listed in Table 4.3. These dimensions were chosen to correspond with a time very close to the end of the burn time to allow for the thermal model to represent conditions when the motor is hottest. They were chosen to estimate the condition of the motor towards the end of its burn time of 1.75 seconds to duplicate results from the previous report, and to allow for comparison between the two to be made [19].

Table 4.3 COMSOL Domain Descriptions and Dimensions

<b>Domain Number</b>	<b>Domain Description</b>	<b>Domain Width</b>	<b>Domain Height</b>
Domain 1	Combustion Gas Domain (Fluid domain)	21.875 mm (0.86 in)	252.22 mm (9.93 in)
Domain 2	Combustion Flame Zone (Volumetric Heat Source)	1 mm (0.039 in)	252.22 mm (9.93 in)
Domain 3	Propellant Grain	1 mm (0.039 in)	252.22 mm (9.93 in)
Domain 4	Motor Case	3.125 mm (0.123)	252.22 mm (9.93 in)

The second domain – the heat source domain – was defined in COMSOL as a general heat source which required an input of the magnitude of the heat generated inside the flame combustion volume. This volume was calculated as 31.11 cm<sup>3</sup> (2.1425 inches cubed) using the dimensions listed in Table 4.3 for domain 2. Equation (45) was used to calculate the heat generated per unit volume based on the results from the Cantera model completed for the PTSS Analysis Task 1.

$$Q_0 = \frac{\dot{m}_{fuel} h_{RP}}{V} \quad (45)$$

The mass flow rate as well as the heat of reaction were taken from the Cantera model. Equation (45) made it possible to calculate the heat generation. The heat of reaction,  $h_{RP}$ , was found from the Cantera model by taking the difference in enthalpy before and after the equilibrium reaction was performed and is provided on a per unit mass of aluminum basis. The variables used in Equation (45) are defined in Table 4.4.

The heat generation had to be calculated so that the second domain could be modeled as a heat source. It is calculated by taking the mass flow rate of fuel multiplied by the heat of reaction and then dividing by the volume of the domain to make sure it is specific to domain 2 of the model. These values are summarized in Table 4.4.

Table 4.4 Heat Source and Mass Flow Rate Calculations

Variable	Description	Value	Units
$h_{RP}$	Heat of reaction (per kilogram of fuel)	850320	J/kg
$V$	Volume of domain 2	3.5109E-5	m <sup>3</sup>
$Q_0$	Heat generation in domain 2	3.2648E9	W/m <sup>3</sup>
$\dot{m}$	Mass flow rate	0.1348	kg/s

Once the above values were tabulated, each domain was assigned its material and initial conditions as seen in Table 4.5.

Table 4.5 Descriptions, Properties, and Initial Conditions of Domains

Domain	Description	Material	Width	Height	Initial Conditions
1	Combustion Gas Fluid	Fluid	0.861 in (21.875 mm)	9.93 in (252.22 mm)	P = 14.7 psia (0.101 MPa) $\mathbf{u} = -6.23$ ft/s (-1.9 m/s) T = 527.67 R (293.15 K)
2	Heat Source	Fluid	0.039 in (1 mm)	9.93 in (252.22 mm)	T = 527.67 R (293.15 K) $Q_0 = 1.03E9$ Btu-h-ft <sup>3</sup> ( $Q_0 = 3.265E9$ W/m <sup>3</sup> )
3	Solid Propellant Grain	Propellant	0.039 in (1 mm)	9.93 in (252.22 mm)	T = 527.67 R (293.15K)
4	Motor Case	Aluminum	0.123 in (3.125 mm)	9.93 in (252.22 mm)	T = 527.67 R (293.15 K)

Ambient conditions were included in the thermal model, which were also used as the initial conditions for the first domain. They were a pressure of 14.7 psia and a temperature of 527.67 R. These values were chosen because they were the default values for the environment. The material properties of aluminum were taken from the data provided in the COMSOL material database. The properties for the combustion fluid were determined from the Cantera model. All the properties are summarized in Table 4.6. The density, heat capacity at constant pressure, specific heat ratio, thermal conductivity, and dynamic viscosity of the combustion products were determined from the results of the motor performance model from Section 4.1.1.

Although the exact composition of the solid propellant grain and material properties were unknown, the thermal conductivity and heat capacity at constant pressure were estimated assuming all species are perfect gases. These estimated values were the same as those from Design, Analysis, and Test of a High-Powered Model Rocket – 2 [19] for their propellant. Their values were used because the team had already estimated the values for their propellant grain from NASA’s space shuttle solid rocket booster which was a reputable source for estimation.



Table 4.6 Material Properties for Domains of COMSOL Model

Material	Density	Heat Capacity, $C_p$	Thermal Conductivity	Dynamic Viscosity	Specific Heat Ratio
Aluminum (casing)	168.55 lbm/ft <sup>3</sup> (2700 kg/m <sup>3</sup> )	0.215 Btu/lbm-R (900 J/kg·K)	137.6 Btu/hr-ft-R (238 W/m-K)	-	-
Propellant	210.92 lbm/ft <sup>3</sup> (1937 kg/m <sup>3</sup> )	0.353 Btu/lbm-R (1476.26 J/kg·K)	0.352 Btu/hr-ft <sup>2</sup> -R (0.6096 W/m-K)	-	-
Combustion Products	0.1354 lbm/ft <sup>3</sup> (2.17 kg/m <sup>3</sup> )	0.711 Btu/lbm-R (1635.3 J/kg·K)	0.14 Btu/hr-ft <sup>2</sup> -R (0.2371 W/m-K)	6.49E-5 lbm/ft-s (9.660E-5 Pa-s)	1.212

The use of multiple boundary conditions was necessary to create an accurate COMSOL model for the heat transfer in solids and fluid physics as well as the laminar flow physics. The heat transfer portion of the model has four boundary conditions as shown in Figure 103. Boundary 1 aligns with the axis of 2D symmetry and is therefore simply a symmetry boundary. Since it is a symmetry boundary, no heat flux is allowed across the boundary. Boundaries 2 and 4 are the top and bottom of Figure 103 and are thermal insulation boundaries.

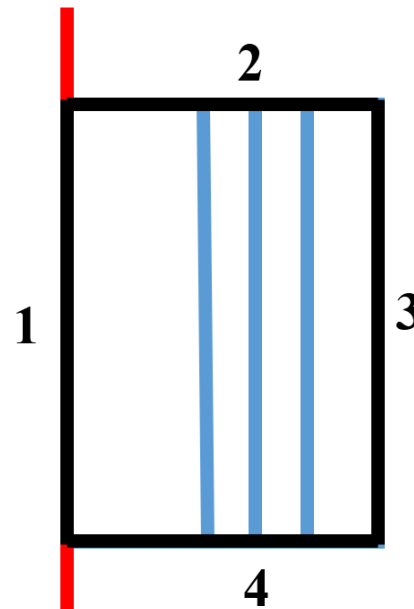


Figure 103 Boundary Conditions Defined in Heat Transfer Model

This means that they also do not allow heat flux across the boundary. This is the default condition COMSOL uses, which allows the user to assume there is no heat flow across the top and bottom of the propellant grain which simplifies the model. Lastly, Boundary 3 was a convective heat flux boundary. Again, to simplify the model, the default values for the aluminum material in COMSOL were used to find the properties for the convective heat transfer coefficient which can be thought of as a convection boundary condition because the aluminum casing is inside of the rocket [69]. The value for this was the same value as that used in Design, Analysis, and Test of a High-Powered Model Rocket – 2 [19]. This was because they had similar material properties, so the value was considered to be a good representation of the coefficient. As mentioned previously, the external temperature around the motor case was set to be the same as the ambient temperature. According to the COMSOL Heat Transfer Module User’s Guide, if the heat flux across the boundary is positive then heat is being added to the system, and if it is negative then the system is losing heat [70]. Table 4.7 provides the boundary conditions as well as their governing equations. The governing equations are further clarified in Appendix E: Equations in COMSOL Model.

Table 4.7 Summary of Heat Transfer Boundary Conditions and Inputs

Boundary	Condition	Governing Equation	Input
1	Symmetry	$-\mathbf{n} \cdot \mathbf{q} = 0$	-
2	Thermal Insulation	$-\mathbf{n} \cdot \mathbf{q} = 0$	-
3	Convective Heat Flux	$-\mathbf{n} \cdot \mathbf{q} = q_0$ $q_0 = h(T_{ext} - T)$	$h = 4.41 \text{ Btu/hr-ft}^2\text{-R}$ ( $25 \text{ W/m}^2\text{-K}$ ), air $T_{ext} = 527.67 \text{ R}$ ( $293.15 \text{ K}$ )
4	Thermal Insulation	$-\mathbf{n} \cdot \mathbf{q} = 0$	-

The laminar flow interface was used in the analysis because it made it possible to simulate the flow of the combustion gas from the burning surface area to the inlet on the nozzle of the rocket. It was assumed that the pressure inside the combustion chamber was held constant at the combustion pressure estimated from the Cantera model. Therefore, the flow was modeled as a weakly compressible flow which utilized the compressible form of the Navier-Stokes equations as well as continuity equation. However, it does neglect the pressure dependence of density, so the density was evaluated at the COMSOL reference pressure, which assisted in simplifying the model [69]. A full summary of the equations used for the COMSOL models are provided in Appendix E: Equations in COMSOL Model.

The default reference temperature of 527.67 R (293.15 K) was used for the laminar flow portion of the model. However, the reference pressure (otherwise known as gauge pressure) was changed from the default to 0 psia so that the absolute pressure of the combustion chamber could

be used. The initial fluid properties remained the same as in Table 4.5. For the laminar flow model, four new boundary conditions were defined as seen in Figure 104.

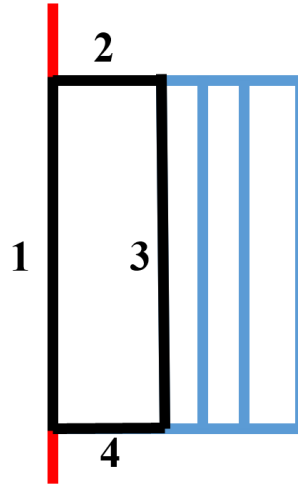


Figure 104 Boundary Conditions Defined in Fluid Flow Model

Boundary 1 again aligns with the axis of 2D symmetry and is therefore simply a symmetry boundary. Since it is a symmetry boundary, no fluid flow is allowed across the boundary. Boundary 2 is a wall node which is typically used in COMSOL to describe flow conditions at a wall that is either stationary, moving, or leaking. Since it is used to model a solid wall in this case, a no slip condition with zero translation velocity was imposed. This means that the fluid velocity relative to the wall velocity is zero. Boundary 3 was an inlet with an initial mass flow rate over the surface area. This is because fluid flows from domain 2 into domain 1. The mass flow rate for this boundary was retrieved from the results of the Cantera model for the flow rate of the fuel. This is the mass flow rate mentioned previously in Table 4.4. Lastly, boundary 4 was defined as an outlet to represent the fluid flow into the inlet of the rocket motor nozzle. The relative pressure at this boundary was calculated by subtracting the reference pressure of 1 atm from the estimated chamber pressure of the rocket motor. During the creation of the COMSOL model, the “suppress backflow” check box was selected which adjusted the outlet pressure to assure no fluid would enter the domain from the outlet boundary to further simplify the model [69]. Table 4.8 provides the boundary conditions as well as their governing equations. In the governing equations for boundary 3 and 4, the vector  $\mathbf{K}$  is the viscous stress tensor, and  $d_{bc}$  is the channel thickness, which is defined as the area across which the mass flow occurs. The value of the integral for boundary 3 changes as the height of the domains change since the variable of integration ( $S$ ) is based on the height of the domain.

Table 4.8 Summary of Fluid Flow Boundary Conditions and Inputs

Boundary	Condition	Governing Equation	Input
1	Axial Symmetry	-	-
2	Wall	$\mathbf{u} = 0$	-
3	Inlet	$-\int_{\partial\Omega} \rho(\mathbf{u} \cdot \mathbf{n})d_{bc}dS = \dot{m}$	$\dot{m} = 0.2972 \text{ lb/s}$ (0.1348 kg/s)
4	Outlet	$[-p\mathbf{I} + \mathbf{K}]\mathbf{n} = -\hat{p}_0\mathbf{n}$ $\mathbf{K} = \mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)\mathbf{n}$ $\hat{p}_0 \leq p_0$	$\hat{p}_0 = 245.81 \text{ psi}$ (1.694774 MPa)

#### 4.1.4 Recovery Bay

The recovery bay, in Figure 105, was designed to house the electronics controlling the recovery system within the airframe body. The outside of the recovery bay is a coupler tube which is housed within the airframe body tube. The Easy-Mini altimeters as well as the batteries are mounted on the sled of the bay. The sled is 3D-printed allowing for small parts to easily be connected.

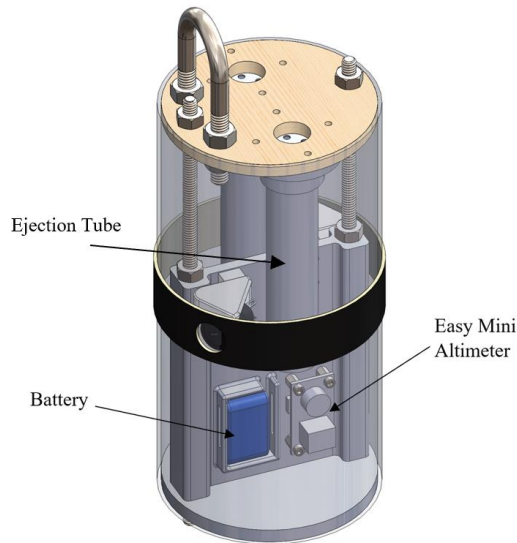


Figure 105 Isometric View of the Recovery Bay

To assure there was room for the CO<sub>2</sub> ejection tubes and canisters as well as the necessary threaded rods, the sled of the bay was created with a dog-bone shaped cross-section and only about half the height of the entire bay. The CO<sub>2</sub> ejection tubes are the tubes that the CO<sub>2</sub> canisters go inside of for the recovery system. The canisters are required to fit snug in the tubes so that they are not loose during flight. The tubes are secured to the recovery bay and are the part of the bay the

canisters are secured to. In Figure 105, one Lithium-Polymer battery can be seen as well as one easy mini altimeter. There is one more of each on the other side for redundancy. The sled can be seen in Figure 106, which is the main part of the recovery bay. The part was printed using PLA. To attach items to the sled using screws, heat-set threaded inserts were mounted within the sled.

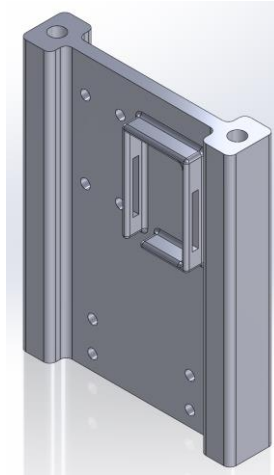


Figure 106 The Recovery Bay Sled

#### 4.1.5 CO<sub>2</sub> System (PTSS Analysis Task 3)

Compressed CO<sub>2</sub> was chosen as the method to separate the airframe by pressurizing the recovery compartment, to release the main parachute at apogee. This section is broken up into five subsections which detail the requirements for separation to occur, the initial mass and volume of liquid and vapor CO<sub>2</sub>, two transient flow phases of the CO<sub>2</sub> as it exits the canister, and the pressurization of the recovery compartment. The temperature of the launch conditions that were used in this analysis resulted in the CO<sub>2</sub> being an unknown ratio of liquid and vapor which needed to be solved in order to complete the analysis. Phase I flow describes the release of CO<sub>2</sub> while it is mixed phase and constant pressure and temperature until there is only CO<sub>2</sub> vapor left in the canister. Phase II flow describes the vapor-only regime where pressure and temperature are not constant. Table 4.9 summarizes all values that are either previously calculated or known that are used in this section, and Table 4.10 summarizes all variables that were used in the equations in this section.

Table 4.9 Provided and Known Values for Flow Analysis

<b>Name</b>	<b>Variable</b>	<b>Value</b>	<b>Units</b>
Area of Bulkhead	$A_{bh}$	0.0081	$m^2$
Area of CO <sub>2</sub> Canister Exit	$A_{exit}$	$1.9635 \cdot 10^{-5}$	$m^2$
Atmospheric Pressure at 4000ft Elevation	$P_{atm}$	87500	$Pa$
Bulkhead Diameter	$d_{bh}$	0.1016	$m$

Critical Pressure Ratio of CO <sub>2</sub>	$P_{crit}$	$\geq 1.83$	$\sim$
Density of Liquid CO <sub>2</sub>	$\rho_{CO_2,liq}$	1103	$kg/m^3$
Diameter of CO <sub>2</sub> Canister Exit	$d_{exit}$	0.0025	$m$
Factor of Safety	$FoS$	1.5	$\sim$
Force from Shear Pins	$F_{pins}$	511.3870	$N$
Mass of CO <sub>2</sub>	$m_{CO_2,i}$	12	$g$
Molar Mass of Air	$\bar{M}_{air}$	28.97	$g/mol$
Molar Mass of CO <sub>2</sub>	$\bar{M}_{CO_2}$	44.01	$g/mol$
Recovery Comp. Length	$L_{rec}$	0.3048	$m$
Specific Heat Ratio of CO <sub>2</sub>	$\gamma$	1.289	$\sim$
Temperature	$T$	273	$K$
Universal Gas Constant	$\mathfrak{R}$	8.3140	$J/mol \cdot K$
Van der Waals Constant “a”	$a$	0.3658	$Pa \cdot m^6/mol^2$
Van der Waals Constant “b”	$b$	0.00004286	$m^3/mol$
Vapor Pressure of CO <sub>2</sub> at 32° F	$P_{vap}$	3475500	$Pa$
Volume of CO <sub>2</sub> Canister	$V_{CO_2}$	$2.5728 \cdot 10^{-5}$	$m^3$
Volume of Recovery Comp.	$V_{rec}$	0.0025	$m^3$

Table 4.10 Nomenclature and Units for Flow Analysis

Name	Variable	Units
Force due to Atmospheric Pressure	$F_{P,atm}$	$N$
Mass Flow Rate for Vaporization	$\dot{m}_{vaporize}$	$kg/s$
Mass Flow Rate of CO <sub>2</sub> Exit	$\dot{m}_{exit}$	$kg/s$
Mass of Air	$m_{air}$	$g$
Mass of CO <sub>2</sub> Liquid	$m_{liq}$	$g$
Mass of CO <sub>2</sub> Vapor	$m_{vap}$	$g$
Mass of Mixed Gas in Recovery Comp.	$m_{gas,r}$	$g$
Minimum Mass of CO <sub>2</sub> for Separation	$m_{CO_2,min}$	$g$
Moles of Air	$n_{air}$	$mol$
Moles of CO <sub>2</sub>	$n_{CO_2}$	$mol$
Moles of CO <sub>2</sub> Vapor	$n_{vap}$	$mol$
Pressure at Separation	$P_{sep}$	$Pa$
Pressure of CO <sub>2</sub>	$P_{CO_2}$	$Pa$
Pressure of Mixed Gas in Recovery Comp.	$P_{gas,r}$	$Pa$
Temperature of CO <sub>2</sub>	$T_{CO_2}$	$K$
Time to Empty CO <sub>2</sub> Canister	$t_{empty}$	$s$
Time to Separate	$t_{sep}$	$s$
Time to Vaporize	$t_{vaporize}$	$s$
Volume of CO <sub>2</sub> Liquid	$V_{liq}$	$m^3$
Volume of CO <sub>2</sub> Vapor	$V_{vap}$	$m^3$

### Separation Requirements

Successful separation occurs when the pressure supplied to the recovery compartment by the CO<sub>2</sub> canister surpasses the total force keeping the airframe together. The total force is the sum of the atmospheric pressure at apogee, the height at which the main parachute will deploy, and the force from the shear pins used to keep the rocket together during launch. Any forces due to friction were not considered and assumed to be negligible. The ideal gas law given by Equation (48) relates the total pressure, calculated in Equations (46) and (47), and recovery compartment volume to the universal gas constant and initial temperature of the CO<sub>2</sub> canister. It is used in conjunction with Equation (49) to find the minimum mass of CO<sub>2</sub> required for separation.

$$F_{P,atm} = A_{bh}P_{atm} \quad (46)$$

$$P_{sep} = \frac{F_{P,atm} + F_{pins}}{A_{bh}} \quad (47)$$

$$n_{CO_2} = \frac{P_{sep}V_{rec}}{\mathfrak{R}T} \quad (48)$$

$$m_{CO_2,min} = n_{CO_2}\bar{M}_{CO_2}FoS \quad (49)$$

Equation (49) is the minimum amount of CO<sub>2</sub> needed, as the CO<sub>2</sub> will be commercially bought and comes in standard sizing, most commonly 8, 12, or 16 g canisters. With an assumed Factor of Safety (FoS) of 1.5, the minimum amount of CO<sub>2</sub> required is 10.8 grams, thus 12 grams of CO<sub>2</sub> was the basis for all subsequent calculations and models.

### CO<sub>2</sub> Canister Initial Conditions

The initial conditions that needed to be calculated for the analysis of the transient CO<sub>2</sub> flow are the total mass and volume of CO<sub>2</sub>, the mass and volume of CO<sub>2</sub> liquid, and the mass and volume of CO<sub>2</sub> vapor. The planned launch site was Lake Winnepesaukee in New Hampshire, which has an elevation of 500 feet above sea level and an average daytime temperature in February of 32°F (0°C) [70,71]. The vapor pressure of CO<sub>2</sub> at 0°C is 3475.5 kPa [73]. As this is a high-pressure system, along with the saturation condition of CO<sub>2</sub> being dependent on pressure and temperature, there will be a mix of CO<sub>2</sub> liquid and vapor of an unknown ratio inside the CO<sub>2</sub> canister. The assumption that the CO<sub>2</sub> is an ideal gas will therefore not be valid due to the high-pressure.

The total initial mass and volume is known, leaving four unknown conditions. As previously stated, the CO<sub>2</sub> cannot be assumed to be ideal, thus the van der Waals equation of state was used as opposed to the Ideal Gas Law. The Van der Waals equation takes the compressibility of fluids into consideration with constants “a” and “b,” and is shown in Equation (52). For simplicity, the mole number was not converted to an equivalent mass inside Van der Waals equation, making it a fifth unknown. Equations (50) through (54) represent the system of five, nonlinear equations that were used to solve for the five unknowns.

$$V_{vap} + V_{liq} = V_{CO_2} \quad (50)$$

$$m_{vap} + m_{liq} = m_{CO_2} \quad (51)$$

$$\left( P_{vap} + \frac{n_{vap}^2 a}{V_{vap}^2} \right) (V_{vap} - n_{vap} b) = n_{vap} \mathcal{R} T \quad (52)$$

$$m_{CO_2} = n_{vap} \bar{M}_{CO_2} \quad (53)$$

$$V_{liq} = \frac{m_{liq}}{\rho_{CO_2, liq}} \quad (54)$$

The system was solved using MATLAB's *vpasolve* function [74], which symbolically solves an equation or system of equations for the desired variable. A symbolic solver was necessary, as opposed to a numerical solver such as *fsolve* [75], for this system of equations because it was not possible to use any guessed values. The *double* function [76] was then used to convert the solutions from symbolic to numerical values that could be used in MATLAB functions that do not accept symbolic values.

#### *Phase I CO2 Transient Flow*

For simplicity, the flow conditions are described as consisting of three phases. Phase I is the regime in which the CO<sub>2</sub> flow is choked and exiting at a constant pressure. Since it is initially stored at high pressure, when the CO<sub>2</sub> vapor exits the liquid CO<sub>2</sub> will boil and vaporize. It is assumed that the phase change is instantaneous, so the vapor is replaced at the same rate at which it exits and thus creates a constant pressure region. This assumption is valid due to Le Chatelier's principle, in which a liquid substance will flash to maintain the vapor space in a system that was closed and was then opened/punctured [77]. In addition, this analysis assumes that the phase change occurs at a constant temperature (isothermal) when at saturation conditions. Due to the conditions described, the initial pressure will correspond to the vapor pressure of CO<sub>2</sub> at the temperature of the surroundings. The pressure will remain at this value until all the liquid CO<sub>2</sub> has vaporized, and thus saturation conditions are no longer met. The mass flow rate was calculated using the choked mass flow equation shown in Equation (55) [78].

$$\dot{m}_{exit,1} = const = \frac{A_e P_{vap}}{\sqrt{T}} \sqrt{\frac{\gamma}{\mathcal{R} \bar{M}_{CO_2}}} \left( \frac{\gamma + 1}{2} \right)^{-\left(\frac{\gamma+1}{2(\gamma-1)}\right)} \quad (55)$$

It is assumed that the temperature and pressure of the CO<sub>2</sub> is constant in this regime, resulting in the mass flow rate being constant as well. The rate of CO<sub>2</sub> vaporization is assumed equivalent to the exit mass flow rate, as seen in Equation (56). The time it will take for the CO<sub>2</sub> to



vaporize, Equation (57), will therefore be the initial mass of liquid CO<sub>2</sub> divided by this rate. The mass throughout Phase 1 flow is described by Equation (58).

$$\dot{m}_{vaporize} = \dot{m}_{exit,1} \quad (56)$$

$$t_{vaporize} = \frac{m_{liq}}{\dot{m}_{vaporize}} \quad (57)$$

$$m_{CO_2,1} = m_{CO_2,i} - \dot{m}_{exit,1}(t_0 : t_{vaporize}) \quad (58)$$

### Phase II CO<sub>2</sub> Transient Flow

Phase II is the regime in which the CO<sub>2</sub> flow is still choked but is not exiting at a constant pressure or temperature anymore. The choked flow mass flow rate equation is used again, however in Equation (59), Equation (60), and Equation (61), the mass flow rate, the rate of pressure change, and the rate of temperature change are functions of time. To solve a system of three interrelated equations, Van der Waals equation was rewritten as a function of time and the conservation of energy equation was simplified. Appendix F shows the key steps for deriving Equation (60), and Equation (61).

$$\dot{m}_{exit,2}(t) = -\frac{A_e P_{CO_2}(t)}{\sqrt{T_{CO_2}(t)}} \sqrt{\frac{\gamma}{\Re \bar{M}_{CO_2}}} \left(\frac{\gamma+1}{2}\right)^{-\left(\frac{\gamma+1}{2(\gamma-1)}\right)} \quad (59)$$

$$\begin{aligned} \dot{P}_{CO_2,2}(t) = & \frac{\dot{m}_{exit,2}(t) \Re T_{CO_2}(t) + m_{CO_2}(t) \Re \dot{T}_{CO_2,2}(t)}{\bar{M}_{CO_2} V_{CO_2} - m_{CO_2}(t) b} \\ & + \frac{\dot{m}_{exit,2}(t) m_{CO_2}(t) \Re T_{CO_2}(t) b}{(\bar{M}_{CO_2} V_{CO_2} - m_{CO_2}(t) b)^2} - \frac{2 m_{CO_2}(t) a}{\bar{M}_{CO_2}^2 V_{CO_2}^2} \end{aligned} \quad (60)$$

$$\dot{T}_{CO_2,2}(t) = \frac{1}{m_{CO_2}(t)} (\gamma + 1) (\dot{m}_{exit}(t) T_{CO_2}(t)) \quad (61)$$

$$t_{empty} = t(m_{CO_2}(t) \approx 0) \quad (62)$$

Equations (59) through (61) are ordinary differential equations and were solved utilizing *ode45*, one of MATLAB's ode solvers [79]. A function was created with all the equations, where the initial mass, pressure, and temperature were the final values of mass, pressure, and temperature from the previous regime. Equation (62) was used to calculate the time at which the CO<sub>2</sub> canister was emptied. Due to the limitations of *ode45*, it may not be possible to actually reach no mass without the simulation failing, which is why it was approximated. The CO<sub>2</sub> flow becoming

unchoked would indicate the beginning of the Phase III CO<sub>2</sub> Transient Flow regime, but this was not within the scope of this project.

### *Recovery Compartment Pressurization*

The crucial result from the pressurization equations was finding the time it would take for the total pressure forces keeping the airframe together to be surpassed. The CO<sub>2</sub> canister needed to start to discharge before the rocket reached apogee, allowing the parachute to deploy exactly at apogee. The initial pressure inside the recovery compartment is the atmospheric pressure at the elevation of apogee and the initial temperature is still 0°C. Knowing this, along with the volume of the recovery compartment where the parachute is stored, the initial mass of air inside the compartment can be found using the Ideal Gas Law shown in Equation (63). The van der Waals equation did not need to be used because the rocket body is not a high-pressure system, thus the ideal assumption is valid.

$$n_{air} = \frac{P_{atm}V_{recovery}}{\mathfrak{R}T_{air}} \quad (63)$$

$$m_{air} = n_{air}\bar{M}_{air} \quad (64)$$

It was assumed that temperature was a constant, and it was also assumed that the airframe of the rocket is airtight. It was not possible to create a model of fluid escaping the airframe while simultaneously pressurizing the rocket without making different assumptions about gaps in the airframe, exit area/ velocity of the fluid, etc.

Similar to the CO<sub>2</sub> transient flows, once the initial pressure and mass of the airframe were known, the constant pressure and non-constant pressure regimes could be modeled. During Phase I, the exit mass flow of the CO<sub>2</sub> canister was constant for the vaporization time, so the mass of fluid in the rocket for this period was modeled using Equation (65).

$$m_{gas,r,1}(t) = m_{air} + (m_{CO_2,i} - m_{CO_2,1}(t)) \quad (65)$$

The rocket pressure during the Phase I regime was calculated using the mass of the rocket found in Equation (65) in the Ideal Gas Law equation in Equation (66).

$$P_{gas,r,1} = \frac{\mathfrak{R}Tm_{gas,r,1}}{\bar{M}_{air}V_{rec}} \quad (66)$$

The same equation for mass flow rate from Phase II CO<sub>2</sub> flow was used for modeling the fluid flow into the recovery compartment in Equation (67). The initial pressure is the final pressure from the recovery compartment Phase I flow, and the rest of the values were kept at the CO<sub>2</sub> values. The mass rate is positive in this case however, because the CO<sub>2</sub> is entering the compartment. The mixing of air and CO<sub>2</sub> was not considered.

$$\dot{m}_{gas,r,2}(t) = \frac{A_e P_{gas,r}(t)}{\sqrt{T}} \sqrt{\frac{\gamma}{\mathfrak{R} \bar{M}_{CO_2}}} \left(\frac{\gamma + 1}{2}\right)^{-\left(\frac{\gamma+1}{2(\gamma-1)}\right)} \quad (67)$$

The pressure is still able to be calculated using the Ideal Gas Law in Equation (68) as the recovery compartment will not become a high-pressure system.

$$\dot{P}_{gas,r,2}(t) = \frac{\mathfrak{R}T}{\bar{M}_{air} V_{rec}} \dot{m}_{gas,r,2}(t) \quad (68)$$

The parachute would deploy at the time at which the pressure of the rocket was equal to the total pressure forces keeping the airframe together as seen in Equation (69).

$$t = t_{separation} \text{ when } P_{rocket}(t) = P_{total} \quad (69)$$

## 4.2 Results

### 4.2.1 Motor Performance Model (PTSS Analysis Task 1)

This section outlines the results of the motor performance model and compares it to published data provided by the manufacturer [64]. The code for this analysis was produced using MATLAB Cantera. A similar analysis was performed using NASA CEA (Chemical Equilibrium with Applications), a program utilized to obtain chemical equilibrium compositions of complex mixtures.

#### *MATLAB Cantera Equilibrium Model*

This analysis assumes that both the fuel and oxidizer are already in the gas phase, so that the input file from the previous MQP could be used that loads species information in the gas phase. This assumption helped to simplify the analysis and would likely not introduce a large error since the heat of phase transformation is much smaller than the heat of combustion. The input file was used to provide the transport properties of the required species in the equilibrium model [19]. More information about this file can be found in Section 4.1.1. The input file creates a gas object that is modified by setting three initial conditions: temperature, pressure, and the mole

fraction of the reactant species. The temperature was set to 288 degrees C, the ignition temperature of the motor. The pressure was set to the standard sea level pressure 101,325 Pa. The mole fraction was set to a ratio of 80% ammonium perchlorate moles to 20% aluminum moles. To find the heat of reaction, the initial enthalpy is recorded after the gas object is declared and the initial conditions are set. The equilibrium method is then run assuming fixed specific internal energy and specific volume. Rocket performance properties are then calculated in MATLAB using the methods and equations in Section 4.1.1. The results for the Cantera and CEA analysis can be found in Table 4.11 including the heat of reaction.

### *NASA CEA Equilibrium Model*

A CEA analysis was performed in conjunction with the Cantera analysis to confirm and compare motor performance data. The CEA analysis required the chamber pressure as an input. The chamber pressure from the Cantera model was used for this. Next aluminum and ammonium perchlorate were designated as the fuel and oxidizer starting at a temperature of 300 K. As discussed in the methodology section, both the fuel and oxidizer are assumed to be in a gaseous phase for this calculation. The last argument of the equilibrium analysis is the oxidizer to fuel weight ratio. This weight ratio was determined by finding 80% the molecular weight of ammonium perchlorate (93.992 g/mol) and dividing it by 20% of the molecular weight of aluminum (5.396) to obtain an oxidizer to fuel weight ratio of 17.41. The output of the NASA model is shown in Appendix D: NASA CEA Results.

### *Mole Fraction Results*

The equilibrium models produced similar resulting mole fractions. These results are presented in Figure 107. The similar results validate the Cantera equilibrium model methodology.

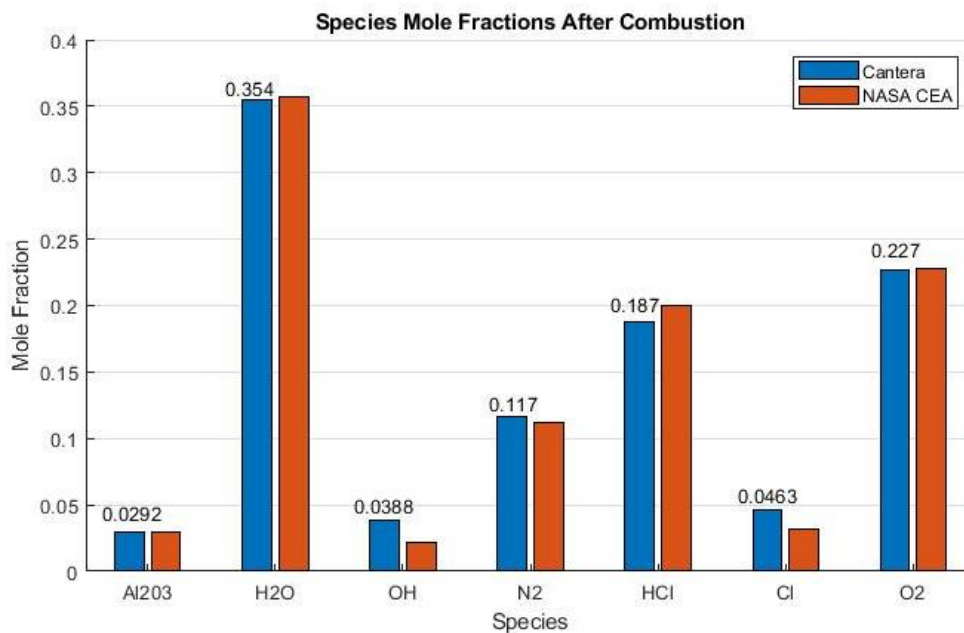


Figure 107 Combustion Product Mole Fractions from CEA and NASA Models

The difference in the amount of trace species included in both models is likely the reason for the differences in the resulting mole fractions. The NASA CEA equilibrium model considered many more trace species than the Cantera model, which produced the resulting mole fractions considering only the seven species included in the input, which are shown in Figure 107.

*Motor Performance Properties Results*

The chamber and exit properties produced by both the Cantera and CEA model are compared in Table 4.11. The Cantera model produced chamber conditions. The methods and equations described in Section 4.1.1 were used to produce the Cantera exit conditions, as well as thrust and specific impulse to estimate the motors performance. The ratio of chamber pressure to exit pressure from the Cantera model was also passed to the CEA model so that it could produce exit conditions and rocket performance information to compare to the Cantera model.

Table 4.11 Chamber and Exit Property Results

Chamber Properties	Cantera	NASA CEA	Units
Temperature	2891	2630	K
Pressure	1.796	1.796	Mpa
Density	2.17	2.478	kg/m <sup>3</sup>
Cp	1635	2379	J/kg K
Molecular Weight	29	29.27	g/mol
Specific Heat Ratio	1.21	1.158	-
Exit Properties	Cantera	NASA CEA	Units
Temperature	1426	1442	K
Pressure	0.3177	0.3179	MPa
Density	0.0778	0.08127	kg/m <sup>3</sup>
Exit Velocity	2189	692.6	m/s
Mach Number	3.111	3.051	-
Heat of Reaction	850320	-	J/kg

Table 4.12 Motor Performance Results

Motor Properties	Units	Cantera	Manufacturer Data
Thrust	N	240.0	757.7
Isp	s	181.5	224.07

The Cantera and NASA CEA results appear to match well, except for the exit velocity. The discrepancies between models are likely due to the assumptions the NASA CEA model uses. The NASA CEA program has little and poor documentation, making it hard to understand exactly how the program works, and what assumptions it is using. However, both analyses did assume that the propellant was in the gas phase. This is likely the reason why the Cantera results do not closely match the manufacturer data.

#### 4.2.2 Thermal Model (Analysis Task 2)

The results of the thermal and velocity distribution models that were developed in COMSOL are presented in this section. These results were based on the inputs provided by the Cantera model. As mentioned in the methodology for Analysis Task 2 in section 4.1.2, the results were obtained at 1.7 seconds so that there was still a thin boundary for the fuel grain to be able to include it in the model. This boundary would still exist since the burn time is 1.75 seconds. The temperature distribution of the motor is shown in Figure 108.

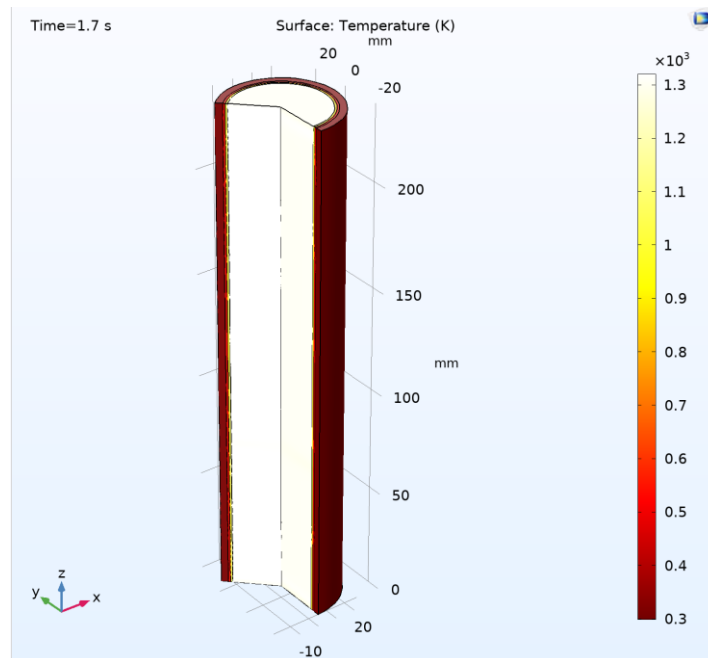


Figure 108 Motor Temperature Distribution at  $t = 1.7s$

The temperature of the fluid combustion domain was determined to be approximately 1,300K (1880.33 F) using Figure 108. In COMSOL, there is an option to create a 1D plot of temperature over time at a specific point in the model. This was used to find the temperature on the outside of the motor case to be 300 K (80.33 F). This result was important because it satisfies the National Fire Protection Association (NFPA) code 1125 which does not allow rocket motors to exceed a temperature of 498 K (436.73F) on the exterior of the motor case either during or after burning [80]. The expectation that the motor casing would be the hottest at the end of the

burn was confirmed with the model in Figure 108. As mentioned in the methodology in section 4.1.3, the temperature distribution model was coupled with a fluid flow model and the velocity distribution can also be observed at a time of 1.7 seconds. The results of this model are shown in Figure 109.

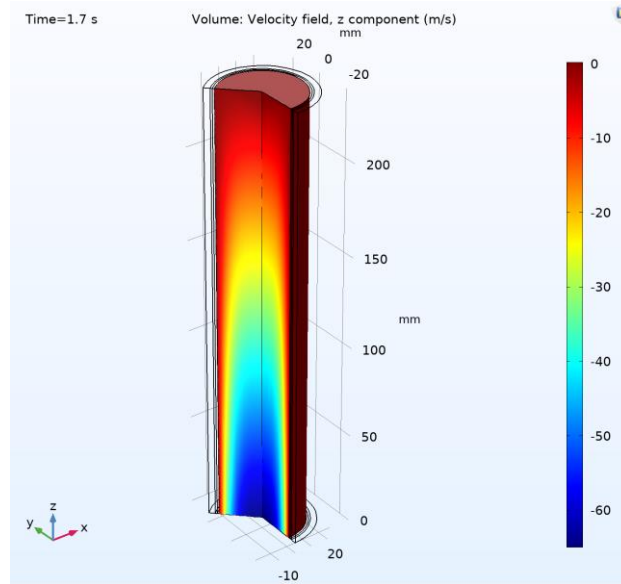


Figure 109 Velocity Distribution in the Motor at  $t = 1.7s$

The fluid is accelerated through the combustion chamber resulting in the velocity distribution being higher at the outlet as verified by the results. In the model seen in Figure 109, the negative velocity sign means that the fluid travels in the negative z-direction as defined in COMSOL, which is in the direction of the outlet boundary. The direction of this flow can be further verified by observing the streamlines of the flow as seen in Figure 110.

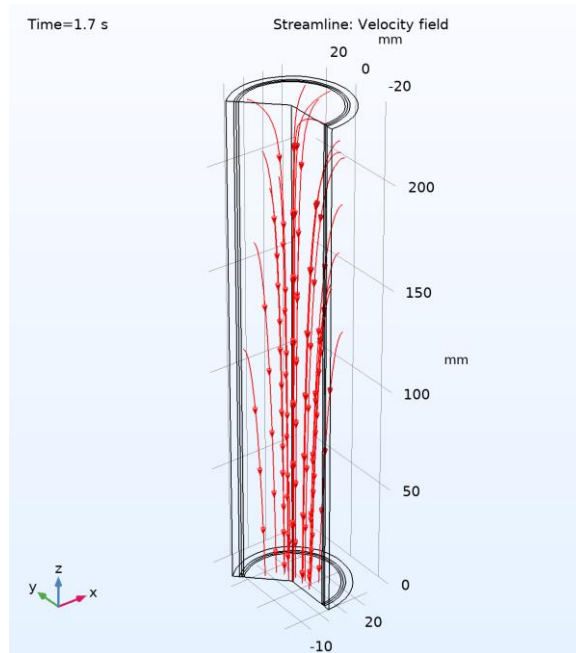


Figure 110 Velocity Streamlines for the Velocity Distribution in the Motor Case at  $t = 1.7s$

The streamlines shown in red point in the direction of flow and confirm that the fluid travels from the inlet on the fuel grain wall to the outlet at the bottom of the motor. This outlet also corresponds to the inlet of the rocket motor nozzle.

#### 4.2.3 CO<sub>2</sub> Separation Model (PTSS Analysis Task 3)

##### *Calculated CO<sub>2</sub> Canister Initial Conditions*

The system of five equations that was solved calculated the volume of CO<sub>2</sub> liquid and vapor, the mass of CO<sub>2</sub> liquid and vapor, and the moles of CO<sub>2</sub> vapor. All of the values of the system were dependent on the total volume of CO<sub>2</sub>, the total mass of CO<sub>2</sub>, and the vapor pressure of CO<sub>2</sub> at the ambient temperature. The starting conditions and the following results are summarized in Table 4.13 below.

Table 4.13 Summary of CO<sub>2</sub> Canister Initial Conditions

Name	Variable	Value	Units
Total Volume of CO <sub>2</sub>	$V_{CO_2}$	$2.573 \cdot 10^{-5}$	$m^3$
Total Mass of CO <sub>2</sub>	$m_{CO_2}$	12	$g$
Vapor Pressure of CO <sub>2</sub> at 0°C	$P_{vap}$	3475500	$Pa$
Volume of CO <sub>2</sub> Liquid	$V_{liq}$	$9.933 \cdot 10^{-6}$	$m^3$
Volume of CO <sub>2</sub> Vapor	$V_{vap}$	$1.580 \cdot 10^{-5}$	$m^3$
Mass of CO <sub>2</sub> Liquid	$m_{liq}$	10.9	$g$
Mass of CO <sub>2</sub> Vapor	$m_{vap}$	1.1	$g$



### Phase I Flow Results

For this phase of the CO<sub>2</sub> flow analysis, it was assumed that the pressure was constant for the mixed liquid-vapor system until all the liquid had vaporized. Due to the CO<sub>2</sub> inside the canister being at saturation conditions and the assumption of the flow being isothermal, this also meant that the temperature would be constant during the phase change. In addition, as previously discussed, the mass flow rate would also be constant. It was attempted to model this phase with a non-constant pressure and temperature using a system of six differential equations: mass flow rate of CO<sub>2</sub> liquid, mass flow rate of CO<sub>2</sub> vapor, change in pressure of CO<sub>2</sub> vapor, change in temperature of CO<sub>2</sub> liquid, change in temperature of CO<sub>2</sub> vapor, and change in volume of CO<sub>2</sub> vapor. This system of equations is different than the system that was solved for the initial CO<sub>2</sub> conditions since all values would change with time and the rate at which they change would not be constant. Due to time constraints, it was not feasible to simulate this system properly and accurately in MATLAB, thus the constant pressure system was pursued. While this may be a constant pressure system for this phase, but the assumption was not able to be verified.

The final values for mass, pressure, and temperature were used as the initial conditions for the following phase. In addition, the time it took for the CO<sub>2</sub> liquid to vaporize was the end condition of this phase and was the initial time for the next phase. The initial and final values for the desired parameters are summarized in Table 4.14.

Table 4.14 Summary of CO<sub>2</sub> Initial and Final Conditions for Phase I Transient Flow

Name	Variable	Value	Units
Mass Flow Rate of CO <sub>2</sub>	$\dot{m}_{CO_2,1}$	1.136	<i>g/s</i>
Initial Total Mass of CO <sub>2</sub>	$m_{CO_2,1,i}$	12	<i>g</i>
Final Total Mass of CO <sub>2</sub>	$m_{CO_2,1,f}$	1.101	<i>g</i>
Initial Pressure of CO <sub>2</sub>	$P_{CO_2,1,i}$	3475500	<i>Pa</i>
Final Pressure of CO <sub>2</sub>	$P_{CO_2,1,f}$	3475500	<i>Pa</i>
Initial Temperature of CO <sub>2</sub>	$T_{CO_2,1,i}$	273	<i>K</i>
Final Temperature of CO <sub>2</sub>	$T_{CO_2,1,i}$	273	<i>K</i>
Time to Vaporize	$t_{vaporize}$	9.599	<i>s</i>

### Phase II Flow Results

The vapor-only system for the CO<sub>2</sub> flow began when the mixed liquid-vapor system ended at  $t = 9.599s$ . This phase considered the change in mass flow rate, change in pressure, and change in temperature. As described in Section 4.1.5, the rate at which the pressure and temperature change also changes, and the three parameters were described as a system of three differential equations. This phase ended when the flow unchoked. Choked flow exists for a specific ratio of upstream pressure to downstream pressure, i.e., canister pressure to recovery compartment

pressure, and will thus be discussed in the following section. Figure 111, Figure 112, and Figure 113 show the change in pressure, mass, and temperature of CO<sub>2</sub>. In these figures, the conditions are displayed up until the condition of the mass being approximately zero was reached even though the flow will unchoke before this point.

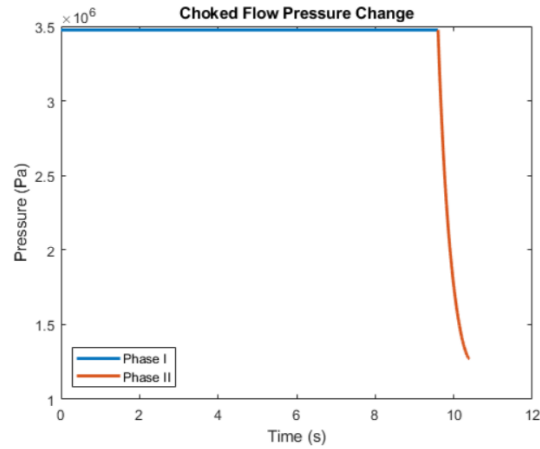


Figure 111 Change in CO<sub>2</sub> Pressure for Transient Choked Flow Phases

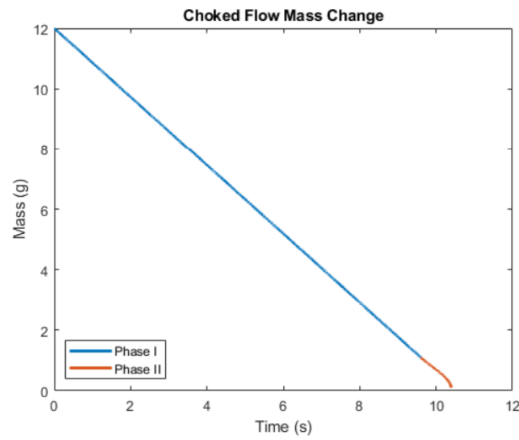


Figure 112 Change in CO<sub>2</sub> Mass for Transient Choked Flow Phases

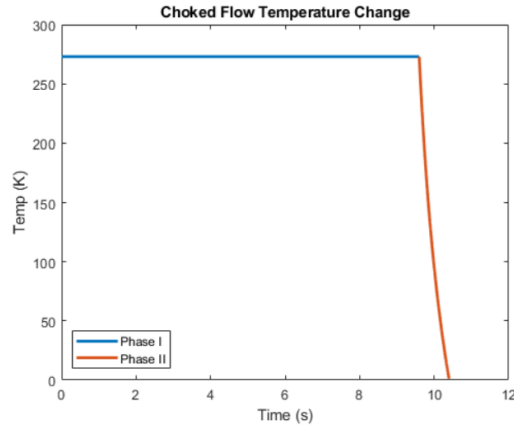


Figure 113 Change in CO<sub>2</sub> Temperature for Transient Choked Flow Phases

It is known that it is unrealistic for the final temperature to be 1.090 K. At a certain temperature the ideal gas assumption will breakdown and the CO<sub>2</sub> would turn back into liquid well after the flow would have become unchoked. As will be discussed below, the recovery compartment will reach the necessary pressure to separate during the first phase, so this was not a concern. These results were sensitive to the exit diameter of the canister. The exit diameter that was used was approximately the size of the hole that would be created once punctured, but the actual diameter could vary for each test. A hole that is 9 mm in diameter for example, which is the total diameter of the tip of the canister, would have depleted in less than 1s.

*Recovery Compartment Pressurization Results*

The initial mass of the air inside the recovery compartment is 2.760g at an atmospheric pressure of 87500 Pa. During the Phase I CO<sub>2</sub> flow, the pressure inside the compartment reaches 433,000 Pa, with a combined air and CO<sub>2</sub> mass of 13.659g. By the end of Phase II, the pressure will reach 437,000 Pa with a mass of 13.773g. Figure 114 and Figure 115 show the change in pressure and mass inside the recovery compartment over time.

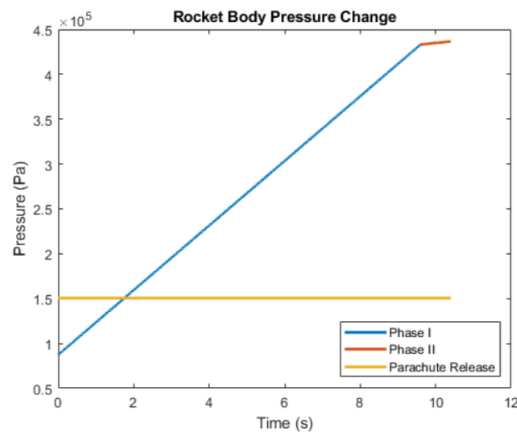


Figure 114 Change in Pressure in Recovery Compartment

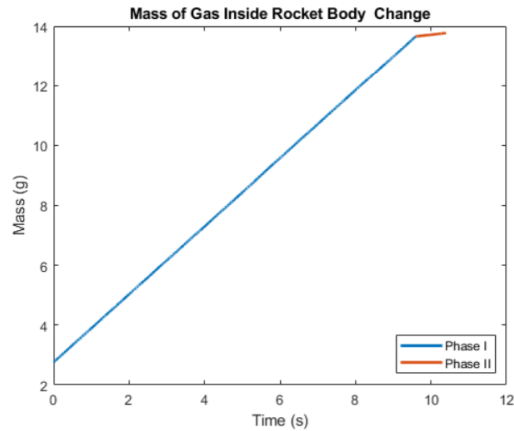


Figure 115 Change of Gas Mass in Recovery Compartment

The yellow line in Figure 114 is the pressure required to overcome the forces keeping the airframe together during flight, which was previously calculated to be 150,577 Pa. Separation occurred at time  $t = 1.752s$ . After this point, the pressure, and thus gas mass, inside the rocket would not actually continue to increase. The figures above plot until the mass is approximately zero, however Phase II will end when the flow becomes unchoked. For  $CO_2$ , the critical pressure ratio is 1.83 and this ratio will be reached some point after separation occurs, so this was not a concern.

## 5 Summary, Recommendations, and Broader Impacts

Presented below are the broader impacts and takeaways from the individual subteams. This paper discussed the design and analysis of a high-powered model rocket and the innovative design aspects therein. The rocket was designed with an innovative recovery system that used a secondary retro-firing motor to arrest the descent of the rocket before main parachute deployment, and the behavior of the rocket in response to this motor firing was modeled using a six degree of freedom flight simulator designed by the FDA subteam in MATLAB. Ansys Fluent, COMSOL/MATLAB, and Cantera were used to determine the pressure loads on the rocket due to various flight events and conditions such as body separation, maximum flight velocity, and main motor peak thrust. These loads were then used in Ansys Static Structural to determine the maximum von Mises stress experienced by the rocket when under these loads. These design and analysis tasks are presented in previous sections. These analyses also included the time from separation system activation to the separation of the upper and lower airframes as well as an analysis of the thermal performance of the motor. The analysis of the separation system both provided pressure load data as well as the data needed to accurately time the separation event. The motor thermal performance showed that the motor casing would not become a fire hazard to the rocket itself.

### 5.1 Airframe and Recovery System

The ARS subteam was responsible for designing the recovery system and main rocket structure and analyzing the structural integrity of the rocket in flight. This involved the design of two innovative systems: a retromotor mounting system and a canard actuation system and the execution of various analysis tasks. These tasks included evaluation of the motor mount load at peak motor thrust, the parachute bay load at maximum pressure before separation, the load on the main fins at maximum airspeed, the pressure on the canards at various flight conditions, and the stress on the canards due to these pressures.

The analysis of the motor mount and parachute bay were successful in showing that the parts would not fail under load. The canard pressure analysis was also successful, using Ansys Fluent to model the fluid flow around the canard. The team did run into significant issues with importing the pressures directly from Fluent into Ansys Static Structural, which was necessary for a more comprehensive analysis of the stress on the fins and canards. The team was able to solve the problem, but it did take up a very significant period of time. A future team should make sure to go through the effort of using an Ansys Mechanical mesh rather than a Fluent mesh when running their CFD analysis to reduce the amount of troubleshooting necessary. Future teams could also expand upon the composite material model. Assumptions were made regarding the number of plies, the specific type of glass fibers, and the epoxy matrix. Future teams would greatly expand upon the accuracy of this material model if they were able to determine the specifics of the fiberglass body tubes.

The team was able to explore the coupling of Ansys Fluent with Static Structural in the course of solving the pressure import issues. Future teams may be able to further explore the coupling of Fluent and Structural analysis into a comprehensive dynamic system that allows the deformation of parts to affect the fluid simulation.

Unfortunately, logistics issues prevented the team from receiving the parts needed to construct the rocket within the time limits of this project. Future teams can aim to resolve this issue by ensuring that parts orders are placed far earlier in the term so that they do not need to wait for parts to arrive.

## 5.2 Flight Dynamics Analysis

Overall, development of the dynamic simulator and verification of the vehicle's stability after retromotor firing was accomplished successfully. The simulation results demonstrate a successful implementation of a 6DOF dynamic model of the vehicle. The use of object-oriented programming resulted in a simulator that can be easily expanded to implement new features in future years. Additionally, a rotating ellipsoid earth model was developed for the simulation, increasing the accuracy of longer and higher flight simulations, an improvement on previous simulators which simulated only a flat earth model.

With this simulator, there are numerous pathways future MQP teams could take to continue development. A more detailed aerodynamic model could overcome some of the issues with dynamics at high angle of attack. Implementing Monte-Carlo simulations and a weather model based on current or forecast data could allow for a more accurate prediction of vehicle landing zone. The current code is also not optimized, resulting in long simulation times. Future teams could also develop a GUI (graphical user interface) for the simulator for improved ease of use. The current simulator architecture provides ample opportunity for further expansion and improvement.

The CFD simulation of the aerodynamic loads on the vehicle provided reasonable and useful results that were consistent with the physical system. The dynamic and static pressure data, the lift and drag data, and the majority of the moment coefficient data, were as expected given the geometry and physical characteristics of the system. This accuracy provides credibility to the half-cylinder domain used for the simulation. However, an unexpected trend was seen in the roll moment coefficient, as it was shown to increase with the angle of attack. In future aerodynamic simulation, it could be useful to examine the roll moment data more closely with the model geometry and setup to see if more accurate results could be obtained.

Additionally, the simulations conducted in Fluent were limited by the fin and canard orientation used in the model. Due to the symmetry conditions of the domain, the fins and canards needed to be configured properly in order for the results to be accurate. Future MQP teams could build off of this model to simulate flight cases with varying fin orientation, in addition to varying angle of attack and flight speed, to create a more comprehensive aerodynamic model for use in the dynamic simulator and airframe stress analysis.

The canard control system designed for this vehicle is reasonable for the desired expectations. The position, velocity, and time from motor burnout to apogee of the system was able to align with the OpenRocket model, proving it fairly accurate. The simulation of this system was able to demonstrate a successful implementation of a PID controller to decrease the roll rate to zero. Future MQP teams would be able to expand upon this in multiple ways. A non-linear state space could be used in the Simulink model in order to obtain better tuning parameters.

Implementation of a physical bench test model would be beneficial to further tune the control and collect data with how the system works in a physical application. In addition to this, if a future team is not able to launch, they could instead run a test in a wind tunnel for the same purposes noted prior.

### 5.3 Propulsion, Thermal, and Separation Systems

The motor performance model produced results in line with the team's expectations. What differences there were between the Cantera model, and the NASA CEA model were attributed to the NASA CEA model including a greater number of trace species than the Cantera model. A future team could expand upon this work by not using a zero-heat loss assumption. The team chose to assume fixed specific internal energy and specific volume to simplify the Cantera model but not making this assumption would further refine the accuracy of the motor model. A future team could also attempt to include the binding compounds in the motor grain as well, also improving the accuracy of the model. Modeling this analysis in a solid phase, rather than a gas phase, would be a better method, but would require use of different Cantera objects, and the user would need to provide solid state thermodynamic properties. Overall, future teams could attempt to reduce the number of assumptions made in the model presented in this paper in order to improve the accuracy of the results.

The motor thermal model produced accurate results in line with the team's expectations. It proved that the motor chosen would not overheat the motor casing and potentially damage the interior of the rocket. The analysis was heavily dependent on the results from the motor performance analysis and future teams should take care when moving data between the two to ensure that the thermal model accurately represents the data produced by the motor performance model. In particular, it is important for future teams to be sure about whether the heat of reaction given by the Cantera model is for the entire product or a single reactant.

The CO<sub>2</sub> separation system model accurately modeled, subject to the assumptions made, the fluid flow behavior of the CO<sub>2</sub> canisters used in the separation system through the point when the parachute bay pressure overcame the shear pins holding it together and the CO<sub>2</sub> canisters became exposed to open air. The performance of the canister model was heavily reliant on the size of the puncture hole. A future team could test different puncture sizes and determine an optimal hole size for the model. The model also made certain assumptions, namely constant pressure and constant temperature, that somewhat reduced the accuracy of the model. A future team could try to model the more complex system that these assumptions were used to avoid.

### 5.4 Project Broader Impacts

During the design and analysis of the team's High Powered Model Rocket, the team encountered many of the societal, economic, and environmental impacts that the rocket the team designed and model rocketry in general have. This section explores these impacts and discusses their importance.

The model rocketry hobby is an expansive one. It encompasses very basic rockets that parents can easily introduce to their children to large rockets of the scale designed by the team to massive rockets that push the boundaries of the atmosphere. This broad range of experience and ages

creates an environment where children are taught engineering skills from a young age and people from all walks of life come together over a shared passion. This community helps foster successive generations of engineers and rocket hobbyists.

The economic impact of the team's rocket is not insignificant. While the team was not able to design a rocket that was capable of a fully powered recovery, it does nonetheless serve as an example of a reusable rocket. There are currently commercial rockets in use, most famously the Falcon 9, that recover the lower stages of the rocket for reuse using powered descent. This method of recovery significantly lowers the cost of putting mass into orbit, expanding the availability of orbital delivery service. The rocket the team designed, as well as most model rocketry overall, is almost entirely reusable, regardless of the recovery system used. Even a basic single parachute system is designed to allow the user to relaunch the rocket, provided they have another motor. This keeps the hobby more accessible by reducing the operating costs of the model rockets.

Model rocketry can be polluting. The products of the chemical reaction that occurs within the rocket motor are toxic to human health and the environment overall. Often, model rocket launches take place in fields that are used for agriculture or on frozen bodies of water that are used for fishing and other water sports. Because of this, it is critical to ensure that the byproducts of combustion are contained at the launch site. The chemicals are not so dangerous that PPE must be used when launching a rocket, but they can contribute to hazardous farm runoff or pollution of bodies of water.



## 6 References

- [1] Cavender, D., *High Powered Video Series Counterpart Documents [PDF File]*, (n.d.). Retrieved September 20, 2021, from [https://www.nasa.gov/sites/default/files/atoms/files/sl\\_video\\_instruction\\_book.pdf](https://www.nasa.gov/sites/default/files/atoms/files/sl_video_instruction_book.pdf)
- [2] *7.5 G12 Fiberglass Filament Wound Tube 48 Long*. (n.d.). Apogee Components. Retrieved December 19, 2021, from [https://www.apogeerockets.com/Building\\_Supplies/Body\\_Tubes/Fiberglass\\_Tubes/7-5in\\_G12\\_Fiberglass\\_Filament\\_Wound\\_Tube\\_48in\\_Long?cPath=42\\_284\\_289&](https://www.apogeerockets.com/Building_Supplies/Body_Tubes/Fiberglass_Tubes/7-5in_G12_Fiberglass_Filament_Wound_Tube_48in_Long?cPath=42_284_289&)
- [3] Sr, G. A. C. *The Descriptive Geometry of Nosecones*, (n.d.). Retrieved September 20, 2021, from [http://servidor.demec.ufpr.br/CFD/bibliografia/aerodinamica/Crowell\\_1996.pdf](http://servidor.demec.ufpr.br/CFD/bibliografia/aerodinamica/Crowell_1996.pdf)
- [4] Mann, D., and Mann, L. *4-H Basic Model Rocketry Units 1 & 2*, (n.d.). Retrieved September 20, 2021, from <https://co4h.colostate.edu/projects/manuals/Model-Rocketry-Basic-U1-2.pdf>
- [5] *Intro to Dual Deployment in Rocketry*. (n.d.). Apogee Components. Retrieved December 19, 2021, from <https://www.apogeerockets.com/Intro-to-Dual-Deployment>
- [6] *High Power Rocketry: Dual Deployment*. (n.d.). West Rocketry. Retrieved December 19, 2021, from <http://westrocketry.com/articles/DualDeploy/DualDeployment.html>
- [7] *Chute Release*. (n.d.). Jolly Logic. Retrieved December 19, 2021, from <https://jollylogic.com/products/chuterelease/>
- [8] Dasch, E. J. (2006) *A Dictionary of Space Exploration*. Oxford University Press.
- [9] Flickr. (n.d.). *Falcon 9 First Stage Reusability Graphic*. [Graphic]. Retrieved December 19, 2021, from <https://www.flickr.com/photos/spacex/16892430560/sizes/l/>
- [10] Prodduturi, S. C. *A Six-Degree-of-Freedom Launch Vehicle Simulator for Range Safety Analysis*. (2007) University of Florida.
- [11] Vittaldev, V. *The Unified State Model: Derivation and Applications in Astrodynamics and Navigation*. (2021).
- [12] Clynch, J. R. *Geodetic Coordinate Conversions*. p. 5. (n.d.).
- [13] *Earth-centered, Earth-fixed coordinate system*. (2021) Wikipedia. Retrieved September 30, 2021, from <https://en.wikipedia.org/w/index.php?title=ECEF&oldid=1047184512>
- [14] *Coordinate System*. Basic Air Data. (n.d.). Retrieved October 1, 2021, from <https://www.basicairdata.eu/knowledge-center/background-topics/coordinate-system/>.
- [15] Niskanen, S. *Development of an Open Source Model Rocket Simulation Software*. (2009.)
- [16] Box, S., Bishop, C., and Hunt, H. E. M. (2010) Stochastic Six-Degree-of-Freedom Flight Simulator for Passively Controlled High Power Rockets. *Journal of Aerospace Engineering - J AEROSP ENG* 24. [https://doi.org/10.1061/\(ASCE\)AS.1943-5525.0000051](https://doi.org/10.1061/(ASCE)AS.1943-5525.0000051).
- [17] *RockSim Pro - Subscription*. (n.d.). Apogee Components. Retrieved September 30, 2021, from [https://www.apogeerockets.com/index.php?main\\_page=product\\_software\\_info&cPath=13\\_205&products\\_id=726](https://www.apogeerockets.com/index.php?main_page=product_software_info&cPath=13_205&products_id=726).
- [18] *RASAero II Users Manual*. (n.d.). Retrieved September 30, 2021 from [http://www.rasaero.com/dl\\_manual\\_ii.htm](http://www.rasaero.com/dl_manual_ii.htm).
- [19] Mitkov, A., Bigwood, J., Connor, R., Rutkowski, N., Schrader, C., and Ternet, J. "Design, Analysis, and Test of a High-Powered Model Rocket-2," WPI Major Qualifying Project (MQP) Report JB3 – 2102, Advisor: J. Blandino, 2021
- [20] Benson, T. *Rocket Stability*. (n.d.). NASA GRC. Retrieved December 20, 2021 from <https://www.grc.nasa.gov/WWW/k-12/rocket/rktstab.html>.
- [21] Kundu, P. K., Cohen, I. M., Ph.D, D. R. D. (2012) *Fluid Mechanics*. Academic Press

- [22] Hall, N. *Boundary Layer*. (n.d.). NASA GRC. Retrieved December 20, 2021, from <https://www.grc.nasa.gov/WWW/k-12/airplane/boundlay.html>.
- [23] Hall, N. *Navier-Stokes Equations*. (n.d.). Retrieved December 20, 2021, from <https://www.grc.nasa.gov/WWW/k-12/airplane/nseqs.html>.
- [24] Belliss, M., Braun, T., Brayshaw, P., Matook, G., Moore, K., Otterman, A., Sanchez-Torres, A., and Stechmann, D. “WPI AIAA Research Rocket for the Investigation and Observation of Recovery and Staging (WARRIORS I)” WPI Major Qualifying Project (MQP) Report JB3 – RCK1, Advisor: J. Blandino, 2006.
- [25] Anderson, J. D. (1995). *Computational Fluid Dynamics*. McGraw-Hill Education.
- [26] SimScale. *What Are Boundary Conditions?* (n.d.). SimScale. Retrieved December 20, 2021 from <https://www.simscale.com/docs/simwiki/numerics-background/what-are-boundary-conditions/>
- [27] SimScale. *What Is Computational Fluid Dynamics (CFD)?* (n.d.). SimScale. Retrieved December 20, 2021, from <https://www.simscale.com/docs/simwiki/cfd-computational-fluid-dynamics/what-is-cfd-computational-fluid-dynamics/>
- [28] *Ansys Fluent vs. SimScale*. (n.d.). G2. Retrieved December 20, 2021, from <https://www.g2.com/compare/ansys-fluent-vs-simscale>.
- [29] Benson, T. *Practical Rocketry*. (n.d.). NASA GRC. Retrieved December 20, 2021 from [https://www.grc.nasa.gov/www/k-12/rocket/TRCRocket/practical\\_rocketry.html](https://www.grc.nasa.gov/www/k-12/rocket/TRCRocket/practical_rocketry.html).
- [30] Kundu, A. K., Price, M. A., and Riordan, D. (2018) *Conceptual Aircraft Design: An Industrial Approach*. John Wiley & Sons.
- [31] Ferrante, R. *A Robust Control Approach for Rocket Landing*. Retrieved December 20, 2021, from [https://project-archive.inf.ed.ac.uk/msc/20172139/msc\\_proj.pdf](https://project-archive.inf.ed.ac.uk/msc/20172139/msc_proj.pdf).
- [32] Setyawan, G. E., Kurniawan, W., Gaol, A. C. L. (2019) Linear Quadratic Regulator Controller (LQR) for AR. Drone’s Safe Landing. Presented at the 2019 International Conference on Sustainable Information Engineering and Technology (SIET).
- [33] Alvarez, A., Kelly, E., Gerhardt, G., Whitehouse, J., O’Neill, J. “Design and Integration of a High-Powered Model Rocket-II,” WPI Major Qualifying Project (MQP) Report MAD – 1901, Advisor: M. Demetriou, 2019
- [34] Apogee Components, Rocket Motor Basics. (n.d.). Retrieved December 20, 2021, from <https://www.apogeerockets.com/Rocket-Motor-Basics-Quick-Start-Guide>.
- [35] National Association of Rocketry, Standard Motor Codes. (n.d.). Retrieved December 20, 2021, from <https://www.nar.org/standards-and-testing-committee/standard-motor-codes/>. Accessed Dec. 20, 2021.
- [36] Jacobs, G. Amateur Rocket Motor Ignition & Igniters. (n.d.). Retrieved December 20, 2021, from [http://www.jacobsrocketry.com/aer/ignition\\_and\\_igniters.htm](http://www.jacobsrocketry.com/aer/ignition_and_igniters.htm).
- [37] Amato, N., Huaman, Z., Hyland, A., Koslow, J., Pickunka, J., Procaccini, J., Ranjit, S., Roe, W., Romankiw, J., Santamaria, D., Scarponi, J., St. Jacques, B., and Tappen, J. “Design, Analysis, and Test of a High-Powered Model Rocket,” WPI Major Qualifying Project (MQP) Report JB3 – 2001, Advisor: J. Blandino, 2020
- [38] Cook, D. Model Rocket Igniter Controller. (n.d.). Retrieved December 20, 2021, from <http://robotroom.com/Rocket-Ignition-System-1.html>.
- [39] Murphy, C., Furter, J., Tavares, J., Handy, M., Songer, N., Farias, N., Halfrey, S., and Legere, V. “Design and Integration of a High-Powered Model Rocket-III,” WPI Major Qualifying Project (MQP) Report JB3 – 1901, Advisor: J. Blandino, 2019
- [40] Stine, G. H., and Stine, B. *Handbook of Model Rocketry*. Wiley, 2004.

- [41] Tinder Rocketry's – Peregrine Exhaustless CO2 Ejection System. (n.d.). Retrieved December 20, 2021, from [https://www.apogeerockets.com/downloads/PDFs/Tinder\\_Peregrine.pdf](https://www.apogeerockets.com/downloads/PDFs/Tinder_Peregrine.pdf).
- [42] Composite Warehouse, G12 Fiberglass. (n.d.). Retrieved December 21, 2021, from [http://www.compositewarehouse.com/index.php?route=product/category&path=25\\_28](http://www.compositewarehouse.com/index.php?route=product/category&path=25_28).
- [43] SendCutSend, G-10 (FR-4 grade). (n.d.). Retrieved December 21, 2021, from <https://sendcutsend.com/g10-fr4/>. Accessed Dec. 21, 2021.
- [44] McGinty, B. Von Mises Stress, Continuum Mechanics. (n.d.). Retrieved December 21, 2021, from <https://www.continuummechanics.org/vonmisesstress.html>.
- [45] McGinty, B. Hooke's Law, Continuum Mechanics. (n.d.). Retrieved December 21, 2021, from <https://www.continuummechanics.org/hookeslaw.html>.
- [46] University of Washington, Young's Modulus. (n.d.). Retrieved December 21, 2021, from [https://depts.washington.edu/matseed/mse\\_resources/Webpage/Biomaterials/young%27s\\_modulus.htm](https://depts.washington.edu/matseed/mse_resources/Webpage/Biomaterials/young%27s_modulus.htm).
- [47] Cai, Z., and Ross, R. J. "Mechanical Properties of Wood-Based Composite Materials." *Wood handbook : wood as an engineering material: chapter 12. Centennial ed. General technical report FPL ; GTR-190. Madison, WI : U.S. Dept. of Agriculture, Forest Service, Forest Products Laboratory, 2010: p. 12.1-12.12.*, Vol. 190, 2010, p. 12.1-12.12.
- [48] "Composites Workflow & Best Practices for FSAE using ANSYS ACP". LEAP Australia, March 2018. Retrieved February 14, 2022, from [https://www.youtube.com/playlist?list=PLvsJbyBB0CMd4MRh5\\_BCWxVdnD6jaaIQf](https://www.youtube.com/playlist?list=PLvsJbyBB0CMd4MRh5_BCWxVdnD6jaaIQf).
- [49] ANSYS FLUENT 12.0 Theory Guide, 26.1 Overview of Using the Solver. (n.d.). Retrieved December 21, 2021, from <https://www.afs.enea.it/project/neptunius/docs/fluent/html/ug/node776.htm>.
- [50] ANSYS FLUENT 12.0 Theory Guide, 18.1.1 Pressure-Based Solver. (n.d.). Retrieved December 21, 2021, from <https://www.afs.enea.it/project/neptunius/docs/fluent/html/th/node361.htm>.
- [51] Reynolds Number for Turbulent Flow. (n.d.). Retrieved December 21, 2021, from <https://www.nuclear-power.com/nuclear-engineering/fluid-dynamics/reynolds-number/reynolds-number-for-turbulent-flow/>.
- [52] SparkFun MicroMod Teensy Processor - DEV-16402 - SparkFun Electronics. (n.d.). Retrieved March 7, 2022, from <https://www.sparkfun.com/products/16402>.
- [53] Microchip Technology, MCP2515. (n.d.). Retrieved March 7, 2022, from <https://ww1.microchip.com/downloads/en/DeviceDoc/MCP2515-Stand-Alone-CAN-Controller-with-SPI-20001801J.pdf>.
- [54] Philips Semiconductors, TJA1050. (n.d.). Retrieved March 7, 2022, from <https://www.nxp.com/docs/en/data-sheet/TJA1050.pdf>.
- [55] TDK Corporation, (n.d.). Retrieved March 7, 2022, from ICM-20948. [https://invensense.tdk.com/wp-content/uploads/2016/06/DS-000189-ICM-20948-v1.3.pdf?ref\\_disty=digikey](https://invensense.tdk.com/wp-content/uploads/2016/06/DS-000189-ICM-20948-v1.3.pdf?ref_disty=digikey).
- [56] NXP Semiconductors, MPL3115A2. (n.d.). Retrieved March 7, 2022, from <https://www.nxp.com/docs/en/data-sheet/MPL3115A2.pdf>.
- [57] u-blox, NEO-M9N-00B. (n.d.). Retrieved March 7, 2022, from <https://www.u-blox.com/en/docs/UBX-19014285>.

- [58] Atmel, ATmega328P. (n.d.). Retrieved March 7, 2022, from [https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf).
- [59] XTX Technology Limited, XTSD01G/XTSD02G/XTSD04G/XTSD08G. (n.d.). Retrieved March 7, 2022, from [https://cdn-shop.adafruit.com/product-files/4899/2005251034\\_XTX-XTSD04GLGEAG\\_C558839\(2\).pdf](https://cdn-shop.adafruit.com/product-files/4899/2005251034_XTX-XTSD04GLGEAG_C558839(2).pdf).
- [60] Semtech, SZ1276/77/78/79. (n.d.). Retrieved March 7, 2022, from [https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R0000001Rbr/6EfVZUorrpoK FfvaF\\_Fkpgp5kzjiNyiAbqcpqh9qSjE](https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R0000001Rbr/6EfVZUorrpoK FfvaF_Fkpgp5kzjiNyiAbqcpqh9qSjE).
- [61] National Imagery and Mapping Agency (NIMA). Department of Defense World Geodetic System 1984: Its Definition, and Relationship with Local Geodetic Systems, TR8350.2, Third Ed. (n.d.). Retrieved December 8, 2021, from <https://gis-lab.info/docs/nima-tr8350.2-wgs84fin.pdf>.
- [62] Portland State Aerospace Society, Roll Control. (n.d.). Retrieved December 21, 2021, from <https://archive.psas.pdx.edu/rollcontrol/>.
- [63] MathWorks Help Center, PID Controller (2DOF). (n.d.). Retrieved December 21, 2021, from <https://www.mathworks.com/help/simulink/slref/pidcontroller2dof.html>.
- [64] Benson, T. (2021, May 13). Shape effects on drag. NASA. Retrieved March 22, 2022, from <https://www.grc.nasa.gov/www/k-12/rocket/shaped.html#:~:text=A%20typical%20value%20for%20the,body%20and%20the%20nozzle%20exit>.
- [65] Cesaroni - P54-3G White Thunder (J760). (n.d.). Retrieved March 7, 2022, from [https://www.apogeerockets.com/Rocket\\_Motors/Cesaroni\\_Propellant\\_Kits/54mm\\_Motors/3-Grain\\_Motors/Cesaroni\\_P54-3G\\_White\\_Thunder\\_J760](https://www.apogeerockets.com/Rocket_Motors/Cesaroni_Propellant_Kits/54mm_Motors/3-Grain_Motors/Cesaroni_P54-3G_White_Thunder_J760).
- [66] Goodwin, D., Speth, R., Moffat, H., and Weber, B. Cantera: An Object-Oriented Software Toolkit for Chemical Kinetics, Thermodynamics, and Transport Processes. (2021). Retrieved December 21, 2021, from <https://doi.org/10.5281/4527812>.
- [67] Snyder, C. Chemical Equilibrium with Applications. (n.d.). Retrieved December 21, 2021, from <https://cearun.grc.nasa.gov/>.
- [68] Manion, J., Huie, R., Levin, R., Burgess, D., Orkin, V., Tsang, W., McGivern, W., Hudgens, W., Knyazev, V., Atkinson, D., Chai, E., Tereza, A., Lin, C., Allison, T., Mallard, W., Westley, F., Herron, J., Hampson, R., and Frizzell, D. NIST Chemical Kinetics Database. (n.d.). Retrieved December 21, 2021, from *National Institute of Standards and Technology*. <https://kinetics.nist.gov/>.
- [69] Grimm, A. A Learning Guide for Model Rocket Launch Systems. Retrieved December 21, 2021, from [https://estesrockets.com/wp-content/uploads/Educator/2811\\_Estes\\_Model\\_Rocket\\_Launch\\_Systems.pdf](https://estesrockets.com/wp-content/uploads/Educator/2811_Estes_Model_Rocket_Launch_Systems.pdf).
- [70] CFD Module User's Guide. Retrieved December 21, 2021, from <https://doc.comsol.com/5.4/doc/com.comsol.help.cfd/CFDModuleUsersGuide.pdf>.
- [71] Heat Transfer Module User's Guide. Retrieved December 21, 2021, from <https://doc.comsol.com/5.4/doc/com.comsol.help.heat/HeatTransferModuleUsersGuide.pdf>.
- [72] National Centers for Environmental Information (NCEI), Lake Winnepesaukee Weather Averages. (n.d.). Retrieved December 21, 2021, from <http://www.ncei.noaa.gov/node>.
- [73] Weather Atlas, Wolfeboro, NH - February Weather Forecast and Climate Information. (n.d.). Retrieved December 21, 2021, from <https://www.weather-us.com/en/new-hampshire-usa/wolfeboro-weather-february>.

- [74] Carbon Dioxide. (n.d.). Retrieved December 21, 2021, from [https://en.wikipedia.org/w/index.php?title=Carbon\\_dioxide&oldid=1059209704](https://en.wikipedia.org/w/index.php?title=Carbon_dioxide&oldid=1059209704).
- [75] MathWorks Help Center, MATLAB Vpasolve. (n.d.). Retrieved December 21, 2021, from <https://www.mathworks.com/help/symbolic/sym.vpasolve.html>.
- [76] MathWorks Help Center, MATLAB Fsolve. (n.d.). Retrieved December 21, 2021, from <https://www.mathworks.com/help/optim/ug/fsolve.html>.
- [77] Chemistry LibreTexts, Le Chatelier's Principle. (2020). Retrieved January 26, 2022, from [https://chem.libretexts.org/Bookshelves/Physical\\_and\\_Theoretical\\_Chemistry\\_Textbook\\_Maps/Supplemental\\_Modules\\_\(Physical\\_and\\_Theoretical\\_Chemistry\)/Equilibria/Le\\_Chate liers\\_Principle#:~:text=Le%20Ch%C3%A2telier's%20principle%20states%20that,change%20to%20reestablish%20an%20equilibrium](https://chem.libretexts.org/Bookshelves/Physical_and_Theoretical_Chemistry_Textbook_Maps/Supplemental_Modules_(Physical_and_Theoretical_Chemistry)/Equilibria/Le_Chate liers_Principle#:~:text=Le%20Ch%C3%A2telier's%20principle%20states%20that,change%20to%20reestablish%20an%20equilibrium).
- [78] MathWorks Help Center, MATLAB Double. (n.d.). Retrieved December 21, 2021, from <https://www.mathworks.com/help/symbolic/double.html>.
- [79] Hall, N. Mass Flow Choking. NASA GRC. (n.d.). Retrieved December 21, 2021, from <https://www.grc.nasa.gov/www/k-12/airplane/mflchk.html>.
- [80] MathWorks Help Center, MATLAB Ode45. (n.d.). Retrieved December 21, 2021, from <https://www.mathworks.com/help/matlab/ref/ode45.html>.
- [81] NFPA 1125: Code for the Manufacture of Model Rocket and High-Power Rocket Motors. (n.d.). Retrieved March 6, 2022, from <https://www.nfpa.org/codes-and-standards/all-codes-and-standards/list-of-codes-and-standards/detail?code=1125>.

## 7 Appendices

### 7.1 Appendix A: Equations in ARS Analysis Task 1

Condition	Equation	Variables
Hooke's Law (Relation of Stress to Strain)	$\epsilon_{ij} = \frac{1}{E} \{ (1 + \nu) \sigma_{ij} - \nu \delta_{ij} \sigma_{kk} \}$	$\epsilon_{ij}$ – Strain Tensor (unitless) $E$ – Young's Modulus (Pa) $\nu$ – Poisson's Ratio (unitless) $\sigma_{ij}$ – Stress Tensor (Pa) $\delta_{ij}$ – Identity Matrix $\sigma_{kk}$ – Principal Stresses (Pa)
von Mises Stress	$\sigma_v = \sqrt{\frac{1}{2} [(\sigma_{11} - \sigma_{22})^2 + (\sigma_{22} - \sigma_{33})^2 + (\sigma_{33} - \sigma_{11})^2] + 3(\sigma_{12}^2 + \sigma_{23}^2 + \sigma_{31}^2)}$	$\sigma_{11}, \sigma_{22}, \sigma_{33}$ – Principal Stresses (Pa) $\sigma_{12}, \sigma_{23}, \sigma_{31}$ – Off-Axis Stresses (Pa)

### 7.2 Appendix B: ARS Analysis Task 1 Material Properties

Material	Properties
Epoxy E-Glass, UD	$E_x = 4.5E + 10 \text{ Pa}$ $E_y = 1E + 10 \text{ Pa}$ $E_z = 1E + 10 \text{ Pa}$ $\nu_{xy} = .3$ $\nu_{yz} = .4$ $\nu_{xz} = .3$
Plywood	$E = 2.278E + 10 \text{ Pa}$ $\nu = .3742$

### 7.3 Appendix C: FDA Analysis Task 2 Fluent Simulation Data

Table 7.1 Aerodynamic Coefficient Data for Varying Angle of Attack and Flight Speed

Angle of Attack (degrees)	Flight Speed (m/s)	Cd	Cl	C <sub>m,roll</sub>	C <sub>m,pitch</sub>	C <sub>m,yaw</sub>
0	20	0.69	-0.0031	0.0000555	-0.00262	0.748
	60	0.659	-0.00682	0.0000747	-0.00512	0.742
	100	0.647	-0.0086	0.000087	-0.00681	0.741
	140	0.64	-0.0094	0.0000915	-0.00745	0.74
	180	0.636	-0.0101	0.0000965	-0.00798	0.74
2	20	0.719	0.593	0.0136	0.308	0.766
	60	0.686	0.596	0.014	0.311	0.76
	100	0.673	0.596	0.0141	0.311	0.759
	140	0.666	0.596	0.0142	0.311	0.758
	180	0.661	0.596	0.0142	0.311	0.757
4	20	0.794	1.2	0.028	0.623	0.815
	60	0.758	1.21	0.0286	0.63	0.809
	100	0.745	1.21	0.0288	0.632	0.807
	140	0.737	1.21	0.029	0.633	0.806
	180	0.731	1.21	0.0291	0.634	0.806
6	20	0.915	1.85	0.0426	0.961	0.893
	60	0.875	1.85	0.0436	0.97	0.886
	100	0.861	1.85	0.0439	0.972	0.885
	140	0.852	1.85	0.0441	0.974	0.884
	180	0.846	1.85	0.0442	0.974	0.884
8	20	1.07	2.51	0.0574	1.31	1
	60	1.03	2.52	0.0587	1.32	0.996
	100	1.02	2.52	0.0591	1.33	0.995
	140	1.01	2.52	0.0593	1.33	0.994
	180	0.999	2.52	0.0595	1.33	0.994
10	20	1.27	3.17	0.0719	1.66	1.14
	60	1.22	3.18	0.0736	1.67	1.14
	100	1.21	3.18	0.0741	1.68	1.13
	140	1.2	3.17	0.0744	1.68	1.13
	180	1.19	3.17	0.0746	1.68	1.13
12	20	1.5	3.83	0.0859	2	1.31
	60	1.45	3.82	0.0877	2.02	1.31
	100	1.44	3.82	0.0883	2.02	1.31
	140	1.43	3.82	0.0887	2.03	1.31
	180	1.42	3.82	0.0889	2.03	1.31
14	20	1.79	4.49	0.0986	2.36	1.53
	60	1.74	4.48	0.101	2.38	1.53

14	100	1.72	4.48	0.101	2.38	1.53
	140	1.71	4.47	0.102	2.38	1.53
	180	1.7	4.47	0.102	2.39	1.53
16	20	2.12	5.16	0.11	2.73	1.79
	60	2.06	5.13	0.112	2.74	1.79
	100	2.04	5.12	0.112	2.75	1.79
	140	2.02	5.12	0.113	2.75	1.79
	180	2.02	5.11	0.113	2.75	1.79
18	20	2.48	5.79	0.118	3.09	2.07
	60	2.42	5.75	0.12	3.09	2.08
	100	2.39	5.73	0.121	3.09	2.08
	140	2.38	5.72	0.121	3.09	2.08
	180	2.37	5.72	0.122	3.09	2.09
20	20	2.88	6.38	0.124	3.43	2.38
	60	2.81	6.32	0.126	3.43	2.4
	100	2.78	6.3	0.127	3.43	2.41
	140	2.77	6.29	0.127	3.43	2.41
	180	2.76	6.29	0.127	3.43	2.41

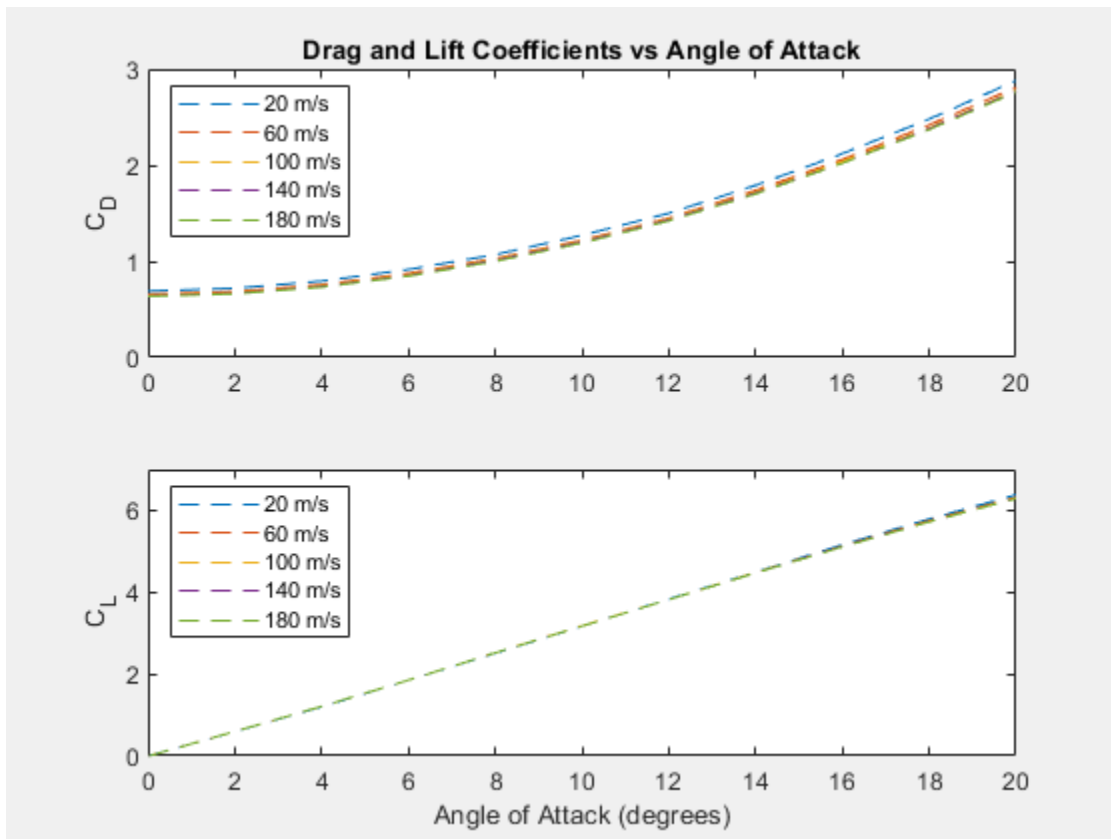


Figure 116 Drag and Lift Coefficients vs. Angle of Attack for Varying Flight Speed



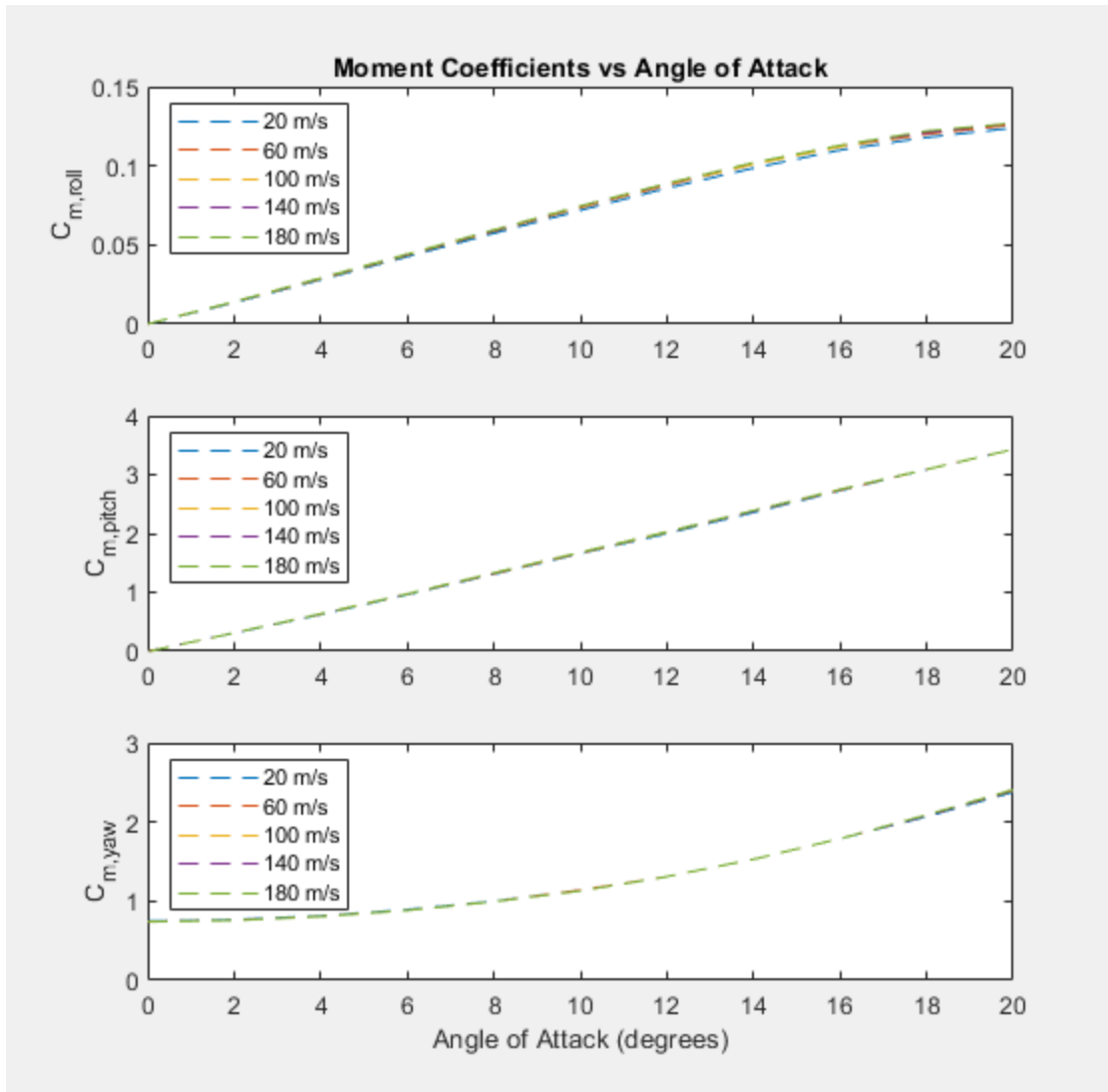


Figure 117 Moment Coefficients vs. Angle of Attack for Varying Flight Speed

## 7.4 Appendix D: NASA CEA Results

```

NASA-GLENN CHEMICAL EQUILIBRIUM PROGRAM CEA2, FEBRUARY 5, 2004
  BY BONNIE MCBRIDE AND SANFORD GORDON
REFS: NASA RP-1311, PART I, 1994 AND NASA RP-1311, PART II, 1996
*****

### CEA analysis performed on Mon 28-Feb-2022 15:51:57

# Problem Type: "Rocket" (Infinite Area Combustor)

prob case=_____6267 ro equilibrium

# Pressure (1 value):
p,bar= 17.96
# Chamber/Exit Pressure Ratio (1 value):
pi/p= 56.5

# Oxidizer/Fuel Wt. ratio (1 value):
o/f= 17

# You selected the following fuels and oxidizers:
reac
fuel AL          wt%=100.0000  t,k= 300.000
oxid NH4CLO4(I)  wt%=100.0000  t,k= 300.000

# You selected these options for output:
# short version of output
output short
# Proportions of any products will be expressed as Mole Fractions.
# Heat will be expressed as siunits
output siunits

# Input prepared by this script:/var/www/sites/cearun.grc.nasa.gov/cgi-bin/CEARU
N/prepareInputFile.cgi

### IMPORTANT: The following line is the end of your CEA input file!
end

      THEORETICAL ROCKET PERFORMANCE ASSUMING EQUILIBRIUM
      COMPOSITION DURING EXPANSION FROM INFINITE AREA COMBUSTOR

Pin = 260.5 PSIA
CASE = _____

      REACTANT              WT FRACTION      ENERGY          TEMP
      (SEE NOTE)           KJ/KG-MOL        K
FUEL      AL              1.0000000       330039.566       300.000
OXIDANT   NH4CLO4(I)     1.0000000       -295529.716      300.000

O/F= 17.00000  %FUEL= 5.555556  R,EQ.RATIO= 0.529825  PHI,EQ.RATIO= 0.153686

      CHAMBER  THROAT  EXIT
Pinf/P      1.0000  1.7509  56.500
P, BAR      17.960  10.257  0.31788
T, K        2630.72  2443.87  1442.36
RHO, KG/CU M 2.4776 0 1.5298 0 8.1267-2
H, KJ/KG    -1696.08 -2086.79 -3928.54
U, KJ/KG    -2420.97 -2757.27 -4319.69
G, KJ/KG    -25234.7 -23953.6 -16834.2

```

S, KJ/(KG)(K)	8.9476	8.9476	8.9476
M, (1/n)	30.175	30.306	30.660
MW, MOL WT	29.273	29.393	29.722
(dLV/dLP)t	-1.00529	-1.00374	-1.00061
(dLV/dLT)p	1.1075	1.0795	1.0104
Cp, KJ/(KG)(K)	2.3797	2.1941	1.4942
GAMMAS	1.1584	1.1655	1.2265
SON VEL, M/SEC	916.4	884.0	692.6
MACH NUMBER	0.000	1.000	3.051

PERFORMANCE PARAMETERS

Ae/At	1.0000	7.8753
CSTAR, M/SEC	1328.1	1328.1
CF	0.6656	1.5911
Ivac, M/SEC	1642.5	2298.2
Isp, M/SEC	884.0	2113.0

MOLE FRACTIONS

ALCL3	0.00003	0.00002	0.00000
ALOCL	0.00001	0.00000	0.00000
ALOHCL2	0.00026	0.00015	0.00000
AL(OH)2CL	0.00013	0.00007	0.00000
AL(OH)3	0.00007	0.00003	0.00000
*CL	0.03230	0.02729	0.00275
CLO	0.00068	0.00048	0.00002
CL2	0.00090	0.00090	0.00162
*H	0.00044	0.00019	0.00000
HCL	0.19952	0.20611	0.23288
HOCL	0.00024	0.00018	0.00002
HO2	0.00008	0.00005	0.00000
*H2	0.00188	0.00103	0.00000
H2O	0.35733	0.36082	0.36131
H2O2	0.00001	0.00000	0.00000
*NO	0.01124	0.00835	0.00039
NOCL	0.00001	0.00000	0.00000
NO2	0.00004	0.00003	0.00000
*N2	0.11201	0.11395	0.11926
*O	0.00297	0.00162	0.00000
*OH	0.02179	0.01448	0.00014
*O2	0.22817	0.23412	0.25099
AL2O3(a)	0.00000	0.00000	0.03060
AL2O3(L)	0.02988	0.03012	0.00000

\* THERMODYNAMIC PROPERTIES FITTED TO 20000.K

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

## 7.5 Appendix E: Equations in COMSOL Model

Condition	Equation	Variables
Heat transfer in Solids	$\rho C_p \frac{\partial T}{\partial t} + \rho C_p \mathbf{u} \cdot \nabla T + \nabla \cdot \mathbf{q} = Q + Q_{ted}$ $\mathbf{q} = -k \nabla T$	<p><math>\rho</math> – solid density (kg/m<sup>3</sup>)</p> <p><math>C_p</math> – solid heat capacity at constant pressure (J/kg-K)</p> <p><math>k</math> – solid thermal conductivity (W/m-K)</p> <p><math>\mathbf{u}</math> – velocity field (m/s)</p> <p><math>Q</math> – heat source (W/m<sup>3</sup>)</p> <p><math>Q_{ted}</math> – thermoelastic damping (W/m<sup>3</sup>)</p>
Heat transfer in Fluids	$\rho C_p \frac{\partial T}{\partial t} + \rho C_p \mathbf{u} \cdot \nabla T + \nabla \cdot \mathbf{q} = Q + Q_p + Q_{vd}$ $\mathbf{q} = -k \nabla T$	<p><math>Q_p</math> – pressure work (W/m<sup>3</sup>)</p> <p><math>Q_{vd}</math> – viscous dissipation (W/m<sup>3</sup>)</p> <p>See “Heat transfer in Solids” for other variable definitions</p>
Symmetry and Thermal Insulation Boundary	$-\mathbf{n} \cdot \mathbf{q} = 0$	<p><math>\mathbf{q}</math> – heat flux (W/m<sup>2</sup>)</p> <p><math>\mathbf{n}</math> – normal vector</p>
Convective Heat Flux Boundary	$-\mathbf{n} \cdot \mathbf{q} = q_0$ $q_0 = h \cdot (T_{ext} - T)$	<p><math>\mathbf{q}</math> – heat flux (W/m<sup>2</sup>)</p> <p><math>\mathbf{n}</math> – normal vector</p> <p><math>h</math> - heat transfer coefficient (W/m<sup>2</sup>-K)</p> <p><math>T_{ext}</math> – external temperature (K)</p> <p><math>T</math> – domain temperature (K)</p>
Temperature Boundary	$T = T_0$	<p><math>T_0</math> – boundary temperature (K)</p>
Weakly Compressible Laminar Flow	$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho (\mathbf{u} \cdot \nabla) \mathbf{u} = \nabla \cdot [-p \mathbf{I} + \mathbf{K}] + \mathbf{F}$ $\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u})$ $\mathbf{K} = \mu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) - \frac{2}{3} \mu (\nabla \cdot \mathbf{u}) \mathbf{I}$	<p><math>\mathbf{I}</math> – Identity matrix</p> <p><math>\mathbf{u}</math> – velocity vector (m/s)</p> <p><math>\rho</math> – solid density (kg/m<sup>3</sup>)</p> <p><math>p</math> – pressure (Pa)</p> <p><math>\mathbf{F}</math> – external force (N)</p> <p><math>\mu</math> – dynamic viscosity (Pa-s)</p>
Wall Boundary (no-slip)	$\mathbf{u} = 0$	<p><math>\mathbf{u}</math> – fluid velocity (m/s)</p>
Inlet Boundary	$\mathbf{u} = \mathbf{u}_0$	<p><math>\mathbf{u}_0</math> – initial velocity field (m/s)</p>

<p>Outlet Pressure Boundary</p>	$[-p\mathbf{I} + \mathbf{K}]\mathbf{n} = -\hat{p}_0\mathbf{n}$ $\mathbf{K} = \mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T) - \frac{2}{3}\mu(\nabla \cdot \mathbf{u})\mathbf{I}$ $\hat{p}_0 \leq p_0$	<p><math>\mathbf{I}</math> – identity matrix  <math>\mathbf{u}</math> – velocity vector (m/s)  <math>p_0</math> – absolute pressure (Pa)  <math>\mu</math> – dynamic viscosity (Pa-s)  <math>\mathbf{n}</math> – normal vector</p>
---------------------------------	---	--

## 7.6 Appendix F: Equation Derivations used for PTSS Analysis Task 3

Derivative of Van der Waals Equation with mass, pressure, and temperature as a function of time for CO<sub>2</sub>

$$P(t) = \frac{m(t)\mathfrak{R}T(t)}{\bar{M}V - m(t)b} - \frac{m^2(t)a}{\bar{M}^2V^2}$$

$$\dot{P}(t) = \frac{d}{dt} \left( \frac{m(t)\mathfrak{R}T(t)}{\bar{M}V - m(t)b} \right) - \frac{d}{dt} \left( \frac{m^2(t)a}{\bar{M}^2V^2} \right)$$

$$\dot{P}(t) = \frac{\frac{d}{dt}(m(t)\mathfrak{R}T(t)) \cdot (\bar{M}V - m(t)b) - \frac{d}{dt}(\bar{M}V - m(t)b) \cdot (m(t)\mathfrak{R}T(t))}{(\bar{M}V - m(t)b)^2} - \frac{2m(t)a}{\bar{M}^2V^2}$$

$$\dot{P}(t) = \frac{\dot{m}(t)\mathfrak{R}T(t) + m(t)\mathfrak{R}\dot{T}(t)}{\bar{M}V - m(t)b} + \frac{\dot{m}(t)m(t)\mathfrak{R}T(t)b}{(\bar{M}V - m(t)b)^2} - \frac{2m(t)a}{\bar{M}^2V^2}$$

Derivative of Conservation of Energy Equation with mass, pressure, and temperature as a function of time for CO<sub>2</sub>

$$\frac{d}{dt} \iiint_V \rho \left( e + \frac{v^2}{2} + g_z \right) = - \iint_{A_{exit}} \rho \left( h + \frac{v^2}{2} + g_z \right) \vec{v} \cdot d\vec{A}$$

$$\frac{d}{dt} (m(t)e(t)) = -\rho v_{exit} A_{exit} \left( h + \frac{v^2}{2} \right)$$

$$C_v \frac{d}{dt} (m(t)T(t)) = -\dot{m}(t)C_p T(t)$$

$$\frac{1}{\gamma} (m(t)\dot{T}(t) + T(t)\dot{m}(t)) = -\dot{m}(t)T(t)$$

$$\dot{T}(t) = -\frac{1}{m(t)} (\gamma + 1) (\dot{m}(t)T(t))$$