JB3-2101

Design, Analysis, and Test of a High-Powered Model Rocket-1

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

Sophie Balkind　Matthew Barrett　Matthew Karns

Sophie Balkind　Matthew Barrett　Matthew Karns

Brendan Marion　Mark Payne　David Robie

Brendan Marion　Mark Payne　David Robie

Adam Sochacki　Krystina Waters

Adam Sochacki　Krystina Waters

April 13th, 2021

Approved By: John J. Blandino
John J. Blandino, Advisor
Professor, Aerospace Engineering Department
Worcester Polytechnic Institute

Abstract

This paper describes the design of a high-powered model rocket (HPMR) that incorporates innovative subsystems for stage separation, and recovery. The first stage of the HPMR is separated using a set of airbrakes deployed in flight. The second stage uses an autorotation system to control the descent velocity. 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. The electrical and thermal characteristics of the e-match ignitor were also modeled using COMSOL. Aerodynamic loads on the vehicle airframe, autorotation blades, air brakes, and fins used for spin stabilization were evaluated using computational fluid dynamic (CFD) tools in Ansys Fluent. Results from the CFD analysis were used as inputs to a dynamical simulation of the vehicle trajectory, implemented in MATLAB. Ansys Workbench was used for structural analysis. Results are presented from these analyses as well as a description of prototype construction and testing completed at the subsystem level.

Acknowledgements

We would like to thank the following individuals for their help and support throughout the entirety of this project:

- Project Advisor Professor John Blandino for his guidance, wisdom, and encouragement throughout the project
- Professor Adriana Hera for training and helping the team in Ansys, Fluent, MATLAB, etc
- Ms. Tina Stratis for aiding the team in purchasing materials used for the project

Table of Contents

Table of Contents iii

List of Figures ix

List of Tables xv

Table of Authorship xvi

Nomenclature ii

1 Introduction 3

1.1 Project Goals 6

1.2 Project Design Requirements, Constraints, and Other Considerations 8

1.3 Safety Goals 9

1.4 Tasks and Timeframe 9

2 Literature Review 17

2.1 Airframe and Recovery System 17

2.1.1 Airframe 18

2.1.2 Nose Cone 18

2.1.3 Fins 21

2.1.3.1 Fin Design 21

2.1.3.2 Fin Can 24

2.1.4 Avionics Bay 26

2.1.5 Payload Bay 27

2.1.6 Recovery System 29

2.1.6.1 Autorotation Recovery 29

2.1.6.2 Parachute Recovery 33

2.1.7 Motor Retention.. 37

2.2 Propulsion, Thermal, and Separation Systems (PTSS).. 37

2.2.1 Arc Lighters.. 37

2.2.2 Stage Separation and Air Brakes.. 39

2.2.3 Model Rocket Motors... 41

2.3 Flight Dynamic Analysis 43

2.3.1 Models for Attitude and Trajectory.. 43

2.3.2 Stabilization and Control Systems... 46

3 Methodology 48

3.1 Airframe and Recovery System.. 48

3.1.1 Airframe.. 49

3.1.2 Fins 50

3.1.3 Avionics Bay 52

3.1.4 Autorotation System... 56

3.1.5 Payload Bay... 58

3.1.6 Nose Cone... 60

3.2 Propulsion, Thermal, and Separation Systems 61

3.2.1 Motor Selection 61

3.2.2 Motor Tube Design.. 62

3.2.3 Air Brake Separation System 63

3.2.4 Ignition System 65

3.3 Flight Dynamic Analysis 68

3.3.1 FDA Analysis Task 1 68

3.3.2 FDA Task 2 78

3.3.3 FDA Analysis Task 3 81

4 Analysis 86

4.1 Airframe and Recovery System 86

4.1.1 ARS Analysis Task 1: Airframe Stress Distribution 86

4.1.2 ARS Analysis Task 2: Autorotation System Model 99

4.1.3 ARS Analysis Task 3: Autorotation Blade Aerodynamic Loads 112

4.1.4 ARS Analysis Task 4: Stabilizing Fin Stress Distribution 117

4.2 Propulsion, Thermal, and Separation Systems 120

4.2.1 PTSS Analysis Task 1: Motor Performance Model 121

4.2.1.1 First Stage Motor Results 124

4.2.1.2 Second Stage Motor Results 126

4.2.2 PTSS Analysis Task 2: Temperature Distribution 128

4.2.2.1 Heat Transfer Analysis 131

4.2.2.2 Laminar Flow Analysis 134

4.2.2.3 COMSOL Analysis Results 136

4.2.3 PTSS Analysis Task 3: Mechanical Separation System Model 141

4.2.4 PTSS Analysis Task 4: Electrical Match Model 148

4.3 Flight Dynamic Analysis 156

4.3.1 FDA Analysis Task 1: Vehicle Dynamics and Performance Model 156

4.3.2 FDA Analysis Task 2: Vehicle Aerodynamic Loads – Simulation............... 166

4.3.3 FDA Analysis Task 3: Stabilization Fin Aerodynamic Loads 170

5 Summary, Recommendations, Broader Impacts 174

5.1 Airframe and Recovery System: Summary and Recommendations............................ 176

5.2 Propulsion Thermal Separation Systems: Summary and Recommendations 177

5.3 Flight Dynamic Analysis: Summary, Recommendations and Broader Impacts.......... 179

5.4 Overall Project Broader Impacts.. 181

6 References 184

7 Appendices 190

7.1 Appendix A Gantt Chart 190

7.2 Appendix B MATLAB Code for Autorotation Model 190

7.2.1 Autorotation System Model Rocket Team 1 2020-2021............................... 190

7.2.2 Defining Initial Rocket Parameters 191

7.2.3 Data Acquired from XFLR5.. 191

7.2.4 HPMR Flight Events 191

7.2.5 Initial conditions.. 192

7.2.6 Variable and Equation List.. 192

7.2.7 Execute 193

7.2.8 calc_coefficients.m 195

7.2.9 calcalpha.m 197

7.2.10 calcd_D.m 197

7.2.11 calcd_L.m 198

7.2.12 calcphi.m 198

7.2.13 calcr_e.m 198

7.2.14 calcv_rel.m 199

7.3 Appendix C Data File for Cantera Equilibrium Analysis 200

7.4 Appendix D Cantera and Motor Analysis MATLAB Code 202

7.5 Appendix E FDA Task 1 Codes 203

7.5.1 20’-21’ Rocket team 1 Main Analysis code for FDA Translational Solver .. 203

7.5.2 Point_Mass_ascent_Stage_1 208

7.5.3 Point_Mass_ascent_stage_2 209

7.5.4 Body2Inertial.m 211

7.5.5 Inertial2Body.m 211

7.5.6 Atmo.m 211

7.6 Appendix F Dynamic Pressure Contours for Full Rocket Body 213

7.7 Appendix G Dynamic Pressure Contorts for Each Fin Case 219

7.8 Appendix H Lift and Drag vs Velocity for Each Fin Case 221

7.9 Appendix I Spin vs Time for Each Fin Case 225

7.10 Appendix J Air Brake Moment and Drag Force Fluent Graphs 229

List of Figures

Figure 1 SOLIDWORKS Model of the Rocket 4

Figure 2 SOLIDWORKS Model of the Rocket, X-Z Plane 4

Figure 3 Rocket Configuration 5

Figure 4 Mission Profile 5

Figure 5 Air During Laminar Flow. © (n.d.) Georgia Tech, Ramblin’ Rocketry Club 19

Figure 6 HAACK Nose Cone Example © 1996 Gary A. Crowell Sr. 20

Figure 7 Nose Cones and Their Calculated Drag Force using a Wind Tunnel. © 2013 Apogee Components, Inc 20

Figure 8 Comparison of Drag Characteristics for Various Nose Cone Shapes © 1996 Gary A. Crowell Sr. 21

Figure 9 Fin Tang. © NASA High Powered Video Series 2015 22

Figure 10 Two Types of Fin Slots. © NASA High Powered Video Series 2015 24

Figure 11 One Component Fin Can [10] © Jim Jarvis 2009 25

Figure 12 Avionics Bay Using Threaded Rods. © WPI University Student Launch Initiative 2018-2019 27

Figure 13 Avionics Bay Using Locking Mechanism. © WPI University Student Launch Initiative 2019-2020 27

Figure 14 Clamshell Payload Bay. © WPI Battle of the Rockets 2017-2018 28

Figure 15 Flowering Payload Fairing. © WPI University Student Launch Initiative 2018-2019 29

Figure 16 Blade Lift Free Body Diagram. © (n.d.) National Association of Rocketry 30

Figure 17 Blade Rotor Disk © Federal Aviation Administration, 2018 31

Figure 18 Blade Force Vector © Federal Aviation Administration, 2018 32

Figure 19 Illustration of Parachute Recovery © WPI Rocket MQP 2019-2020 34

Figure 20 Drag Recovery © Apogee Rockets, 2009 36

Figure 21 Streamer Recovery © Apogee Rockets, 2009 36

Figure 22 Comparison of Arc Lighter and E-Match Fundamentals 39

Figure 23 Blue Origin's New Sheppard Reusable Rocket Booster. © Blue Origin 2019 41

Figure 24 Movable Fins, Gimbaled Thrust, Vernier Rockets, and Thrust Vanes. © NASA 2014 47

Figure 25 Airframes of the Rocket 49

Figure 26 Stage 1 Fin, NACA 4414.. 51

Figure 27 Stage 2 Fin, NACA 0414.. 51

Figure 28 Raven 3.0 Altimeter (Featherweight Altimeters) ©, FA 2009 53

Figure 29 RD KL25-AGMP01 data collection board (Freescale Semiconductor) ©2015, NXP. 55

Figure 30 Autorotation CAD 57

Figure 31 First Stage Separation System Phases 64

Figure 32 Air Brake Bay Separated and Joined with Air Brakes Deployed................................ 65

Figure 33 Arc Ignitor Configuration Design Options.. 66

Figure 34 Body Fixed-Frame Definition of a Model Rocket, [54] © Springer, 2004 71

Figure 35 Relationship Between Body-Fixed Axis, Earth-Fixed Axis, and Attitude Angles [54] © Springer, 2004.. 71

Figure 36 State Space ODE Solver... 77

Figure 37 Pressure-Based Segregated Algorithm vs Coupled Algorithm © Ansys, 2009 79

Figure 38 Airframe in Fluent 80

Figure 39 Fin Design in XFLR5 83

Figure 40 Isometric View of 2414 Fin 84

Figure 41 2414 Fin in Design Modeler 84

Figure 42 Force to Spin Rate Flowchart 85

Figure 43 Meshed Motor Tube 1 88

Figure 44 Boundary Conditions for Motor Tube 1 89

Figure 45 Von Mises Stress on Motor Tube 1 90

Figure 46 Meshed Motor Tube 2 91

Figure 47 Boundary Conditions for Motor Tube 2 92

Figure 48 Von Mises Stress on Motor Tube 2 92

Figure 49 Small Nylon Shear Pins © 2019 Apogee Components, Inc. All Rights 93

Figure 50 Large Nylon Shear Pins © 2019 Apogee Components, Inc. All Rights 94

Figure 51 Boundary Conditions Astronaut Capsule 95

Figure 52 Boundary Conditions Upper Airframe 95

Figure 53 Von Mises Stress on Astronaut Capsule w/ Small Nylon Shear Pins 96

Figure 54 Von Mises Stress on Astronaut Capsule w/ Large Nylon Shear Pins 96

Figure 55 Von Mises Stress on Upper Airframe w/ Small Nylon Shear Pins 97

Figure 56 Von Mises Stress on Upper Airframe w/ Large Nylon Shear Pins 97

Figure 57 Boundary Conditions at Max. Recommended Pressure for Black Powder Charge 98

Figure 58 Max. Von Mises Stress at Max. Recommended Pressure for Black Powder Charge .. 98

Figure 59 Autorotation MATLAB Model Map 101

Figure 60 Flight Events Modeled Across Time Span, t 104

Figure 61 Acceleration of Rocket Over Time During Descent 110

Figure 62 Angular Speed of Blades Over Time 110

Figure 63 Rocket Height Over Time During Descent 111

Figure 64 Speed of Rocket Over Time During Descent 112

Figure 65 Ansys Fluent Simulation Diagram 113

Figure 66 Velocity Along the Blade 114

Figure 67 Static Pressure on the Top of the Blade 115

Figure 68 Static Pressure on the Bottom of the Blade 116

Figure 69 Lift and Drag vs. Angle of Attack 117

Figure 70 Von Mises Stress Distribution NACA 4414 Fin, Right View 118

Figure 71 Von Mises Stress Distribution NACA 4424 Fin, Left View 119

Figure 72 Von Mises Stress Distribution NACA 0414, Right View 119

Figure 73 Von Mises Stress Distribution NACA 0414, Left View 120

Figure 74 Schematic of the Aerotech K1000T-P Rocket Motors © [67], (n.d.) 124

Figure 75 Temperature Distribution Domains 130

Figure 76 Boundary Conditions in COMSOL Model 132

Figure 77 3D Velocity Gradient in chamber 137

Figure 78 Streamlines and 2D Velocity Gradient 137

Figure 79 Temperature Distribution in Chamber 139

Figure 80 Pressure Distribution in Chamber 139

Figure 81 Density Gradient with Isothermal Lines 140

Figure 82 2D Temperature Distribution inside Aluminum Casing 140

Figure 83 2D Temperature Distribution inside Unburnt Propellant (Left) and Aluminum Casing (Right) 141

Figure 84 Air Brake Force Diagrams Open (left) Closed (right) 143

Figure 85 Air Brake Variants 144

Figure 86 50° Air Brake Inside Ansys Fluent Simulation 144

Figure 87 Extension Spring Selected for Air Brakes, © McMaster-Carr Extension Spring 3630N389, 2020 146

Figure 88 E-Match 3D Scale Model 149

Figure 89 E-Match Boundary Conditions 150

Figure 90 Zoomed in E-Match Mesh 152

Figure 91 Temperature of Pyrogen at Multiple Points on its Surface 153

Figure 92 Temperature of E-Match Components 153

Figure 93 Temperature of Pyrogen in Proximity to Bridge Wire 154

Figure 94 Current in Bridge Wire vs. Time 155

Figure 95 Temperature Distribution of Pyrogen Cross Section 155

Figure 96 The Three Distinct Stages of the Code 156

Figure 97 X-Direction Displacement During Flight 161

Figure 98 Y-Direction displacement during flight 161

Figure 99 Altitude during flight 162

Figure 100 Elevated Perspective of the Flight 163

Figure 101 Perspective showing purely the motion in the X direction and altitude 164

Figure 102 Elevated perspective of the flight from alternate perspective 164

Figure 103 Vertical Velocity component until apogee 165

Figure 104 Moment about Vertical Axis vs. Angle of attack 167

Figure 105 Full Rocket Drag vs Angle of Attack 168

Figure 106 Dynamic Pressure at 100 m/s and 0 degrees 169

Figure 107 Dynamic Pressure at 100 m/s at 25 degrees 170

Figure 108: Lift and Drag vs Velocity for 2414 Fin 172

Figure 109: Spin Rate versus Time, NACA 4414 173

Figure 110: Dynamic Pressure on “Top” Surface of NACA 0414 174

Figure 111: Dynamic Pressure on “Top” Surface of NACA 4414 174

List of Tables

Table 1 Airframe and Recovery System (ARS) Subteam Analysis Tasks 9

Table 2 Propulsion, Thermal and Separation Systems (PTSS) Subteam Analysis Tasks 11

Table 3 Flight Dynamic Analysis (FDA) Subteam Analysis Tasks 14

Table 4 Analysis Tasks Section Table 16

Table 5 Properties of Standard Fiberglass and Carbon Fiber 23

Table 6 Break-down of State Space Control Variables 45

Table 7 Nomenclature and Units for MATLAB Model Map 101

Table 8 Nomenclature and Units for Flight Events Across Time Span, t 104

Table 9 Fluent Diagram Key 113

Table 10 Results of First Stage Cantera Analysis 125

Table 11 Comparison of Model and Manufacturer First Stage Motor Performance Data 125

Table 12 Results of Second Stage Cantera Analysis 127

Table 13 Comparison of Model and Manufacturer Second Stage Motor Performance Data 127

Table 14 Material Properties of Aluminum used in COMSOL Analysis 130

Table 15 Properties of Materials in the Thermal Modeling 131

Table 16 Air Brake Ansys Fluent Results 145

Table 17 Analytical Spring Constant "k" Variables Provided by McMaster-Carr 147

Table 18 COMSOL E-Match Physical Parameters 148

Table 19 Code Required for This Task 156

Table 20 Fin Lift and Drag Results 171

Table of Authorship

Section #	Section Title	Authorship
1	Introduction	Krystina Waters
1.1	Project Goals	Sophie Balkind
1.2	Project Design Requirements, Constraints, and Other Considerations	All
1.3	Safety Goals	All
1.4	Tasks and Timeframe	All
2	Literature Review	Krystina Waters
2.1	Airframe and Recovery System	Krystina Waters
2.1.1	Airframe	Sophie Balkind
2.1.2	Nose Cone	Krystina Waters
2.1.3	Fins	Sophie Balkind
2.1.3.1	Fin Design	Brendan Marion, Sophie Balkind
2.1.3.2	Fin Can	Sophie Balkind
2.1.4	Avionics Bay	Krystina Waters
2.1.5	Payload Bay	Krystina Waters
2.1.6	Recovery System	Matthew Barret
2.1.6.1	Autorotation Recovery	Matthew Barret, Krystina Waters

2.1.6.2	Parachute Recovery	Matthew Barret, Krystina Waters
2.1.7	Motor Retention	Matthew Barret
2.2	Propulsion, Thermal, and Separation Systems (PTSS)	Adam Sochacki
2.2.1	Arc Lighters	Adam Sochacki
2.2.2	Stage Separation and Air Brakes	David Robie
2.2.3	Model Rocket Motors	Matthew Karns
2.3	Flight Dynamic Analysis	Mark Payne
2.3.1	Models for Attitude and Trajectory	Mark Payne
2.3.2	Stabilization and Control Systems	Brendan Marion
3	Methodology	Sophie Balkind
3.1	Airframe and Recovery System	Sophie Balkind
3.1.1	Airframe	Sophie Balkind
3.1.2	Fins and Fin Can	Sophie Balkind Brendan Marion
3.1.3	Avionics Bay	Mark Payne
3.1.4	Autorotation System	Matthew Barrett
3.1.5	Payload Bay	Krystina Waters
3.1.6	Nose Cone	Krystina Waters
3.2	Propulsion, Thermal, and Separation Systems	David Robie
3.2.1	Motor Selection	Matthew Karns
3.2.2	Motor Tube Design	Matthew Karns

3.2.3	First Stage Airbrake Separation System	David Robie
3.2.4	Arc Ignitor System	Adam Sochacki
3.3	Flight Dynamic Analysis	Mark Payne
3.3.1	FDA Analysis Task 1	Mark Payne
3.3.2	FDA Analysis Task 2	Brendan Marion Mark Payne
3.3.3	FDA Analysis Task 3	Brendan Marion
4	Analysis	Krystina Waters
4.1	Airframe and Recovery System	Krystina Waters
4.1.1	ARS Analysis Task 1: Airframe Stress Distribution	Krystina Waters
4.1.2	ARS Analysis Task 2: Autorotation System Model	Krystina Waters Sophie Balkind
4.1.3	ARS Analysis Task 3: Autorotation Blade Aerodynamic Loads	Matthew Barrett
4.1.4	ARS Analysis Task 4: Stabilizing Fin Stress Distribution	Sophie Balkind
4.2	Propulsion, Thermal, and Separation Systems	Karns, Robie and Sochacki
4.2.1	PTSS Analysis Task 1: Motor Performance Model	Matthew Karns
4.2.2	PTSS Analysis Task 2: Temperature Distribution	Matthew Karns
4.2.3	PTSS Analysis Task 3: Mechanical Separation System Model	David Robie
4.2.4	PTSS Analysis Task 4: Electrical Match Model	Adam Sochacki

4.3	Flight Dynamic Analysis	Mark Payne Brendan Marion
4.3.1	FDA Analysis Task 1: Vehicle Dynamics and Performance Model	Mark Payne
4.3.2	FDA Analysis Task 2: Vehicle Aerodynamic Loads – Simulation	Brendan Marion
4.3.3	FDA Analysis Task 3: Stabilization Fin Aerodynamic Loads	Brendan Marion
5	Summary	ALL
5.1	Airframe and Recovery System: Summary and Broader Impacts	ARS Subteam
5.2	Propulsion Thermal Separation Systems: Summary and Broader Impacts	PTSS Subteam
5.3	Flight Dynamics Analysis: Summary and Broader Impacts	FDA Subteam
5.4	Overall Project Broader Impacts	Krystina Waters

Nomenclature

ARS: Airframe and Recovery Systems

PTSS: Propulsion Thermal and Separation Systems

FDA: Flight Dynamic Analysis

FO: Flight Operations

TBD: To Be Determined

MQP: Major Qualifying Project

NAR: National Association of Rocketry

COTS: Commercial off the Shelf

1 Introduction

Model rocketry became a hobby in the late 1950s at the peak of the space race [1]. The hobby has continued to evolve, becoming the basis for a variety of advanced rocketry competitions, educational applications, and personal projects worldwide. The NAR defines two main categories of rocketry, these are “Class 1 Model Rocketry” and “Class 2 High Power Rocketry”. For this Major Qualifying Project (MQP), the team has been tasked with designing, testing, and analyzing the equivalent of a Class 2 High Power Rocket.

High Powered Rocketry is a subcategory within model rocketry with a focus on advanced propulsion power, size, weight, and complexity of the model [2]. The National Association of Rocketry (NAR) has a set of regulations which define high power rocketry. A rocket is considered High Powered if it uses a motor with more than 80 N average thrust, exceeds 125 g of propellant, weighs more than 1,500 g, uses a motor that is either hybrid or emits sparks, includes ductile metal as part of its airframe components, or uses a motor with more than 160 N-s of total impulse or multiple motors that exceed 320 N-s total impulse [2].

To accommodate all areas and goals of this MQP, the team was divided into three subteams: Airframe and Recovery System (ARS), Flight Dynamics Analysis (FDA), and Propulsion Thermal and Separation Systems (PTSS). The ARS subteam was responsible for the design and integration of the vehicle’s innovative autorotation recovery system and carbon fiber manufacturing technique of the airframe and fins. The subteam was additionally responsible for the integration of the entire rocket and all its additional systems as determined by the FDA and PTSS subteams. The FDA subteam was responsible for the analysis of the rocket’s overall performance and the aerodynamic loads on the vehicle during its flight. Furthermore, the subteam was also tasked with the selection, integration, and simulation of sensors as well as providing simulations of the rocket’s flight dynamics. The PTSS subteam was responsible for the design and fabrication of an innovative stage separation system between the vehicles first and second stage, as well as the selection, modeling, and test of commercially available high-powered rocket motors to be used for this project. Additionally, the PTSS team was responsible for conducting analysis on the thermal loads generated within the vehicle during flight by its motors.

With all these subsystems and tasks in mind, the team formulated a series of goals and objectives to guide the formulation of specific tasks over the course of the project terms. Figures 1-4 show the computer aided design (CAD) model, a cross-sectional view, the rocket configuration, and mission profile, respectively.

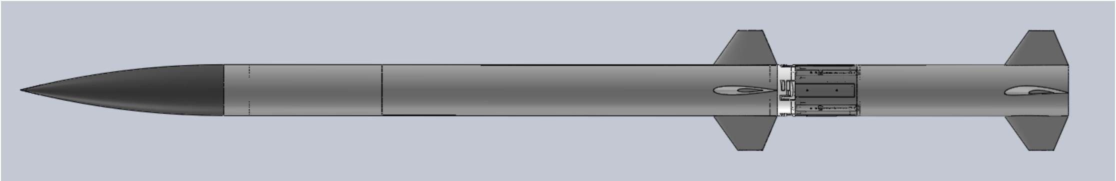


Figure 1 SOLIDWORKS Model of the Rocket

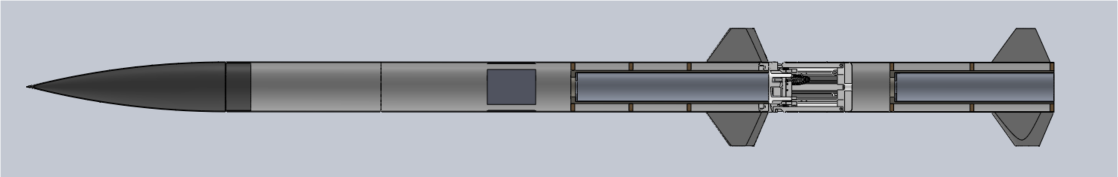


Figure 2 SOLIDWORKS Model of the Rocket, X-Z Plane

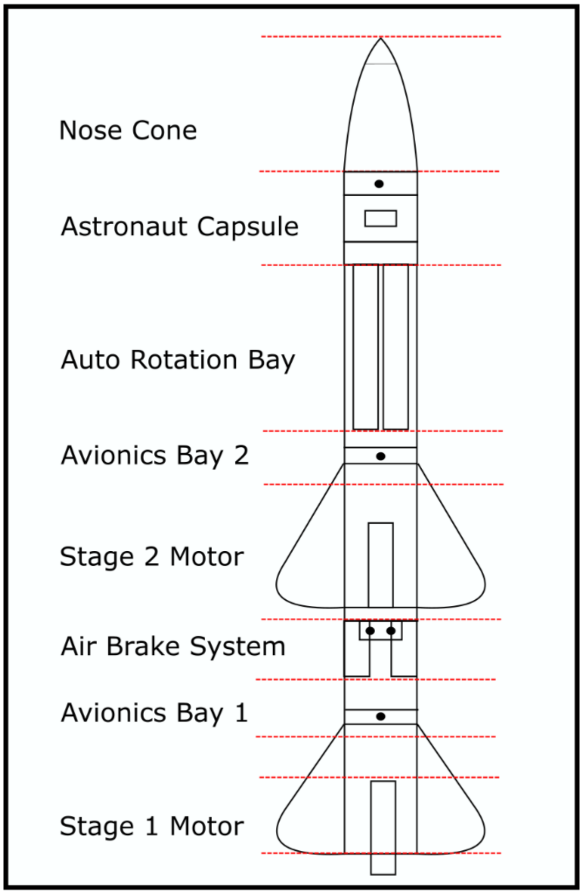


Figure 3 Rocket Configuration

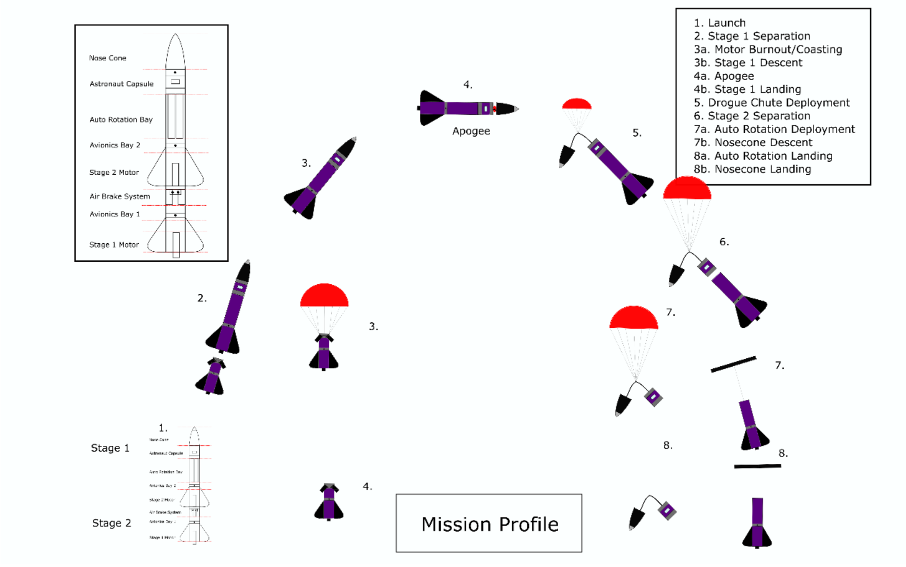


Figure 4 Mission Profile

1.1 Project Goals

The requirements for the entire MQP were divided between four subteams as previously described. In addition to the three subteams already discussed, there was a Flight Operations (FO) subteam whose responsibility was to be shared among all team members if a flight opportunity became available.

The overall project goals were as follows:

1. Design, build, and (if permitted) fly a reusable rocket to an altitude of 1500 ft.
2. Work as a team to design, build, and test a moderately complex aerospace system in which overall vehicle performance is critically tied with the mass and performance of the individual components and assemblies.
3. Learn and apply software tools such as:
   - MATLAB, Ansys – Static Structural Analysis, Fluent, and Dynamic Analysis, Cantera, COMSOL

In addition to these overall goals, each subteam had its own set of objectives:

- Airframe and Recovery System (ARS)
  1. Design and fabricate the airframe structure.
  2. Design, fabricate, and test both a baseline and innovative recovery system.
  3. Lead integration of recovery, payload, staging, avionics, propulsion subsystems.
- Propulsion, Thermal, and Separation Systems (PTSS)
  1. Propulsion - compare commercially available motors with respect to size (impulse), cost, and thrust, to choose the best combination of motors to achieve project goals.

2. Thermal and Ignition - research, design, and prototype an innovative arc lighter ignition system for the first stage of the rocket.
3. Stage Separation System - design, fabricate, and test an innovative stage separation system.
4. Select, model, and test (commercially available) motors, mounts, and ignition system for two-stage rockets.
5. Performance an analysis of thermal loads from the motor(s) during flight.
6. Support integration of propulsion and staging systems.

- Flight Dynamic and Analysis (FDA)
    1. Perform analysis of aerodynamic loads on vehicle in flight.
        - Lead selection and integration of avionics, including accelerometers, gyros, and altimeters.
    2. Perform simulation of rocket flight dynamics (center of mass trajectory and attitude analysis).
    3. Create numerical model to perform simulation of rocket performance (altitude, range, etc.) in support of design activities and flight planning.
    4. Support integration of other payload, recovery, staging, avionics, and propulsion subsystems.
    5. Analyze the spin generating behavior of cambered and angled fins on the rocket.
    6. Evaluate aerodynamic forces and moments generated by a single fin on the airframe with stabilizing fins
- Flight Operations (FO)

1. Select and prepare a payload.
2. Identify candidate NAR and Tripoli launch events and communicate with organizers if needed.
3. Support flight tests(s) including preflight planning, launch operations, and recovery.
4. Maintain MQP Lab

1.2 Project Design Requirements, Constraints, and Other Considerations

The design requirements for this MQP were identified and agreed upon by the whole team and consist of the following:

- Use a camera as part of a simulated payload to record rocket's flight.
- Return an "astronaut" to the ground at a safe speed such that there is no damage to the craft or components inside.
- Use parachutes to recover the first stage and nosecone/astronaut capsule.
- Use an autorotation System to recover Stage 2
- Use airbrake based separation system after Stage 1 burnout
- Ensure airbrake based separation system induces minimal forces on second stage
- Ensure stage separation requires no expendable parts and minimal effort to reset
- Use an arc lighter ignition system for the first stage rocket motor
- Ensure chosen motors fulfill thrust delivery needs and are compatible with rocket structures and housing
- Simulate the aerodynamic forces on the rocket and estimate the dynamics and trajectory of the rocket
- Use on-board sensors to collect flight data

The design constraints for this MQP were as follows:

- The Rocket must have a sufficient thrust to weight ratio in order to ensure a reliable stability
- All components must be easily accessible for disassembly
- Components must be contained within the airframe upon liftoff
- The rocket must be launched by a person holding the same certification level as its motor
- The Rocket must be built with lightweight, durable, and reliable materials

1.3 Safety Goals

- Ensure parachutes are properly folded and protected so they successfully deploy
- Ensure all portions of the rocket are correctly assembled to prevent unexpected spontaneous disassembly
- Ensure safe and stable flight through simulations of the rocket's performance
- Ensure a safe landing by simulating the rocket's possible landing area

1.4 Tasks and Timeframe

A set of specific analysis tasks were identified for each subteam to inform and support the team's effort to meet the project goals listed in Section 1.1. The analysis tasks were broken up by subteam and can be seen in Table 1, Table 2, and Table 3. The locations of descriptions and results for each analysis task can be found in Table 4.

Table 1 Airframe and Recovery System (ARS) Subteam Analysis Tasks

ARS Analysis Task 1: Airframe Stress Distribution
Problem Statement:
Identify critical locations of high stress throughout the airframe and internal structure during peak acceleration loads.
Solution Methodology:

• Tool(s): Ansys Mechanical Workbench / Dynamic Analysis, SOLIDWORKS Stress Analysis • 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 power or spring actuated)
Analysis Products:
• Map of stress distribution • Identification of critical locations/joints
Use of Results:
• Stress data will be used to determine weak points within airframe design during 1) maximum acceleration and 2) separation events • Determine method(s) to improve structural integrity of the airframe and internal structures e.g. electronics bay and motor bay
ARS Analysis Task 2: Autorotation System Model
Problem Statement:
• Create a simplified autorotation blade model to estimate lift forces to decrease the decent velocity
Solution Methodology:
• Tool(s): MATLAB, XFLR5, and SOLIDWORKS • Required Inputs: vehicle mass and airframe geometry, spanwise blade geometry (assumed for design or from commercially available blades) • Formulate model for lift as a function of blade design, vehicle weight, and descent velocity
Analysis Products:
• Vehicle descent velocity as a function of time after deployment and blade design
Use of Results:
• Model results will be applied to original design as well as commercially available blades to estimate velocity at time of impact. Sensitivity to deployment altitude will also be investigated
ARS Analysis Task 3: Autorotation Blade Aerodynamic Loads
Problem Statement:

• Evaluate the aerodynamic forces and moments on a single blade
Solution Methodology:
• Tool: Ansys Fluent • Required Inputs: initial and boundary conditions (flight velocity, etc.), fluid properties, mesh characteristics, blade airfoil geometry, number and placement of fins on the airframe • Evaluate forces and moments produced by single blade
Analysis Products:
• Net forces and moments on a blade.
Use of Results:
• For each case, the forces and moments generated over a range of flight velocities and angles of attack will be curve-fit to produce a function that can be used as an input in the autorotation system model
ARS Analysis Task 4: Stabilizing Fin Stress Distribution
Problem Statement:
• Identify critical locations of high stress throughout the fins during peak loads.
Solution Methodology:
• Tool(s): Ansys Mechanical Workbench / Dynamic Analysis, SOLIDWORKS Stress Analysis • Required Inputs: Fin solid model including information on materials, flight speed (to determine aerodynamic loads on the fin) • Estimate stresses on the fin as a function of flight velocity and attitude
Analysis Products:
• Map of stress distribution
Use of Results:
• Determine method(s) to improve structural integrity of fin and fin-airframe attachment method

Table 2 Propulsion, Thermal and Separation Systems (PTSS) Subteam Analysis Tasks

PTSS Analysis Task 1: Motor Performance Model

Problem Statement:

Create a simplified model for the selected motors, that can be used to estimate performance (thrust and Isp) that can be compared with published data

Solution Methodology:

- Tools: Cantera, MATLAB
- Major Assumptions: chemical equilibrium in chamber, frozen flow, steady state, isentropic flow in nozzle
- Required Inputs: Propellant composition and properties, chamber and nozzle geometry, ambient conditions (p, T)
- Formulate model that couples equilibrium chemistry with flow through nozzle

Analysis Products:

- Composition of combustion products, chamber pressure and temperature, mass flow rate, thrust, specific impulse

Use of Results:

- Compare predicted performance with manufacturer data
- Estimate heat flux for thermal analysis

PTSS Analysis Task 2: Temperature Distribution

Problem Statement:

- Estimate the temperature distribution through the propellant grain, motor casing, and rocket body subject to heat flux from the motor

Solution Methodology:

- Tool: COMSOL
- Required Inputs: Material and property data for motor, casing, and rocket body. External, flight boundary conditions (velocity, ambient pressure, temperature, air properties)

Analysis Products:

- Temperature and heat flux distribution through motor and airframe structure

Use of Results:

- 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.)

PTSS Analysis Task 3: Mechanical Separation System Model

Problem Statement:

- Create a simplified model for the spring-based stage separation system that can be used to determine an optimal airbrake/spring configuration

Solution Methodology:

- Tools: MATLAB, Excel, Ansys, force balance (scale), Wind Tunnel
- Formulate simple spring-hinge model using Hooke's Law, calculate number and size (length diameter) of springs, spring constant, compression distance. Use this model to determine multiple spring and air brake design combinations.
- Experimentally measure spring force and drag forces as a function of airspeed for each spring/air brake combination considered. Use this analysis to recommend an optimal combination pair.
- Required Inputs: Aerodynamic force on air brake, spring constant and geometry (length, coil and wire diameters), variable airspeeds.

Analysis Products:

- Aerodynamic forces induced on air brakes, spring forces required to counteract aerodynamic forces.

Use of Results:

- Evaluate design options to determine an optimal spring/air brake design pair

PTSS Analysis Task 4: Electrical Match Model

Problem Statement:

- Create a simplified model for the electrical match

Solution Methodology:

- Tools: MATLAB
- Required Inputs: physical dimensions of wires and pyrogen, voltage and current characteristics of electrical components

• Create simple mode of thermal energy delivered as a function of time for a specific set of electrical circuit parameters
Analysis Products:
• Comparison of thermal energy delivered as a function of time with a comparison to baseline (pyrogen) ignitors
Use of Results:
• Assess feasibility of concept and compare performance (energy delivered vs time, reliability, cost, complexity) with baseline ignitors

Table 3 Flight Dynamic Analysis (FDA) Subteam Analysis Tasks

FDA Analysis Task 1: Vehicle Dynamics and Performance Model
Problem Statement:
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)
Solution Methodology:
• Tool: MATLAB • 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 • Formulate model consisting of *two, coupled systems* of nonlinear ODEs, one for attitude dynamics (Euler solver) and one for the equations of motion (Newton’s 2nd law) describing the vehicle trajectory. Euler equations are solved at each time step as the equations of motion are solved for the trajectory
Analysis Products:
• Simulation of rocket trajectories, capturing statistically randomized variation in wind speed and direction • 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

Use of Results:

- Landing probability distribution plot (safety plot)
- Data that can be used to compare stability and performance of innovative fin design with baseline
- Provide estimates of perturbation on vehicle attitude (angles and rates) when subject to transient wind disturbances.
- Estimate upper limit on wind disturbance to maintain stable flight
- Evaluation of effectiveness of innovative fin design, i.e. compare vehicle’s ability to maintain stable flight with baseline design

FDA Analysis Task 2: Vehicle Aerodynamic Loads – Simulation

Problem Statement:

- Estimate the aerodynamic forces and moments on the vehicle as a function of velocity and vehicle attitude

Solution Methodology:

- Tools: SimScale, Ansys Fluent
- 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

Analysis Products:

- Pressure contours, plots of forces and moments acting on vehicle of given design and flight state (attitude and velocity)
- Estimate of forces and moments acting on vehicle of given design and flight state (attitude and velocity) when subject to wind disturbances

Use of Results:

- Provide load estimates to be used in structural stress analysis
- Provide forces and moment data that can be used in “table-lookup” for simulation of vehicle trajectory and attitude

• Provide estimates of perturbation on vehicle attitude (angles and rates) when subject to transient wind disturbances.
FDA Analysis Task 3: Stabilization Fin Aerodynamic Loads
Problem Statement:
• Evaluate the aerodynamic forces and moments on a single fin and on the airframe with a set of stabilizing fins
Solution Methodology:
• Tool: Ansys Fluent • Required Inputs: rocket geometry and inertia properties, initial and boundary conditions (flight velocity, etc.), fluid properties, mesh characteristics, fin airfoil geometry, number and placement of fins on the airframe • Evaluate forces and moments produced by two fin options to provide spin stabilization during flight. The two fin options are: (1) symmetric airfoil with non-zero AOA, (2) cambered airfoil
Analysis Products:
• Net forces and moments on airframe for each "case" considered. A case consists of a combination of fin type, placement, number, and flight conditions. • Estimate of loads on single fin
Use of Results:
• For each case, the moment generated over a range of flight velocities will be curve-fit to produce a function that can be used as an input in the vehicle dynamics model • Results will be part of a trade study (effectiveness vs. complexity) to down select option for innovative rocket • Loads on single fin will be used as input in stabilizing fin stress analysis.

Table 4 Analysis Tasks Section Table

Airframe and Recovery System (ARS) Subteam Analysis Tasks	**Section**
ARS Analysis Task 1: Airframe Stress Distribution	4.1.1

ARS Analysis Task 2: Autorotation System Model	4.1.2
ARS Analysis Task 3: Autorotation Blade Aerodynamic Loads	4.1.3
ARS Analysis Task 4: Stabilizing Fin Stress Distribution	4.1.4
Propulsion, Thermal, and Separation (PTSS) Subteam Analysis Tasks	**Section**
PTSS Analysis Task 1: Motor Performance Model	4.2.1
PTSS Analysis Task 2: Temperature Distribution	4.2.2
PTSS Analysis Task 3: Mechanical Separation System Model	4.2.3
PTSS Analysis Task 4: Arc-Ignitor Model	4.2.4
Flight Dynamic Analysis (FDA) Subteam Analysis Tasks	**Section**
FDA Analysis Task 1: Vehicle Dynamics and Performance Model	4.3.1
FDA Analysis Task 2: Vehicle Aerodynamic Loads – Simulation	4.3.2
FDA Analysis Task 3: Stabilization Fin Aerodynamic Loads	4.3.3

2 Literature Review

To become more familiar with various design parameters, applications, and methods associated with high powered model rockets, the team conducted an in-depth literature review. This review provided the necessary background knowledge needed to meet the objectives for this MQP. The team researched topics within model rocketry ranging from the general construction of high-powered model rockets to specific topics relevant to this MQP such as spin stabilization, autorotation recovery systems, air brake separation systems, and arc lighters.

2.1 Airframe and Recovery System

This section provides a background on the main components and compartments that make up a high-powered model rocket. Research was conducted with a focus on mechanical structures including airframes, nose cones, fins, and recovery systems. The integration of each of these systems into the rocket was essential for providing the outer framework and structural integrity required for flight.

2.1.1 Airframe

The airframe is the main structural component of a rocket that supports the nose cone, fins, and many other features depending on the complexity of the vehicle. The airframe of a model rocket is usually composed of a number of sections connected via couplers, bulkheads, or a combination of both. The airframe needs to be rigid and strong enough to support the internal and external forces it is experiences to during flight. For example, the rocket might experience moments when it is not pointing directly straight up. When this occurs, the vehicle will experience a large force from the air it is pushing out of the way [3]. In order to reinforce the airframe, components, such as couplers and centering rings, can be used to add strength and increase reliability.

Common airframe materials are phenolic tubing, plastic tubing, fiberglass, and carbon fiber [3]. Blue Tube 2.0 is also frequently used for a high-powered model rocket of the size of the project. Typically, couplers are the same material as your airframe and bulkheads are made of plywood, composites, or aluminum.

2.1.2 Nose Cone

The nose cone is the most forward part of a rocket. Nose cones are used in all projectiles, e.g. rockets, missiles, etc. for the purpose of reducing aerodynamic forces, such as drag, on a vehicle. Drag is an aerodynamic force that opposes thrust and thus resists the upward motion of a rocket. The amount of drag imparted on the vehicle during its flight is directly related to the shape of the nose cone and speed of the rocket [4]. Therefore, nose cones come in various shapes and sizes with some being more suitable for high-speed applications and others more suitable for low-speed applications. This is because the ideal shape of the nose cone for a particular vehicle is related to the speed at which the vehicle is designed to fly [5]. The amount of drag on the rocket is minimized when the air traveling past the vehicle is smooth. This smooth flow of air, with well-defined streamlines, is referred to as laminar flow. In laminar flow, air begins to pass over the vehicle in smooth layers where each layer has little to no mixing as can be seen in Figure 5 [5], [6].

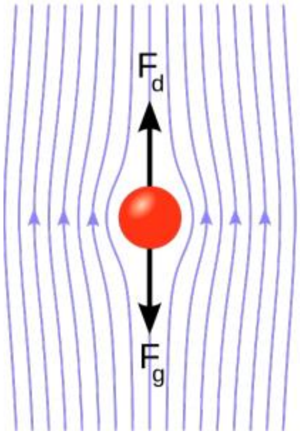


Figure 5 Air During Laminar Flow. © (n.d.) Georgia Tech, Ramblin' Rocketry Club

The most common types of nose cones include, parabolic, ogive, conical elliptical, and blunt/cylinder. Within these categories are sub-types of nose cones such as the LV-HAACK ogive and Von Karmen Ogive as can be seen in Figure 6. where the value of C is 1/3 for LV-HAACK and 0 for the Von Karmen. When C=0, it signifies the minimum drag for the given length and diameter, and when C=1/3, it indicates the minimum drag for a given length and volume [7]. Figure 7 shows the shape of various nose cone types [8].

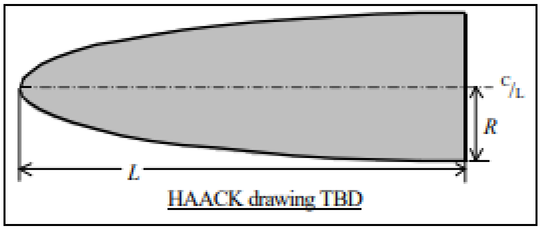


Figure 6 HAACK Nose Cone Example © 1996 Gary A. Crowell Sr.

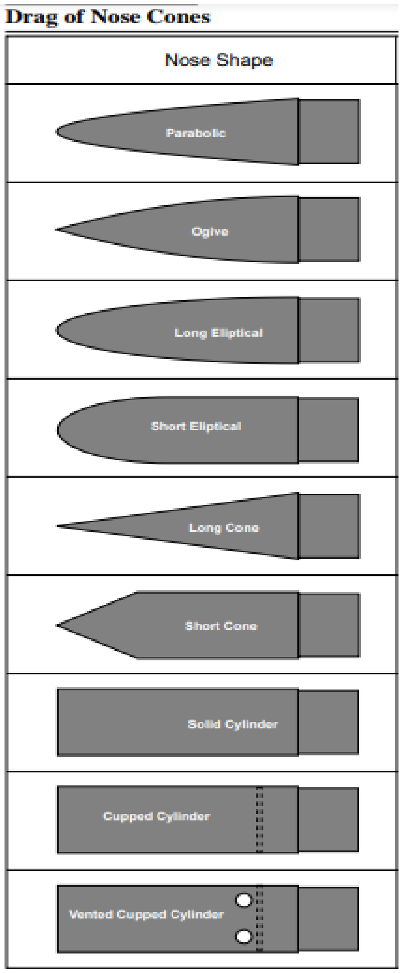
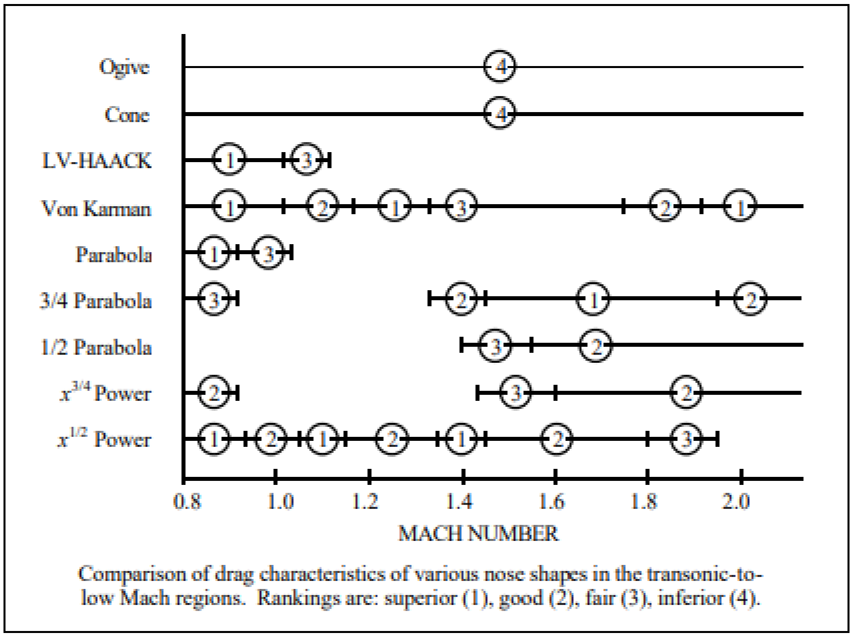


Figure 7 Nose Cones and Their Calculated Drag Force using a Wind Tunnel. © 2013 Apogee Components, Inc

The most ideal nose cone type can be determined based on the drag characteristics at a particular speed. Figure 8 demonstrates how desirable a specific nose cone shape is at a particular

Mach number using a ranking system of (1) for superior, (2) for good, (3) for fair, and (4) for inferior. Thus, based on this diagram, the most desirable nose cone in cases where the vehicle is traveling at a speed less than the speed of sound, or subsonic speed, is a more rounded nose cone shape such as parabolic [7].



Comparison of drag characteristics of various nose shapes in the transonic-to-low Mach regions. Rankings are: superior (1), good (2), fair (3), inferior (4).

Figure 8 Comparison of Drag Characteristics for Various Nose Cone Shapes © 1996 Gary A. Crowell Sr.

2.1.3 Fins

When designing the fins for a high-powered rocket, it is important to understand how fin design parameters will enhance the vehicle's flight, and how they can be attached to the rocket. To better understand these concepts, the team researched various fin designs and applications such as fin stabilization. These concepts are explored in depth in the following sections.

2.1.3.1 Fin Design

Fins of the rocket are typically fixed structures that extend from the airframe of the rocket, providing stability while the rocket is in flight. Fins help stabilize the rocket by keeping the rocket's center of pressure behind its center of mass. Fins come in a variety of shapes, including, trapezoidal, elliptical, delta, cropped delta, and tapered wings but "most fins are tapered or delta type due to ease of manufacturing" [4]. While it is common for fins to be attached through the wall of the rocket via a "tang", as seen in Figure 9, the team has researched another method in which the fins are manufactured out of carbon fiber and then are adhered to the airframe. This process is described in Section 2.1.3.2.

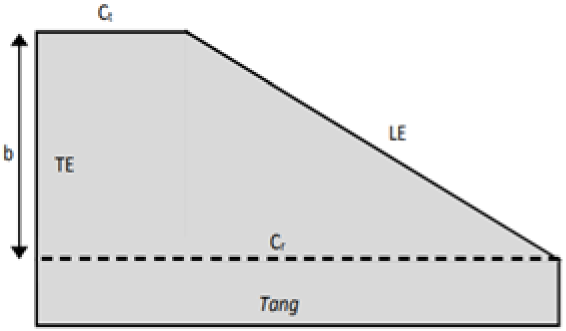


Figure 9 Fin Tang. © NASA High Powered Video Series 2015

The orientation and cross-sectional shape, or airfoil, of the fins also have effects on the rocket dynamics. Fins with a cambered airfoil, or a non-zero angle of attack, can produce a lift force perpendicular to the rocket's path. This force can induce a spin on the rocket, which can help stabilize it during flight. However, with a lift force comes a drag force, which hinders the rocket's velocity. Overall, the fin design plays an essential part in not only the rocket's stability but the aerodynamic forces the rocket experiences as well.

There are a variety of materials available for the manufacturing of fins for high-powered model rockets, including G-10 fiberglass, birch plywood, and carbon fiber [3]. Lighter materials can be used as a core to reduce weight of the fins while still adding structure and stronger materials can be used as an exterior “skin”. Typical core materials are foam, balsa wood, and G-10 Fiberglass and typical skin materials are fiberglass, carbon fiber, and thin hardwood [4]. With respect to fiberglass, typically E Glass is what is being referred to for model rocketry. Fiberglass is stronger than carbon fiber, meaning it has a higher ultimate tensile strength (UTS), but carbon fiber has a higher stiffness. The properties of standard fiberglass and carbon fiber are summarized by F1 engineer Kyle Forster and displayed in Table 5 [9]. The values show that the strength of standard carbon fiber and fiberglass are similar, close enough that it doesn’t make a difference for a model rocket. In contrast, the stiffness is significantly different. The strength-to-weight of carbon fiber is significantly higher so if weight is the primary concern then carbon fiber would be the better choice.

Table 5 Properties of Standard Fiberglass and Carbon Fiber

Property	**Fiberglass**	**Carbon Fiber**	**Units**
Ultimate Tensile Strength (UTS)	3500	3100	MPa
Stiffness	75	225	GPa
Strength-to-Weight	0.5	0.65	N/A

Depending on the material chosen to construct the fins, different construction methods are necessary. Different materials and methods are, flat plate, airfoiled, core materials oven, and room temperature. The flat plate fins are relatively easy to manufacture and can be done by making a “sandwich” of the core and skin using epoxy. The sandwich can then be trimmed to size.

2.1.3.2 Fin Can

As mentioned in Section 2.1.3.1, there are numerous methods to attach fins including “through- the-wall" and surface mounted [4]. For all “through-the-wall" designs, the airframe needs a slot cut in the airframe through it for the fins to attach. As shown in Figure 10, there are two types of fin slots, End-Slotted and Tang-Slotted [10]. Fins are either pushed up through the aft end of the airframe or perpendicularly through the airframe. End Slots are often used in conjunction with a fin can, while Tang Slots are most often used to permanently secure the fins to the motor tube.

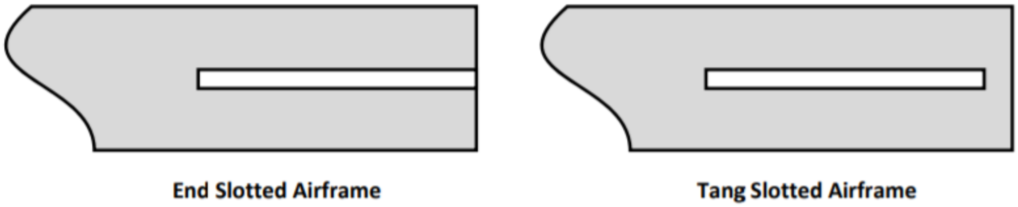


Figure 10 Two Types of Fin Slots. © NASA High Powered Video Series 2015

A fin can is a method of integration of the fins into the airframe, seen in Figure 11. A fin can may be designed to come apart with fasteners or designed to be one component. As an alternative to securing fins surface-to-surface with an adhesive, fin cans are less prone to shearing off or breaking at a mounting point.

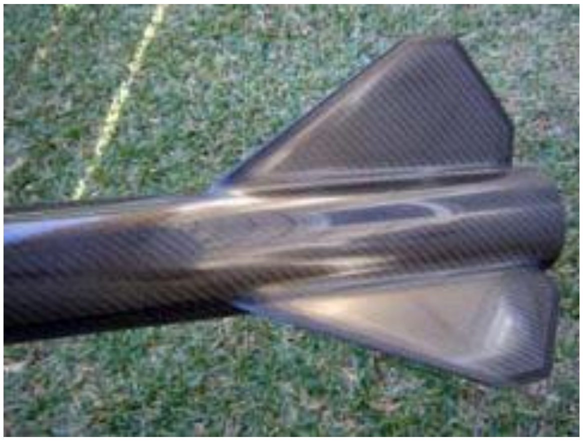


Figure 11 One Component Fin Can [10] © Jim Jarvis 2009

Removable designs typically use bolts to secure the fins via a fin can or other mounting mechanisms while fixed designs will typically use an adhesive such as epoxy. Surface mounted fins are created by using epoxy to secure the root of the fin to the airframe. Fillets can be used for additional reinforcement, as well as a lamination of “tip-to-tip" fiberglass or carbon fiber.

An excellent example of surface mounted fins can be seen in the “The Jarvis Illustrated Guide to Carbon Fiber Construction” which uses a carbon fiber airframe and carbon fiber, fiberglass core fins. This guide covers the process of manufacturing the fins, attaching the fins to the airframe, adding epoxy fillets, and finally the “tip-to-tip” carbon fiber lamination. The fins are made with “a G-10 core, and 5 layers of carbon … laminated to each side” [11]. Once the epoxy is cured, the fins can be sanded to add any additional shape. The fins are then attached to the airframe using a jig and epoxy. The jig is designed to keep the fins properly aligned as they are secured to the airframe. This could be done with fins one at a time or all at once. The fillets are created using a dowel (in his case), or any properly sized circular object, and epoxy with milled glass and Kevlar fillers. Once cured, the fillet ends are sanded to the proper shape. He describes a concave rounded leading edge.

Once the fins are correctly shaped and attached, a final “tip-to-tip” lamination is performed with two layers of carbon fiber. Once cured to the “leather” stage, the excess carbon from this layer can be trimmed. After fully cured, the fin can may be sanded, and a thin layer of epoxy can be added on the exposed edges of the fins as reinforcement.

2.1.4 Avionics Bay

The avionics bay is a compartment within the model rocket that houses and protects the selected flight electronics, or avionics, for a particular mission. Depending on the application, the vehicle may have multiple avionics bays with varying purposes and devices such as flight computers, accelerometers, batteries, wiring, gyroscopes, and other sensors.

In model rocketry, an avionics bay is typically designed using a coupler and bulkhead system. A coupler is a section of airframe that’s outer diameter is equal to the inner diameter of the outer airframe. The purpose of a coupler is to attach two pieces of airframe typically denoted as the upper and lower airframes. This is accomplished by fixing the coupler tube to one section of airframe and then attaching the other side via a separation system such as shear pins. A bulkhead is a piece of plywood typically between 0.25”-0.5” thick that acts as the cap for each end of the coupler. It can, however, also be made from strong 3D printed filament. A section of tubing equal in diameter to that of the outer airframe is wrapped around and epoxied to the center of the coupler. A fixed switch is installed onto this piece of tubing in order to toggle the electronics from outside the rocket. Figure 12 shows an example of an avionics bay designed by the WPI 2018-2019 University Student Launch Initiative team [12] using the methods described.

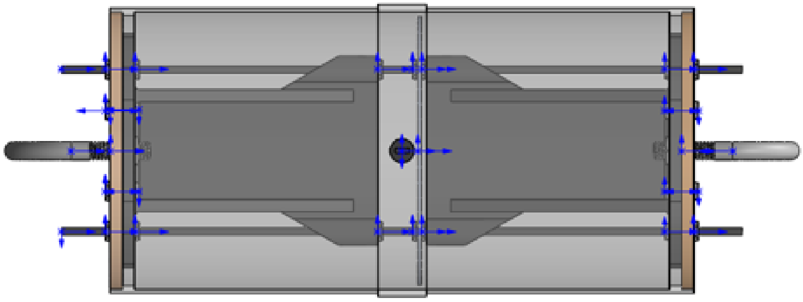


Figure 12 Avionics Bay Using Threaded Rods. © WPI University Student Launch Initiative 2018-2019

This common design for avionics bays is typically held together by threaded rods, washers, nuts, and bolts. However, this option does not always provide the best ease of use and thus more innovative designs may be desirable such as, a fully 3D printed system or a locking mechanism to easily twist the system in and out of the airframe. Figure 13 shows a design by the same team in 2019-2020 [13] that incorporates a locking mechanism. This locking mechanism allows the electronics bay to be more easily removable and eliminates the need for threaded rods.

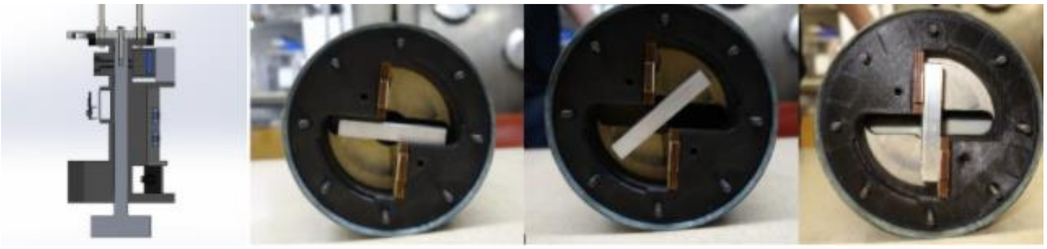


Figure 13 Avionics Bay Using Locking Mechanism. © WPI University Student Launch Initiative 2019-2020

2.1.5 Payload Bay

The payload bay is a compartment within the rocket that houses and protects the selected payload for a particular mission. In most situations this compartment is located within the upper airframe and is built into the tubing of the airframe. Payload bays are often custom to the application of the project or vehicle it is designed for and can be used to house cameras, rovers, UAVs, recovery components, sensors, etc. This can be a crucial component of the overall rocket especially for high-powered rocketry competitions where teams are often asked to design a specific payload and thus must design a payload bay that can house and deploy that payload. There are many different options for designing a payload bay and designs can range from simple to complex depending on the application. Figure 14 shows a clamshell design used by the WPI 2017-2018 Battle of the Rockets team [12] to house a rover and Figure 15 shows a flowering payload fairing used to house a UAV by WPI's 2018-2019 University Student Launch Initiative team [12].

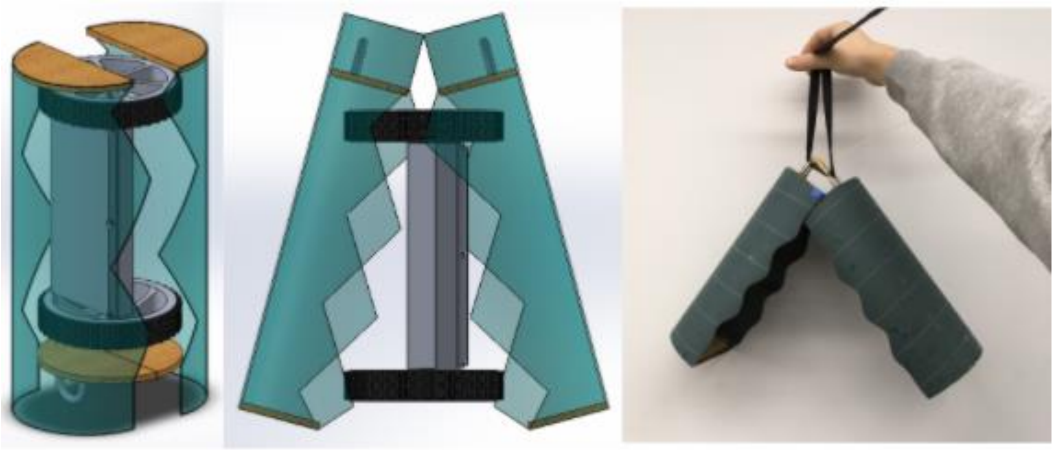


Figure 14 Clamshell Payload Bay. © WPI Battle of the Rockets 2017-2018

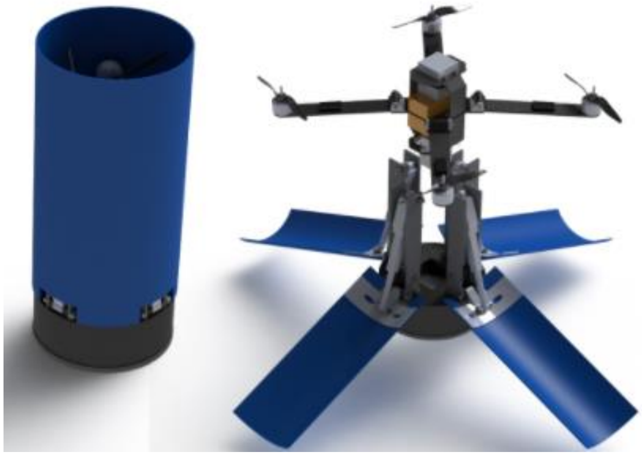


Figure 15 Flowering Payload Fairing. © WPI University Student Launch Initiative 2018-2019

2.1.6 Recovery System

Recovery systems are essential for making any rocket consistently reusable. Without them, the forces a rocket experiences during its impact with the ground would damage it beyond repair. Typically, recovery systems are designed such that the landing velocity of the craft is between 3.5 and 4.5 m/s [14, 15]. In the following sections we examine an innovative recovery system along with a more traditional option.

2.1.6.1 Autorotation Recovery

For a helicopter or rotary-winged aircraft, autorotation is the state in which the rotor system is being turned by air running through the blades as opposed to being driven by an engine or electric motor. In autorotation, the free-spinning rotor generates drag the slows the descent of the vehicle. For helicopters, performing an autorotation landing is done in emergency circumstances, however NASA has also proposed autorotation recovery systems to safely land crewed capsules and unmanned craft [3]. For a small rocket, autorotation can be used to safely control its descent to the ground. Due to the fact that the vehicle is falling when the recovery system is deployed, the blades will spin up to create resistance to slow the decent of the rocket enough to prevent damage. For the purpose of this MQP the ARS subteam decided to focus on an autorotation recovery system for the descent of Stage 2 of the rocket. In order to accomplish this, the subteam plans to build upon the work completed by last year's MQP team and the year prior.

The blades of an autorotation system must be made of material strong enough to survive deployment and withstand the forces exerted on them. During the rocket’s descent the angle of attack of the blades needs to be increased in order to create lift and drag [16]. The component of lift that is perpendicular to the long axis of the model then causes the blades to rotate. In turn, the component of lift parallel to the long axis of the model offsets the weight of the rocket thus slowing its descent as shown in Figure 16 [17] provides an example of a free body diagram of a blade undergoing lift. The lift on the blade can be calculated using Eq. 1:

$$L = 0.5\rho V^2 C_l S \tag{1}$$

In Eq. 1, L represents the force of lift generated on the blade in units of Newtons; the Greek symbol ρ represents the density of air measured in kg/m^3; V represents the true airspeed squared which is the square of the velocity of the vehicle relative to oncoming air in units of m/s; S represents the area of the rotor blade in m^2; and C_l represents the coefficient of lift at a specific angle of attack [16].

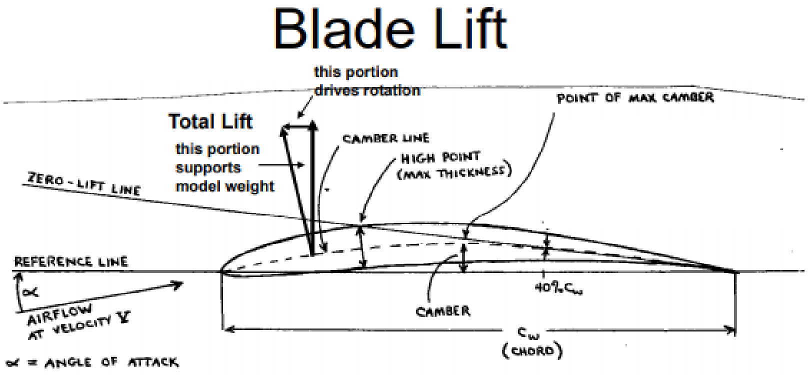


Figure 16 Blade Lift Free Body Diagram. © (n.d.) National Association of Rocketry

In order to create the maximum amount of drag possible, the autorotation system must achieve a fast rotation rate. This Is because the blade’s net velocity is directly affected by the rotation rate of the blade as can be seen in Eq. 2 [17].

$$V = \omega r \tag{2}$$

Where V represents the net velocity on the blades, ω represents the blade rotation rate, and r represents the radial distance of the spanwise coordinate measured from the root. The area swept by the rotor blades or rotor disk is divided into three main radially concentric regions: the driving region, driven region, and stalling region as can be seen in Figure 17 [16].

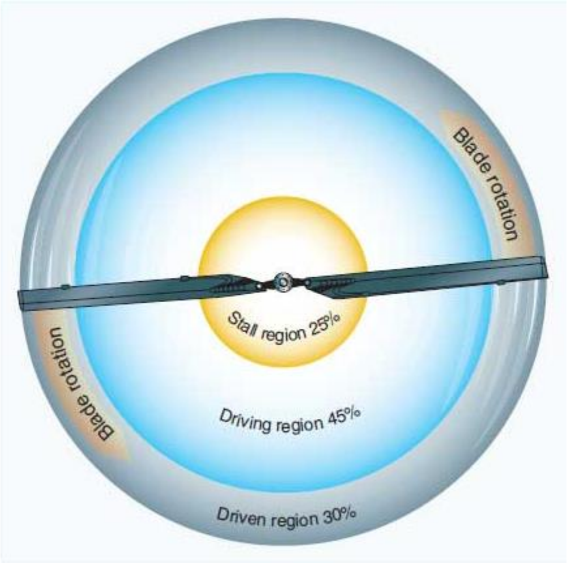


Figure 17 Blade Rotor Disk © Federal Aviation Administration, 2018

These three regions experience varying force vectors due to the relative wind velocity being slower near the stalling region and faster near the driven region. This results in a changing angle of attack as demonstrated in Figure 18. In the stall region of Figure 18, diagram E, the angle of attack is very high due to the blade going slower in this area. Due to the high angle of attack this results in a high drag force and stall. Furthermore, diagram C demonstrates the driving region where the angle of attack is nominal to create lift. In diagram A, the driven region, the blade is spinning very fast which results in a shallow angle of attack and relatively little lift to compensate for the drag. Diagram B/D of Figure 18 shows the equilibrium points on the blades. Each rotor blade will experience two points of equilibrium located between the driven and driving, denoted by point B, and between the driving and stalling regions of the rotor disk, denoted by point D. At these two points the lift and drag force cancel each other out. In addition, Figure 18 illustrates that the driven and driving regions are mainly responsible for creating the aerodynamic forces responsible for slowing the rocket's descent whereas the stalling region does not contribute significantly to the drag slowing the descent [16].

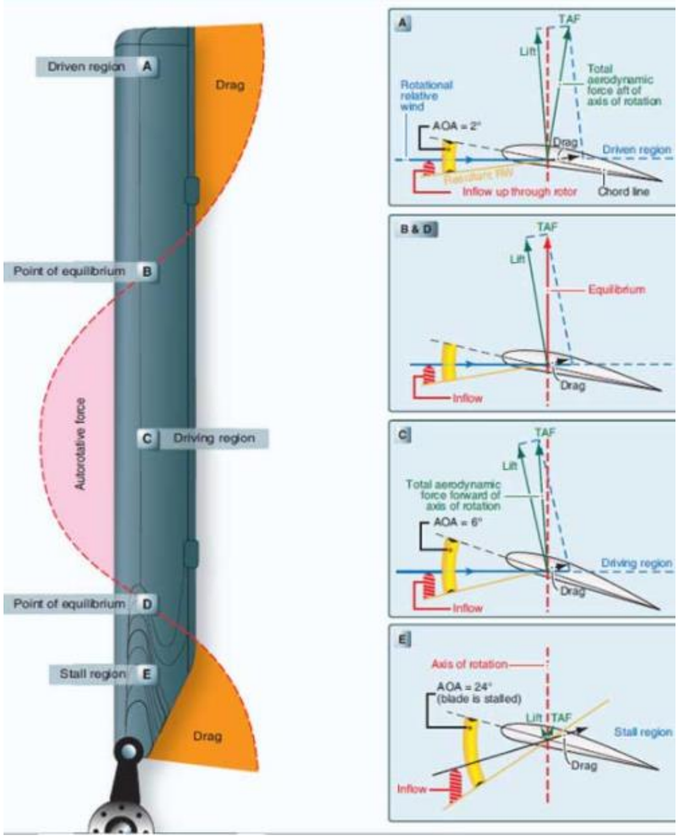


Figure 18 Blade Force Vector © Federal Aviation Administration, 2018

In order to avoid limiting the rate of rotation of the blades the team considered lightweight options for blade material in order to keep them smooth and thin and decrease their moment of inertia in order to increase the rate of rotation. The team will also have to consider the stability of the vehicle as it descends under the autorotation system by attempting to keep the rocket's center of gravity low (i.e., as far aft as possible) during its descent. Furthermore, one specific design choice that was evaluated was the possibility of stalling as a result of an excessive pitch angle on the blades [17].

Like the 2019 MQP team [18], the ARS team for this MQP researched different options for autorotation recovery in high-powered model rockets. There are three types of autorotation models currently used in model rocketry. The first option incorporates external blades that rest along the outer airframe of the vehicle until they are deployed. These blades are typically held in place via burn-strings until ejection [17]. Although this is an easier option in terms of construction, having the blades attached to the outer airframe creates higher levels of drag during the start of the rockets flight. This in turn will result in a lower apogee which may impeded the team's goal of reaching an apogee of at least 1500 ft. The second option involves manufacturing the blades as part of the rocket's outer airframe. This presents complications however due to the sharp edges created by cutting the airframe. In addition, using a portion of the airframe itself as blades is a less reliable option compared to using a dedicated airfoil. The third option involves internal blades that are housed within the outer airframe of the vehicle. In this case the blade size is constrained due to the amount of space available within the airframe. The blades can be pushed out of the airframe via a piston or CO2 canister, or dragged out using a small drogue parachute. This option is the most complex but eliminates the drag increase or reliability concerns inherent in the other two options.

2.1.6.2 Parachute Recovery

Parachute recovery is the most common form of recovery system for model rockets. For this MQP, other stages of the rocket that are too small to warrant the use of an independent autorotation system will make use of this more traditional method of recovery. Parachutes work to create drag to slow the decent of the vehicle or of a particular stage. NASA uses parachutes extensively to land its payloads and crewed capsules [19] [20].

It is common in traditional parachute recovery applications to have a drogue parachute and a main parachute. Drogue parachutes are essential for high-speed applications where the vehicle may be expected to reach higher altitudes or use a dual deployment system [21]. Drogue parachutes are designed to be more rigid, with thicker suspension lines and constructed from more durable material than larger parachutes [21]. The drogue parachute is always smaller than the main parachute and is ejected at apogee where it begins to slow the rockets initial descent to a moderate pace. Later in the rockets flight, a much larger main parachute, is deployed in order to slow the vehicle's descent to a safe speed for landing. This can be seen in Figure 19 [18]. The figure illustrates a recovery profile used by the 2019-2020 MQP team [18].

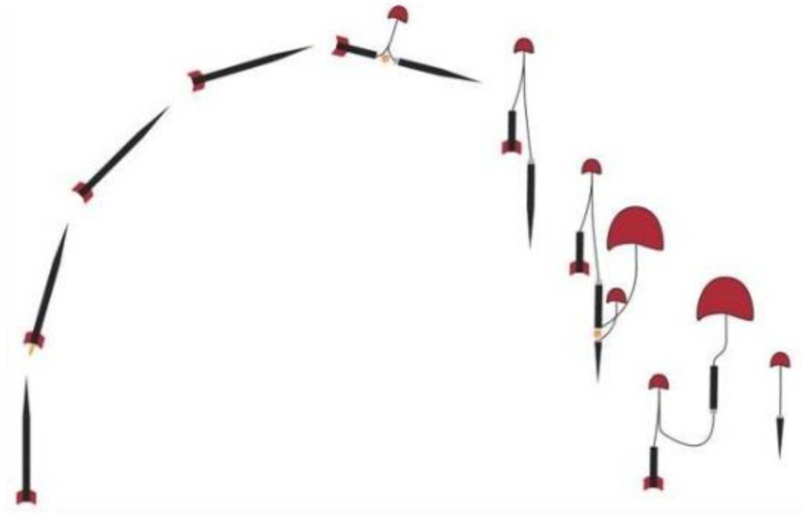


Figure 19 Illustration of Parachute Recovery © WPI Rocket MQP 2019-2020

The most common material used for model rocket parachutes is nylon. Nylon is resistant to ripping and can withstand cold and heat better than plastic, this along with its lightweight nature makes it an ideal choice as a parachute material [22].

Although parachute recovery is the most common option, there are also other recovery options such as gliders, the deployment of objects will a large surface area, and streamers. Rockets that use gliding as a form of recovery are typically designed to stay completely intact for the duration of their flight and can drift quite a far distance from the launch site [23]. There are many options for glider recovery such as rocket-glide recovery, flexie-glider recovery, and lifting-body recovery. Rocket-glide recovery involves the model staying intact for the duration of the gliding descent. Thus, because the glider carries the weight of its motor it descends very quickly [23]. Flexi-gliders use a flexible fabric or plastic wing that folds up for launch in order to create lift during descent. This helps to reduce the frontal area of the model [23]. Lifting-body recovery involves reliance on the fuselage body in order to create lift and slow the descent of the rocket [23]. Recovery that involves the deployment of a large surface area towards the front of the vehicle helps to inflict an upwards drag force on the descending model, thus slowing its speed for landing [23]. An example of a large surface area recovery can be seen in Figure 20 and is often comparable in design to that of a UFO. In order to deploy this sort of recovery system it could be a spring-loaded mechanism that radially folds out. When it deploys air resistance would help to open it up similar to how a parachute works.

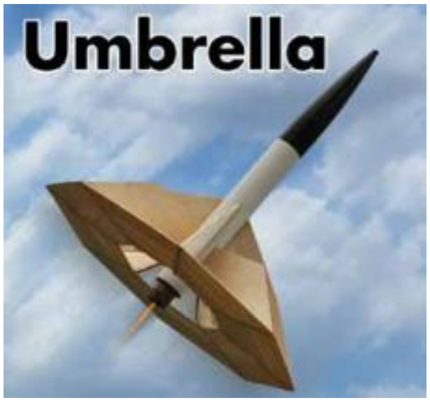


Figure 20 Drag Recovery © Apogee Rockets, 2009

Streamer recovery is typically used for smaller model rocketry applications, rockets using less than 125 grams of burning propellant and weighing with propellant no more than 454 grams [15], due to their lightweight build. Streamer recovery consists of releasing a thin rectangular strip of material that produces drag as it flutters in the wind during the rocket's descent [24]. The typical size of a model rocket that uses streamer recovery is around 56 grams as can be seen in Figure 21.

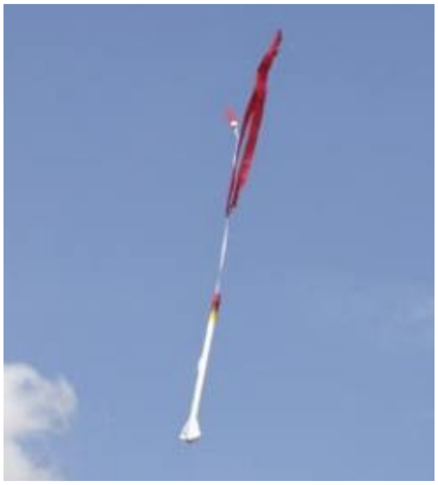


Figure 21 Streamer Recovery © Apogee Rockets, 2009

2.1.7 Motor Retention

The motor mount is an extension of the airframe and serves to connect and center the motor within the rocket. It is housed within the Airframe or Fin Can if such a component is used on a rocket. The connection between the mount and the airframe is held in place by centering rings. The motor tube needs to be made of a material that can withstand temperatures up to and often beyond 100°C due to the heat produced by the motor. In addition, both the motor tube and the centering rings need to be able to withstand the mechanical forces exerted by the motor while it is generating thrust [25].

2.2 Propulsion, Thermal, and Separation Systems (PTSS)

Propulsion, thermal analysis, and stage separation are integral to a successful model rocket. The propulsion system provides the source of thrust for the rocket, allowing it to reach the established goal altitude of 1500 ft for this project. Thermal analysis determines if the structure can survive the high temperatures during the motor burn and helped determine if structural changes needed to be made. As part of this design challenge, innovative stage separation and ignition methods were considered. The requirements for stage separation added a significant level of complexity to this project. Ignition of the initial propulsion stage is an area with potential for innovation that the team also investigated. The PTSS team focused on making sure each of these critical areas were researched and addressed in this project for compatibility with other elements of the system, efficiency, and reliability.

2.2.1 Arc Lighters

Arc lighters are a portable ignition that generate a plasma conduit between separated electrodes to produce heat and initiate combustion. The arc is then placed in contact with a flammable substance resulting in ignition. The basic principle behind arc lighters is that an electrical energy is stored in a capacitor or generated using the piezoelectric effect. This electrical energy is then used to create a high voltage across two electrodes, resulting in a breakdown in the gas between the electrodes. The stored electrical energy is then discharged through a highly conducting column of plasma (ions, electrons, and neutral atoms). This current creates results in heating of the surrounding material. Two metal nodes form the gap, which is less than a millimeter in length [26]. For gases at atmospheric pressure, a high voltage - approximately 3,000V per millimeter in distance - is required to surpass the breakdown voltage of air but, with the right transformer configuration with alternating currents, an arc can be produced with as little as 3V. This form of ignition has been used since the early 1800s in the first forms of electric lighting [26]. The first widespread arc lighter system specifically meant for portable ignition was only recently introduced in 2015 as a successful Kickstarter. Once this brought arc lighters into the eyes of the public, several companies began mass producing these products, and they are now readily available for commercial use. Compared to standard, flint-spark ignited butane lighters, arc lighters do not require chemical fuel, they are windproof, and they produce arcs which are hotter than regular flames.

With their increasing popularity, many resources in the form of videos, articles, and websites have become available for the assembly and use of home-made arc lighters [27]. The construction of a low voltage igniter is accomplished with a combination of a battery, resistors, capacitors, and various other electrical components to create an operational arc lighter with on/off switches and controllable currents. For model rocketry, ignition systems are usually composed of electrical matches, or e-matches, which can be either commercially bought or home-made. E-matches use an electric current running through a wire within a combustible compound to create ignition. Current is sent through a designated ‘bridge wire’, which is highly resistant and causes the combustible compound to ignite.

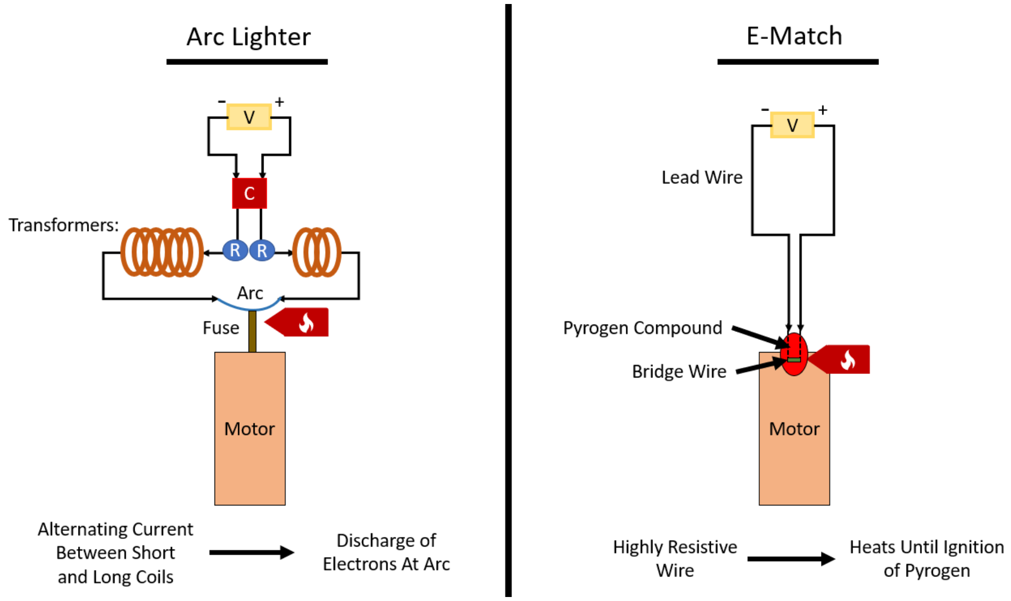


Figure 22 Comparison of Arc Lighter and E-Match Fundamentals

Figure 22 above shows the fundamental differences between the two methods. While homemade e-matches are less expensive than commercially available options, they do involve more complexity in their assembly and are less reliable [28]. However, e-matches require much stronger power sources in comparison to arc lighters with variations between 9V and 12V batteries. Although ignition times may be slightly longer due to the need for a charge up of electricity, arc lighter ignition is still acceptable for model rocketry needs, especially if there is only one motor to ignite at a time.

2.2.2 Stage Separation and Air Brakes

One of the goals for a high-powered model rocket is to have its payload reach the highest apogee possible. The practice of ditching dry mass stage separation is common both in full scale rocketry and model rocketry. Typically stages separate upon completion of an engine burn on the first stage, and the spent dry mass is discarded midflight [29]. Newton's second law states that an object has a higher acceleration with a constant force if mass decreases. When a first stage burn ends, there is a short period of time before the first stage has separated and the second stage engine has lit. In these few seconds, drag induced by air resistance slows the rocket's velocity. Therefore, the most efficient way to keep the rocket's momentum going is to light the second stage immediately at first stage burnout. For a solid motor typically used with model rockets this is usually when thrust begins to decrease towards the end of its burn time. However, immediately lighting the second stage motor could damage the stage separation mechanisms or the recovery systems on the first stage. To avoid this, there needs to be a safe distance between the two stages.

Another issue found during stage separation is maintaining control. If a staging system induces a disturbance, shock, or perturbation to the vehicle's orientation during staging the vehicle could deviate from its intended course and potentially be destroyed. It is common in large scale rockets to see small solid motors separate the vehicle's stages, such as the Saturn V with its eight retrorockets on the first S-IC stage [30]. It is also common to see resettable systems used, such as the Falcon 9's collets and a pneumatic pusher system. Both systems use mechanical components to induce a force which ensures a distance is achieved before next stage engine ignition while not inducing a roll component of their motion.

Adding an angular velocity to the complexity of the problem aggravates this separation event. In the 1960s the United states was investigating intercontinental ballistic missiles, which were based off early sounding rockets. These rockets were often spin stabilized and were separated with a guide shoe and guide rail system. The Athena flight test is a well-known declassified test flight, and some investigators in 1970 published that an abnormal stage separation event caused an “excessive hinging motion” which led to an overall mission failure after the rocket did not meet its intended trajectory [31].

The use of air brakes on rockets is a common method to produce a large drag. This technique is used on Blue Origin’s New Sheppard rocket, featured in Figure 23 [32]. After stage separation, the upper section of New Sheppard’s first stage opens air brakes, inducing a large drag which slows the booster before landing. In Blue Origins case, they use airbrakes to reduce the energy required from the rocket engine to land safely. Another application of air brakes is to slow the spent first stage down to get clear from the second stage. This large drag slows the first stage quicker than the second stage in this short period. It also allows the second stage to ignite its motor sooner, reducing the amount of momentum lost in free flight.

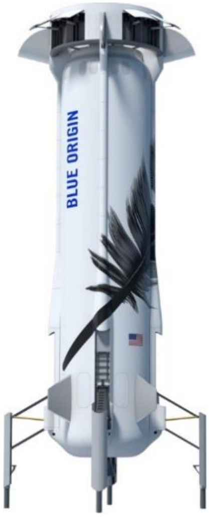


Figure 23 Blue Origin’s New Sheppard Reusable Rocket Booster. © Blue Origin 2019

2.2.3 Model Rocket Motors

Model rocket motors function on the same principles as large solid rocket boosters, that are used for spaceflight. A model rocket motor is just a scaled version of one of these boosters. To ensure safety and consistency, model rocket motors are commercially manufactured and sold [29]. To properly communicate a model rockets specifications and capabilities the National Association for Rocketry developed a three-part code. The first part of this code is a letter, which indicates the total impulse provided by the rocket motor. For model rockets this letter ranges from 1/8A to 0, with total impulses of 0.3125 to 40960 N*sec, respectively [33]. To ensure proper usage and safety, letters H-O are restricted to people and groups who have an appropriate certification. Due to the groups level-2 certification, rocket motors up to class L, with a cap of 5,120 Ns total impulse, can be utilized.

The second part of the code is a number, which may be either one digit or more digits. This number represents the average thrust that the rocket motor provides. While the last number of the code represents the time delay between the end of the motor's burn and when it deploys its ejection charge, releasing the recovery system. Some rockets will have a zero in this spot to indicate that they are booster motors, and do not have an ejection charge. This allows them to be used as lower stage engines in multi-stage rockets or in the final stages of rockets with unique recovery systems [33].

The chemical composition of model rocket motors varies between black powder, ammonium perchlorate, and hybrid motors. Black powder motors are often used for low power rockets, usually below class D. While some black powder motors can provide a higher impulse, they are not common. Instead, ammonium perchlorate, or ammonium chlorate composite propellent (ACAP), is often used in high-powered rockets as it can supply a larger impulse [33]. Another benefit of ACAP motors is that they are often reloadable, whereas black powder motors are not. A reloadable rocket motor comes with a permanent casing, often made of aluminum, that can be refilled with motor packs. While reloadable motors have a higher initial cost, they are often more cost effective in the long run as the refill packs are less expensive than single use motors [29].

Due to the design specifications of the rocket, motors that will be of particular interest will include two reloadable motors of class K or L, that have a significant average thrust, and no delay charge. These motors will likely use ACAP as a fuel, due to their high total impulse, reloadability, and average thrust.

2.3 Flight Dynamic Analysis

This section provides a background on the main analysis methods and control systems that will be employed by the rocket. Research was conducted with a focus on designing efficient modelling systems and creating a passive structure through spin stabilization that allows for stable ascent and flight.

2.3.1 Models for Attitude and Trajectory

In order to make complex decisions on the attitude and trajectory of a high-powered model rocket, it is important to have a thorough understanding of the vehicle's behavior. This process involves creating a system to model the flight and orientation of the rocket starting from ignition and ending with the final landing position that can be determined through simulation.

An effective model of a rocket in flight must include all the pertinent state parameters. This means the analysis requires knowledge of the rocket's position, velocity, and attitude. These values can be modeled by evaluating each parameter's state derivatives and state vectors and iteratively calculating the way they change during flight. The flight parameters are all defined with respect to the fixed frame of the rocket's body. For this project, the model will be created using yaw-pitch-roll angles and will follow the general guiding principles outlined in "Stochastic Six-Degree of Freedom Flight Simulator for Passively Controlled High-Power Rockets" by Simon Box [34]. This model simplifies the system by making some key assumptions to allow for easier simulation. As per Box, the model assumes: the rocket is passively controlled, the center of pressure of the rocket is constant and all forces act through it, the rocket is axisymmetric, gravity is constant at all altitudes, and the moments of inertia are constant throughout flight (might be altered). While the force of gravity is constant with respect to mass according to our assumptions, the force of gravity in the body fixed frame is relative to the rocket's orientation with respect to the Earth.

Using the state derivatives, vectors, and specific rocket parameters, a simulation can be conducted to iteratively update the values of the state vectors and derivatives. Using a small time-step in between iterations allows for a more accurate model. This simulator will employ the Runge-Kutta-Fehlberg method for the numerical solution of ordinary differential equations in MATLAB. These values can then be plotted to create a flight map as well as a velocity and orientation profile during the flight.

The main methods of addressing the problem of coupled trajectory and orientation modelling are derived through Euler angles, quaternions, and yaw-pitch-roll analysis. The analysis presented in this report will employ the yaw-pitch-roll approach as outlined in "Orbital Mechanics for Engineering Students" by Howard D. Curtis [35]. This will allow for a coupled analysis of the trajectory and orientation to achieve a more accurate model of the rocket's behavior. It also avoids a problem with the Euler angle method, which is that the system becomes singular when the nutation angle is zero.

The rocket the team will focus on in this report does not boast any active control systems, so the analysis will be solely reliant on passive control systems such as the spin stabilization system and the dynamic stability created by the rocket's geometry. The essential figures for the model analysis are listed in Table 6 below.

Table 6 Break-down of State Space Control Variables

Vector	**Vector Components**	**Vector Description**
Position: X	X, Y, Z	**Position of rocket, geodetic frame**
Linear Velocity: V	u, v, w	**Linear Velocity of rocket, geodetic frame**
Force: F	F_x, F_y, F_z	**Net Force Acting on Rocket**
Acceleration: a	a_x, a_y, a_z	**Net acceleration caused by Force**
Angular Rate: ω	p, q, r	**Angular Rates about body-fixed X, Y, Z axes**
Euler Angles:	φ, θ	**Rocket Euler Angles**

The position and velocity vectors **X** and **V** are both with respect to the constant geodetic frame, and the model neglects the effects of the rotation of the Earth. The force vector, **F**, is time varying based on aerodynamic loads, the changing force of gravity due to the rotation of the rocket, and the - assumed constant - thrust output by the rocket. The acceleration, **a**, is time varying dependent on the depletion of mass due to thrust, as well as the varying forces experienced by the vehicle. The angular rate, **ω**, and Euler angles determine the attitude of the rocket. Due to the thrust forces from the motor that act axis-symmetrically on the rocket, the attitude with respect to the Earth is essential for determining the flight path taken by the vehicle. The body-fixed frame will only experience changing forces due to aerodynamic changes (changing velocity causing more drag) and gravity (different orientation causes gravity to act on the rocket's frame in varying ways), with a relatively constant thrust component. The thrust will naturally contribute to the vertical displacement of the rocket, but this changing attitude will cause it to propel horizontally from the launch site to some degree.

Analyzing different parameters in different reference frames requires coordinate transformations that are conducted using transformation matrices [36]. In each iteration, the flight parameters will be adjusted between the geodetic and body-fixed coordinate frames to model the dynamics and attitude of the rocket.

The ideal model will include: a moving position for the center of gravity that adjusts with respect to the rate of mass depletion, coupled attitude and dynamics models, an animated flight path and a time-related velocity profile. Completion of this model will allow for informed decision making in determining certain design parameters such as fin design and recovery system design.

2.3.2 Stabilization and Control Systems

Stability in a projectile is essential to keeping the object flying along the right path. Stability can be achieved in two ways, active stabilization, and passive stabilization. Active stabilization and control systems change the direction of the rocket to either compensate for outside forces effecting the rocket's trajectory or change the rockets trajectory all together. Some examples of active control systems are moveable fins, jet vanes, vernier rocket, gimbaled engines, see Figure 24. Moveable fins take advantage of aerodynamic forces to create a resultant force that changes the direction of the rocket [37]. Jet vanes are placed in the exhaust of the engine and deflect the direction of the engine's exhaust, causing the rocket to change direction. Vernier Rockets are smaller rockets placed on the sides and around the bottom of the rocket. By firing one or a combination of these Vernier Rockets, the rocket's heading can be changed or corrected. Gimbaled thrust systems, swivel the nozzle of the rocket in order to change the direction of the thrust, and therefore the rocket's trajectory.

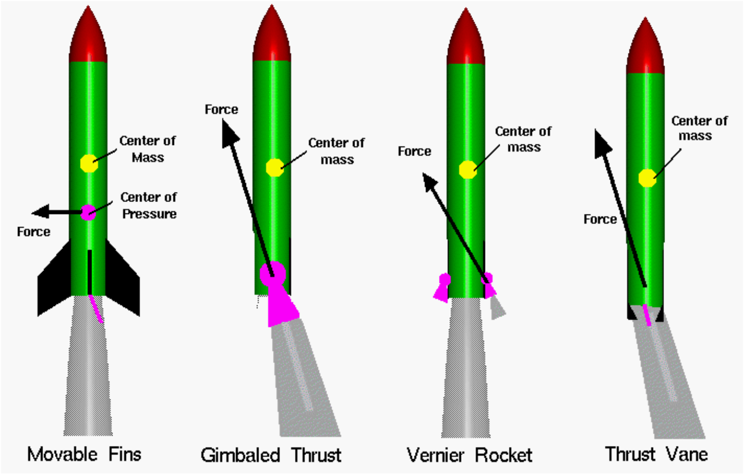


Figure 24 Movable Fins, Gimbaled Thrust, Vernier Rockets, and Thrust Vanes. © NASA 2014

Passive stabilization systems or controls “are fixed devices that keep rockets stabilized by their very presence on the rocket's exterior.” [38]. Passive controls work mainly to keep the rocket’s center of pressure behind the rocket’s center of mass. “It is extremely important that the center of pressure in a rocket be located toward the tail and the center of mass be located toward the nose.” [37]. If these two are very close two each other, or if the center of mass is behind the center of pressure, the rocket will try to tumble through the air about the center of mass in the pith and yaw direction. The use of fins helps to keep the center of pressure behind the center of mass due to their large surface area. Fins can also be used to spin the rocket along the roll axis, which can help stabilize the rocket.

This type of stabilization, called spin stabilization, is commonly seen in spiraling footballs and bullets shot from a gun barrel with rifling. The spinning of a mass creates a gyroscopic effect that resists outside forces that would otherwise knock the mass off course, for example a large gust of wind. The 2019-2020 Rocket Design MQP chose to evaluate the moment of three different fin design categories [18]. Fins with varying areas of attack, fins with a cambered airfoil, and fins with an adjustable control surface on the rear of the fin, called a rolleron, were all evaluated to see the moments they would produce on the rocket. However, due to time constraints, they were not able to consider the drag and spin rate that each fin design would produce. These measurements are extremely valuable in determining the ideal fin for the rocket, as the drag produced by these different fin types can slow the rocket and lower its peak altitude [38]. The ideal fin would produce a high moment, causing sufficient spin while keeping drag low, in order to not significantly hinder the rocket's performance. This can be done by testing the designed fins in Ansys Fluent and calculating the drag, and the spin rate that each fin causes. From this, the ideal fin design can be determined and implemented by the team into the rocket.

3 Methodology

This section of the paper outlines the design and construction (when applicable) of each component of the rocket. The components of the rocket fall into three subdivisions, the Airframe and Recovery System, the Propulsion, Thermal, and Separation Systems, and Flight Dynamics Analysis. The rocket, as designed, can be seen in Figure 1 and 2. In total, the rocket has two stages and carries a payload of an astronaut capsule and camera. Initial designs were created using OpenRocket, a program in which a model rocket can be designed and simulated. Then the final design was created through iteration, calculation, and analysis.

3.1 Airframe and Recovery System

The Airframe and Recovery System of the rocket include the airframe, the fins and fin can, the avionics bay, the autorotation system, the payload bay, and the nose cone.

Throughout the design process certain considerations were kept in mind. These considerations were the feasibility of manufacturing, ease of assembly, and size — more specifically, length budget inside the rocket.

3.1.1 Airframe

The airframe of this rocket was designed to have an internal diameter of 5 inches. The overall length of the airframe is 77.4 inches and is broken into three airframe sections referred to as airframe one, airframe two, and airframe three, with airframe one being the first following the nose cone. The rocket has two stages; airframe one and airframe two are components of Stage 2 and airframe three comprises Stage 1. The airframe sections can be seen in Figure 25.

Airframe one is 16 inches long and houses the vehicle's payload, the parachute system for this airframe, and the nose cone. Airframe two is 40 inches long and houses the autorotation system designed to recover airframe two, the Stage 2 avionics bay, and the Stage 2 motor and retention system. Airframe three is 24.5 inches long and houses the stage separation and airbrake system, the parachute recovery for Stage 1, the avionics for Stage 1, and the motor and retention system for Stage 1.

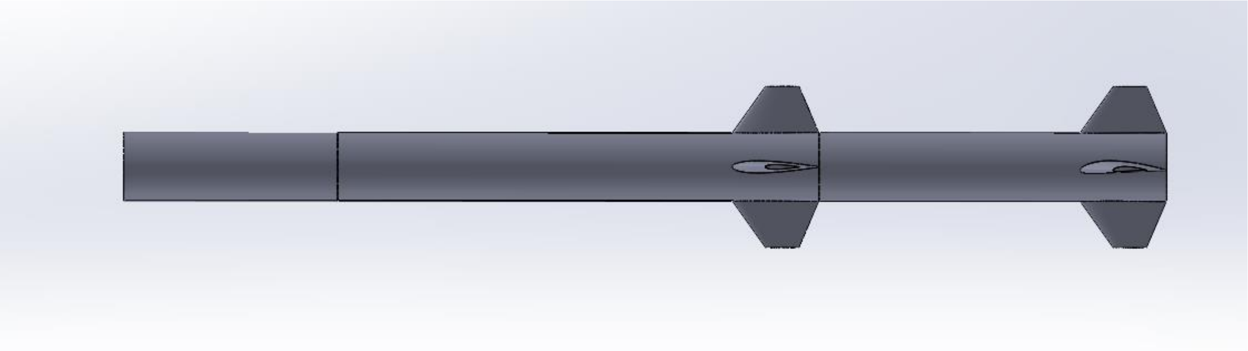


Figure 25 Airframes of the Rocket

The airframe of the rocket is made of filament wound fiberglass. The airframe is designed and manufactured by Madcow Rocketry [39]. As mentioned previously, the inner diameter of the airframes is 5 inches. The airframes have a thickness of 0.075 inches and weigh 19 oz/ft. The airframe couplers are of the same material and are also available from Madcow Rocketry. Each coupler has an inner diameter of 4.815 inches, a thickness of 0.0915 inches. The airframe is commercially available for the cost of $170 per 60-inch segment. The couplers are available for the cost of $53 per 12-inch segment [39].

To assemble the rocket, the team must cut airframe sections to length, drill holes for bolts, cut the astronaut capsule window, and cut sections of the airframes for brakes for the separation system. To do this, the team will use a Dremel tool and a drill. Fiberglass and carbon fiber dust, like all dust, is harmful to the lungs when inhaled. All cutting of these materials will be done in well ventilated spaces with proper personal protective equipment (PPE).

The details of the airframe design, such as airframe section lengths and bolt hole locations, are determined by the subsystems and components that make up the rocket. Therefore, details of integration between the subsystems and components will be detailed in these upcoming sections.

3.1.2 Fins

The rocket has two sets of four trapezoidal fins that were designed to induce spin. The fins are both the same dimensions but utilize two different NACA airfoils to serve each stage most efficiently. The fins have a root chord of 6.5 inches, a tip chord of 2.5 inches, a sweep angle of 55°, and the airfoils correspond to the NACA 4414 for the first stage and NACA 0414 for the second stage. Figure 26 and Figure 27 show the first and second stage fins, respectively. These airfoils are the same in all aspects except for their camber, which is 4% and 0%, respectively. To determine which airfoils would create the appropriate level of induced spin it is important to understand the differences between these cases. The team chose to investigate changes in cant and camber only to limit independent variables and simplify experimentation. By doing this, the team was able to better understand the relationships between the differences in the fins and the differences in the results. The complete process of analyzing and then selecting the fins for the project is described in Section 3.3.3 and Section 4.3.3, respectively.

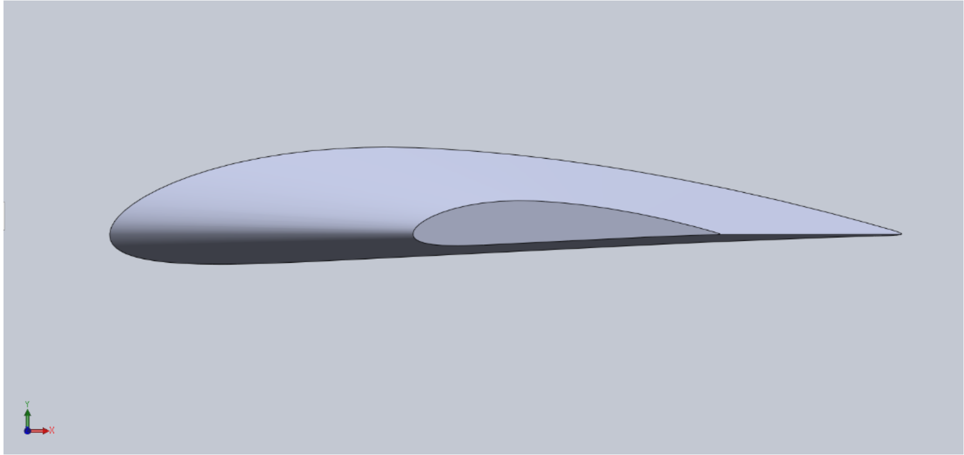


Figure 26 Stage 1 Fin, NACA 4414

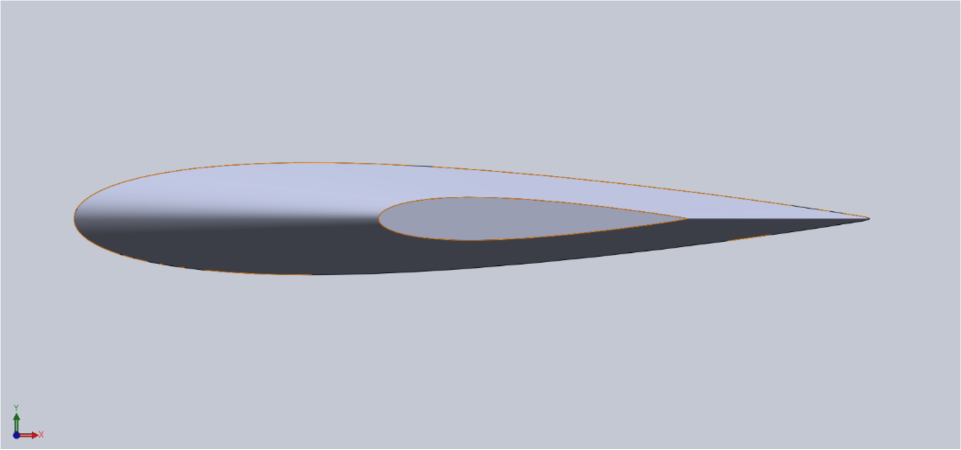


Figure 27 Stage 2 Fin, NACA 0414

For a rocket of the size and with the anticipated flight speed the team has designed, an elliptical fin is the ideal shape because it produces the least amount of induced drag from fin tip vortices [40]. Because most of the lift force is generated at the base of the elliptical fin, less air flows around the tip which reduces the fin tip vortices [40]. However, with the equipment available to the team, this shape is difficult to manufacture due to its curved nature. Trapezoidal fins are used instead because they provide many of the same benefits as the elliptical fins while being easier to manufactured. The main benefit being that, like an elliptical fin, most of the lift force is still generated closer to the airframe.

The fins were designed to be made of fiberglass composite with a wire cut foam core. The team chose to use these materials because of the material strength provided by the carbon fiber composite and the ability to shape the foam into the design airfoil shape. The lightness of both materials is also beneficial because weight on the aft end of the rocket decreases stability. The stiffness of the materials combined should be enough to negate potential flutter of the fins. While it is not a primary concern, there is a chance of the rocket landing on a fin during its descent. However, because fiberglass has a relatively flexibility (lower stiffness), it should be able to withstand a more-forceful-than-anticipated landing.

The fins are surface mounted to the airframe using epoxy and a tip-to-tip lamination of fiberglass. They transition seamlessly to the surface of the airframe with a 0.5-inch radius fillet.

3.1.3 Avionics Bay

The avionics bay is the central housing unit for the electronic and power systems for the rocket. The primary flight computer for the rocket will be a Raven 3.0 altimeter (Figure 28), which is manufactured by Featherweight [41]. The Raven serves several purposes, acting as a microprocessor, inertial measurement unit, and altimeter. The primary rationale for this choice is to reduce cost, as this altimeter was previously used by the 2019-2020 team [18] and is sufficient for the mission requirements.

Figure 28 Raven 3.0 Altimeter (Featherweight Altimeters) ©, FA 2009

The Raven altimeter serves as the team’s main computer and controls all on-board computational needs. For the team’s purposes, this includes the following operations: Initiation of separation between Stage 1 and Stage 2, ignition of the Stage 2 motor using an e-lighter, ignition of black powder charges to separate nose cone and deploy the autorotation device. The altimeter initiates these different mission phase changes through the following protocols. The following approach is theoretical and planned, as the team has been unable to physically handle the altimeter during the time this project was undertaken.

The first stage deployment, separation from Stage 1 to Stage 2, will be engaged by an abrupt decrease in acceleration. The Raven altimeter can detect a change in acceleration, so when the acceleration’s direction and magnitude change due to the drop in thrust resulting from the first stage motor cut-out, the stage separation mechanism is engaged [41]. The ignition of Stage 2 will follow shortly after the release of the first stage using a time delay of a few seconds to allow for complete separation. The stage must give sufficient time for the team’s innovative separation system to fully separate so as to not damage the mechanical system, as well as to ensure a safe burn that is unperturbed. The nose cone separation will occur at, or shortly after, apogee. This can be initiated using the pressure sensor on the altimeter [41]. As altitude increases, air pressure decreases, and the Raven altimeter will engage the nose cone separation once pressure begins to increase. The altimeter documentation notes that this may be a problem in supersonic flight, and a separate velocity channel is thus required as a back-up check, but due to the known limitations of our rocket, it will not reach supersonic conditions.

The altimeter is limited to a maximum of four event triggers, a limitation which is not a concern for our mission profile [41]. When a deployment phase is initiated, the altimeter can deliver a maximum current of 25A, a considerable throughput. The altimeter has a built-in internal power system of a Lithium-polymer (LiPo) battery that is sufficient for all necessary operations.

In addition to the altimeter, a data collection board (RD-KL25-AGMP01) (Figure 29) will be used to collect and organize data from the flight. While the Raven altimeter can also record data, the collection board serves as both a backup and an extension of the capabilities of the altimeter. The RD-KL25 boasts a FXOS8700CQ 6-axis e-compass, FXAS21002C 3-axis digital angular rate gyroscope, MPL3115A2 Precision altimeter and temperature sensor, and Kinetis KL25Z and Kinetis K20 microcontrollers [42]. For our purposes, the microcontrollers are valuable as they allow the data collection board to work independently of the altimeter.

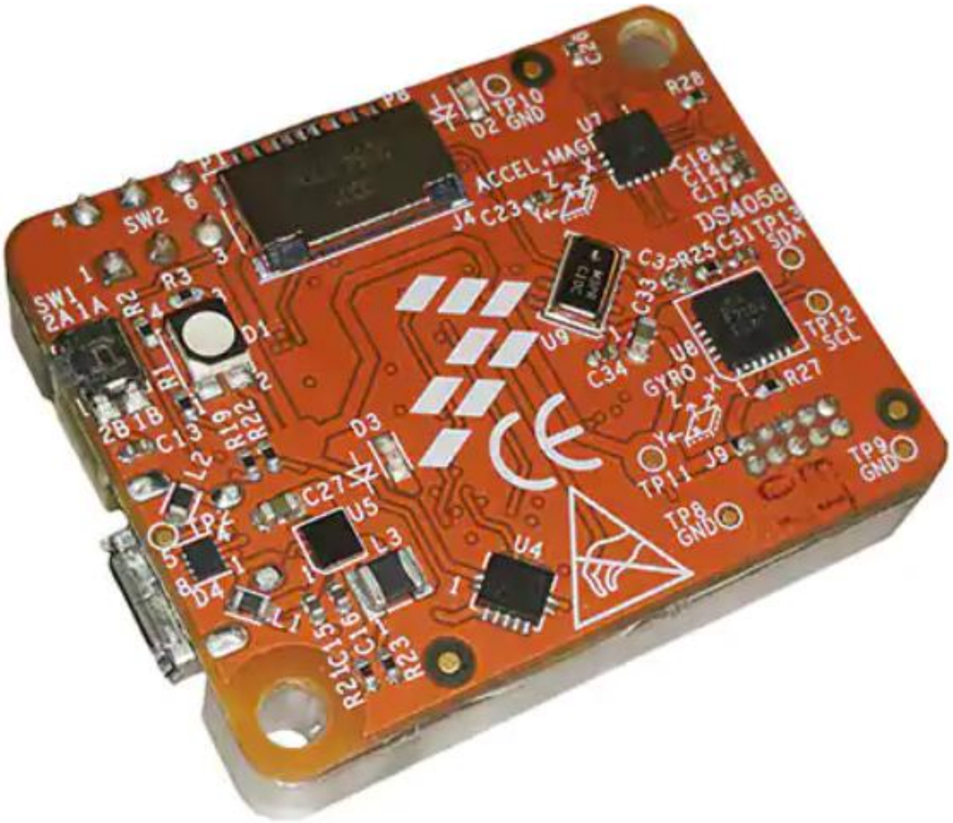


Figure 29 RD KL25-AGMP01 data collection board (Freescale Semiconductor) ©2015, NXP

The final data analysis will consist of the acceleration, gyroscopic, and barometric data from the RD-KL25-AGMP01 and the barometric and acceleration data from the Raven altimeter. Having backups of the data is valuable, as multiple sensor readings allows the team to determine the actual behavior of the rocket more accurately, while also being a safe backup in case of failure.

The avionics bay was not designed in detail in a CAD software. The intent is to have a cylinder where the height is equal to the radius (5 inches) with the components on two layers parallel to the bulkheads. To accommodate for the long autorotation blades, the avionics bay will have with three vertical slots spaced equally radially in the exterior of the avionics bay case. The avionics bay was intended to 3D printed out with plywood bulkheads. On the forward end of the avionics bay, there are black powder caps, a U-bolt, and terminal blocks. To secure the avionics bay into the airframe, the avionics bay was intended to be lowered into the airframe from the forward end.

Although it was not designed, the intent for securing avionics into the avionics bay was to 3D print a “sled” to fit the components.

3.1.4 Autorotation System

Autorotation is a system used by helicopters to perform unpowered landings using the upward motion of the air during descent to turn the main rotor. The system works much the same in a rocket recovery system. Blades are deployed from a static position during descent and the upward rushing air forces them to rotate, creating lift that slows the descent velocity to a safe speed.

Following the drogue event after the rocket's apogee, an autorotation recovery system is deployed to slow the second stage of the rocket’s fall. The system will be made of three fixed airfoils hinged to a mechanism that relied on a shock cord to deploy. The drogue event pulls the deployment mechanism out of the airframe to begin the autorotation portion of descent. The airfoils chosen by the ARS subteam are 780mm Align Carbon Fiber Blades [43]. The blades have a length of 780mm, and are 64mm wide. The airfoil has a thickness of 8.5mm. These blades were chosen while considering their weight and length to ensure they would not significantly decrease the rocket’s apogee or fail to fit within the confines of the airframe. Figure 30 shows the CAD model of the autorotation recovery system.

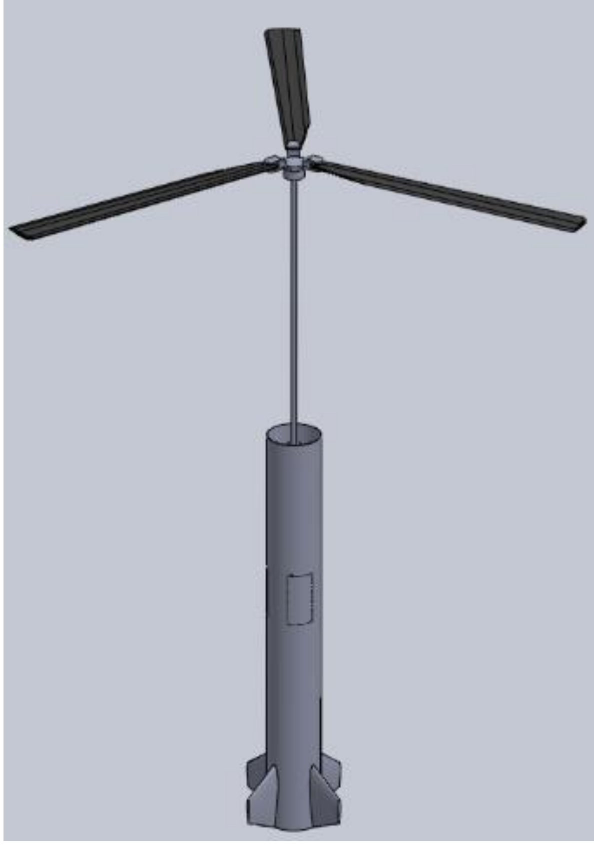


Figure 30 Autorotation CAD

The blades are contained in the upper airframe under the astronaut capsule to prevent them from causing any drag on the rocket's exterior during ascent. The deployment mechanism is secured to the interior of the airframe and slides out during the drogue release. After their deployment, the force of the air unfurls the blades into position and starts rotation. The lift created by the rotating blades will slow the rocket's decent velocity ensuring the second stage lands safely. There were considerations in the design of the deployment mechanism to help decrease the descent velocity of the rocket. One was the angle of attack of the blades (AOA), an optimal AOA provides the most lift to slow the rocket down, however air conditions such as wind forces on the rocket can alter the AOA and prove detrimental to the autorotation recovery. In addition, unlike with helicopter autorotation, the blades of the autorotation system start with no initial rotational velocity. This means there is a period after the deployment event that the system is not providing its optimal lift while it spins up to speed.

3.1.5 Payload Bay

This year the MQP team chose an astronaut and camera to meet the payload requirement for the model rocket. These payloads were to be housed in a compartment referred to by the team as the "astronaut capsule." This capsule design was based off the designs of prior MQP teams such as the 2019-2020 MQP team [44]. The camera chosen was a GoPro, the same one used by the 2019-2020 MQP team. This is because the camera was readily available and already owned by the Aerospace Engineering Department. By reusing it, the team would be able to save money for other parts. The size of the GoPro was approximately 2.6x1.8x1.4 inches and weighed approximately 118 grams with a battery and memory card installed [45]. The selected "astronaut" was to be an Among Us plushie toy that was approximately 4 inches in height [46]. Because one of the team's goals was to ensure the astronaut landed safely in his capsule, he would have had an accelerometer attached for the duration of the rocket's flight.

The astronaut capsule or payload bay was designed such that it was made from a stock G-12 fiberglass outer airframe purchased from Madcow Rocketry. It is located between the nosecone and the upper airframe and attached to each via fiberglass couplers and shear pins. The astronaut capsule represents approximately 16 inches of the total outer airframe. Of those 16 inches, 5 inches are occupied by the nose cone coupler, 2.5 inches by the coupler connecting the astronaut capsule to the upper airframe, 2 inches by the packed parachute, and 6 inches accommodate the camera and astronaut.

Because the payload includes a camera, the ARS subteam decided to design and fabricate a window similar to the one used by the 2019-2020 MQP team [44]. Their window was epoxied into the airframe by heating a polycarbonate sheet cut to accommodate the area they were trying to work with and molding it into a cut section of the airframe. As they did this, they found that it took many attempts to ensure the polycarbonate did not end up damaged or scratched during this process. This year, the ARS team wanted to optimize this design and come up with an efficient way to avoid the same obstacles faced by the 2019-2020 MQP team. The team researched various options for protecting polycarbonate and plastic materials from scratching while being used to build the capsule. The team ultimately decided that they would use a Plexus Plastic Cleaner Protectant Polish [47], a product designed to protect aircraft windshields and painted surfaces, while being commercially available at a cheap price. Unfortunately, due to limitations caused by the COVID-19 Pandemic, the team did not have the opportunity to test this in practice. Theoretically, the team would have applied the polish to the polycarbonate sheet before cutting and fabricating it to fit within the fiberglass airframe.

When cutting fiberglass, it is important to consider the possible safety risks. Cutting fiberglass can become dangerous when splintering, chipping, or harmful fibers get into the air. To protect against these harmful fibers, the team would have needed to wear face shields, protective gloves, protective clothing, and masks [48]. When constructing the astronaut capsule, the team would have begun by marking the 4.5x2 inch dimensions of the window on the airframe. Then the team would have placed a piece of tape over the surface of fiberglass that was to be cut. The tape would have helped to prevent chipping around the outside of the cut. Next, a starter hole would have been drilled such that a Dremel tool with a 561 multipurpose cutting bit could then be used to safely cut the hole for the window out of the fiberglass tubing.

A piece of polycarbonate would have been cut using a hacksaw to ensure it would fit within the hole in the airframe. This could have prevented the problem encountered by the 2019-2020

MQP team, in which the polycarbonate was hard to cut straight due to its flimsy nature [44]. The team would have used a similar method to stabilize the polycarbonate, i.e. using a piece of wood as a brace while cutting it.

Next would have been the molding process. To protect the edges of the airframe the team would have needed to use tape as done by the 2019-2020 MQP team. The team would have needed to use a coupler to mold the polycarbonate correctly to the shape of the airframe. Furthermore, to protect the coupler, it also would have had to be wrapped in tape. In order to mold the window into place the polycarbonate would have needed to be heated sufficiently. Polycarbonate begins to bend between temperatures of 155 and 165 degrees Celsius [49]. Once the polycarbonate window had been heated to this temperature it would have become malleable enough to mold. Once the window had been molded to the correct shape, the team would have applied a layer of epoxy on the edges. The window would then have been held into place using clamps while the epoxy cured and stuck to the airframe.

3.1.6 Nose Cone

The team chose a pre-made nose cone for this project that was compatible with the G-12, 5-inch diameter fiberglass tubing selected for the airframe. The selected nose cone is a fiberglass, 5-inch filament wound metal tip nose cone with an ogive shape. The nose cone comes with a separate coupler that the team must measure and epoxy into place at a specific location on the nose cone. This location is such that the end of the coupler sticks out about 5-inches from the end of the nose cone. Once this is done the team would have then epoxied a bulkhead into place at the bottom of the coupler. To ensure the coupler would still slide easily into the airframe, the team would sand away the excess epoxy while ensuring proper safety measures were taken such as wearing masks, gloves, protective clothing, and goggles. The bulkhead would have had a U-bolt to attach the nosecone to the drogue parachute.

3.2 Propulsion, Thermal, and Separation Systems

The Propulsion, Thermal, and Separation Systems of the rocket include the Stage 1 and Stage 2 motors, the ignition system, and the mechanical air brake separation system. Throughout the design process, certain considerations were kept in mind. These considerations were the feasibility of manufacturing, ease of assembly, and size, or more specifically, length budget inside the rocket.

3.2.1 Motor Selection

Since at least one member of the team had a class 2 certification, we were able to select motors with a combined total impulse up to 5,120 N·s. To accomplish this, a catalog of potential rocket motors was created based on OpenRocket's database of rocket motors, which listed many COTS rocket motors and their impulses. The motors that had a total impulse within the target range were then researched online to determine if they were available, i.e. still in production and viable for the project. If so, these motors were put into the team's catalog along with motor performance information, such as I_{sp}, average thrust, maximum thrust, and burn time, among others.

The motor catalog provided a pre-vetted list of potential motor options with varying specifications that could be selected based on what the mission required. However, to know which motor would be best suited for the mission the team needed to estimate the mass of each stage of the rocket. A spreadsheet was created to track the masses and estimated masses of the stages so the craft's total mass could be properly estimated as the design progressed. This spreadsheet also provided the minimum required thrust for each stage, by multiplying the total weight of each stage by five, for a five to one thrust to weight ratio. A greater than 5:1 thrust to weight ratio is required by local rocketry clubs and launch sites as it allows the rocket to be stable and safely leave the launch rod [50]. This required thrust further helped narrow down the potential options for the motors of the respective stage.

The first stage motor was selected first, as it had the higher required thrust which also meant it would usually carry the larger impulse value. Having this impulse value already established narrowed down the potential options for the second stage motor, as its impulse value could not exceed the 5120 N·s minus the impulse of the first stage motor.

Through this process, the team determined that the first stage rocket motor would be the AeroTech K1000T-P with a total impulse of 2497 N·s and an average thrust value of 1066 N [51]. Furthermore, the second stage motor was selected to be the Ceasaroni K661 with a total impulse of 2430.4 N·s. This motor has an average thrust value of 660.5 N [52].

3.2.2 Motor Tube Design

To ensure that the motor was secured into the rocket body and pointed in line with the axis, it was necessary to construct a motor tube which not only centered the motor in the rocket but also transferred the motor's thrust to the rest of the rocket.

At the time that the motor tube was first modeled the exact dimensions of the motors were unknown. Therefore, a generic motor tube was designed with approximated dimensions for the motors that were being considered. The design that was used for the motor tube was based on images of 2019-2020 MQP team's design motor tube design [18]. This design had three main components: the motor casing, the motor tube, and three centering rings to ensure stability and properly distribute thrust [18]. The current design followed this approach, incorporating the same three pieces of the motor tube. However, since our references were only pictures, the thicknesses and dimensions of the tube were unclear. To ensure structural stability the parts had to be thick enough to withstand the expected forces and support the system.

The motor casing is a part that comes with the rocket motor, and therefore has a thickness determined by the manufacturer. The average thickness of this case could not be found; therefore, we decided to use a stand-in thickness for the case of 1.5 mm. The tube that the case was in had a thickness of 5 mm while the centering rings were 10 mm thick and equally spaced along the assembly.

3.2.3 Air Brake Separation System

The goal of the stage separation design was to solve two problems as defined in section 2.2.2:

- Staging induced instability
- Rapid distancing of two stages

The solution for the first problem is to design a mechanism which “lets go” of the second stage instead of pushing it away. This forceless separation helps mitigate any perturbations, especially with a craft that is spinning at a rapid rate. However, a forceless separation does not clear the spent stage away from the next stage’s motor ignition. To solve this distancing problem, the spent stage needs to move away from the upper stage on its own.

The design chosen consists of a system which uses a hinged arm on the first stage to release the second stage while deploying an air brake. This system solves both problems by not imparting any forces on the second stage while getting out of the way quickly using only drag for second stage motor separation.

Figure 31 displays the sequence of this deployment event. The first stage flight controller sends a signal to pull a pin located at the base of the air brake bay. This pin releases cables tied to the bottom of each air brake arm. The air brake arm is on a hinge with the upper arm section under constant tension from a spring. With the cables released, the arm swings open letting go of the second stage while simultaneously deploying aerodynamic surfaces on the air brake arm. The air brakes induce a large drag force on the first stage, allowing it to slow down while the second stage continues coasting upwards. A small string attached to the second stage helps remove the parachute from the canister located in the center of the air brake bay between the springs. After the second stage ignition, the first stage quickly reaches it apogee and begins to descend. The air brakes also help stabilize the descent of the first stage by passively assisting with orientation. The parachute now is fully open, and the first stage can gently touch down.

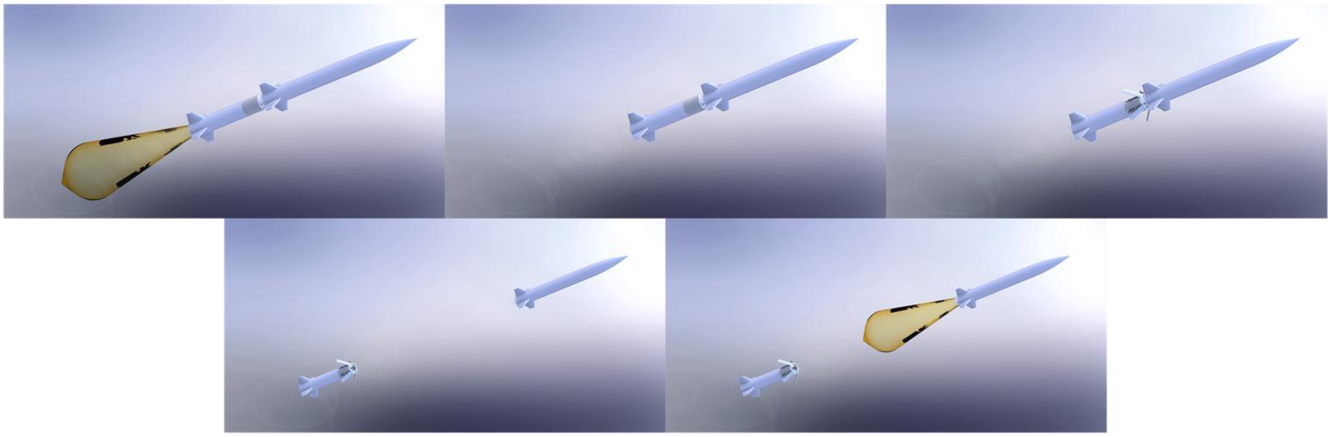


Figure 31 First Stage Separation System Phases

The air brake bay in Figure 32 was modeled in SOLIDWORKS and a prototype was manufactured on Ultimaker 3 and Lulzbot Taz 6 3D printers in PLA, a common plastic-based filament. Parts were fastened together with M3 Alloy Steel Screws and the air brakes themselves are reinforced fiberglass repurposed from the airframe. Under constant tension, the air brake arms are attached to extension springs on the lower section of the air brake bay and fasted to the bay with 1" x 1/2" Mortise-Mount Hinges with nonremovable pins. These hinges and springs were purchased from McMaster Carr, an online supplier.

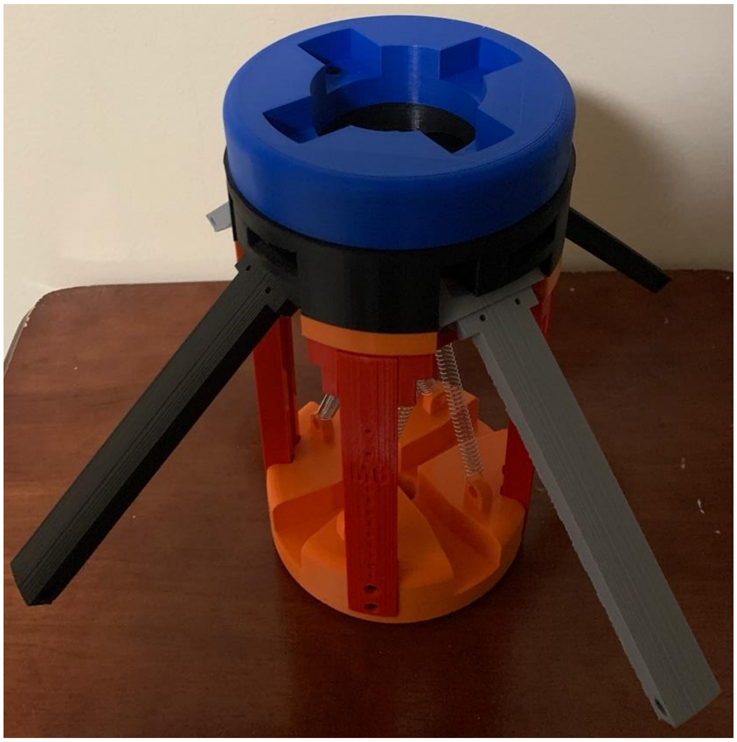


Figure 32 Air Brake Bay Separated and Joined with Air Brakes Deployed

3.2.4 Ignition System

The goal of this subsystem was to create a simplified model for the electric match (E-Match) ignition system described in Section 3.2.4. The E-Match was modeled in COMSOL with a combination of current and heat source simulations. The COMSOL model allowed for the evaluation of several parameters to ultimately minimize the time it takes for a complete ignition of the pyrogen. This model was useful to ensure that the ignition of the model rocket would occur successfully, and it would be completed as efficiently as possible. The other ignition method considered was the arc ignitor.

There are several major design constraints with the arc ignitor system that had to be considered in the design process. The main constraint is that an electric field of 3,000V per millimeter of gap distance is needed to breakdown the resistance of air for the arc to be formed between nodes. Any wires or components within the plume of the rocket motor will be incinerated upon ignition so it is critical that the ignition works as intended to ensure a thorough ignition of the motor grain. As shown in Figure 22 in Section 2.2.1, the exposed electrodes need to be in direct contact with a flammable material for ignition to occur. The team determined that having the point of ignition as deep as possible within the motor grain would help in the ignition of the remainder of the motor. The team also determined that a relatively small power source of three volts would not result in a strong enough electric arc that far away from the electronics housing. Instead, a larger power source of nine volts could be used that would run separately through a circuit configuration for a more complex ignition method. As shown in Figure 33 below, several options were considered to determine the best configuration for the integration within the rocket motor.

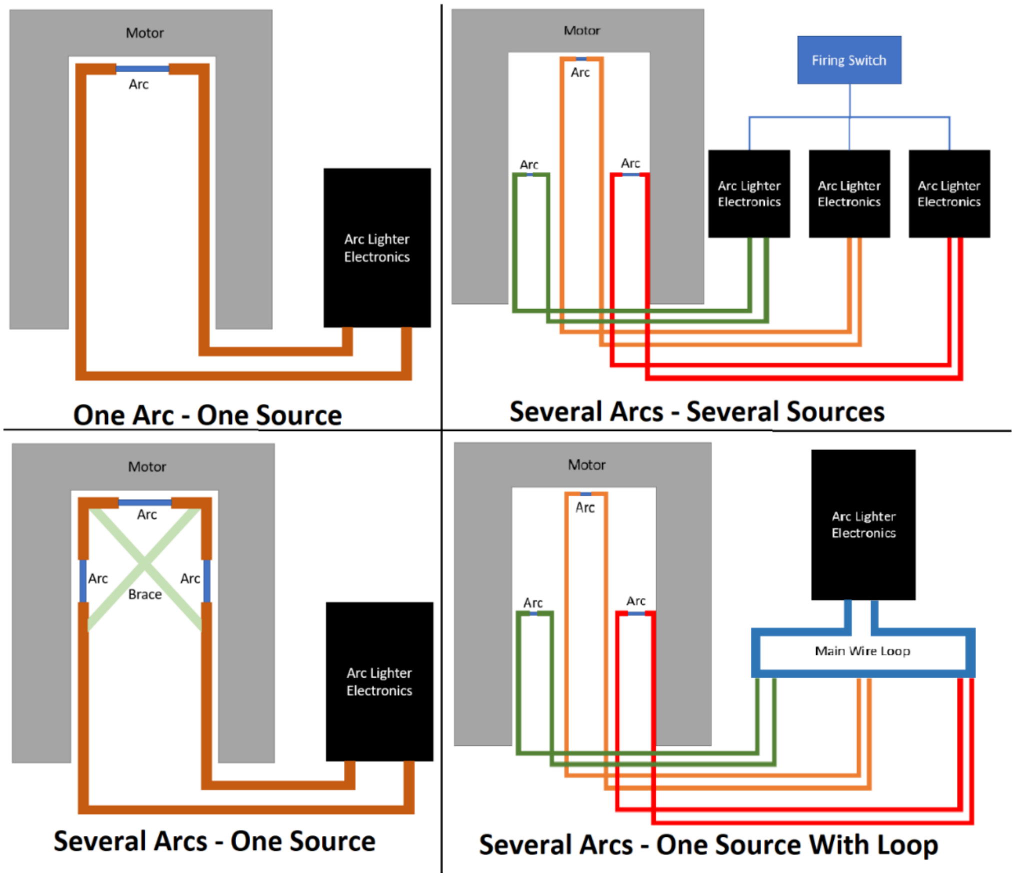


Figure 33 Arc Ignitor Configuration Design Options

These options include arcs formed from one source (top left), in series with one source and flammable braces (bottom left), in parallel with multiple sources (top right), or in parallel with one source and a feedback loop (bottom right). To determine the efficacy of these designs, electrical diagrams would need to be constructed similar to other high voltage arc ignitor diagrams [53]. Then, their respective PSPICE models would show which configuration resulted in a successful production of a singular arc or multiple arcs. This would be determined by the output of probes placed before any gaps. If the observed voltages resulted in an electric field of at least 3,000V per millimeter of gap distance, then theoretically, the configuration should work.

For the E-Match ignitor option, COMSOL was used to create a model that would be used to calculate the time required to achieve full ignition of the pyrogen. The team determined that simulating the transfer of heat within the wire from the current along with the pyrogen would be too computationally intensive and not required. Instead, the approach the team used was to model a spherical ball of pyrogen with a two-dimensional line acting as the wire and singular source of heat. Using the wire's known current, voltage, and resistance, the heat produced by the wire was determined. With the wire as a heat source, the team used COMSOL to determine when the temperature at the outer surface of the pyrogen reached the ignition temperature of the material. In COMSOL, the wire is modeled as copper and the pyrogen is boron due to the unavailability of nichrome for the wire and any other conventional pyrogen choices such as metal oxidizers or intermetallic compounds.

The ideal results of these simulations would be used to determine which designs to ultimately use in the ignition systems of the rocket. The COMSOL model would show the time it took for a full ignition of the pyrogen. It would also show that the relationship between the wire as a heat source from the current passing through it translated correctly to an output temperature.

After a successful simulation, parameters such as wire thickness, material properties, and pyrogen volume could be altered to determine which combination of parameters resulted in the most efficient ignition. By taking all of these results into consideration, the final decisions for an actual arc ignitor system and electric matches could be made for integration into the first and second stage motors respectively. A consistent and reliable ignition of the motors with correct and precise timing is critical to ensuring the intended launch sequence is followed and that the entire mission is successful.

3.3 Flight Dynamic Analysis

The Flight Dynamic Analysis subteam's areas of responsibility included fin design and analysis of the aerodynamic behavior of the rocket. The team completed the analysis tasks which helped guide the subsystem design. In order to create accurate models, the FDA team had to consider the inputs of all other teams, and the impacts. The team had to consider the effect of rocket geometry, motor performance, stage separation speed, mass of the rocket, and the autorotation recovery system when analyzing the performance metrics.

3.3.1 FDA Analysis Task 1

The purpose of Task 1 is to develop a model of the flight location and attitude of a model rocket during powered and unpowered flight. This model evaluates the location and attitude of the rocket using a set of coupled ordinary differential equations. This solution must consider the dynamically updating values of rocket flight, such as the mass, velocity, rotation rates, and attitude, and use these state values to determine the subsequent states. The solution must also incorporate the results from many other tasks from other sections of the MQP, such as the aerodynamic load results from FDA Tasks 2 and 3, the motor performance results from PTSS

Task 1, and the autorotation system model from ARS Tasks 2 and 3. Information from all these tasks is essential to developing an accurate representation of the flight of the rocket.

The objective for this code is simple and clear. The simulation is to provide an accurate model of the behavior of the rocket during its flight. This system will inform us of the maximum velocity experienced during the flight, the apogee, and the distance travelled during the flight, all important performance metrics.

When designing the analysis code for this task, there were many considerations that had to be made. First and foremost, we had to decide the method that would be employed to define the system. In rocket modelling, there are multiple descriptive approaches (i.e. angle sets) that can be used to analyze the coordinate transformations and attitude of the rocket in-flight. The three approaches that were considered for this project were Euler angles, the Direction Cosine Method, and quaternions. All three approaches require the definition of rotation matrices: a matrix for converting from one frame of reference to another by using the attitude data from both frames [54].

All three approaches have their own drawbacks. Euler angle systems suffer from a mathematical singularity as the pitch angle, θ, approaches 90° [54]. This singularity can be very problematic and is best to be avoided by using other methods when possible. The Direction Cosine Method does not have this singularity problem, but it does rely on several constraint equations, and the team decided that a quaternion approach would be more appropriate. The quaternion system does not suffer from any singularities, as it is a quadruple of real numbers and is written as a three-dimensional vector:

$$[Q] = q_0 + iq_1 + jq_2 + kq_3 = (q_0, q_1, q_2, q_3) = (q_0, \boldsymbol{q}) \tag{3}$$

q_0, q_1, q_2, q_3 are real numbers and i, j, k defines the quaternion vector space. The orthogonality of quaternions also requires that

$$q_0{}^2 + q_1{}^2 + q_2{}^2 + q_3{}^2 = 1 \tag{4}$$

These quaternion terms are defined by the attitude angles φ, θ, ψ,

$$q_0 = \cos\left(\frac{\psi}{2}\right)\cos\left(\frac{\theta}{2}\right)\cos\left(\frac{\varphi}{2}\right) - \sin\left(\frac{\psi}{2}\right)\sin\left(\frac{\theta}{2}\right)\sin\left(\frac{\varphi}{2}\right) \tag{5}$$

$$q_1 = \cos\left(\frac{\psi}{2}\right)\sin\left(\frac{\theta}{2}\right)\sin\left(\frac{\varphi}{2}\right) + \sin\left(\frac{\psi}{2}\right)\cos\left(\frac{\theta}{2}\right)\cos\left(\frac{\varphi}{2}\right) \tag{6}$$

$$q_2 = \cos\left(\frac{\psi}{2}\right)\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\varphi}{2}\right) - \sin\left(\frac{\psi}{2}\right)\cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\varphi}{2}\right) \tag{7}$$

$$q_3 = \cos\left(\frac{\psi}{2}\right)\cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\varphi}{2}\right) + \sin\left(\frac{\psi}{2}\right)\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\varphi}{2}\right) \tag{8}$$

Classically, the quaternion values are described as above, using q_0, q_1, q_2, q_3. For this derivation, the terms $A, B, C, D,$ were used, respectively, in their place. These quaternion values can then be used to create a direction cosine matrix (DCM) that can transform between two coordinate systems. For instance, it is possible to convert from the body fixed coordinates into navigational coordinates using:

$$C_b^n = \begin{bmatrix} A^2 + B^2 - C^2 - D^2 & 2(BC - AD) & 2(BD - AC) \\ 2(BC + AD) & A^2 - B^2 + C^2 - D^2 & 2(CD - AB) \\ 2(BD - AC) & 2(CD + AB) & A^2 - B^2 - C^2 + D^2 \end{bmatrix} \tag{9}$$

This conversion allows the simulation to evaluate the state parameters of the rocket with respect to both the body-fixed coordinate system and an observer coordinate system. This is an important part of the simulation, as final displacement values can be observed from the initial observer, while also evaluating the body velocities experienced in the body-fixed frame. Figure 34, retrieved from George Siouris' "Missile Guidance and Control Systems" book, shows the body-fixed directions in a clear way, and Figure 35, shows the relationship between the body fixed frame (directions with subscript “b”) and the earth-fixed frame (subscript “e”) through the rocket’s Euler angles.

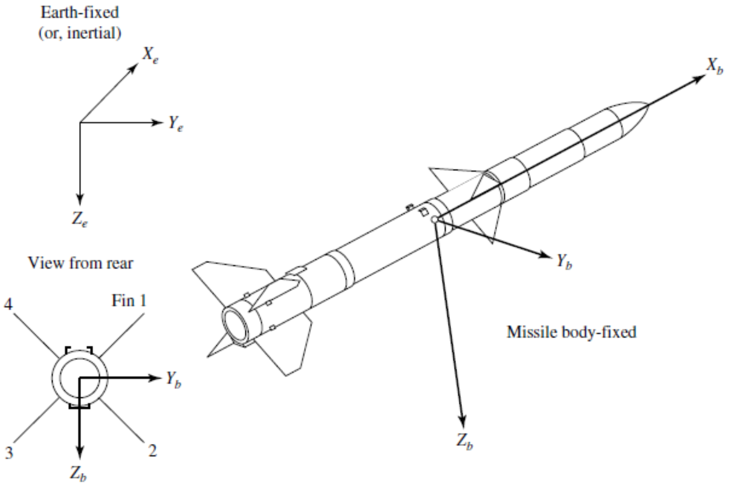


Figure 34 Body Fixed-Frame Definition of a Model Rocket, [54] © Springer, 2004

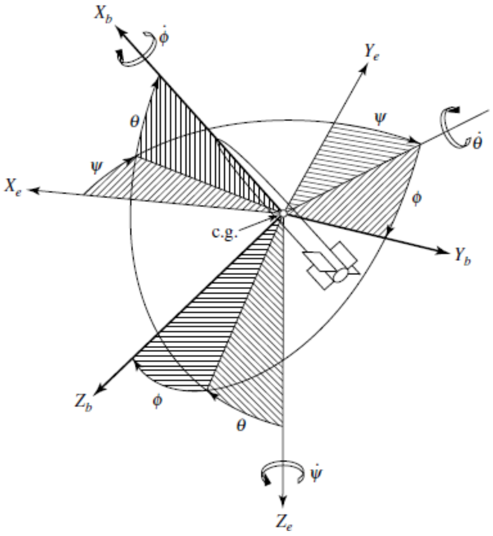


Figure 35 Relationship Between Body-Fixed Axis, Earth-Fixed Axis, and Attitude Angles [54] © Springer, 2004

After deciding upon the quaternion system as the choice for coordinate transformation, the equations of motion for the flight of the rocket were derived. As previously stated, the main parameters that are under investigation are the translational motion and the attitude of the rocket. These two derivations are linked and, for ease of explanation, will be discussed one at a time.

The translational motion of the aircraft is, at its core, predicated on the forces experienced by the aircraft. These forces can be described as a single summation force vector that acts independently in the three axis directions of the craft.

$$F = F_x + F_y + F_z \tag{10}$$

$$F_x = m\left(\frac{du}{dt}\right) \tag{11}$$

$$F_y = m\left(\frac{dv}{dt}\right) \tag{12}$$

$$F_z = m\left(\frac{dw}{dt}\right) \tag{13}$$

$$\left(\frac{dV_m}{dt}\right)_{body} = \frac{F}{m} \tag{14}$$

The value “m” represents instantaneous rocket mass. u, v, and w represent velocities in the body-fixed X, Y, and Z directions, respectively. V_m represents the missile’s linear velocity.

This simple derivation leaves out the main nuance of the problem, and that is: What do the forces in each direction consist of? The net force acting on the rocket consists of the propulsive thrust force, the aerodynamic forces, and the gravitational force. This derivation assumes that the thrust force acts longitudinally along the X-axis, and thus will only be relevant in that direction.

To analyze the behavior of the aerodynamic forces and gravitational forces, though, the full force equation must be considered in the earth-fixed frame.

As the rocket rotates during its flight, the body will experience the same net gravitational force, but the contributions are distributed based on the attitude of the body. This force can be treated as if it is acting on the center of mass, though, as the discrete differences are very minimal. This analysis requires converting from the body fixed frame to the earth-fixed frame using the attitude angles and quaternion scheme described above. Considering the rotation of the rocket leads to the following expressions.

$$\sum \Delta F_x = m(\dot{u} + wQ - vR) \tag{15}$$

$$\sum \Delta F_y = \ m(\dot{\mathrm{v}} \ + uR - \mathrm{wP}) \tag{16}$$

$$\sum \Delta F_z = m(\dot{w} + vP - uQ) \tag{17}$$

P, Q, R represent the roll, pitch, and yaw rates, respectively. $\dot{\mathrm{u}}, \dot{\mathrm{v}}, \dot{\mathrm{w}},$ represent the time derivative of u, v, w. I.e., the acceleration in the X, Y, Z direction, respectively.

The forces here are related to not just the translational motion, but also to the rotation and attitude of the rocket coupled with its velocity, affecting the behavior. Another way to describe this system with respect to the translational motion directly is through:

$$\dot{u} = \sum \frac{\Delta F_x}{m} + vR - wQ \tag{18}$$

$$\dot{v} = \sum \frac{\Delta F_y}{m} + wP - uR \tag{19}$$

$$\dot{u} = \sum \frac{\Delta F_x}{m} + uQ - vP \tag{20}$$

This system of differential equations can be solved either solely through the translational mode, or in unison with the attitude mode. If the values for P, Q, and R are set to zero, these terms are eliminated, and the only consideration becomes the direct force terms. Solving for the translational mode allows for simplified calculation as well as an estimate of the rocket's behavior if attitude was held constant. It also establishes a good baseline for reasonable values for maximum velocity, apogee, and downfield range.

With the translational motion being defined through the forces, the team next considered consider how the attitude of the craft is affected. This analysis assumes both a rigid body and symmetry along the body fixed X-Z plane. This means the product of inertia, $I_{XZ} = 0$. With these assumptions, the angular momentum equations for the rocket are defined as [54]

$$\Delta L \ = \dot{\mathrm{P}} I_x \ + \ QR(I_z - I_y) \tag{21}$$

$$\Delta M \ = \dot{\mathrm{Q}} I_y \ + \ PR(I_x - I_z) \tag{22}$$

$$\Delta N \ = \dot{\mathrm{R}} I_z \ + \ PQ(I_y - I_x) \tag{23}$$

Equations 21-23 represent the angular momentum equations. L, M, N are the roll, pitch, and yaw moments, respectively. P, Q, R are the roll, pitch, and yaw acceleration rates, respectively. I_x, I_y, I_z are the moments of inertia in the x, y, z planes, respectively. After re-writing the equations to solve for the roll, pitch, and yaw accelerations:

$$\frac{dP}{dt} = QR\left(\frac{I_y - I_z}{I_x}\right) + \frac{L}{I_x} \tag{24}$$

$$\frac{dQ}{dt} = PR\left(\frac{I_z - I_x}{I_y}\right) + \frac{M}{I_y} \tag{25}$$

$$\frac{dR}{dt} = PQ\left(\frac{I_x - I_y}{I_z}\right) + \frac{N}{I_z} \tag{26}$$

These differential equations are important for the upcoming analysis and solution of the attitude system. Further, the roll, pitch, and yaw rates relate to the Euler angle rates through:

$$\frac{d\psi}{dt} = \frac{Q\ sin\varphi\ +\ R\ cos\varphi}{cos\theta} \tag{27}$$

$$\frac{d\varphi}{dt} = P\ +\ \frac{d\psi}{dt}\ sin\theta \quad OR \quad \frac{d\varphi}{dt}\ =\ P\ +\ \frac{Q\ sin\varphi\ +\ R\ cos\varphi}{cos\theta}\ sin\theta \tag{28}$$

$$\frac{d\theta}{dt} = Q\ cos\varphi\ -\ R\ sin\varphi \tag{29}$$

These equations define the behavior of the 6-degree of freedom (6DOF) rocket model. Through evaluating these equations, we can solve for the location and attitude of the rocket at any point during its flight. It should be noted that there are four distinct phases of the rockets flight: Powered flight of Stage 1, powered flight of Stage 2, unpowered flight, and the descent stage. During the first two phases, there is a thrust component generated by the motor of the rocket. During unpowered flight, there is no thrust component and motion is adjusted solely based on the contribution of the aerodynamic forces and gravity. During the descent stage, the autorotation device acts to control the descent see Section 3.1.4.

These equations must be implemented in a computational solver. We decided to utilize MATLAB to solve this problem. The methodology we employed was to define the ordinary differential equations (as shown above) and use the ode45 solver to find the solutions. Ode45 is a solver that solves non-stiff differential equations. The system works by integrating the system of differential equations through a defined timespan and using initial conditions. The solver must specify the following parameters: mass, thrust profile, drag profile, initial velocity, initial orientation, and moments of inertia. The initial condition consists of the initial mass, velocity, and orientation, while the thrust profile, drag profile, and moments are discretely represented in lookup tables. These are essential to defining the flight behavior and solving the rocket's response to flight conditions.

The team's solution involved creating two ode45 solvers, the first solving for the translational motion of the rocket, while the second solved for the attitude. The computational process followed a repeating loop, described in Figure 36. It uses only the equations mentioned above and follows the steps in MATLAB.

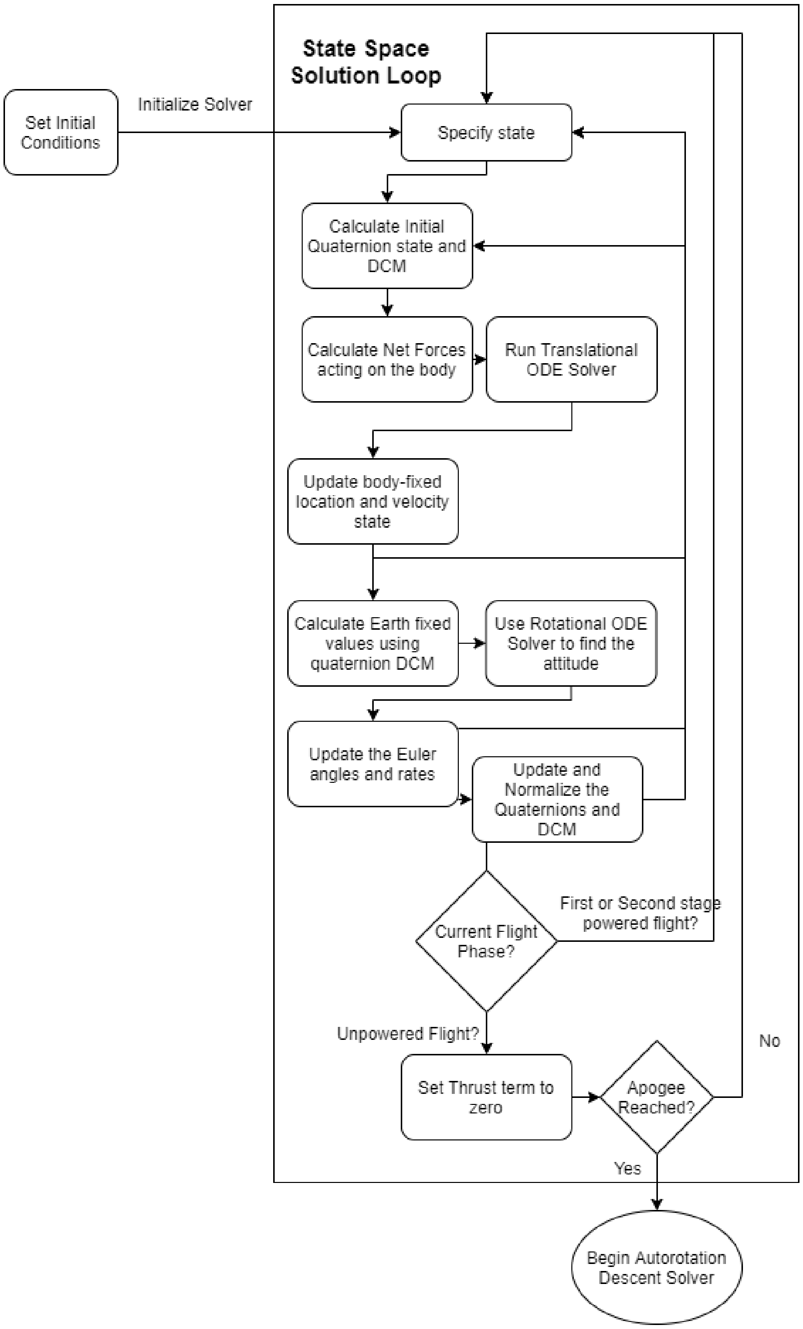

State Space
Solution Loop
Set Initial Conditions
Initialize Solver
Specify state
Calculate Initial Quaternion state and DCM
Calculate Net Forces acting on the body
Run Translational ODE Solver
Update body-fixed location and velocity state
Calculate Earth fixed values using quaternion DCM
Use Rotational ODE Solver to find the attitude
Update the Euler angles and rates
Update and Normalize the Quaternions and DCM
Current Flight Phase?
First or Second stage powered flight?
Unpowered Flight?
Set Thrust term to zero
Apogee Reached?
No
Yes
Begin Autorotation Descent Solver

Figure 36 State Space ODE Solver

The ideal results of this task are the performance metrics of flight range, time of flight, max velocity, potential landing region, general flight stability, and reaction to wind perturbances, among others. The team wanted to know how effective the innovative rocket design was in achieving these metrics. Normally, this process would involve the construction of both an innovative and a baseline model, but due to health concerns the team did not complete in-person construction. For that reason, the model is even more important, as it may be the only way to evaluate the performance of the team's design choices.

An ideal model also allows for perturbation analysis, for example a gust of wind. This gust would act by exerting a force in a random direction for a random amount of time, and the model's reaction to the perturbation indicates the stability of the rocket and the degree to which it can be altered by simple disruptions. The model would also be useful in determining the upper limit of wind that can be experienced by the rocket without causing too much disruption.

Simulation results can also be valuable for improving the rocket design. If the team were to find that one design choice had an adverse effect on rocket performance, they could modify that parameter to improve the result. Further, the team could refine the good choices and find designs that produce certain maximum values for performance metrics. For example, optimizing the launch angle to find the maximum range, or launching such that a velocity threshold is not exceeded.

The team can also use the results to determine the landing probability and generate a probabilistic region surrounding the projected location. This would allow the team to create a "safety plot" of landing locations for the rocket. The team can also use the results to compare our innovative design with a baseline rocket to determine the efficacy of our changes. The team would also be able to directly evaluate the performance of the spin stabilization fins and determine if they are effective for this rocket.

3.3.2 FDA Task 2

The purpose of Analysis Task 2 was to estimate the aerodynamic forces and moments applied on the vehicle as a function of velocity and vehicle attitude. This allowed the team to be able to provide load estimates that were then used as part of the structural stress analysis. Additionally, it allowed the team to have values for forces and moments that the team then used in the dynamic model from FDA Analysis Task 1.

The team used Ansys Fluent's pressure-based solver, which uses mass and momentum conservation to solve for incompressible flows. In Ansys there are two kinds of pressure-based solvers, one that uses a segregated algorithm and one that uses a coupled algorithm, both can be seen below in Figure 37.

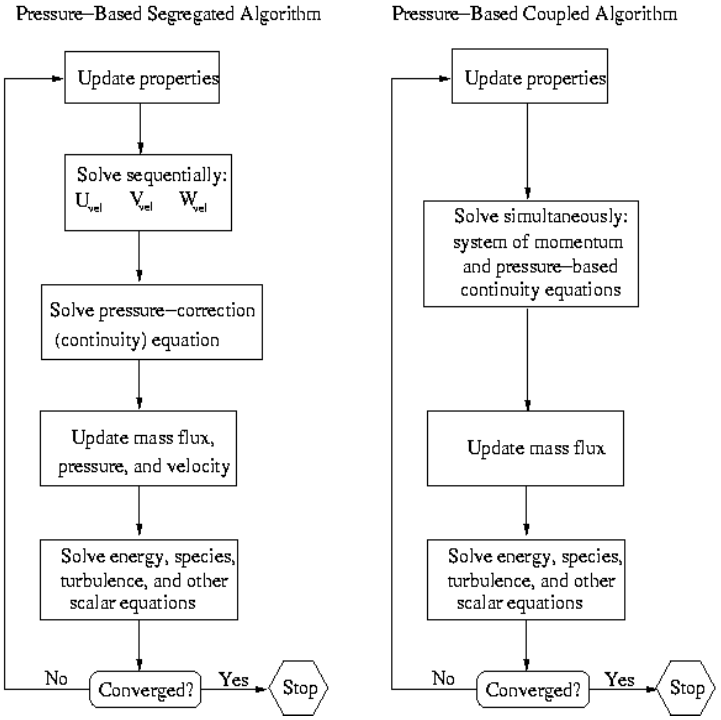

Pressure–Based Segregated Algorithm
Pressure–Based Coupled Algorithm
Update properties
Solve sequentially: U_{vel} V_{vel} W_{vel}
Solve pressure–correction (continuity) equation
Update mass flux, pressure, and velocity
Solve energy, species, turbulence, and other scalar equations
No
Converged?
Yes
Stop
Update properties
Solve simultaneously: system of momentum and pressure–based continuity equations
Update mass flux
Solve energy, species, turbulence, and other scalar equations
No
Converged?
Yes
Stop

Figure 37 Pressure-Based Segregated Algorithm vs Coupled Algorithm © Ansys, 2009

Using the coupled algorithm, the results are more accurate, and the solver takes less iterations to solve, however it requires much more computer memory. Because of its more accurate results, the coupled pressure-based algorithm was used for the solver. The turbulence model used was the SST (shear-stress transport) k-ω turbulence model, which because it combines the standard k-ω and the k-ε models, allows it to better simulate boundary layer separations and makes it a reliable and accurate choice for our model [55]. The boundary layers used for this analysis task can be seen below in Figure 38.

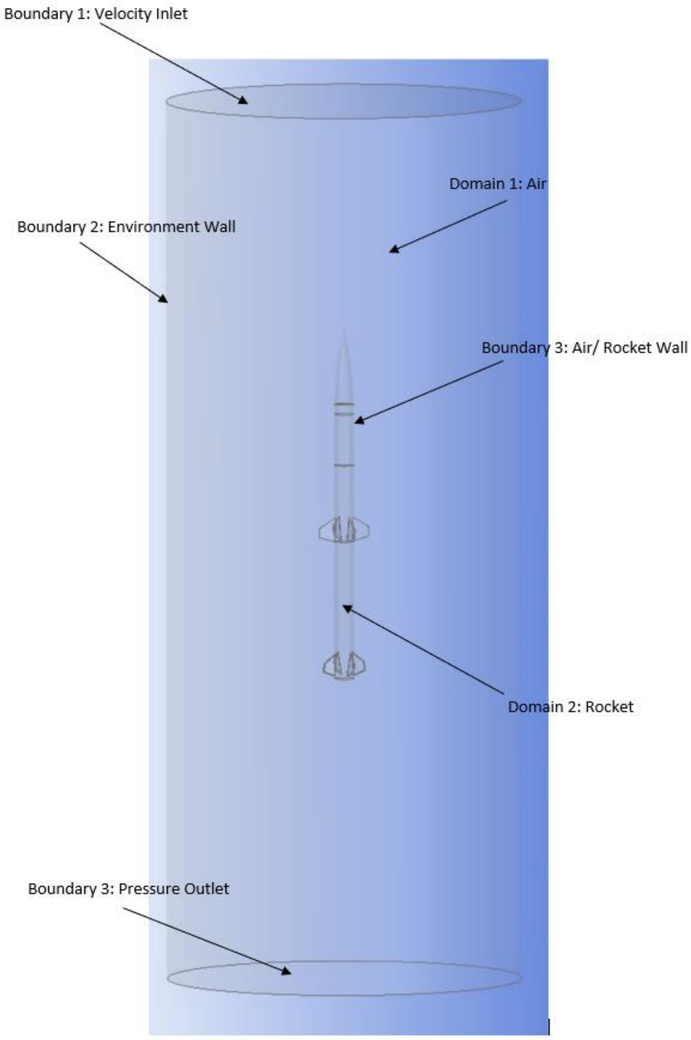

Boundary 1: Velocity Inlet
Domain 1: Air
Boundary 2: Environment Wall
Boundary 3: Air/ Rocket Wall
Domain 2: Rocket
Boundary 3: Pressure Outlet

Figure 38 Airframe in Fluent

The two domains used are the flow environment and the rocket body, while the enclosure wall, pressure outlet, velocity inlet, and the boundary between the flow environment and the rocket body define the boundary conditions. To simulate a change in the rocket's angle of attack, the environment wall domain was used as an inlet. Both the velocity inlet and the environment wall were able to produce velocity at a variety of angles in order for the rocket to be analyzed at different angles of attack. So, while the rocket's model itself is not rotating, the direction of the flow that changes. In addition to the changing angles of attack, the inlet velocities were also changed, to provide estimates of the aerodynamic forces and loads at different points throughout the rocket's trajectory.

Integration using the solver provided the team with detailed drag and moment profiles for different rocket orientations. One possible way the team considered integrating the results into our dynamic model was to find a generalized coefficient of drag term for a single attitude and utilize that for different velocities, but with the Ansys Fluent solver different trials were run for these varying parameters instead. This information can be used to generate a lookup table for FDA Task 1 to accurately model the drag and moment behavior of the rocket throughout the flight. These results allow for improvement in the simulation of the flight, as well as improvement of design parameters. For example, if a certain design is causing too much drag, or is not stable due to the effect of the moments, the design can be modified to make a more stable rocket. The results of this task are explored in greater detail in Section 4.3.2.

3.3.3 FDA Analysis Task 3

The goal of FDA Analysis Task 3 was to evaluate the aerodynamic forces and moments applied by the stabilizing fins. The results from this task were then used to make the dynamic model in FDA Analysis Task 1 more accurate. After the results from each case were put into the dynamic model, the team was able to see how each fin design affected the rocket's performance. Additionally, the spin induced from the moment applied to the airframe by each fin design was evaluated. By comparing the rocket performance, induced spins, and resulting stability for each fin design, the ideal fin for the rocket was found to be the fin with the NACA 4414 airfoil. The results of this task are discussed more in Section 4.3.3.

The first part of this task was to design the fins. The fin design process started in XFLR5, an analysis tool for airfoils, in which the airfoils for all of the fins were designed. For the symmetric airfoil, which was evaluated at different areas of attack as part of this Analysis Task, a NACA 0414 airfoil was selected and the fins were evaluated at 1°, 3°, and 5°. For the cambered airfoils, NACA 2414, NACA 4414, and NACA 6414 airfoils were selected. These were selected because the only difference in these airfoils is their camber percentage, meaning their maximum camber as percentage of the chord. The NACA 0414 has a 0% camber, meaning it is symmetric, the NACA 2414 has a 2% camber, the NACA 4414 has a 4% camber, and the NACA 6414 has a 6% camber. The reason the team chose to only change one aspect of the fins, either the cant or the camber, for each respective case is because the team wanted as few independent variables as possible. By doing this, the team was better able to understand the relationships between the differences in the fins and their effect on the results.

Once the airfoils designs were selected, the size and shape for the rest of the fin was designed as well. OpenRocket was used to estimate what fin dimensions would be needed in order have a center of pressure behind the vehicle’s center of mass and keep the rocket stable. From this baseline, XFLR5’s Plane and Wing designer was used to create a trapezoidal fin with a 6.5 inch root chord length, a 2.5 inch tip chord length, a 3.5 inch span, and a 2.5 inch leading edge offset from the root to the tip of the fin. Figure 39, depicts the fin shape in XFLR5’s fin editor.

With the fins designed in XFLR5, they then needed to be created as 3D models in SOLIDWORKS. The 2D coordinates of the airfoils in XFLR5 were saved as a .dat file which then was opened in Excel. The team then made a program in Excel which turned the base 2D coordinates of the airfoils into 3D coordinates, based on the desired root and tip cord lengths, the fin span, and the root to tip offset. With the 3D coordinates made in Excel, they were then transferred over to SOLIDWORKS via the "Curve through XYZ Points" feature. The "Swept Boss/Base" feature was used to connect the root airfoil and the tip airfoil together, finishing the 3D model.

With the 3D model depicted in Figure 40 created, the aerodynamic forces and moments were then evaluated. Using Ansys Fluent, all of the cases were simulated using the coupled pressure-based solver with the k- ω turbulence model. These were chosen because the segregated pressure-based solver is ideal for subsonic, incompressible flows, and the k-ω viscous model is the best choice for simulating aerodynamic flows, due to its ability to accurately simulate the boundary layer separations [55]. The simulation domain boundaries used for the fin analysis can be seen in Figure 41.

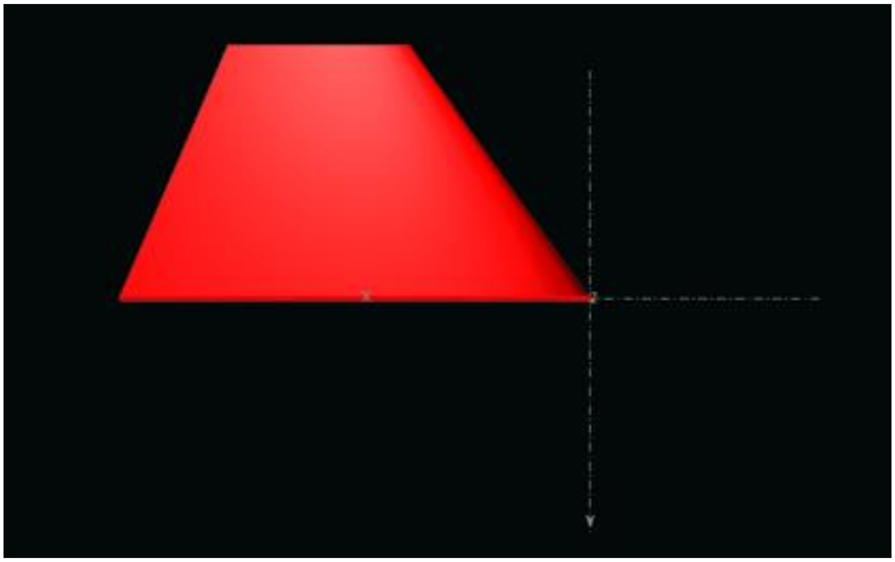


Figure 39 Fin Design in XFLR5

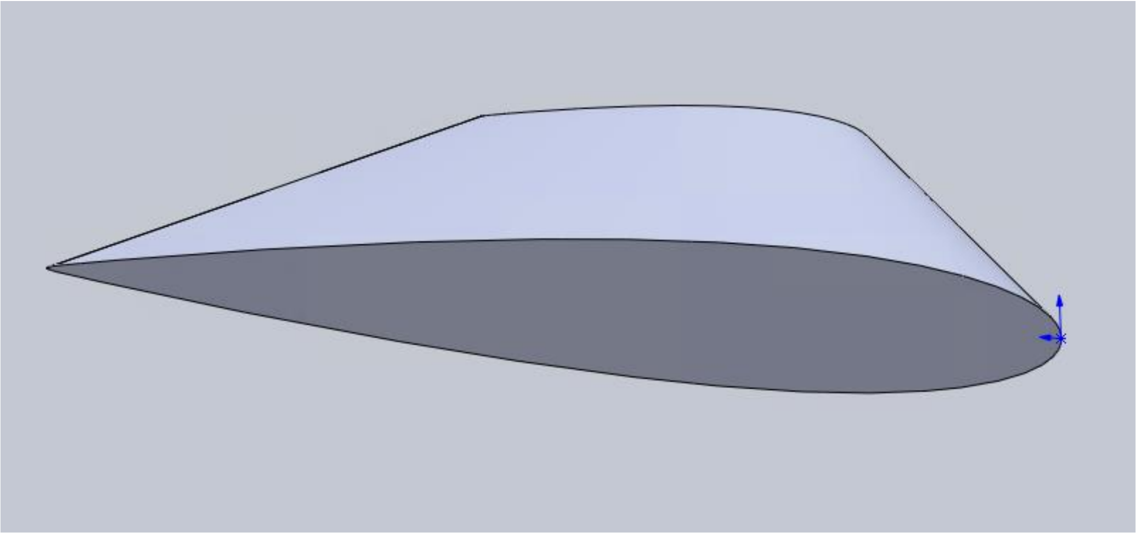


Figure 40 Isometric View of 2414 Fin

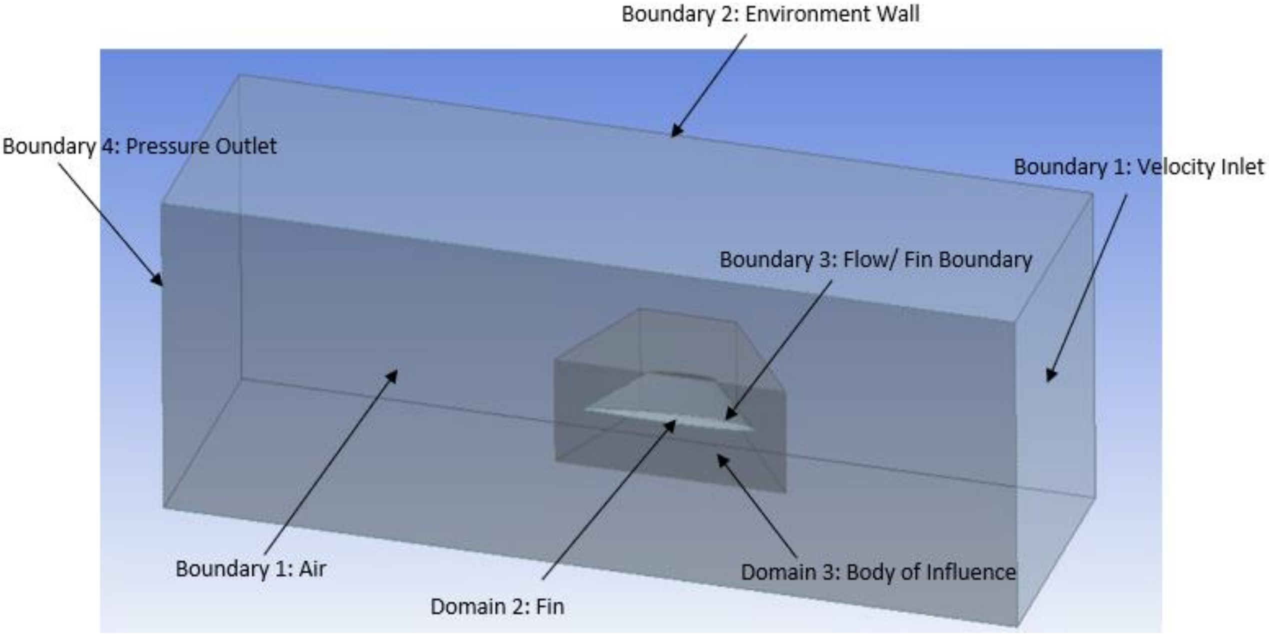

Boundary 2: Environment Wall
Boundary 4: Pressure Outlet
Boundary 1: Velocity Inlet
Boundary 3: Flow/ Fin Boundary
Boundary 1: Air
Domain 3: Body of Influence
Domain 2: Fin

Figure 41 2414 Fin in Design Modeler

Notice that the inner domain in Figure 42, called the Body of Influence was used to allow the mesh size around the fin to be much smaller compared to the mesh in the rest of the domain, allowing the team to obtain accurate results. This was necessary, because if the small mesh size used around the fin was used for the whole environment, it would have required too much of the computer's memory and taken too long to compute.

Once the meshing was completed, the model was solved using the models described in Section 3.3.3. Each fin was analyzed at different velocities, ranging from 35m/s to 130m/s. Results for this task included calculation of the lift and drag produced by each fin as well as the coefficients of lift and drag. From there, these coefficients were used in the dynamic model, to see how the rocket performs with each fin. Additionally, from the moments applied on the airframe by the fin, the spin rate of the vehicle was found, via the method seen in Figure 43. These spin rates are discussed in Section 4.3.3

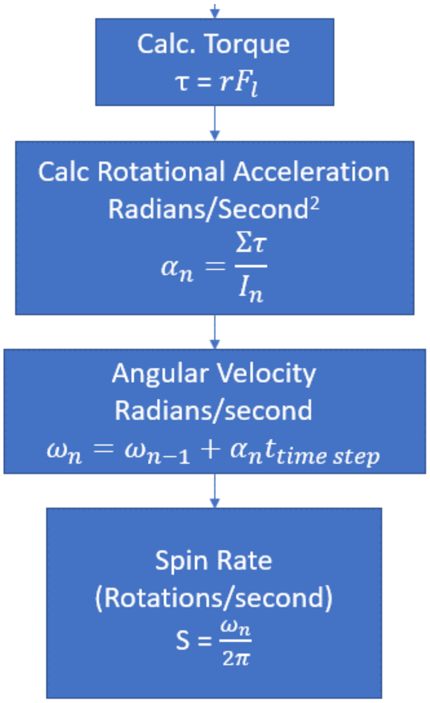

Calc. Torque
$\tau = rF_l$
Calc Rotational Acceleration
Radians/Second^2
$\alpha_n = \frac{\Sigma\tau}{I_n}$
Angular Velocity
Radians/second
$\omega_n = \omega_{n-1} + \alpha_n t_{time\ step}$
Spin Rate
(Rotations/second)
$S = \frac{\omega_n}{2\pi}$

Figure 42 Force to Spin Rate Flowchart

With the vehicle's spin rate found, the team then was able to compare the spins generated by each of the different fins. The ideal fin for the rocket should be the fin that that created the highest spin rate on the rocket, and had a lowest drag. After considering all of these factors, the fin with the 4414 airfoil was found to be the best out of all the fins considered for this project. Further explanation of these results can be found in Section 4.3.3.

4 Analysis

A series of analysis tasks were identified for the three subteams to provide an in-depth understanding of the various components within the rocket and the forces they experience. This Section presents the approach, findings, and results of each subteam.

4.1 Airframe and Recovery System

The ARS subteam was tasked with four main analysis tasks. These tasks centered around an in-depth analysis of the overall stress distribution, aerodynamic loads, and model of the rocket throughout the events of its descent under autorotation. Each team member assumed leadership of at least one analysis task and completed the majority of the work. By completing these analysis tasks, the ARS subteam was able to apply theoretical knowledge from the literature review, as well as training in Ansys Structural and Ansys Fluent to better understand the structural and physical properties of the rocket.

4.1.1 ARS Analysis Task 1: Airframe Stress Distribution

The first analysis task challenged the ARS subteam to determine the stress distribution on various critical locations of the rocket. These critical locations consisted of sections of the rocket that experienced high levels of stress due to peak forces and pressure loads. Testing locations were also selected based on their impact on the integrity and rigidity of the airframe. These critical high areas of stress can be found where black powder charges are used to separate various sections of the vehicle. Other critical locations identified by the team were the Stage one and two motor tubes where the motor ignites and thus produces an upward force on the centering rings.

The team used Ansys Workbench and preformed a transient structural analysis to determine the stress distribution. This software allowed the team to solve complex finite element analysis (FEA) problems to simulate stresses and deformations of the critical locations throughout the rocket. Understanding FEA was important stress or work as intended. When a part is exposed to various force and pressure loads it undergoes deformation. This deformation is classified as either elastic or plastic. Elastic deformation is non-permanent deformation and occurs at relatively low levels of force and pressure. However, when a part is being exposed to excessive forces and pressure loads, its applied stress may exceed this elastic region causing what is known as plastic deformation. Plastic deformation is permanent and means the part will no longer be able to flex and return to its original shape. This occurs at what is known as the yield stress or the point at which elastic deformation becomes plastic deformation [56]. The yield stress is also commonly known as the tensile strength for a given material. In addition, the young's modulus represents the relationship between the applied stress, σ and the induced strain, ε. This is given by the following formula where E represents the material's modulus of elasticity [56].

$$E = \frac{\sigma}{\varepsilon} \tag{30}$$

Understanding how a part goes from elastic deformation to plastic deformation, and finally to yielding was important for the ARS team to understand the analysis being performed by the Ansys software. The team used Ansys structural, and meshed each individual part using FEA and then solved for the Von Mises Stress. The Von Mises Stress is used to determine if a part will yield under a certain force or pressure. When this result is selected in Ansys, the program will determine the maximum Von Mises Stress experienced by that part. This result can then be compared to the maximum tensile strength of the material being used to determine if the part is able to withstand the physical loads and pressures applied to it.

The first two components the team analyzed were the two motor tubes for each stage of the rocket. This is because the centering rings on the motor tube experience a significant upward thrust from the motor. It is essential that the centering ring can withstand this upward force, otherwise the entire motor would blow through the center of the rocket. Figure 44 shows the first stage motor tube after it was loaded into Ansys. Each quarter inch centering ring is made of plywood, and the motor tube is made of Blue Tube 2.0. Thus, these materials were assigned to their respective parts. Once materials were assigned, the part was then meshed in preparation for analysis.

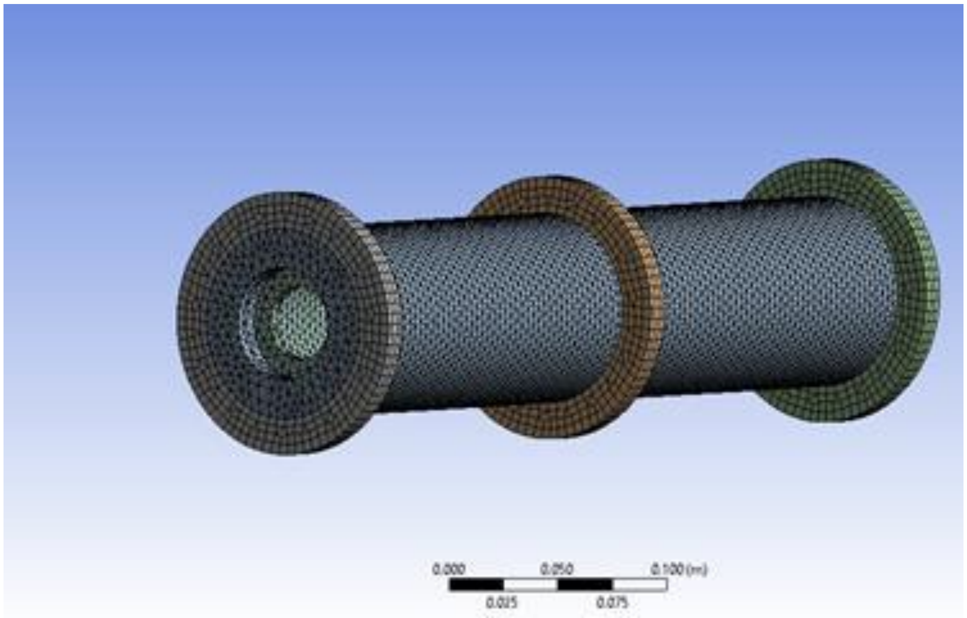


Figure 43 Meshed Motor Tube 1

After meshing, the team added fixed supports on the edge of each centering ring to represent the part being epoxied into place and an upward maximum thrust of 1674 newtons. This upward maximum thrust was determined by the PTSS team to be the maximum thrust imparted on the motor mount by the motor for Stage 1. These conditions can be seen in Figure 44.

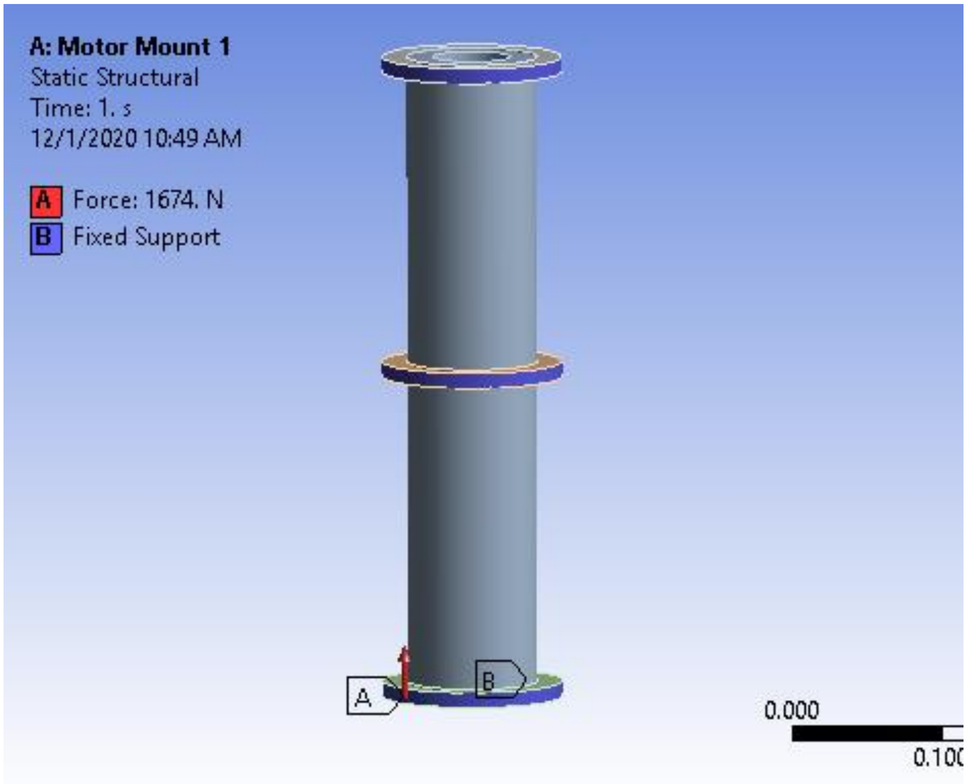


Figure 44 Boundary Conditions for Motor Tube 1

Once the boundary conditions were input Ansys was able to solve for the Von Mises Stress. Figure 45 shows the results of the Von Mises Stress on the Stage 1 motor tube one. Areas in blue represent regions where the stress is low, whereas regions in red represent areas of high stress concentration. In addition, Ansys displays an exaggerated deformation of the part. This deformation can be seen by the curvature of the part in Figure 45 however, the induced strain is not actually significant enough to cause permanent plastic deformation. Thus, the part remains in the region of elastic deformation and will return to its original shape.

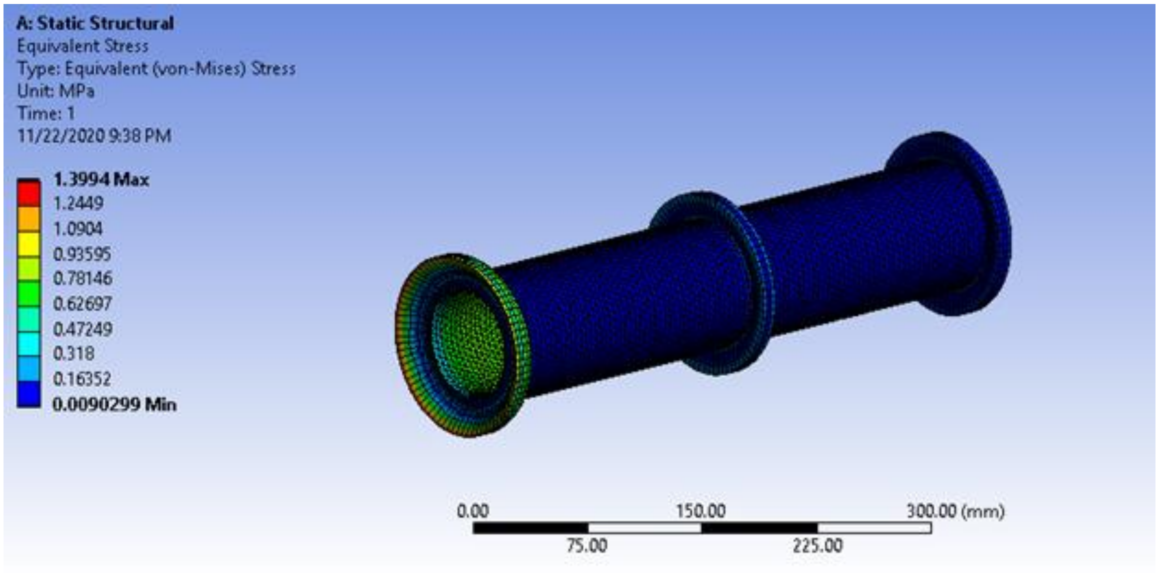


Figure 45 Von Mises Stress on Motor Tube 1

The part experienced a maximum Von Mises Stress of 1.3994 MPa. This is less than the maximum tensile strength of plywood and epoxy which are 27.6 MPa and 35.2 MPa respectively [57], [58]. Thus, this simulation verifies that the part should perform successfully, as the maximum Von Mises Stress of the motor tube does not exceed the maximum tensile strength of either plywood or epoxy.

Figure 46 shows the second stage motor tube after it was loaded into Ansys. Similar to the first motor tube, each quarter inch centering ring is made of plywood and the motor tube is made of Blue Tube 2.0. Thus, these materials were assigned to their respective parts and meshed in preparation for analysis.

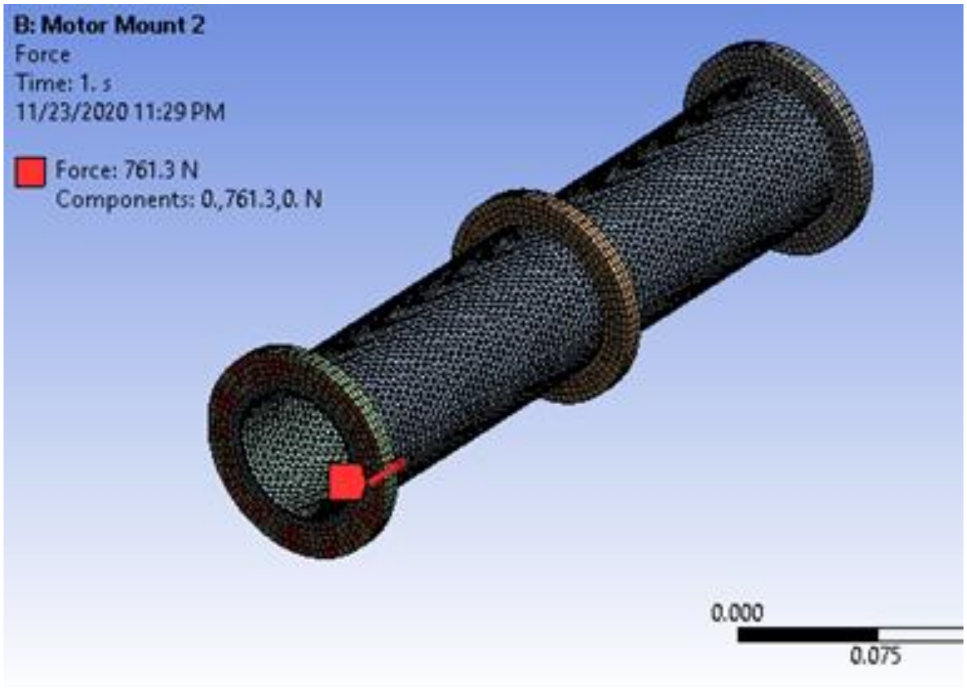


Figure 46 Meshed Motor Tube 2

Once the part was meshed, the team added fixed supports on the edge of each centering ring to represent the part being epoxied into place and an upward maximum thrust of 761.3 newtons. This upward maximum thrust was determined by the PTSS team to be the maximum thrust imparted on the motor mount by the motor for Stage 2. Figure 47 shows the boundary conditions for the Stage 2 motor mount.

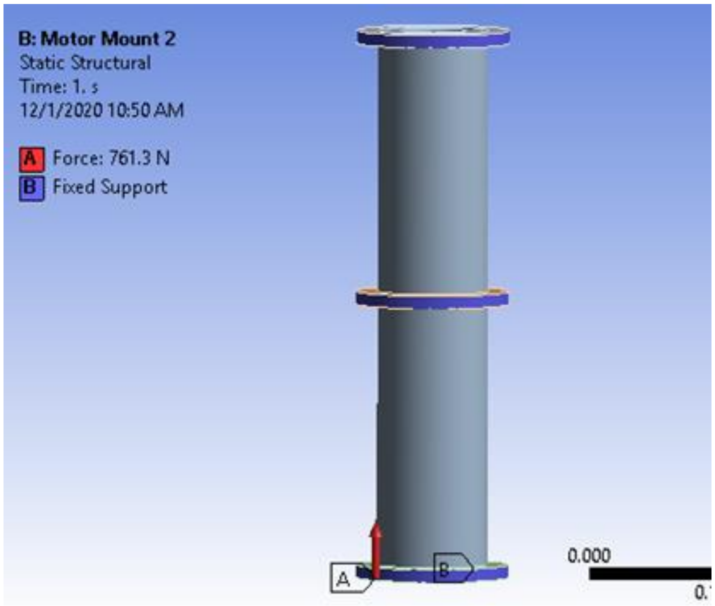


Figure 47 Boundary Conditions for Motor Tube 2

Once the boundary conditions were specified, Ansys was able to solve for the Von Mises Stress. Figure 48 shows the results of the Von Mises Stress on motor tube 2.

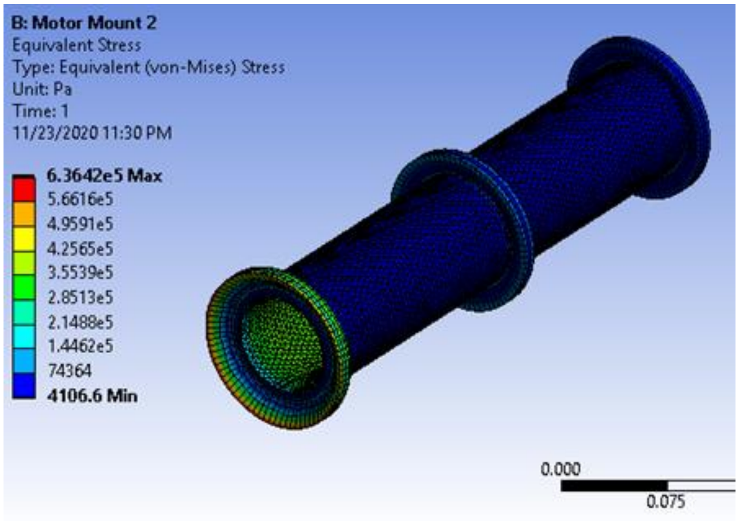


Figure 48 Von Mises Stress on Motor Tube 2

This component experienced a maximum Von Mises Stress of 636.42 KPa, or 0.636 MPa. This is less than the maximum tensile strength of plywood and epoxy which are 27.6 MPa and 35.2 MPa respectively [57], [58]. Thus, because the maximum Von Mises Stress of the motor tube does not exceed the maximum tensile strength of plywood or epoxy, the part should hold successfully.

After performing structural analysis on the motor tubes, the team analyzed two different sections of airframe that make up the rocket. The team only analyzed the astronaut capsule and upper airframe as they were the only two sections that will experience a black powder charge to separate. The lower airframe or first stage will separate via an airbrake system, and thus does not experience a large sudden force. In order to analyze the peak pressure experienced by the airframe, the ARS subteam first determined the maximum force needed from the black powder charge for each event. The black powder charges must break the shear pins and allow the nose cone and astronaut capsule to separate in one case, and the astronaut capsule and upper airframe to separate in the second case.

The ARS team compared, through structural analysis, two different types of shear pins for the rocket. The first option featured small nylon shear pins that break at a minimum force of 285.75 newtons, and the second option features large nylon shear pins that break at a force of 530.2 newtons [59]. Option one and two can be seen in Figures 50 and 51 respectively.

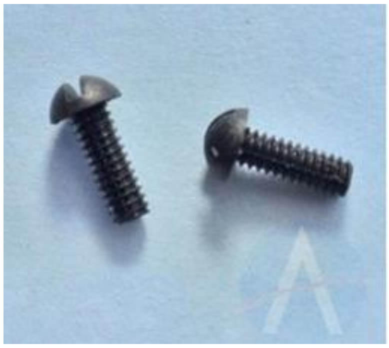
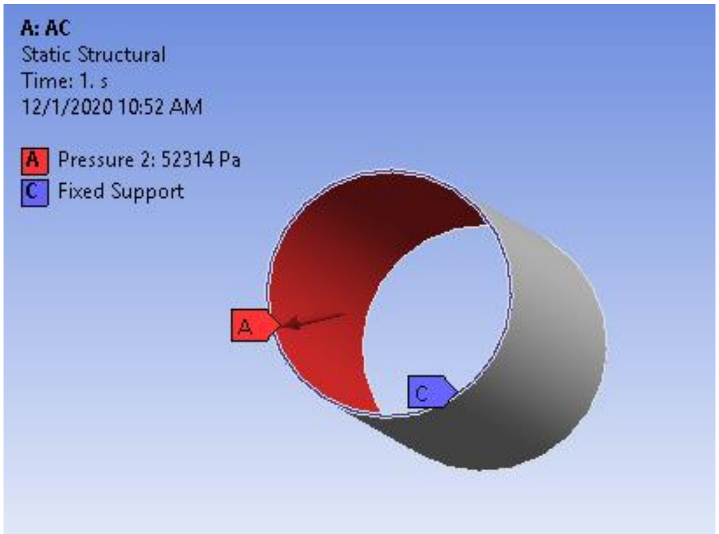


Figure 49 Small Nylon Shear Pins © 2019 Apogee Components, Inc. All Rights

Figure 50 Large Nylon Shear Pins © 2019 Apogee Components, Inc. All Rights

Once this force threshold was determined the team was able to determine the pressure experienced by the airframe imparted by the black powder charge. The force in this case would be the minimum force needed to break the shear pins and the area would be the cross-sectional area of the airframe tube.

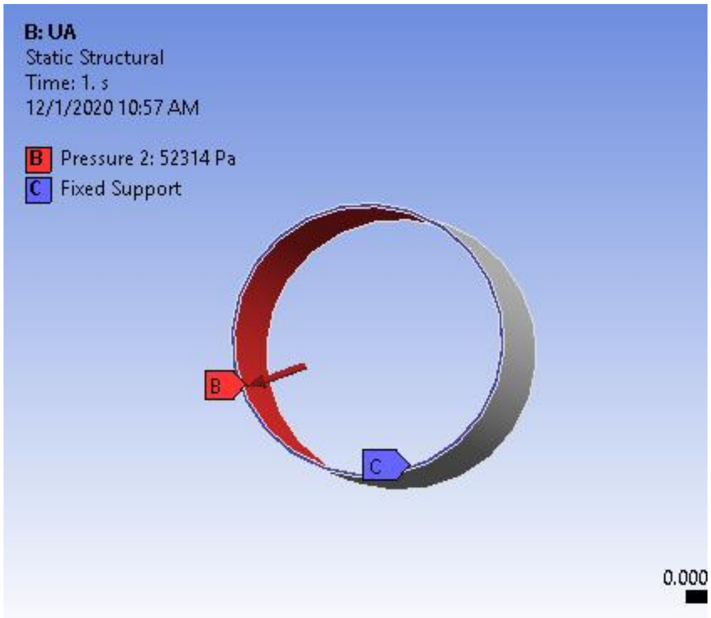
The team determined that the small nylon shear pins needed a minimum pressure of 28,192 Pa to shear. According to an online calculator, this correlates to a black powder charge of approximately 0.66 grams and is below the maximum recommended pressure when using black powder of 103,421 Pa [60]. In turn, the large nylon shear pins needed a minimum force of 530.2 newtons and thus a pressure of 52,314 Pa [61]. This is also below the maximum recommended pressure and correlates to a black powder charge of approximately 1.23 grams [60]. Figures 52 and 53 show the boundary conditions set in Ansys for the astronaut capsule and upper airframe, respectively.



A: AC
Static Structural
Time: 1. s
12/1/2020 10:52 AM

A Pressure 2: 52314 Pa
C Fixed Support

Figure 51 Boundary Conditions Astronaut Capsule



B: UA
Static Structural
Time: 1. s
12/1/2020 10:57 AM

B Pressure 2: 52314 Pa
C Fixed Support

Figure 52 Boundary Conditions Upper Airframe

Once the boundary conditions were set in Ansys, the team was able to solve the model for the Von Mises stress to determine whether the airframe would be able to withstand the force. The structure must be able to absorb the force needed to separate 3 shear pins and ensure the safe deployment of the parachute and autorotation systems. Figures 54 and 55 show the results of the analysis for the two options on the astronaut capsule and Figures 56 and 57 show the results of the analysis on the upper airframe.

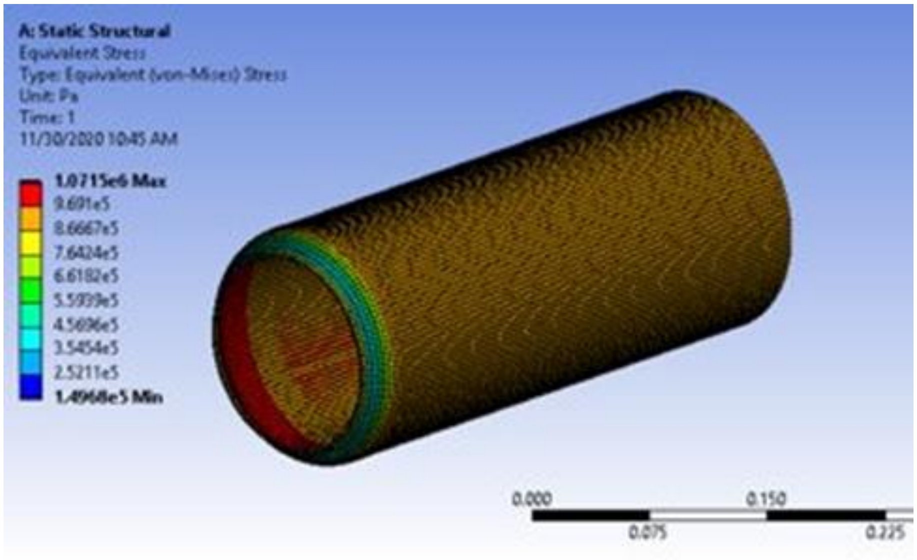


Figure 53 Von Mises Stress on Astronaut Capsule w/ Small Nylon Shear Pins

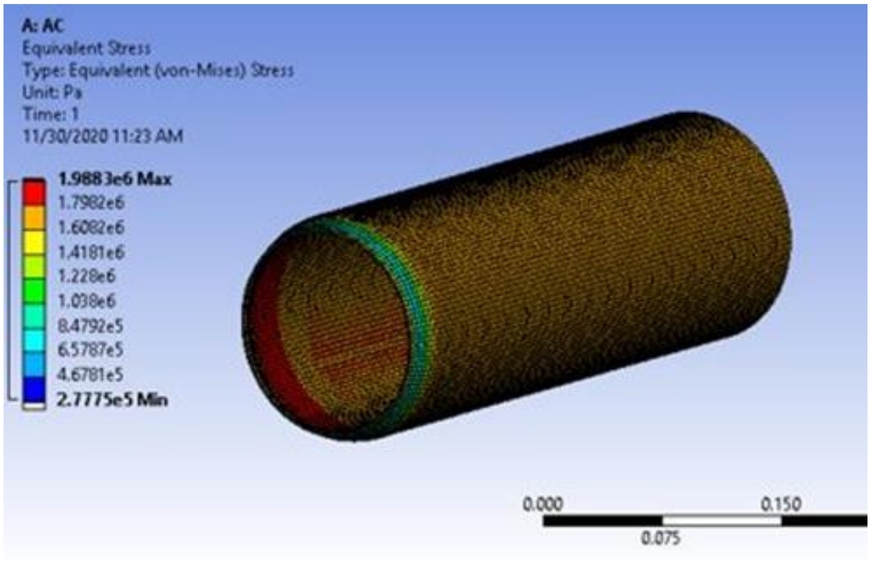


Figure 54 Von Mises Stress on Astronaut Capsule w/ Large Nylon Shear Pins

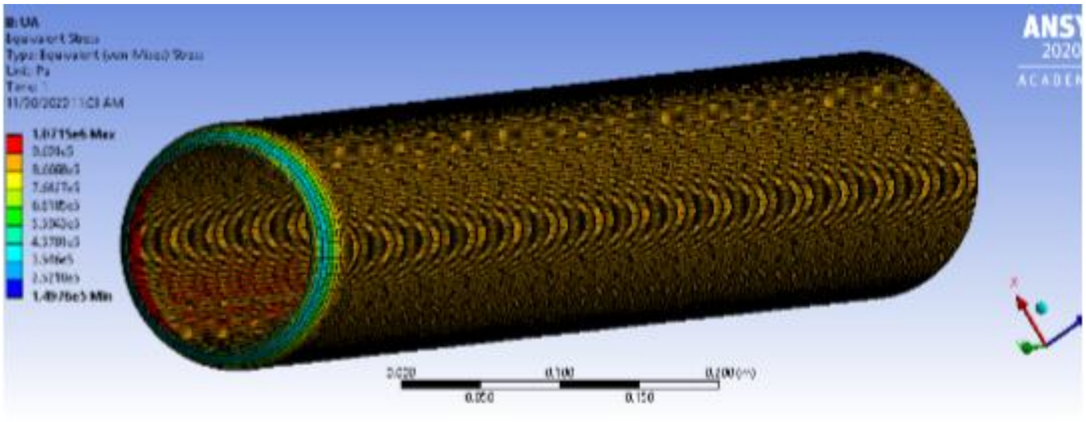


Figure 55 Von Mises Stress on Upper Airframe w/ Small Nylon Shear Pins

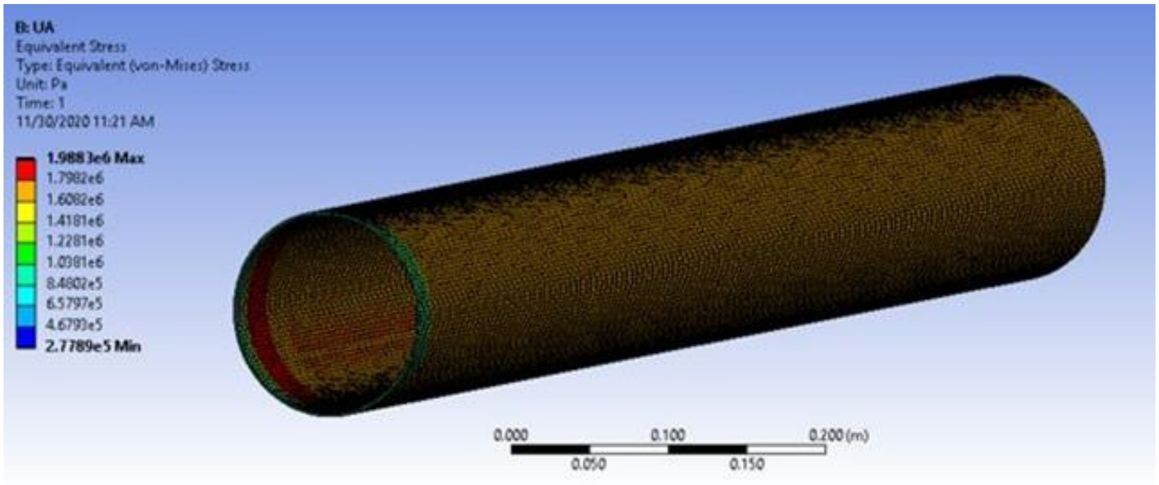


Figure 56 Von Mises Stress on Upper Airframe w/ Large Nylon Shear Pins

The team found that the airframes experienced a maximum Von Mises Stress of 1.0715 MPa in the case of the small shear pins and 1.988 MPa in the case of the large shear pins. Neither option exceeds the maximum tensile strength of E-Fiberglass, the airframe material, with a value of 1770 MPa [62]. Thus, both the astronaut capsule and the upper airframe would be able to withstand the maximum force and pressure of separation via black powder charge for at least 3 shear pins.

To confirm that the airframe could withstand the maximum recommended pressure range of the black powder charge, the team also ran an additional Von Mises Stress simulation under these conditions. Figure 58 shows the boundary conditions and Figure 59 shows the results of the analysis.

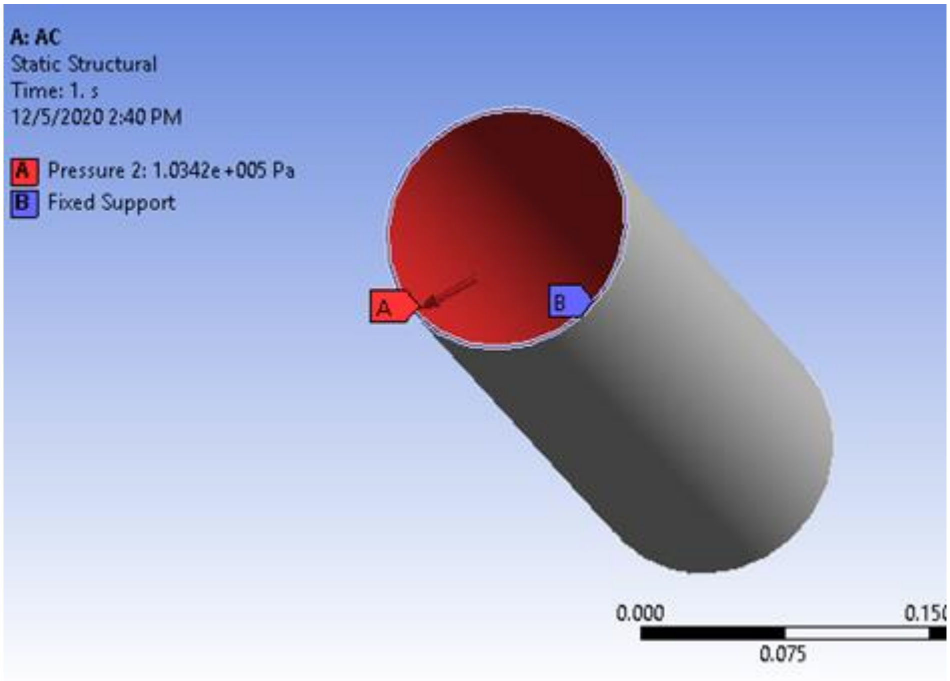


Figure 57 Boundary Conditions at Max. Recommended Pressure for Black Powder Charge

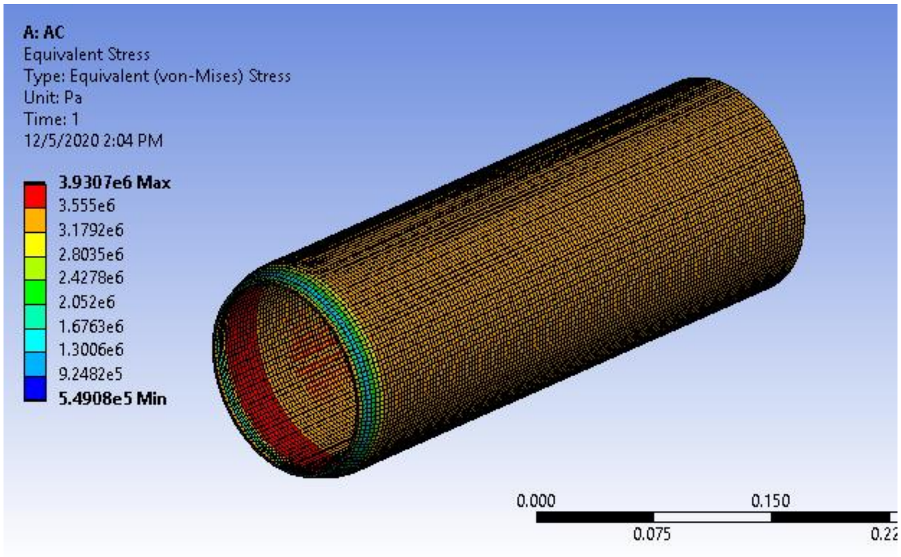


Figure 58 Max. Von Mises Stress at Max. Recommended Pressure for Black Powder Charge

The results show that even at the maximum recommended pressure of 15 psi, or 103,421 Pa, and approximately 2.6 grams of black powder, the E-Fiberglass airframe should hold. The maximum Von Mises stress the part experiences is 3.9 MPa, which is significantly less than the tensile strength of the material at 1770 MPa. This result confirmed that the black powder ignition would not cause any deformation or damage to the airframe during launch even at the maximum recommended pressure. Completing this analysis was critical to the structural integrity of the entire rocket. If the airframe was to deform or experience critical damage from high pressure concentrations, this could result in the entire rocket breaking up in flight. If that were to occur it could lead to the full-scale destruction of other subsystems and the potential safety threat of free-falling debris.

4.1.2 ARS Analysis Task 2: Autorotation System Model

To simulate the descent of the rocket under the autorotation system, a mathematical model was developed that solves for the kinematics of the rocket's descent over time. The kinematics consist of altitude, velocity, angular velocity, and acceleration. Overall, the forces acting on the rocket include gravity pulling the rocket down towards the ground which is counteracted by drag from the parachute and the forces generated by the autorotation system. As the rocket descends under autorotation, it enters a dynamic equilibrium with gravity, where the faster the rocket falls, the faster the blades spin, resulting in an increase in drag, slowing down the descent.

Several assumptions were made in this simulation. First, that the rocket does not travel in any direction perpendicular to its downward trajectory. Thus, the vehicle is considered to have only a velocity in the y-direction. To assess horizontal motion, the descent solver was coupled with the full model (see Section 4.3.1). Additionally, the assumption was made that the rocket remains in an upright orientation for the entirety of the simulation.

The model starts by defining the initial rocket and autorotation system parameters. A commercially available blade was selected for the autorotation model. Thus, the initial descent parameters were the overall mass of the craft, m, the diameter of the parachute, D, the area of the parachute A, and the moment of inertial of the blade system, I. Furthermore, for the autorotation solver, these parameters included the specified number of blades, n, the span of the blade in meters, b, the chord length of the blade in meters, c, and the pitch angle, ψ. The next set of parameters were considered constant: acceleration due to gravity, g, the density of air, ρ, and the viscosity of air, μ. Once the initial parameters were defined, then the initial conditions of the rocket were input. The simulation begins when the rocket is at apogee, with an assumed altitude of the rocket h_0, equal to the apogee height. The value of apogee is determined from FDA Analysis Task 1, results of which can be found in Section 4.3.1. The initial vertical velocity V_0, is assumed to be zero, as the craft will reach a point of zero velocity at the apogee. The angular rotation of the autorotation system is assumed to be zero at the start of the descent as the system has not yet deployed and only then will speed up. Finally, the acceleration at apogee was assumed to be zero. Figure 60 depicts all the equations used for the solution of the descent model while Table 7 provides the nomenclature and units for each variable.

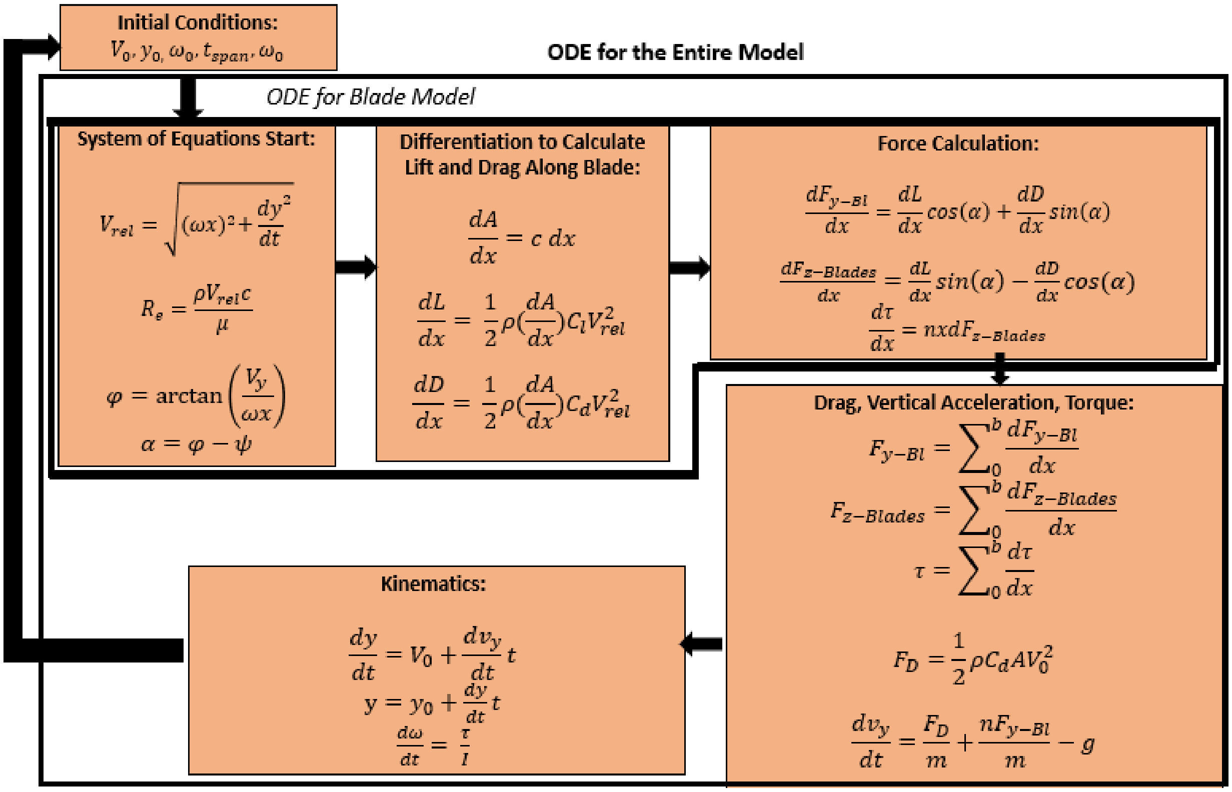

Initial Conditions:
$V_0, y_0, \omega_0, t_{span}, \omega_0$
ODE for the Entire Model
ODE for Blade Model
System of Equations Start:
$V_{rel} = \sqrt{(\omega x)^2 + \frac{dy}{dt}^2}$
$R_e = \frac{\rho V_{rel} c}{\mu}$
$\varphi = \arctan\left(\frac{V_y}{\omega x}\right)$
$\alpha = \varphi - \psi$
Differentiation to Calculate Lift and Drag Along Blade:
$\frac{dA}{dx} = c\,dx$
$\frac{dL}{dx} = \frac{1}{2}\rho(\frac{dA}{dx})C_l V_{rel}^2$
$\frac{dD}{dx} = \frac{1}{2}\rho(\frac{dA}{dx})C_d V_{rel}^2$
Force Calculation:
$\frac{dF_{y-Bl}}{dx} = \frac{dL}{dx}cos(\alpha) + \frac{dD}{dx}sin(\alpha)$
$\frac{dF_{z-Blades}}{dx} = \frac{dL}{dx}sin(\alpha) - \frac{dD}{dx}cos(\alpha)$
$\frac{d\tau}{dx} = nxdF_{z-Blades}$
Drag, Vertical Acceleration, Torque:
$F_{y-Bl} = \sum_0^b \frac{dF_{y-Bl}}{dx}$
$F_{z-Blades} = \sum_0^b \frac{dF_{z-Blades}}{dx}$
$\tau = \sum_0^b \frac{d\tau}{dx}$
$F_D = \frac{1}{2}\rho C_d A V_0^2$
$\frac{dv_y}{dt} = \frac{F_D}{m} + \frac{nF_{y-Bl}}{m} - g$
Kinematics:
$\frac{dy}{dt} = V_0 + \frac{dv_y}{dt}t$
$y = y_0 + \frac{dy}{dt}t$
$\frac{d\omega}{dt} = \frac{\tau}{I}$

Figure 59 Autorotation MATLAB Model Map

Table 7 Nomenclature and Units for MATLAB Model Map

Nomenclature	Parameter	Units
$V, \frac{dy}{dt}$	Velocity in the Vertical Direction	$\frac{m}{s}$
V_0	Initial Velocity	$\frac{m}{s}$
y	Height of the Rocket	m
y_0	Initial Height of the rocket	m
t_0	Initial Time	s
a	Acceleration	$\frac{m}{s^2}$

V_{rel}	Relative Velocity	$\frac{m}{s}$
ω	Angular Velocity	$\frac{rad}{s}$
$\frac{d\omega}{dt}$	Change in Angular Velocity	$\frac{rad}{s^2}$
dx	Differential Blade Length	m
x	Position on Blade	m
b	Blade Length	m
ρ	Density of Air	$\frac{kg}{m^3}$
R_e	Reynolds Number	$-$
c	Chord length	m
μ	Air dynamic viscosity	$\frac{N\ s}{m^2}$
φ	Angle of relative wind velocity	rad
α	Angle of Attack	rad
ψ	Pitch Angle	rad
$\frac{dA}{dx}$	Differential Area	m^2
$\frac{dL}{dx}$	Differential Blade Lift	N
$\frac{dD}{dx}$	Differential Blade Drag	N
C_l	Coefficient of Lift	$-$
C_d	Coefficient of Drag	$-$

$F_{y\ Bl}$	Blade Force Y-Direction	N
$dF_{z\ Bl}$	Blade Force Z-Direction	N
τ	Torque on the Blade	$N\ m$
I	Moment of Inertia of Blade System	$kg\ m^2$
n	Number of Blades	$-$
F_D	Parachute Drag Force	N
A	Parachute Area	m^2
$C_{d\ chute}$	Coefficient of Drag Parachute	$-$
g	Acceleration Due to Gravity	m/s^2
m	Mass of the Rocket	kg

The mathematical model consists of two systems of differential equations. The first represents the forces on the blade and is solved for with respect to increments of blade length, *dx*, over the blade span, *b*. When the forces have been found for each differential blade location, they are summed up to give the overall forces on the blade for one time step. These consist of the torque τ, and force F_y, representing the vertical force on the blade. These forces are then used in the second differential model to solve for the acceleration, a. The acceleration, a, and torque, τ, are then used to calculate the rest of the kinematics: altitude, h_y, velocity, V_y, and the angular velocity of rotation, ω.

Figure 61 shows the flight events modeled by the simulation over the full time span of descent, which terminates upon ground impact as denoted by Figure 60 over a time span of 100 seconds. Depending on the launch angle this termination time may vary. The simulation is marked by four distinct events, apogee (t_0), drogue parachute deployment (t_chute), autorotation deployment (t_ar), and landing (t_landing). The rocket will descend in free fall from apogee for approximately five seconds before the deployment of the drogue parachute. Once the parachute has been deployed, the rocket will descend for another fifteen seconds before the deployment of the autorotation system. This time was selected to give the parachute enough time to fully deploy and orient the rocket vertically. By using simple kinematics, this simulation allowed the ARS subteam to model the initial and final velocities and heights for each flight event. Table 8 lists the nomenclature for the variables over time span, *t*. However, the simulation code can be modified to change the timing of these mission events depending on the mission.

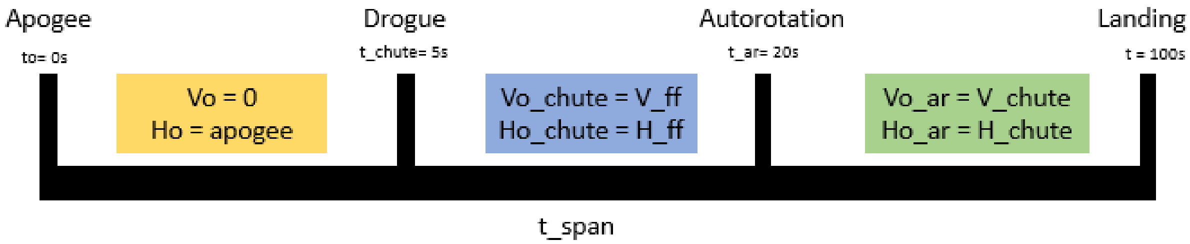


Figure 60 Flight Events Modeled Across Time Span, t

Table 8 Nomenclature and Units for Flight Events Across Time Span, t

Nomenclature	Parameter	Units
V_0	Initial Velocity	*m/s*
h_0	Initial Height (at apogee)	*m*
t_0	Initial Time	*s*
t_chute	Time of chute deployment, end of the free-fall period	*s*
V_ff	Velocity at the end of freefall and at parachute deployment	*m/s*

H_ff	Height at the end of freefall and at parachute deployment	*m*
t_ar	Time of autorotation deployment, end of the chute descent	*s*
V_ar	Velocity at the end of parachute descent and start of autorotation	*m/s*
H_ar	Height at the end of parachute descent and start of autorotation	*m*
t_landing	Time at the end of autorotation descent and at landing	*s*

Now that the differential models have been described, as well as the different time spans involved, the calculation of the kinematics and forces will be described. The first step to solve for the forces in the autorotation system over a single time step is to calculate the angle of attack, α, of the autorotation blades as well as the Reynolds number, R_e. This starts with an equation for the relative velocity of the blades, V_{rel}, with respect to the air. This equation is then used to calculate the Reynolds number, R_e. This number is used to determine the flow patterns in different fluid flow situations, in this case the fluid being air. The angle of attack of the blades is calculated using the constant pitch of the autorotation blades, ψ, and a calculated value representing the angle of the wind velocity relative to the blade, φ.

$$V_{rel} = \sqrt{(\omega dx)^2 + \frac{dy}{dt}^2} \tag{31}$$

$$R_e = \frac{\rho V_{rel} c}{\mu} \tag{32}$$

$$\varphi = \arctan\left(\frac{V_y}{\omega dx}\right) \tag{33}$$

$$\alpha = \varphi - \psi \tag{34}$$

Next, the forces for lift and drag are calculated using the Reynold’s Number and angle of attack from the previous step. The coefficients for lift and drag are interpolated using a lookup table with XFLR5 data based upon the previously calculated Reynolds number and angle of attack. With the coefficient of lift or drag, the density of the air, ρ, differential area, $\frac{dA}{dx}$, and relative wind velocity, V_rel, the differential lift and drag forces, $\frac{dL}{dx}, \frac{dD}{dx}$, can be calculated for a particular point on the blade.

$$\frac{dA}{dx} = c\ dx \tag{35}$$

$$\frac{dL}{dx} = \frac{1}{2}\rho(\frac{dA}{dx})C_l V_{rel}^2 \tag{36}$$

$$\frac{dD}{dx} = \frac{1}{2}\rho(\frac{dA}{dx})C_d V_{rel}^2 \tag{37}$$

The coefficients of lift and drag can be found using the batch analysis feature in XFLR5. An airfoil, in this case NACA 0012, can be input to the Direct Foil Design tool. Then in XFoil Direct Analysis, a batch analysis can be run. To run this, XFLR5 needs a range of Reynold’s numbers, a Mach number, and a range of values for Angle of Attack, AOA. Increments were selected to be a factor of 10 less than the extent of the range. The Reynold's Number, R_e, was calculated at the beginning and end of a period spanning a range of time from apogee to 10 seconds after apogee. Finding the Reynold's Number also required calculating the velocity, V, at the same points in time. The range for the AOA was determined to be -5 to 25 degrees. The remainder of the values for batch analysis could be left as default. After the analysis ran, the correct graphs could be pulled up and the data from them could be exported into a .txt file.

Using the differential lift and drag forces calculated in the previous step, and the angle of attack from the first step, the differential forces on the blade could be calculated in the y-direction, $\frac{dF_{y-Bl}}{dx}$ and z-direction, $\frac{dF_{z-Blades}}{dx}$ with the equations below. Using the number of blades, *n*, and the forces in the z-direction on the blade, the torque can be calculated for a particular point on the blade.

$$\frac{F_{y-Bl}}{dx} = \frac{dL}{dx}cos(\alpha) + \frac{dD}{dx}sin(\alpha) \tag{38}$$

$$\frac{dF_{z-Blades}}{dx} = \frac{dL}{dx}sin(\alpha) - \frac{dD}{dx}cos(\alpha) \tag{39}$$

$$\frac{d\tau}{dx} = nx\frac{dF_{z-Blades}}{dx} \tag{40}$$

All the equations in the previous three steps are integrated with respect to the blade span, b, which allowed the total sum of forces to be calculated across the entire blade. This results in finding the forces in the y-direction, F_{y-Bl}, and torque, τ, which allows the kinematics to be calculated.

With the solution of the forces of lift and drag for a particular time step, the kinematics of the rocket are calculated. Prior to the deployment of the autorotation mechanism, the rocket is slowed by the drogue chute. Therefore, the kinematics of the drogue chute must be calculated. The force of the drag associated with the drogue parachute is assumed to be zero until the parachute is deployed. Below, the equation to solve for F_D, the chute drag force, can be seen. With all the force calculations complete, the acceleration of the rocket, $\frac{dv_y}{dt}$, can be calculated.

$$F_D = \frac{1}{2}\rho C_d A V_0^2 \tag{41}$$

$$\frac{dv_y}{dt} = \frac{F_D}{m} + \frac{nF_{y-Bl}}{m} - g \tag{42}$$

Once the rocket's acceleration is known at a particular time step, the vertical, $\frac{dy}{dt}$, and change in angular velocity, $\frac{d\omega}{dt}$, can be calculated. These equations were integrated with respect to time through the ordinary differential equation solver.

$$\frac{dy}{dt} = V_0 + \frac{dv_y}{dt}t \tag{43}$$

$$\frac{d\omega}{dt} = \ \frac{\tau}{I} \tag{44}$$

The implementation for this simulation was completed in MATLAB, employing the ode45 function, an ordinary differential equation solver. As mentioned before, there are two systems of ODEs that were solved, one with respect to time, and one with respect to the blade length.

The ode45 equation with respect to time, represents the kinematics solution and solves for the height, velocity, angular velocity, and acceleration of the craft over time. The drag force on the craft, and forces generated by, gravity, the autorotation are the variables that significantly change the kinematics of the craft over time. The solution for the forces during a particular time step, is accomplished by an ODE nested within the kinematics ODE. This is referred to as the blade model ODE. Each ODE was executed by putting the calculations in a separate function called by ode45. The run time of the program can increase exponentially due to the fact that the two ODE45 functions are nested, however, this simulation takes about 10 seconds to run, depending on the simulation parameters.

The most intensive processing task during the execution of the program is finding the lift and drag coefficients. These are obtained from the XFLR5 data tables. This data is imported into MATLAB as a 2D matrix of lift and drag coefficients mapped to a particular Reynolds number and angle of attack. When the program is executed, the Reynolds number and angle of attack is calculated, and the corresponding lift and drag coefficient can be interpolated using a spline FIT function.

The purpose of this analysis task was to create a simplified autorotation blade model to estimate lift forces to adequately slow down the descent of the craft. By changing the simulation parameters, certain different rocket configurations can be tested. Once the model was completed, the data was analyzed in four different plots. The finalized code for this analysis task can be found in Appendix B.

Figure 61 shows the acceleration of the rocket’s descent over time. Within the graph are two spikes that represent the two main flight events: first the deployment of the drogue parachute at about 5 seconds, and second the deployment of the autorotation system at about 20 seconds.

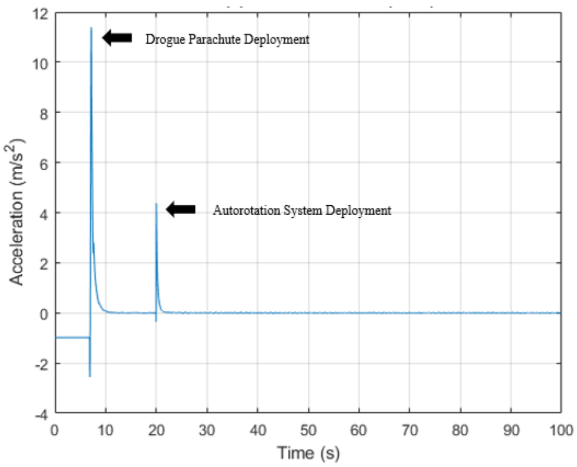


Figure 61 Acceleration of Rocket Over Time During Descent

The next plot Figure 62 shows the angular speed of the blades on the autorotation system over time. The angular speed of the blades increases exponentially meaning it begins by increasing very quickly and settles out over time.

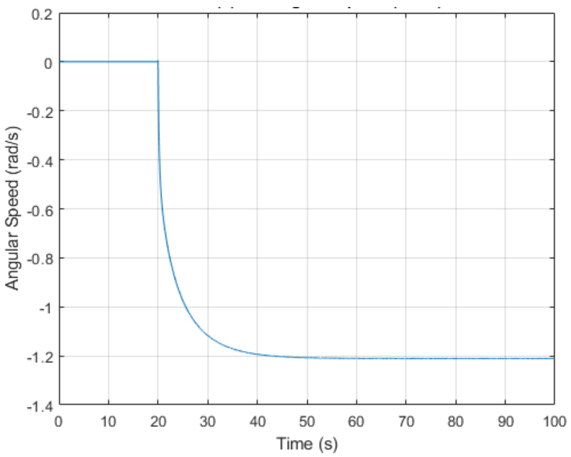


Figure 62 Angular Speed of Blades Over Time

Figure 63 shows the rocket altitude vs. time. The flight events during the rocket’s descent can be seen clearly as the drogue parachute deploys at 5 seconds and the autorotation system deploys at 20 seconds. After the deployment of the autorotation system, the rocket descends at a constant rate.

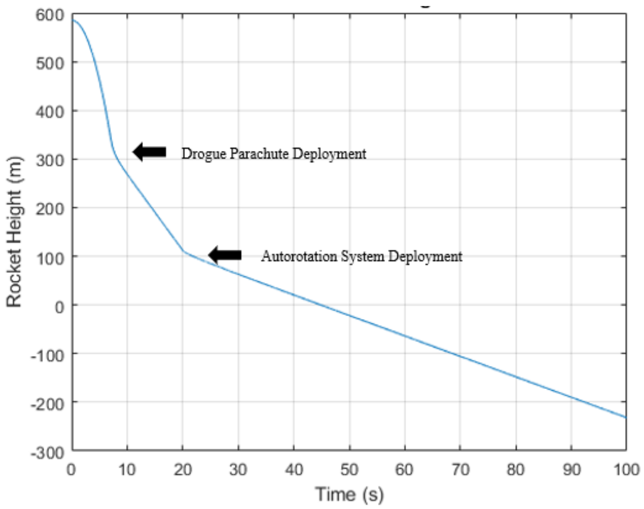


Figure 63 Rocket Height Over Time During Descent

Figure 64 shows the rocket’s vertical speed vs. time. The rocket’s speed increases rapidly as it descends under free fall and as denoted by the spike of the deployment of the drogue parachute, the speed is then quickly reduced. The craft then descends at a constant speed of about 15m/s under the drogue parachute until the autorotation system is deployed. Once the autorotation system deploys at 20 seconds the speed can be seen leveling out at about 4.25 m/s. In order to avoid damage the rocket needs to descend at a speed less than 5m/s. According to the model, the landing velocity is within this safety margin and thus the craft should be able to land successfully under autorotation without damage.

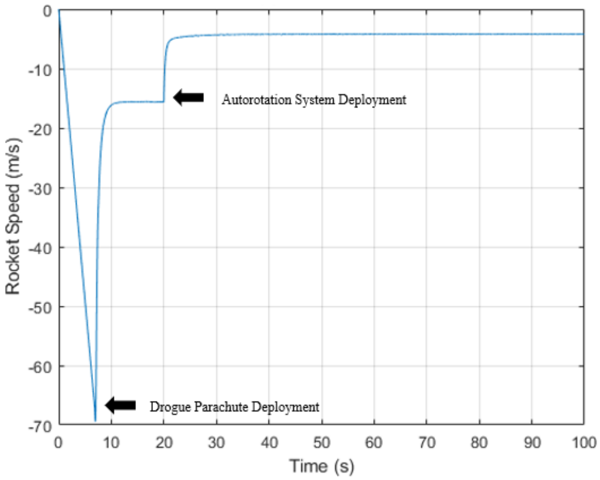


Figure 64 Speed of Rocket Over Time During Descent

4.1.3 ARS Analysis Task 3: Autorotation Blade Aerodynamic Loads

For the third analysis task, the rotor blade chosen was modeled as a 3D airfoil to be used within a Fluent simulation. The method chosen was a Sliding Mesh approach [63]. This differs from the 2019-2020 MQP team which used a Multiple Reference Frame (MRF) approach to model their autorotation system [18]. This method has more complicated user inputs and suffers from less accurate results due the model being based on a stationary airfoil in a domain with variable airspeeds to simulate rotation [63]. The Sliding Mesh approach however simulates the rotation and change of position for the airfoil. This has a higher computational cost but provides more accurate results while using a more concise geometry. The domains and boundaries for the simulation are shown in Figure 66.

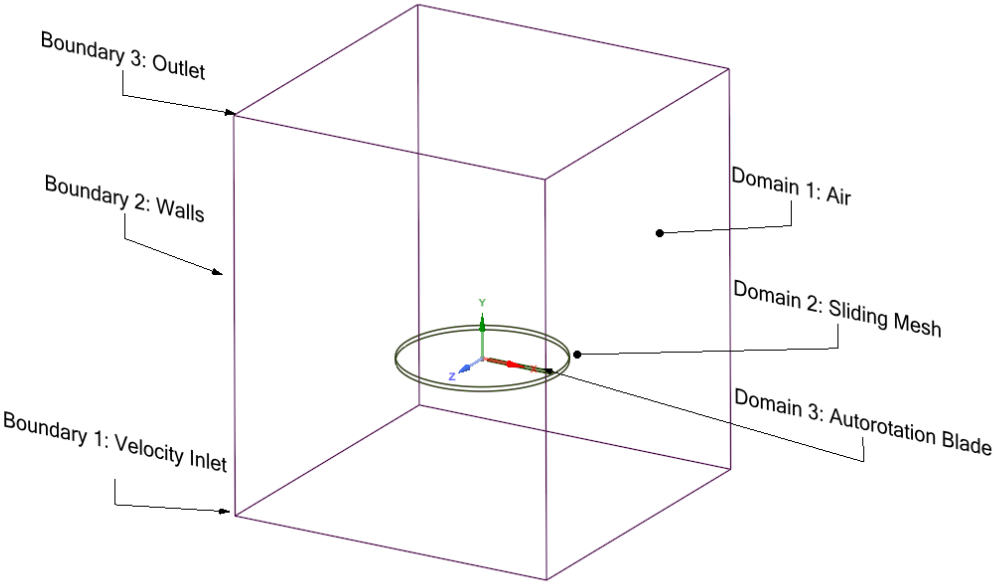


Figure 65 Ansys Fluent Simulation Diagram

Table 9 Fluent Diagram Key

CFD Component	Boundary Condition	Equation or Value	Description
Medium	Air	ρ=1.225 kg/m^2 μ=1.7894E-05 kg/ms	Air to produce Lift/Draft forces on the blade.
Sliding Mesh	Air	1.7 Rad/s	Gives the Blade Rotation
Rotor Blade	Carbon Fiber	ρ=1699.9kg/m^2	A 780mm Align Blade
Inlet	Velocity Inlet	Standard ATM	Simulates Freefall
Outlet	Pressure Outlet	Standard ATM	Simulates no boundary, or ‘External flow’

The simulation had a time step of 1.5E-3 seconds, with 80 iterations per time step and a total of 10,000 steps in total. This allows for the blade to rotate more than once per simulation. That allowed for the simulation to account for the wake and turbulence. Figures 67 - 69 show the pressure and velocity distribution on the blade. The velocity gradient on the airfoil represents the rotational movement of the blade. To do a single revolution around the center axis of the autorotation recovery system, the outer edge of the rotor blade will have to spin faster than the inner edge. The blade itself was positioned a distance away from the rotational axis, to model its connection to the autorotation recovery deployment mechanism. Using Ansys Fluent, the pressure contours on the top and bottom of the blade were found and are shown in Fig. 68 and Fig. 69. The static pressure here represents the blade moving with the fluid through the sliding mesh. The reference pressure used is gauge pressure. A positive value represents areas of high pressure, greater than the reference pressure, while a negative value shows lower pressure zones, below gauge pressure. This difference in pressure across the rotor blade produces its lift.

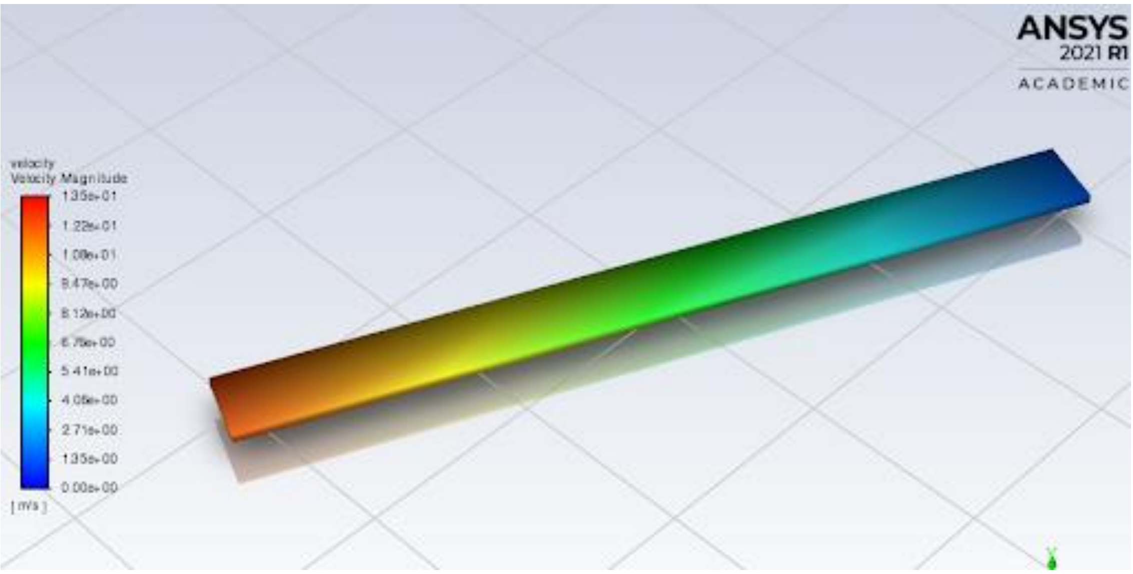


Figure 66 Velocity Along the Blade

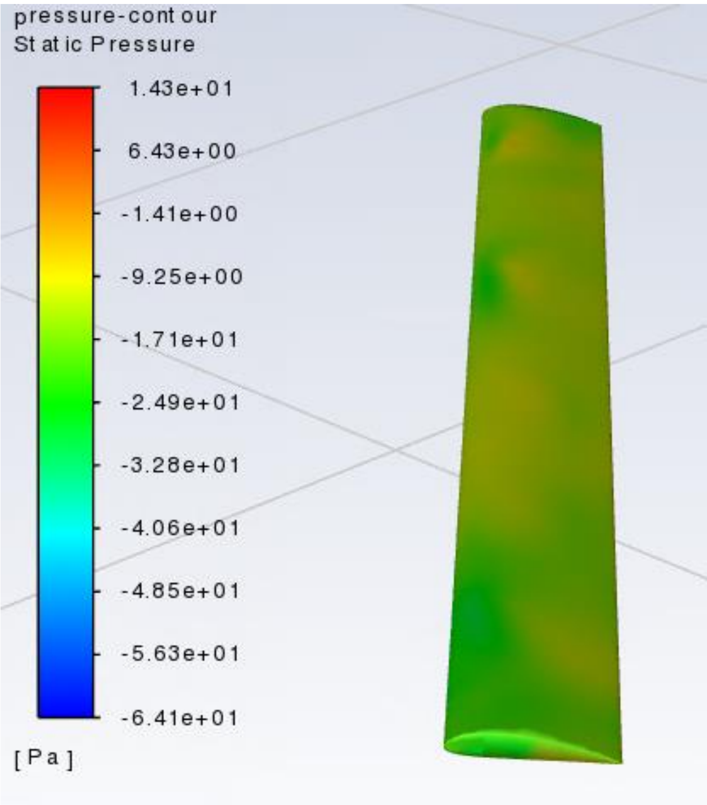


Figure 67 Static Pressure on the Top of the Blade

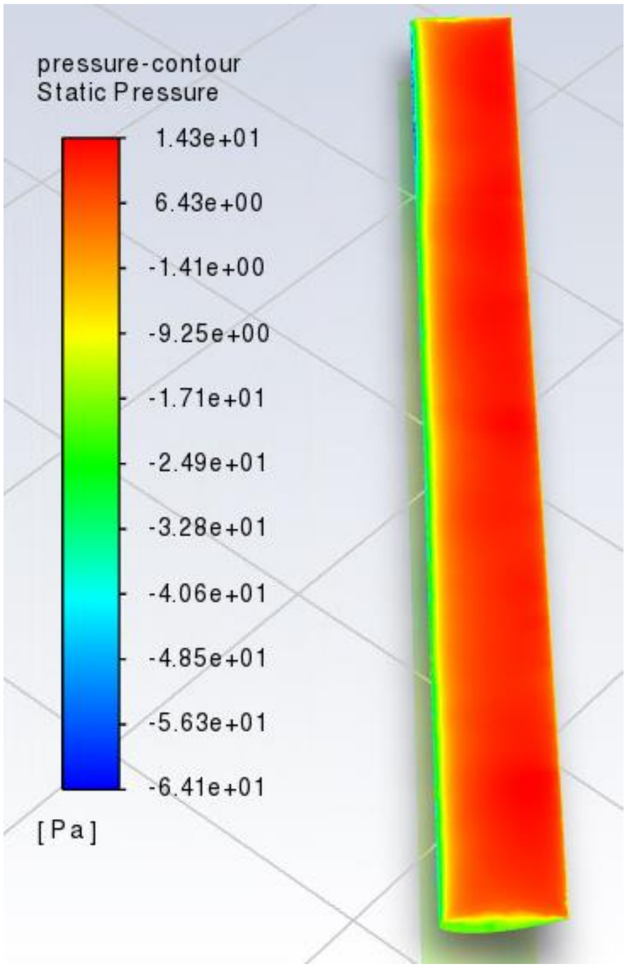


Figure 68 Static Pressure on the Bottom of the Blade

The lift and drag coefficients were found between angles of attack ranging from 0°, to 30° in 5° intervals. This was to determine the effect that the angle of attack has on the aerodynamic forces on the rotor blade during the autorotation recovery. For all cases the same blade was used, with the same angular velocity and conditions. These coefficients can be seen in Figure 70. The functions found by plotting these values can be used as an input in the autorotation system model, allowing for more accurate simulation of the descent speed and allowing the ARS subteam to design an optimal recovery system.

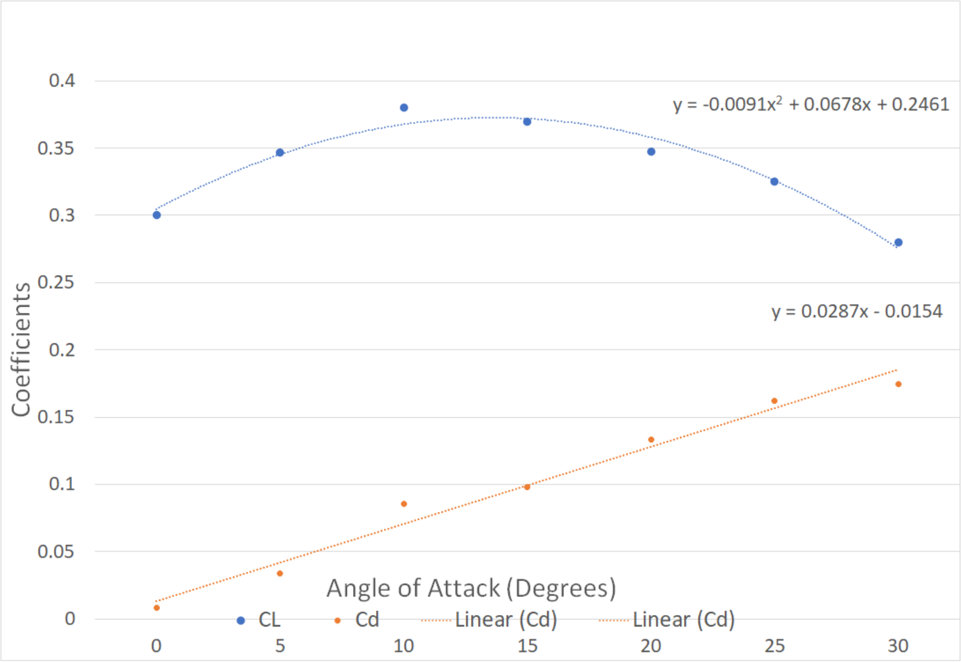


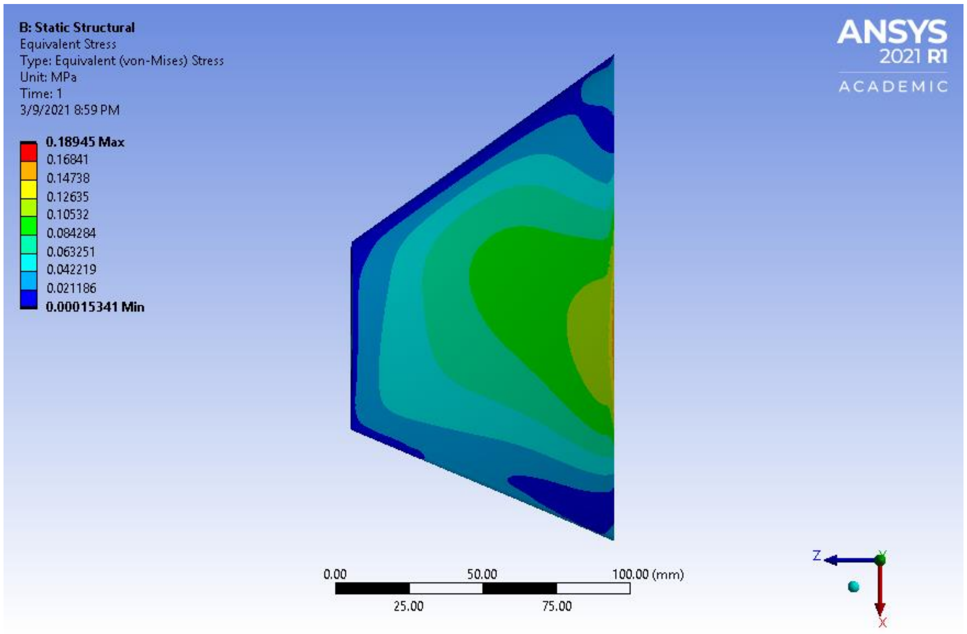
Figure 69 Lift and Drag vs. Angle of Attack

4.1.4 ARS Analysis Task 4: Stabilizing Fin Stress Distribution

The fourth analysis task aimed to determine the stress distribution of each stabilizing fin. The purpose of this was to see if the fins can withstand the loads during the period of the maximum pressure.

The fins were modeled in SOLIDWORKS and imported into Ansys Workbench where an Ansys Fluent analysis was performed as part of FDA's Analysis Task 3, detailed in Section 4.3.3. As described in Section 4.3.3, the maximum stresses were determined to be at the point prior to stage separation. For this task, an Ansys Static Structural system was added and the pressure distribution from the results of the Fluent model could be used as an imported load for the Static Structural model. The rest of the set up for the Static Structural model includes setting the material of the fin to E-Glass Wet from the Composite Materials library, which has the properties of a layer of a wet layup E-Glass, and a fixed support where the fin meets the airframe. The results of the task are viewed in an Von Mises (Equivalent) Stress distribution.

The maximum Von Mises stress the NACA 4414 and 0414 fins experience are 0.18945 MPa and 0.01109 MPa, respectively. The Von Mises stress distributions are displayed in both a right and left view on both NACA foils in Figures 71-74. According to AGATE, the tensile strength of a fiberglass layup at room temperature is 288.54 MPa, and so, it appears that the Ansys model is not representative of the fins as manufactured [64].



B: Static Structural
Equivalent Stress
Type: Equivalent (von-Mises) Stress
Unit: MPa
Time: 1
3/9/2021 8:59 PM

Figure 70 Von Mises Stress Distribution NACA 4414 Fin, Right View

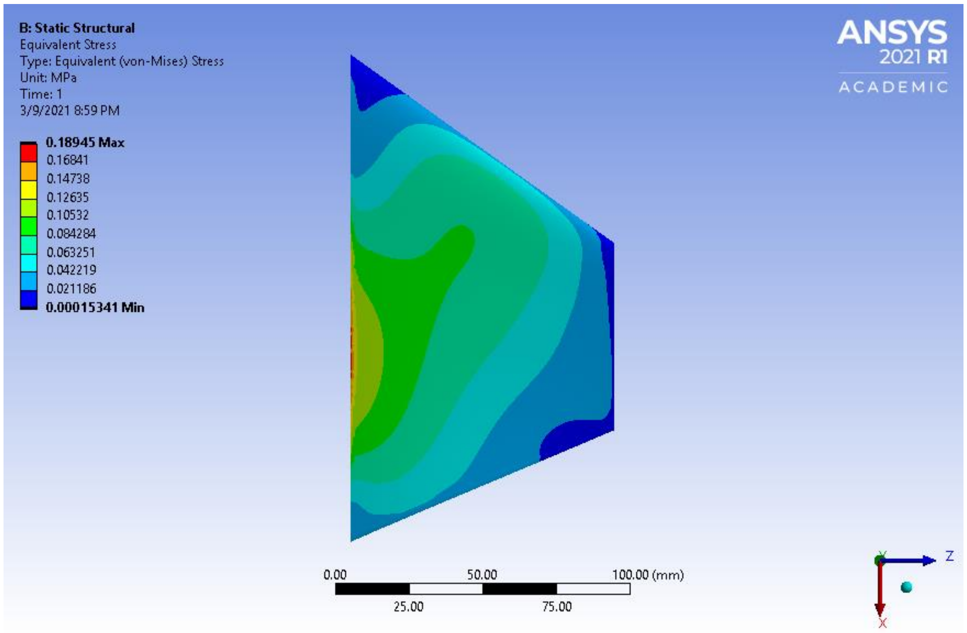


Figure 71 Von Mises Stress Distribution NACA 4424 Fin, Left View

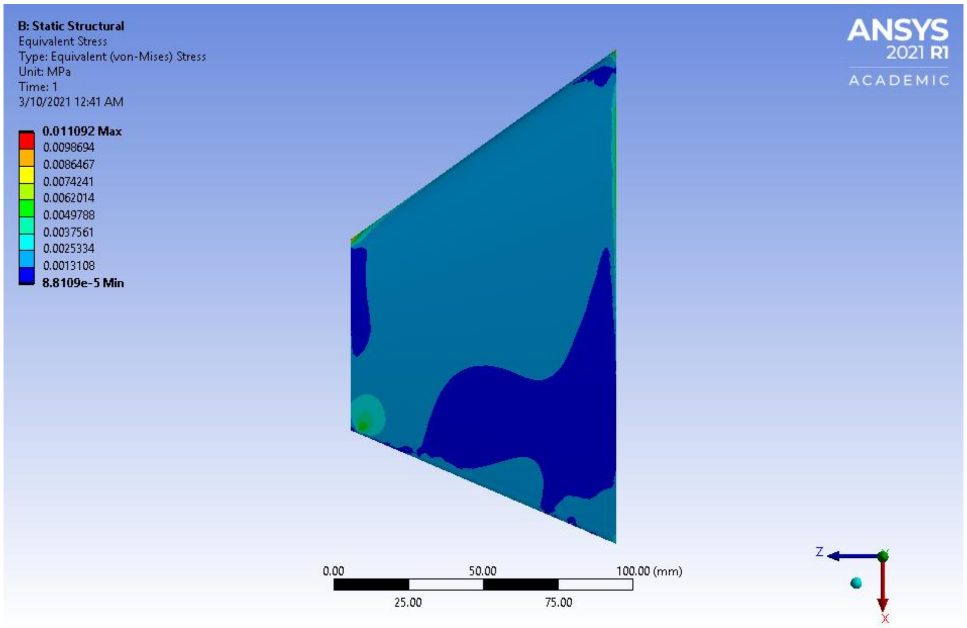


Figure 72 Von Mises Stress Distribution NACA 0414, Right View

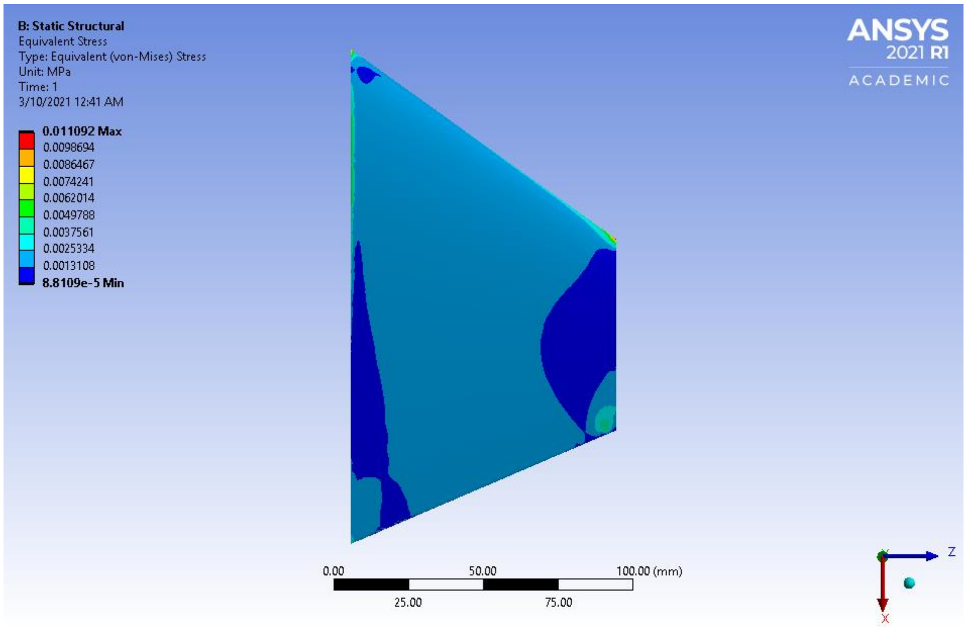


Figure 73 Von Mises Stress Distribution NACA 0414, Left View

The Von Mises stress from the analysis is likely much lower than it would be on the constructed fins. This is because Ansys assumes a solid E-Glass and that the directions of the material align with the coordinate system that the fin imported on. In reality, the material is built up in many plies, or layers, and the material may change x, y, and z directions following the contours of the airfoil, in addition each ply may not be in the same orientation. Not only that, but it has a hot-wire cut foam core to give it the shape of the airfoil. An accurate model would analyze the E-Glass composite ply-by-ply which would show that the stresses are different for each individual layer with the stresses being higher on the outer most layer and lowest on the inner most layers.

4.2 Propulsion, Thermal, and Separation Systems

The PTSS subteam was tasked with four main analysis tasks. These tasks centered around an in-depth analysis of the overall rocket motor performance, temperature distribution, spring and air brake effectiveness, and the electric match model. Each task was assigned a task lead. By completing these analysis tasks, the PTSS subteam was able to apply theoretical knowledge from the literature review, as well as training in Ansys Structural and Ansys Fluent to better understand the structural and physical properties of the rocket.

4.2.1 PTSS Analysis Task 1: Motor Performance Model

The first analysis task for the PTSS team was to create a simplified model for the selected motors that could be used to model the performance of the motors during flight. The reasoning for this analysis was to ensure that the motor performance information provided by the manufacturer was accurate and would provide the thrust necessary to accomplish the mission.

To do this, the Cantera tool suite was downloaded and installed as an extension to MATLAB. Cantera solves problems involving chemical kinetics, thermodynamics, and transport processes. The usage of Cantera focused primarily on its built-in chemical equilibrium solver which uses an element potential method to determine the result of an equilibrium reaction in a non-stoichiometric way [65]. To make use of this solver, it was assumed that the reaction that occurs in the chamber is in equilibrium. This assumption allowed the reaction to be modeled using the equilibrium solver and specifically the “equilibrate()” method. The reaction that was modeled using Cantera was:

$$xNH_4ClO_2 + yAl \rightarrow aH_2O + bHCl + cN_2 + dO_2 + eAl_2O_3 + fTrace$$

A limitation of this reaction was that it did not include the bonding agent that holds the propellant together. As a result of this, the mole fractions of the aluminum and ammonium perchlorate were skewed. To model the reaction, a data file needed to be created contained thermodynamic information about the reactants and potential products in the above reaction. Cantera comes with several of these types of files which provide this information for many substances and compounds. However, the thermodynamic data for the ammonium perchlorate and many of the products that involve the aluminum were not included in this data. Therefore, it was necessary to create a data file for this problem that referenced the data Cantera already has, as well as thermodynamic information for the missing compounds. This data file can be seen in Appendix C. The thermodynamic data that was not included in Cantera's database was found online using NASA's ThermoBuild application which uses the NASA Glenn Research Center thermodynamic database and provides data for use in chemical equilibrium analysis programs [66].

Finally, to use the Cantera equilibrate method, a MATLAB script was created that imported the data file and then ran the equilibrate command with the imported gas. This script can be seen in Appendix D. The equilibrate method required that two properties be held constant during the reaction. For the purposes of modeling, it made the most sense that the internal energy and volume be held constant, as the pressure and temperature were necessary to calculate motor performance. While the volume of the system would increase due to the burning propellant creating more space in the chamber, it was more important to ensure that the temperature and pressure could change freely.

The Cantera Analysis provided a great deal of information about the resulting gas mixture, including the temperature and pressure of the reaction, which is used as the chamber temperature and pressure. The density, average molar mass, enthalpy of reaction, and heat capacities at constant pressure and volume are also provided. From these values the ratio of specific heats and specific gas constant could be calculated. These results could then be used in calculations for motor performance as well as in other analysis tasks.

The main equation of the motor performance model was the thrust equation, which is given in Eq. 45.

$$T = \dot{m}u_e + (p_e - p_a)A_e \tag{45}$$

Where $\dot{m}$ is the mass flow rate through the nozzle, u_e is the nozzle exit velocity, p_e is the pressure at the exit of the nozzle, p_a is the pressure of the surrounding atmosphere, and A_e is the exit area of the nozzle. Aside from p_a these values were either dependent on the results of the Cantera analysis or the design of the nozzle. The nozzle exit velocity could be calculated using the results from the chemical equilibrium analysis based on Eq. 46.

$$u_e = \sqrt{\frac{2\gamma_m}{\gamma_m - 1}\mathcal{R}T_0\left[1 - \left(\frac{p_e}{p_0}\right)^{\frac{\gamma_m}{\gamma m-1}}\right]} \tag{46}$$

Where γ_m is the ratio of specific heats for the propellant exhaust, $\mathcal{R}$ is the mass specific gas constant of the propellant exhaust, and T_0 and p_0 are the chamber temperature and pressure, respectively. This equation provides the speed in meters per second of the exhaust gas as it leaves the nozzle, expanding to the specified exit pressure. Another value was calculated from the Cantera analysis was the mass flow rate of the propellant through the nozzle. Equation 47 is used to calculate the mass flow rate through a chocked nozzle.

$$\dot{m} = \frac{p_0 A_t}{\sqrt{T_0}}\sqrt{\frac{\gamma}{\mathcal{R}}\left(\frac{2}{\gamma + 1}\right)^{\frac{\gamma+1}{\gamma-1}}} \tag{47}$$

Where A_t is the area of the nozzle throat. While the mass flow rate and exit velocity were calculated from the Cantera model, the exit pressure and area were dependent on the geometry of the nozzle. The nozzle of the rocket motor had an expansion ratio, which was the exit area, divided by the area of the throat. This ratio helped to determine A_t and A_e but could also be used in an isentropic flow table to find a pressure ratio between the exit pressure and the chamber pressure. This ratio could then be used in the equation for u_e (Eq. 46) as well as used to find p_e for the thrust equation.

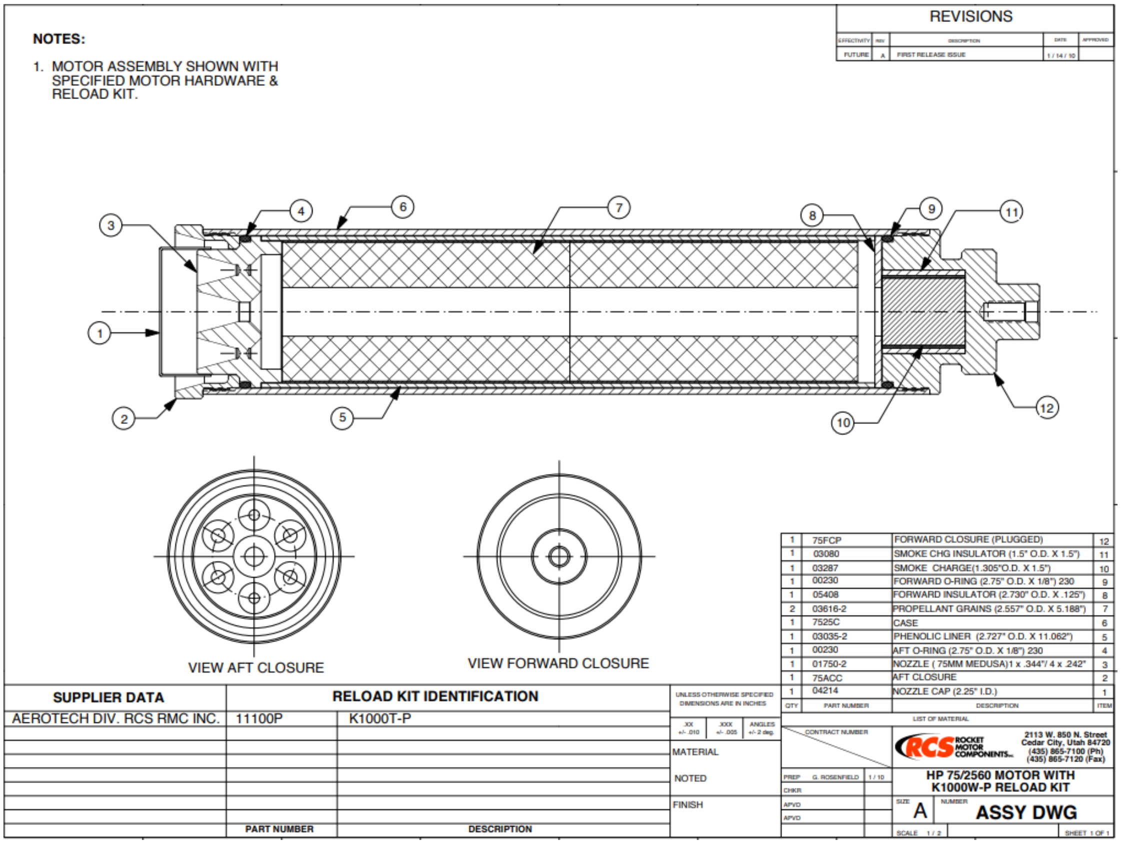
With the thrust of the rocket, other rocket performance parameters could be calculated. These include the total impulse and specific impulse of the motor. The total impulse of the rocket motor could be found by integrating the thrust over the burn time. Because the mass flow rate was assumed constant, the thrust was also assumed constant across the burn time. This result was not realistic due to difference in propellant burn rate in the chamber, which would affect the mass flow rate and therefore the thrust. This caused inaccuracies in the impulse calculation as the thrust may be too high or low over the burn time. The specific impulse is found using Eq. 48.

$$I_{sp} = \frac{T}{\dot{m}g} \tag{48}$$

This equation suffers from similar issues to the total impulse of the motor, where the assumption of a constant mass flow rate causes inaccuracies in the result of the equation.

4.2.1.1 First Stage Motor Results

The motor that was chosen for the first stage was the Aerotech K1000T-P. Aerotech's website provides schematics on all their rocket motors, including part names, as can be seen in Figure 75.


NOTES:

1. MOTOR ASSEMBLY SHOWN WITH SPECIFIED MOTOR HARDWARE & RELOAD KIT.

REVISIONS

EFFECTIVITY	REV	DESCRIPTION	DATE	APPROVED
FUTURE	A	FIRST RELEASE ISSUE	1 / 14 / 10	

QTY	PART NUMBER	DESCRIPTION	ITEM
1	75FCP	FORWARD CLOSURE (PLUGGED)	12
1	03080	SMOKE CHG INSULATOR (1.5" O.D. X 1.5")	11
1	03287	SMOKE CHARGE(1.305"O.D. X 1.5")	10
1	00230	FORWARD O-RING (2.75" O.D. X 1/8") 230	9
1	05408	FORWARD INSULATOR (2.730" O.D. X .125")	8
2	03616-2	PROPELLANT GRAINS (2.557" O.D. X 5.188")	7
1	7525C	CASE	6
1	03035-2	PHENOLIC LINER (2.727" O.D. X 11.062")	5
1	00230	AFT O-RING (2.75" O.D. X 1/8") 230	4
1	01750-2	NOZZLE (75MM MEDUSA)1 x .344"/ 4 x .242"	3
1	75ACC	AFT CLOSURE	2
1	04214	NOZZLE CAP (2.25" I.D.)	1

LIST OF MATERIAL

SUPPLIER DATA	RELOAD KIT IDENTIFICATION	
AEROTECH DIV. RCS RMC INC.	11100P	K1000T-P
	PART NUMBER	DESCRIPTION

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES

.XX +/- .010 .XXX +/- .005 ANGLES +/- 2 deg.

MATERIAL

NOTED

FINISH

CONTRACT NUMBER

RCS ROCKET MOTOR COMPONENTS

2113 W. 850 N. Street
Cedar City, Utah 84720
(435) 865-7100 (Ph)
(435) 865-7120 (Fax)

PREP G. ROSENFIELD 1 / 10

CHKR

APVD

APVD

HP 75/2560 MOTOR WITH K1000W-P RELOAD KIT

SIZE A NUMBER ASSY DWG

SCALE 1 / 2 SHEET 1 OF 1

Figure 74 Schematic of the Aerotech K1000T-P Rocket Motors © [67], (n.d.)

Using the schematic in Figure 75, the nozzle of the motor was identified as the 75 mm Medusa Nozzle. Information about this nozzle's characteristics was found on the manufacturer's website [68], which lists the exit diameter as 0.75 inches. Whereas the exit diameter was a set constant, the throat diameter may be drilled to five different values: 0.344", 0.485", 0.594", 0.627", and 0.634" [68]. Since the mass flow rate, and therefore thrust, is a function of the throat area, it became an input parameter. By altering the oxidizer to fuel ratio and the throat area, the calculated motor performance could be adjusted to try to match the available data. This was done in an iterative process with different throat areas being paired with different oxidizer to fuel ratios, while recording the motor performance and mass flow rate for each iteration. From this iterative process, it was determined that there should be 0.05 moles of aluminum to 0.95 moles of ammonium perchlorate for the Cantera reaction analysis which provided the following results.

Table 10 Results of First Stage Cantera Analysis

T_0 (K)	P_0 (MPa)	ρ (kg m^{-3})	c_p (J kg^{-1} K^{-1})	γ	M_m (g mol^{-1})	$h_{reaction}$ (kJ kg^{-1})
2065.78	2.830	4.617	1540.9	1.239	28.02	-1750.4

Furthermore, the throat area that was determined by this iterative process was 1.992 cm^2 which provides an expansion ratio of 1.431. With this geometry and the results of the Cantera analysis, the performance of the rocket motor was determined and compared to the information provided by the manufacturer. This comparison is outlined in Table 11.

Table 11 Comparison of Model and Manufacturer First Stage Motor Performance Data

	$\dot{m}$ (kg s^{-1})	Max Thrust (N)	Average Thrust (N)	I$_{sp}$ (s)
Cantera/ MATLAB	0.4723	1333.01		287.7
Manufacturer Data	0.5142	1674	1066	206.3

From the data in Table 11 it is apparent that the mass flow rate of the model is less than that of the value calculated from the manufacturer provided data. Despite this, the model still provides a thrust output larger than the average thrust value for the motor. The specific impulse of the model and the manufacturer data are significantly different, since they are calculated based on total impulse, and the constant mass flow assumption causes the thrust to be consistently greater, leading to a much higher total impulse. From this analysis it seems that the motor will be able to perform to the manufacturer provided data. Despite the discrepancies in mass flow rate, the model is still able to outperform the average thrust of the manufacturer data, though it is not quite able to reach the output of the maximum thrust.

4.2.1.2 Second Stage Motor Results

Blueprints and nozzle information were not available for the Ceasaroni K661 motor that was chosen for the second stage. Because of this, a different approach was needed to model the second stage motor. By assuming a constant mass flow rate, the value could be calculated by subtracting the final engine mass, m_f, from the initial engine mass, m_i, and dividing by the engine burn time, τ_b, as seen in Eq. 49.

$$\dot{m}_2 = \frac{m_i - m_f}{\tau_b} \tag{49}$$

These values were provided by the manufacturer [20] and result in a mass flow rate of 0.3411 kg s^{-1}. Knowing this mass flow rate, the throat area could be calculated by rearranging Eq. 47 for mass flow rate. This results in an equation for the area of the throat, Eq. 50.

$$A_t = \sqrt{\frac{\dot{m}^2 T_0}{p_0^2}\left(\frac{\mathcal{R}}{\gamma}\right)\left(\frac{2}{\gamma+1}\right)^{-\left(\frac{\gamma+1}{\gamma-1}\right)}} \tag{50}$$

Using this equation and the calculated mass flow rate the throat area is 1.486 cm^2. By using this value and treating the expansion ratio as a controlled parameter, the exit area and pressure were able to be determined. With these values, the thrust of the second stage motor was able to be calculated. This once again leads to an iterative process, where the expansion ratio and the oxidizer to fuel ratio were altered until the resulting thrust was equivalent to the manufacturer provided thrust. This occurred with a mole ratio of 0.01 moles of aluminum to 0.99 moles of ammonium perchlorate which leads to an equilibrium condition as defined in the Table 12. Note that the heat of reaction listed in this Table is per unit mass of propellant, not just the aluminum fuel.

Table 12 Results of Second Stage Cantera Analysis

T_0 (K)	P_0 (MPa)	ρ (kg m^{-3})	c_p (J kg^{-1} K^{-1})	γ	M_m (g mol^{-1})	$h_{reaction}$ (kJ kg^{-1})
1787.93	2.551	4.765	1487.9	1.252	27.76	-1969.1

The other result of the iterative process was utilizing an expansion ratio of 5.103 which not only allowed us to find the exit area but also allowed us to calculate the exit pressure from the isentropic flow tables. With this information the motor performance was able to be calculated and compared to the product information. This comparison is shown in the Table 13.

Table 13 Comparison of Model and Manufacturer Second Stage Motor Performance Data

	$\dot{m}$ (kg s^{-1})	Max Thrust (N)	Average Thrust (N)	I$_{sp}$ (s)
Cantera/ MATLAB	0.3411	769.9		230.1
Manufacturer Data	0.3411	761.3	660.5	210

For the second stage nozzle, the calculated thrust value was closer to the manufacturer provided maximum thrust value. The mass flow rates are consistent between the two models as they are both calculated from manufacturer provided data. The I_{sp} values, though not the same, are closer than for the first stage motor. Through this analysis, it seems highly likely that the motor will perform as commercially advertised, as it was difficult to scale the performance back to match the manufacturer data. This was in part due to the limitations of the analysis, such as not knowing or being able to model the exact propellant chemistry and not knowing the geometry of the nozzle. However, despite these limitations, the motor should easily be able to perform according to the manufacturer provided data.

4.2.2 PTSS Analysis Task 2: Temperature Distribution

The PTSS subteam's second analysis task was to estimate the temperature distribution inside the motor casing during the motor burn. The primary purpose of this analysis was to determine how much heat is transferred to the motor casing and the motor tube, with the goal of ensuring that the motor casing would maintain structural stability during the flight.

The COMSOL Multiphysics software was used to model the flow and heat transfer of the system inside the motor tube. This was accomplished by creating a two-dimensional axisymmetric model in the COMSOL window. This model was studied using a time-dependent conjugate heat transfer model with a laminar flow. This conjugate heat transfer model paired the "Heat Transfer in Solids" interface with the laminar flow interface that allows the system to model the transfer of heat between the flowing propulsive gas, the unreacted propellant, and the aluminum casing. The coupling of these two modules was accomplished using Eq. 51 and Eq. 52 from the nonisothermal flow module in COMSOL.

$$Q_p = \alpha_p T \left(\frac{\partial p_a}{\partial t} + \vec{u} \cdot \nabla p_a \right) \tag{51}$$

$$\alpha_p = -\frac{1}{\rho} \left(\frac{\partial \rho}{\partial T} \right)_p \tag{52}$$

To simulate the system the geometry of the motor casing had to be modeled inside COMSOL. The motor was modeled in four sections, the aluminum motor casing, the unburnt propellant, a volumetric heat source, and the heated gas. The motor that was modeled was the first stage rocket motor, so all sections have a height of 0.4 meters. The thicknesses of the four domains are 35.47 mm for the propellant gas, 2 mm for the volumetric heat source, 5 mm for the unburnt propellant, and 6 mm for the aluminum casing. A picture of the four domains is shown in Figure 76.

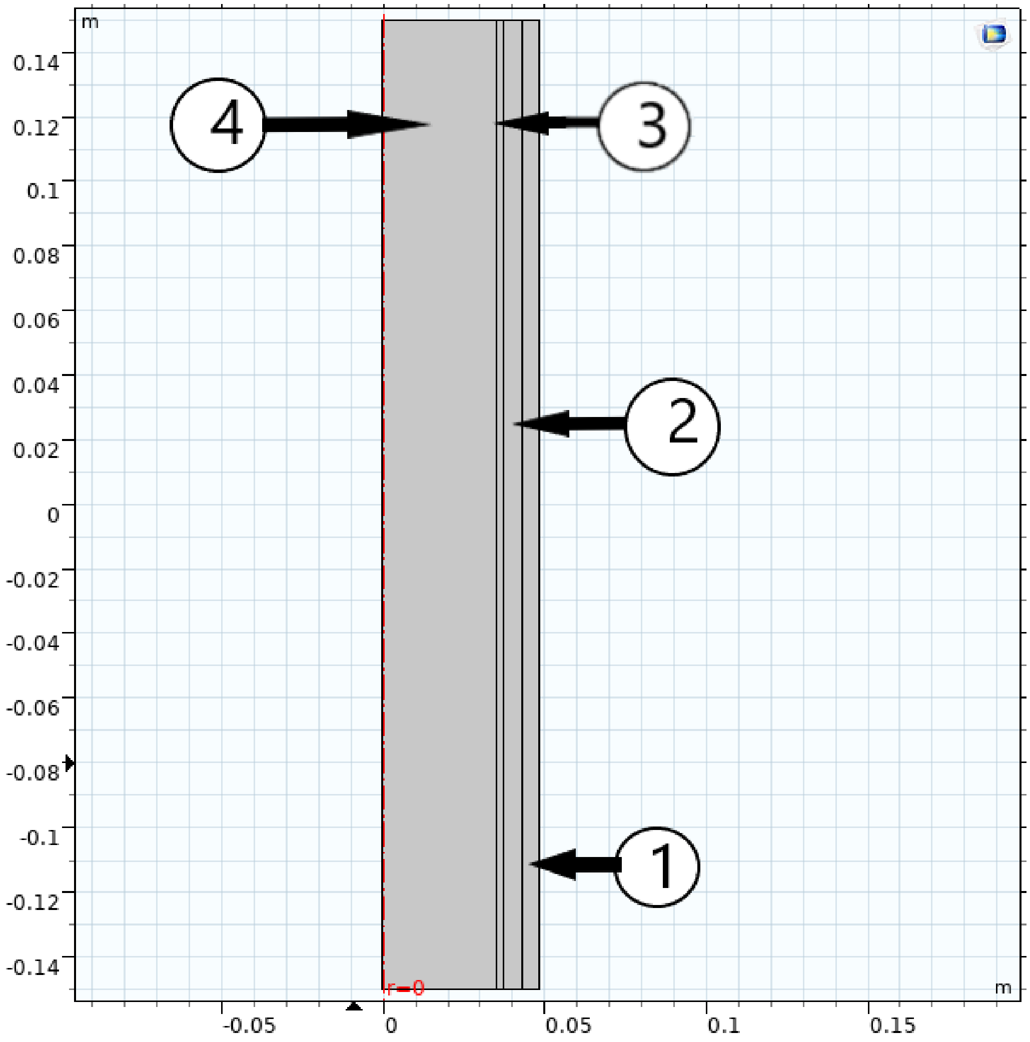


Figure 75 Temperature Distribution Domains

Once the geometry was established, each section needed to be assigned a material to properly model its properties and interactions. The aluminum casing, region one in Figure 76, was assigned the aluminum material property from COMSOL's internal database of materials and their properties. The values provided by COMSOL that were used in the analysis are shown in Table 14.

Table 14 Material Properties of Aluminum used in COMSOL Analysis

Density (kg m^{-3})	C_P (J kg^{-1} K^{-1})	Conductivity (W m^{-1} K^{-1})
2700	900	238

The solid rocket propellant, the reacted propellant, and the volumetric heat source did not have a material in COMSOL's database and therefore a material needed to be created for them. The properties for the solid, unburnt rocket propellant were found from [69] The properties for the volumetric heat source and the propellant gas were found from the Cantera analysis, which provided the density and specific heats of the reacted propellant. The viscosity and thermal conductivity were estimated based upon the viscosities of gases found in the propellant gas and the 2019-2020 MQP projects values [18]. For example, a part of the propellant gas is H_2O, which has a mole fraction of 0.36, has a viscosity of 4.1E-5 Pa s, while O_2, with a mole fraction of 0.28, has a viscosity of 4.47E-5 Pa s, and N_2, with a mole fraction of 0.12, has a viscosity of 3.78E-5 Pa s. The relevant properties of these materials are listed in Table 15.

Table 15 Properties of Materials in the Thermal Modeling

Material	Region	Density (kg m^{-3})	C_P (Jkg^{-1}K^{-1})	Conductivity (Wm^{-1}K^{-1})	Gamma	Viscosity (Pa s)
Unburnt Propellant	2	1949	1608.5	0.4054	N/A	N/A
Propellant Gas	3, 4	4.617	1540.9	0.03	1.239	$3.78*10^{-5}$

With the material properties and the geometry defined, the model was used to determine the temperature distribution and the heat flow through the casing.

4.2.2.1 Heat Transfer Analysis

The COMSOL heat transfer interface allowed for the modeling of heat transfer through conduction, convection, and radiation. To complete this analysis, the four regions of the geometry needed to have defined values and parameters, including the material properties that were described earlier. These regions also need to be defined as either a fluid or a solid region, which each have their own distinct set of critical properties. These regions further needed initial values for the temperature, to have a starting point for the modeling. For the fluid regions, regions 3 and 4, this temperature was 2065.78 K, which was a direct result from the Cantera analysis. The solids, on the other hand, were set at 293.15 K, or room temperature.

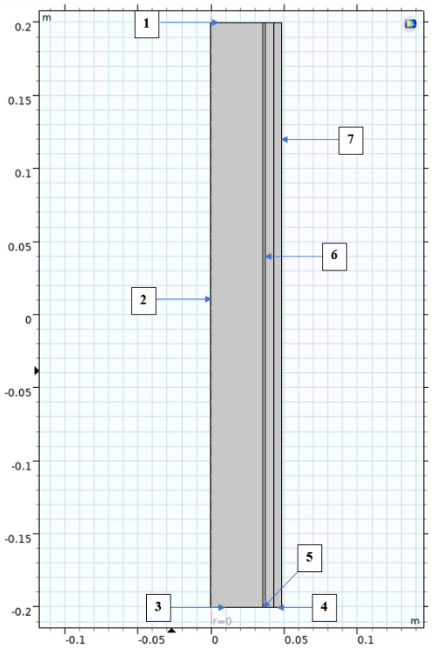


Figure 76 Boundary Conditions in COMSOL Model

To properly model the system, boundary conditions needed to be created and regions needed to be defined. Since the purpose of this study is to find heat transferred through and into the aluminum casing along the side of the motor, a boundary condition was created at the top and the bottom, boundaries 1, 3, 4 and 5 in Figure 77, to insulate heat transfer. This boundary condition applies Eq. 53 to ensure that no heat was lost over the area at the top or bottom of the system.

$$-\vec{n} \cdot \vec{q} = 0 \tag{53}$$

The next boundary condition applied is for a heat flux on the outside of the aluminum motor casing, boundary 7 in Figure 77. This heat flux condition models the loss of heat through the system that the aluminum motor casing would experience due to convection to the surrounding air. This boundary condition was applied by using the “Convective heat flux” model for the heat flux. From there external natural convection was chosen as the proper way to model the aluminum casing inside the rocket body. Wall height was set for the height of the aluminum casing, 0.4 meters, and the absolute pressure and external temperature were set to 1 atmosphere and 293.15 K, respectively. Because the surface of the casing is within the Blue Tube, there is no relative air velocity for the convective heat transfer. These parameters define the value of h that is used in Eq. 54.

$$q_0 = h(T_{ext} - T) \tag{54}$$

The final boundary condition applied to the system was axial symmetry which mirrored all regions and boundaries about the central axis. This boundary condition was applied to boundary 2 in Figure 77.

The driver of the model is the volumetric heat source that simulates the reacting propellant and provides constant heat generation for the motor casing. This volumetric heat source applies Eq. 55 to region 3 in Figure 77 so that it consistently produces the heat of reaction. This is modeled by Eq. 55.

$$Q = Q_0 \tag{55}$$

The value that was used for Q_0 was calculated based on the enthalpy of reaction and mass flow rate from the Cantera analysis. The enthalpy of reaction is in joules per kilogram of propellant as Cantera utilizes both the oxidizer and the fuel mass in the calculation of the specific enthalpy. By using Eq. 56, the value for the general heat source was set to 4.51E9 W m^{-3}.

$$Q_0 = \frac{h_{reaction}\dot{m}}{V} \tag{56}$$

where V is the volume of the volumetric heat source, $\dot{m}$ is the mass flow rate of the exhaust propellant, and $h_{reaction}$ is the enthalpy of reaction for the propellant in J kg^{-1} of propellant. With the boundary conditions and the regions fully defined the time-dependent study was conducted. This study used Eq. 57 and Eq. 58 to determine the heat transfer and final temperature of the system.

$$\rho C_p\left(\frac{\partial T}{\partial t}\right)+\rho C_p\vec{u}\cdot\nabla T+\nabla\cdot\vec{q}=Q \tag{57}$$

$$\vec{q}=-k\nabla T \tag{58}$$

In these equations ρ is the density of the material, C_p is the heat capacity at a constant pressure, k is the thermal conductivity, and $\vec{u}$ is the velocity vector of the fluid. The velocity vector is provided by the laminar flow study, discussed below. The density of the propellant gas is determined using the ideal gas law, as shown below with R being the universal gas constant.

$$\rho=\frac{M_m p}{RT} \tag{59}$$

4.2.2.2 Laminar Flow Analysis

The laminar flow interface in COMSOL uses the equations for conservation of mass, energy, and momentum to model the flow of fluids in a system. As opposed to the heat transfer interface, the laminar flow interface requires definitions for just the fluid regions of the system, the volumetric heat source and the propellant gas. These regions are defined by the material properties of the fluids, as discussed earlier. Furthermore, to properly model the rocket motor chamber, the initial chamber pressure needed to be set, as pressure in the chamber can change due to temperature fluctuations and fluid flow phenomena. This initial value was set to 2.830 MPa, which was the pressure produced by the Cantera analysis.

There were four boundary conditions used for the laminar flow study. The first was a no-slip wall condition at the top of propellant gas, as well as along the top and bottom boundaries of the volumetric heat source. These walls can be seen as boundaries 1 and 5 in Figure 77. This was to prevent the gas from flowing out of these boundaries, as well as to accurately model the system. Eq. 60 applied at these boundaries is below.

$$\vec{u} = 0 \tag{60}$$

The inlet to the system was modeled as a constant pointwise mass flux that simulates the propellant gas entering the system from the burning propellant walls. This inlet occurred on boundary 6 in Figure 77 This is calculated using Eq. 61, with M_f being the mass flux in kg m^{-2} s^{-1}.

$$\vec{u} = -\frac{M_f}{\rho}\vec{n} \tag{61}$$

$$M_f = \frac{\dot{m}}{A} = \frac{\dot{m}}{2\pi r h} \tag{62}$$

The mass flux was calculated from the mass flow rate, $\dot{m},$ found in the Cantera analysis using Eq. 62, with A being the surface area of the boundary.

The final boundary condition, other than axial symmetry, which was applied to boundary 2 in Figure 77, is applied to the outlet at the bottom of the combustion chamber, boundary 3. The boundary condition that was applied was an outlet pressure set to the pressure found from the Cantera analysis, 2.830 MPa. This pressure boundary layer implemented Eq. 63 and Eq. 64 on the lower boundary, to ensure that the pressure at the outlet was 2.830 MPa.

$$\left[-p\vec{I} + \vec{K}\right]\vec{n} = -\widehat{p_0}\vec{n} \tag{63}$$

$$\widehat{p_0} \leq p_0 \tag{64}$$

Eq. 64 represents the prevent backflow condition that was implemented on the boundary, to prevent the flow from returning into the chamber. In these equations p_0 is the set pressure at the boundary, $\widehat{p_0}$ is the pressure outside the boundary, $\vec{K}$ is the viscous shear stress tensor, $\vec{I}$ is the identity matrix, and $\vec{n}$ is the direction normal to the boundary.

With the boundary conditions, flow regions, and initial conditions set, COMSOL applied these conditions and layers to model the laminar flow. As stated above, this model uses the conservation of mass, energy, and momentum equations to determine the flow characteristics and flow field.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0 \tag{65}$$

$$\rho \left(\frac{\partial \vec{u}}{\partial t} \right) + \rho (\vec{u} \cdot \nabla) \vec{u} = \nabla \cdot \left[-p\vec{I} + \vec{K} \right] + \vec{F} \tag{66}$$

$$\vec{K} = \mu (\nabla \vec{u} + (\nabla \vec{u})^T) - \frac{2}{3} \mu (\nabla \cdot \vec{u}) \vec{I} \tag{67}$$

4.2.2.3 COMSOL Analysis Results

The results of the COMSOL simulation yielded flow information and temperature distributions inside the chamber. The results of the fluid flow analysis are presented in Figures 78 and 79, which display the flow velocity in the chamber as well as the streamlines of the flow. These streamlines highlight how the fluid flows in from the walls and out the bottom of the chamber. This result makes intuitive sense as the propellant that is burnt would travel into the chamber and into the nozzle, which is not included in the model.

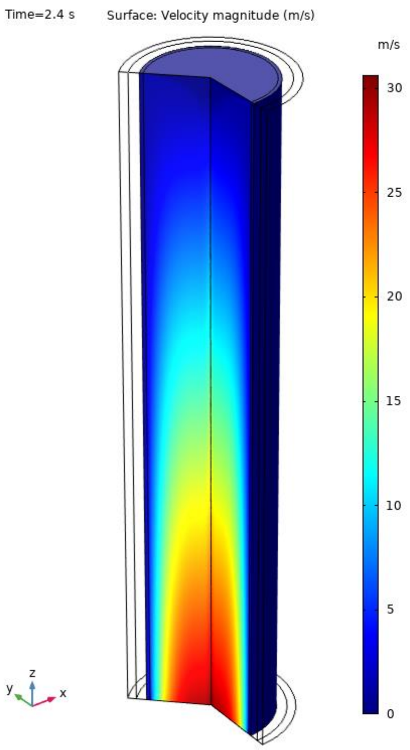


Figure 77 3D Velocity Gradient in chamber

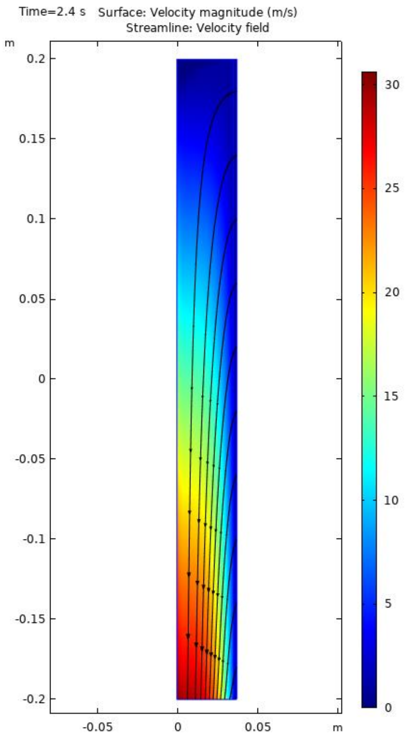


Figure 78 Streamlines and 2D Velocity Gradient

The temperature distribution inside the chamber is displayed in Figure 80. The temperature distribution indicates the effect of the volumetric heat source and the effect it has on the temperature of the fluid and the surrounding solids.

Since the fluid in the chamber is treated as an ideal gas, the density in the chamber varies with the temperature and the pressure. The pressure inside the chamber does not have a significant variation as can be seen in Figure 81. Because of this the variations in density closely match the isothermal curve lines, which is highlighted in Figure 82.

The main goal of this analysis task was to determine the temperature of the aluminum casing to ensure that it would remain structurally sound. Due to the large range of temperatures in the motor, it is difficult to see the temperature distribution inside the aluminum casing. Because of this the aluminum casing was isolated to highlight the temperature gradient and values inside the casing. This can be seen in Figure 83.

These values for the temperature are quite low and do not make physical sense initially. However, due to the way the model is set up, the unburnt rocket propellant remains the same thickness for the entire burn time. Due to the unburnt propellant’s low thermal conductivity, the heat is not transferred easily inside the propellant, leading to a low temperature at the inside boundary of the casing. This can be seen in Figure 84, which is a sliver of the model. If the model was able to incorporate a moving boundary, then the aluminum casing may have a higher temperature, due to a higher temperature at its inner boundary.

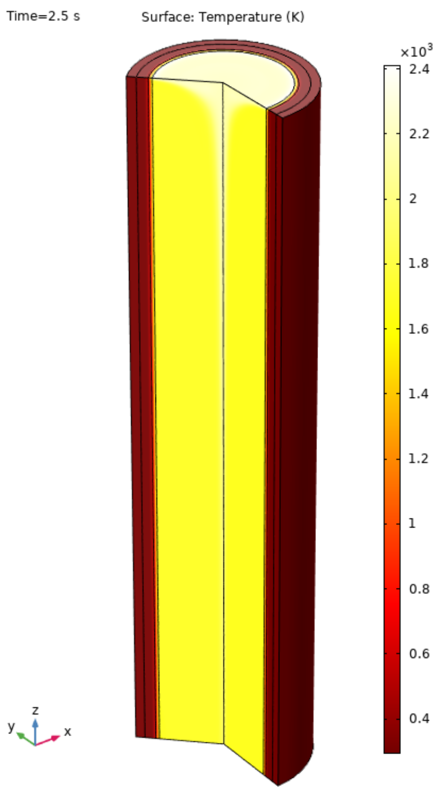


Figure 79 Temperature Distribution in Chamber

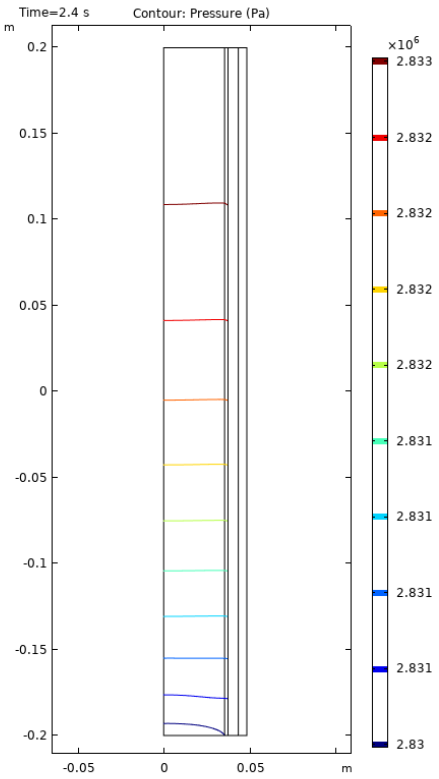


Figure 80 Pressure Distribution in Chamber

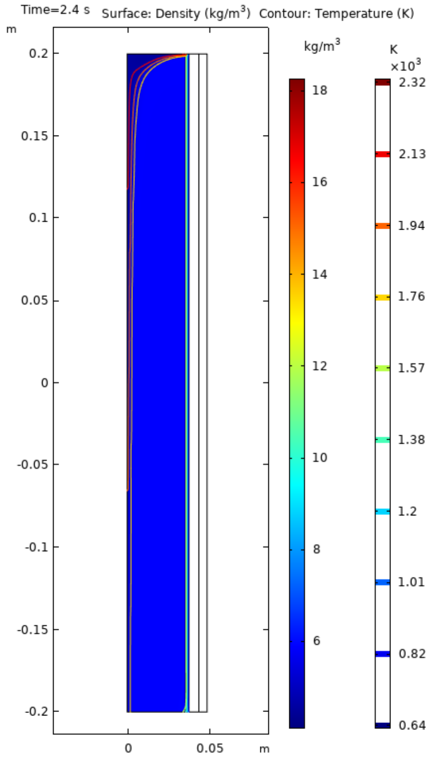


Figure 81 Density Gradient with Isothermal Lines

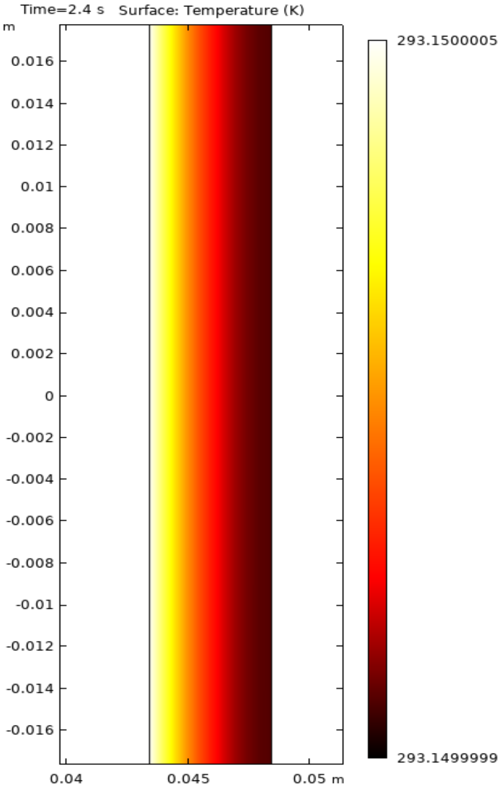


Figure 82 2D Temperature Distribution inside Aluminum Casing

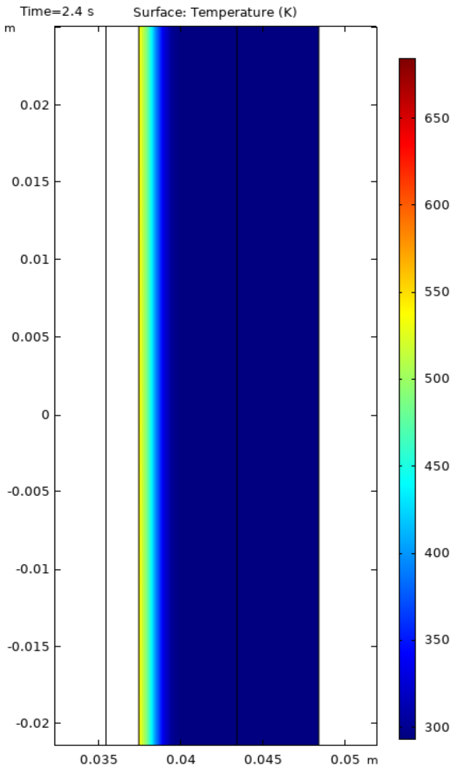


Figure 83 2D Temperature Distribution inside Unburnt Propellant (Left) and Aluminum Casing (Right)

4.2.3 PTSS Analysis Task 3: Mechanical Separation System Model

The goal of PTSS Analysis Task 3 was to create a simplified model for the spring-based air brake stage separation system which was used to determine an optimal airbrake/spring configuration. In other words, the goal was to analytically solve a torque-based physics problem where one side of the moment arm is an aerodynamic surface in an airstream, and opposing that on the other side of the moment arm is a spring force. Variations in air brake width and spring strength were considered, with the goal to provide an optimal drag value ensuring a sufficient separation distance after a given time.

First, design constraints were defined. Based on an estimate of the speed at first stage motor cutoff, an airspeed of approximately 100 m/s was used for analysis. A sufficient separation distance of 10 meters was selected for the second stage motor to ignite without damaging the first stage. By assuming a half-second pause before second stage engine ignition it was found that the air brakes needed to produce a negative acceleration of approximately 20 m/s^2. The mass of the spent first stage is approximately 6.86 kg. Using Newtons' Second Law, the force required to produce this negative acceleration is calculated.

$$F = ma \tag{68}$$

All four air brakes combined must produce approximately 140 N of drag force, with each individually producing approximately 35 N. In addition, the volume budget for the airframe dictated a length of six inches for the aerodynamic surface and the design called for four airbrakes. Then, width limitations were placed on the aerodynamic surface size. The minimum and maximum width of the air brake was determined to be between ten degrees and ninety degrees, measured as an angle from the rocket's center.

Next it was necessary to create the torque force diagram based on the geometry of the design, modeled in Figure 84. The torque center is the black dot, with the red lines representing the moment arms and spring location. It is important to note the torque from the air brake will come from Fluent, so the force diagram is mostly for the spring requirements.

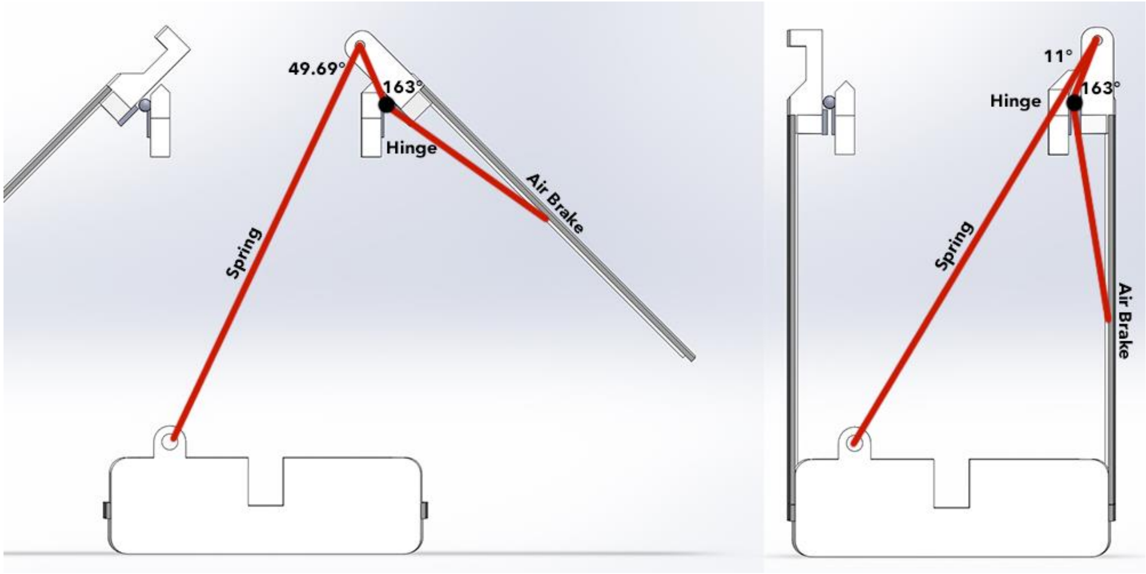


Figure 84 Air Brake Force Diagrams Open (left) Closed (right)

In the open configuration the spring makes a 49.69° angle with the moment arm, and in the closed configuration the spring makes an 11° angle with the moment arm. With constraints set, an Ansys Fluent analysis was carried out on five air brake variants, shown below in Figure 85. These variants are categorized by the angle they make with their geometric center. A larger angle would indicate a wider width. Since there are four air brakes, the upper limit for an air brake variant would be 90°.

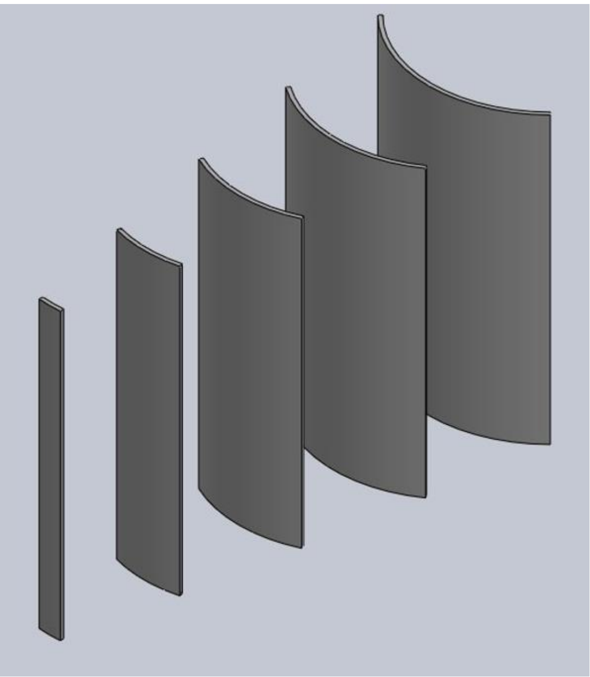


Figure 85 Air Brake Variants

From left to right in Figure 85 airbrakes of widths 10°, 30°, 50°, 70°, and 90° were considered. Each was 3D modeled and inserted into an airstream fluid domain in Fluent at an angle of attack of 45°, as seen in Figure 86. The airstream inlet is represented on the left in blue, and the outlet is represented on the right in red. In SOLIDWORKS, the distance from the geometric center to the center of rotation was measured and inputted into Fluent. Fluent uses this offset to produce drag force and moment calculations. The results of the calculations are below. The full graphs produced by Fluent are included in appendix I.

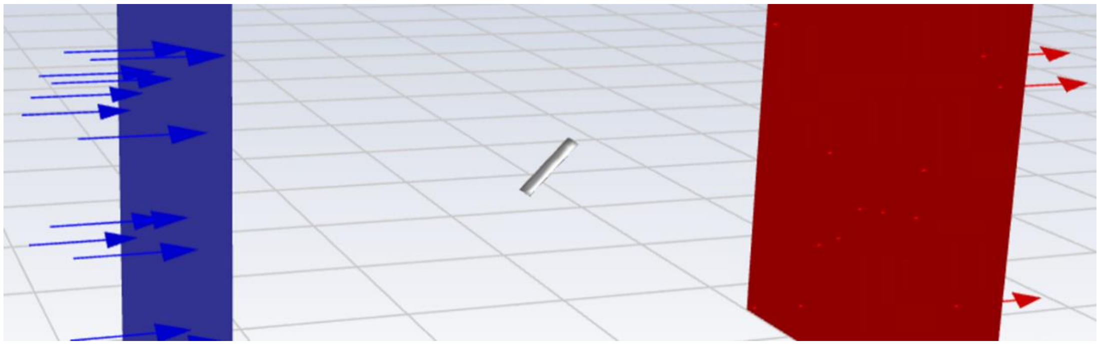


Figure 86 50° Air Brake Inside Ansys Fluent Simulation

Table 16 Air Brake Ansys Fluent Results

Variant	**Drag Force**	**Moment**
10°	9.58 N	3.34 Nm
30°	27.44 N	5.18 Nm
50°	47.96 N	8.82 Nm
70°	63.40 N	11.93 Nm
90°	74.98 N	14.32 Nm

Recall that the desired moment force for an individual air brake is around 35 N. Using the results from Fluent, the ideal air brake variant would be between 30° and 50°. As a result, the analysis was repeated with a 40° air brake variant. Recalculating produces a drag force of 39.09 N with a moment of 7.16 Nm. With four air brakes, this produces 156.36 N of drag force. This is close to the target goal.

The next step was to determine the spring force required to keep the air brake open in this air stream. This means that the spring strength has to produce an opposing moment force greater than 7.16 Nm. The design constraints for the spring were a minimum length of 174.24 mm and a maximum length of 184.15 mm. Keep in mind the spring at its shortest required length must still produce enough force to counter the aerodynamic torque, so the designed minimum length of the spring will be shorter than 174.24 mm. By placing this moment requirement into the force diagram and by using basic trigonometry, we found that the spring needed to have a strength of 364.4 N at a length of 174.24 mm.

Therefore, an air brake of width 40° and a length of 6 inches, and a corresponding spring with a force of 364.4 N at an extension length of 174.24 mm were found to be an optimal pair. To verify the air brake geometry, the total drag forces produced by the air brakes were substituted back into the original acceleration equation and were found to produce -22.79 m/s^2 of relative drag acceleration. To verify that a realistic spring existed with the specified constraints, a search on McMaster-Carr was completed and a spring was found which met the criteria. This selected spring is featured shown in Figure 87.

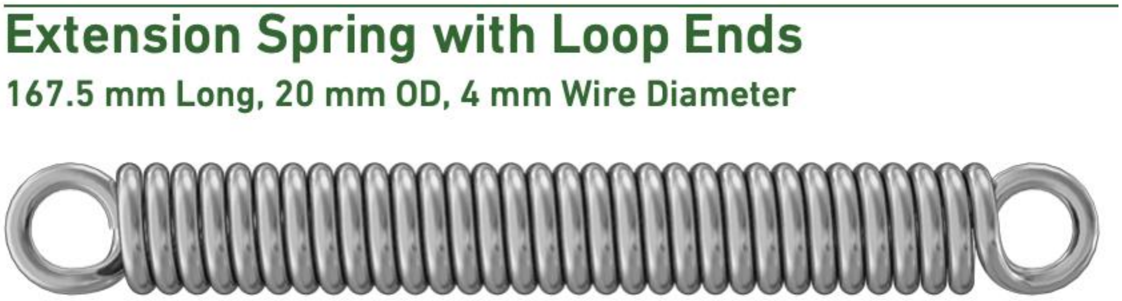

Extension Spring with Loop Ends
167.5 mm Long, 20 mm OD, 4 mm Wire Diameter

Figure 87 Extension Spring Selected for Air Brakes, © McMaster-Carr Extension Spring 3630N389, 2020

After a spring resembling design specifications was found, analytical calculations were performed on the selected spring to test the advertised values. Using the online data sheet provided by McMaster-Carr, the spring constant *k* could be estimated using the common spring constant, Eq. 69.

$$k = \frac{Gd^4}{8nD^3} \tag{69}$$

Where G is the modulus of rigidity of the material, d is the wire diameter, n is the number of turns or coils, and D is the coil diameter – the wire diameter. The variables provided by McMaster-Carr are in

Table 17 below.

Table 17 Analytical Spring Constant "k" Variables Provided by McMaster-Carr

Variable	**Value Provided**
Modulus of Rigidity - G	79,000 Mpa
Wire Diameter - d	4 mm
Number of Coils - n	36
Coil Diameter - D	20 mm
Minimum Force for Displacement - F_0	242.2 N

After substituting this into the equation for the spring constant *k*, it is found that *k* is equal to 19357.83 N/m. Compare this advertised spring constant to Hooke's Law with the selected spring force of 364.4 N (minus the minimum force required for extension) at a displacement of 6.74 mm:

$$F - F_0 = -kx \tag{70}$$

$$\frac{-F}{x} = k \tag{71}$$

After substituting this into Eq. 71 for the spring constant, *k*, it is found that *k* is equal to 18130.56 N/m, which indeed is close to the desired spring. This also proves that the selected spring will be stronger than the project's requirement, ensuring a strong deployment. This is the final verification that proves the selected McMaster-Carr spring will meet our design criteria.

Overall, this analysis shows that a 40° air brake and spring pair can be designed to provide a sufficient drag force of 156.36 Newtons to separate Stage 1 and Stage 2 about 11.39 meters in just one half-second without inducing any forces on the second stage. This meets the design requirements of stage separation with minimal disturbance (instability) and rapid stage distancing before engine lighting.

4.2.4 PTSS Analysis Task 4: Electrical Match Model

The goal of this analysis task was to create a computational model that determined how long it would take for ignition to occur in an electric match. First, the COMSOL simulation for the electrical match time-dependent temperature analysis was done. To determine the ignition time of the pyrogen in an E-match, the Electromagnetic (or Joule Heating) module was used. To begin, the physical parameters of the 3D model were defined.

Table 18 COMSOL E-Match Physical Parameters

Component	**Length (mm)**	**Radius (mm)**
Lead Wire (copper)	25	.644
Bridge Wire (Nichrome)	4	.0305
Pyrogen (Boron – Potassium Nitrate)*	N/A	2.5
Gap Between Lead Wires	1.712	N/A

*Some Material Properties Unavailable

As seen in Table 18, the small geometric parameters used allowed for a realistic scale model to be constructed. These were determined by using measurements from real E-Matches found from various sources and creating an appropriate representation.

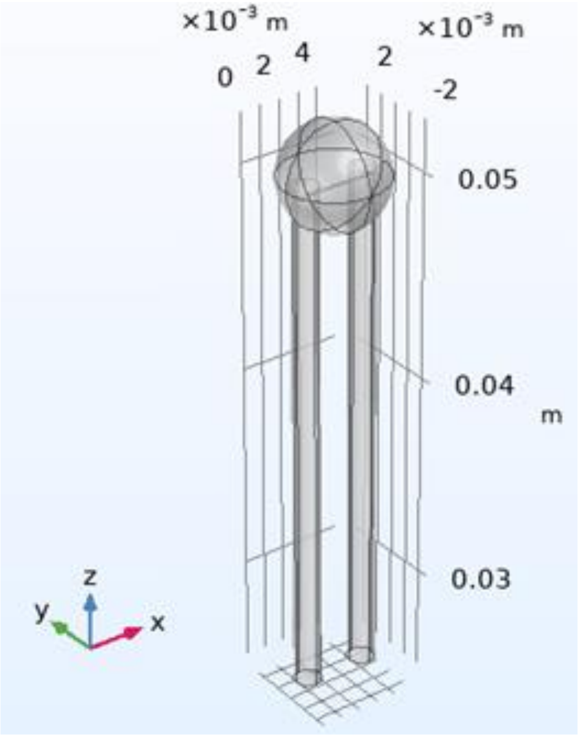


Figure 88 E-Match 3D Scale Model

From an electric match construction article from Skylighter, the nichrome bridge wire should be between 48 to 51 gauge, the copper lead wires are usually 22 gauge, and the ‘match head’ should be 2.5mm thick [70]. In mm, 48 gauge can be expressed as 0.0305mm and 22 gauge is 0.644mm. It can be seen in Figure 88 above but, is important to note that this is a very small scale and as a result, the current flowing through these wires when 9V is applied produce very high temperatures. To begin defining the physics, first the proper boundary conditions needed to be defined.

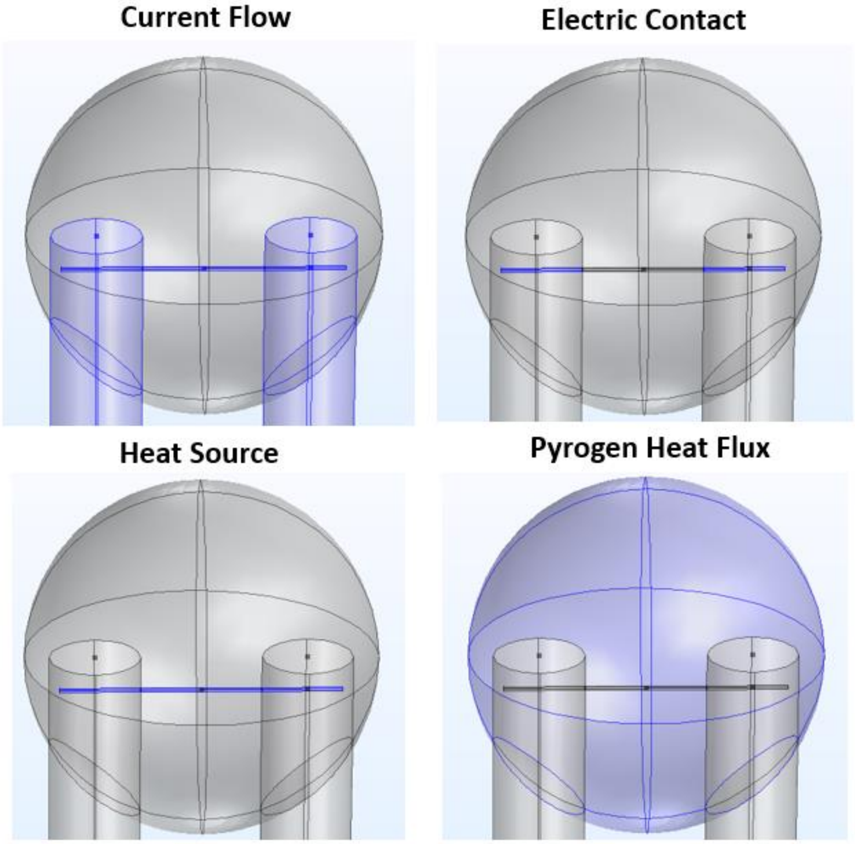


Figure 89 E-Match Boundary Conditions

In Figure 89, all of the boundary conditions used in the simulation are visualized. Electromagnetic Heating in COMSOL can be split up into the 'Electric Current' and 'Heat Transfer in Solids' modules. The Electric Current module manages all of the electrical parameters involved in the module. Here, current conservation and electric insulation conditions are applied to all applicable surfaces. An initial electric potential of 9 V is applied to the open face of the bottom of one of the lead wires. The opposite lead wire's open face is set to a potential of 0 V (ground). This establishes a current to flow through one lead wire, the bridge wire, and then the next lead wire. Electrical contact is then defined between all of the surfaces that are in contact between the bridge wire and the lead wires.

Then, the Heat Transfer in Solids module is configured. Here, thermal insulation and heat transfer capability is applied to all applicable domains to include the entire system. Initially, every domain is at atmospheric temperature (293.1K). First, a boundary heat source is defined to include every surface of the bridge wire within the sphere of pyrogen. This ensures that the main heat source distributing to the pyrogen is from the bridge wire. The wire's resistance is calculated through the wire's length, cross-sectional area, and resistivity. In COMSOL, the resistivity of Nichrome is provided and calculated as a function that increases linearly with temperature. The value provided is 1.042 • 10^6 Ω·m at 293 K and 1.195 • 10^6 Ω·m at 773 K.

$$R = \frac{\rho L}{A} \tag{72}$$

Here, the resistance is calculated as 15.36 Ohms. The power is then calculated from the wire's resistance and current.

$$P = I^2 R \tag{73}$$

The power output from the bridge wire is 5.28 Watts. The heat source q_0 is then defined as the power divided by the bridge wire's surface area.

$$q_0 = \frac{P}{2\pi r L} \tag{74}$$

For the bridge wire, the heat source is calculated as 6.88 • $10^6 \, \frac{W}{m^2}$. In Figure 90, the 'Heat Source' image in the bottom left shows where this value is used. Similarly, a heat flux is introduced throughout the pyrogen that is calculated as the wire's power divided by the pyrogen's surface area. The radius here is the radius of the pyrogen ball.

$$q_0 = \frac{P}{4\pi r^2} \tag{75}$$

This heat source is smaller and calculated as 6.72 • $10^4 \, \frac{W}{m^2}$. In Figure 90, the 'Pyrogen Heat Flux' image in the bottom right shows where this value is used. The main distinction is that current is allowed to flow through all of the wires whereas no current is flowing through the pyrogen. These conditions are all combined and are applied to every applicable domain through the Multiphysics Electromagnetic heating conditions.

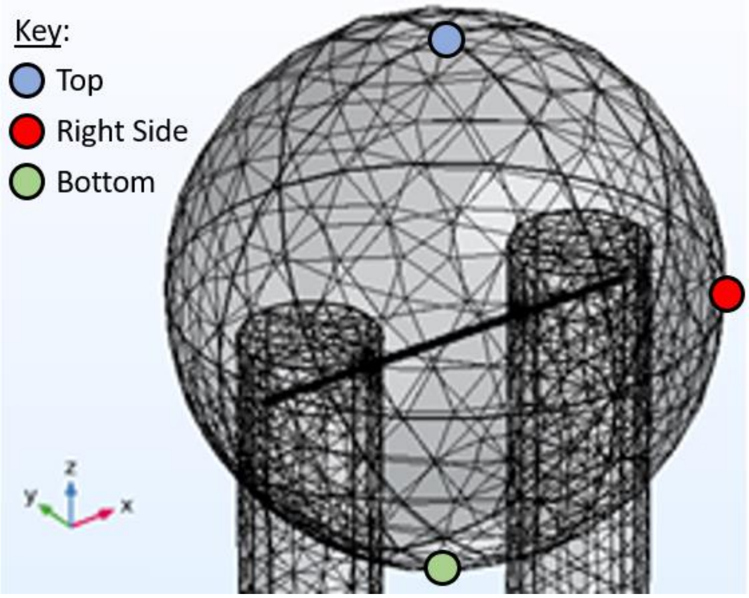


Figure 90 Zoomed in E-Match Mesh

The mesh is then constructed, as shown in Figure 90, in a way that the bridge wire still has enough detail but, the lead wires and pyrogen are not too complex. The results consist of a series of graphs that track specific points on the E-Match. The most desired piece of information is how does the temperature within the pyrogen changes over time.

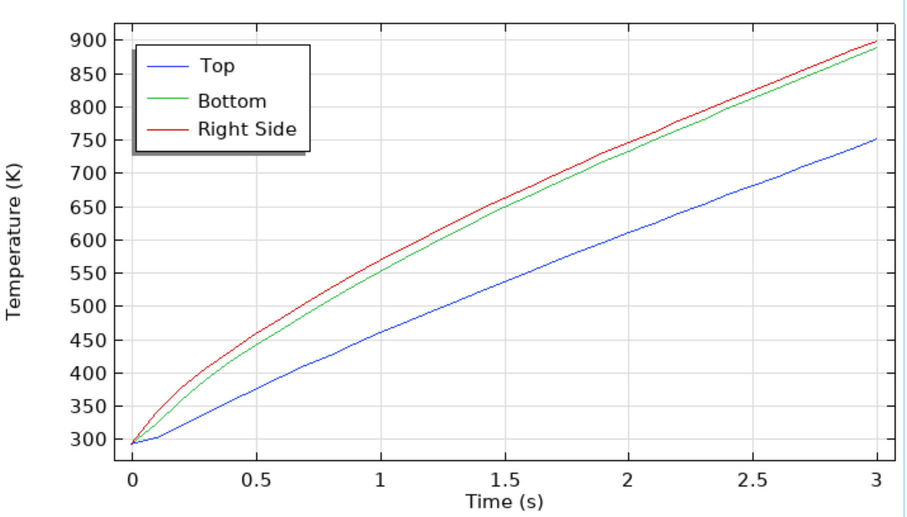


Figure 91 Temperature of Pyrogen at Multiple Points on its Surface

It turns out, the temperature is not the same everywhere in the pyrogen as seen above in Figure 91. It is also important to note what the temperature is in the lead wires and in the bridge wire.

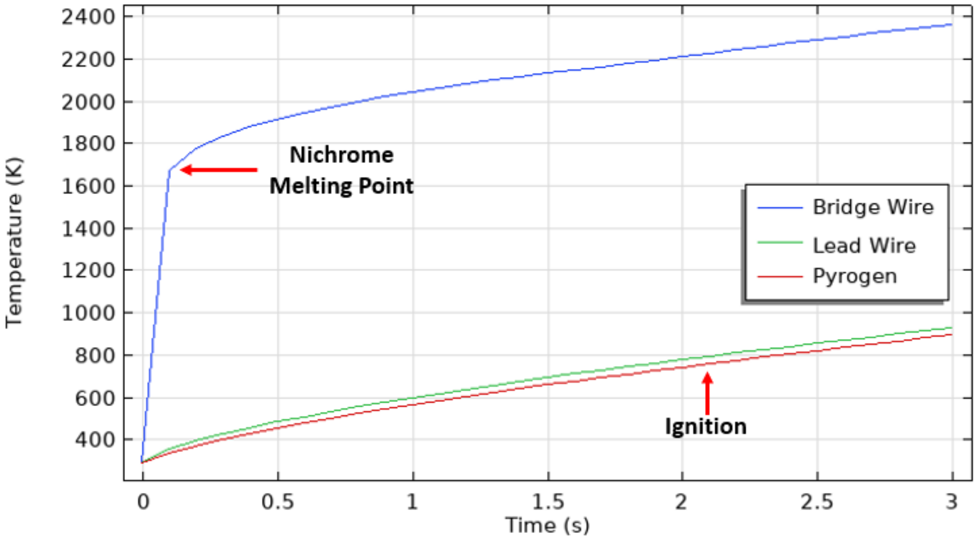


Figure 92 Temperature of E-Match Components

Figure 92 shows multiple notable points. First, the temperature in the bridge wire is significantly higher than in the pyrogen and lead wires. This shows that it is in fact the only source of heat in the pyrogen because, much higher temperatures are needed in order for significant heat flow to occur across the much larger pyrogen structure. Second, the lead wires are not a significant heat source but, they do still contribute some heat dissipation and evidently, the temperature in them increases gradually over time. Another notable result is the current in the bridge wire. With an auto-ignition temperature of 788K for Boron Potassium Nitrate [71], the model shows that it approximately takes 2.1 seconds for the complete ignition of the electric match. This is taking the surface temperature of the pyrogen into consideration though. Another option is looking at the temperature of the pyrogen closer to the bridge wire.

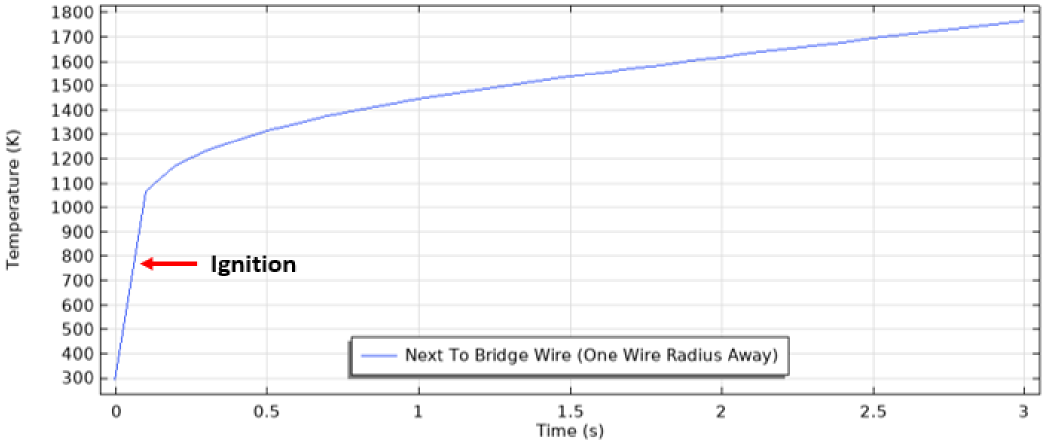


Figure 93 Temperature of Pyrogen in Proximity to Bridge Wire

If this profile, shown in Figure 93, is taken into consideration, the electric match would take approximately 0.1 seconds to ignite – significantly faster. This could be verified by experimentation, but it is safe to assume that the longer time approximation of 2.1 seconds is the most accurate for this model. It is also important to note what the current is within the bridge wire.

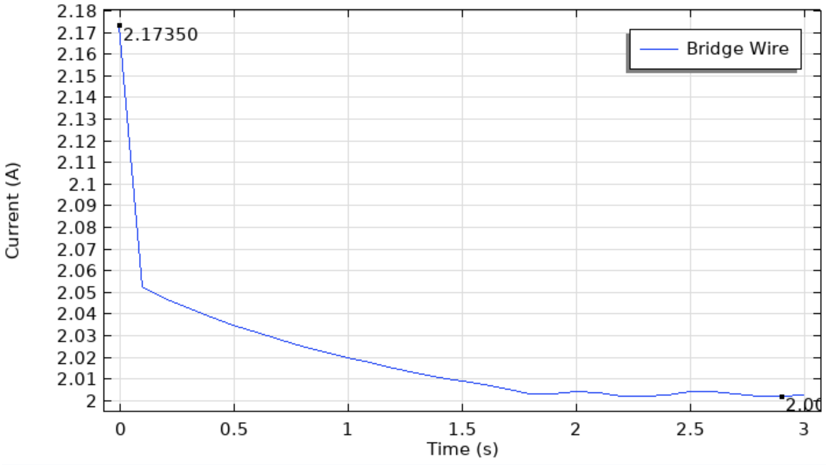


Figure 94 Current in Bridge Wire vs. Time

As shown in Figure 94, there is a slight current drop, but it stays relatively constant at approximately 2 Amps throughout the simulation. One of the more visually useful results is a cross section of the pyrogen showing how the heat is distributed throughout the pyrogen as a function of time.

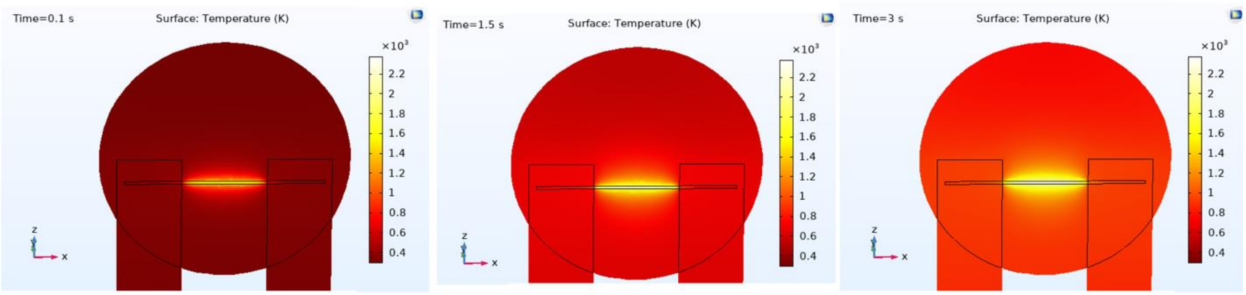


Figure 95 Temperature Distribution of Pyrogen Cross Section

From Figure 95, it is clear that the heat source is located primarily around the bridge wire in the center of the pyrogen. After some time, the heat dissipates evenly throughout the pyrogen. This is where the heat transfer through the pyrogen comes into play. Based on the pyrogen properties (mainly the thermal conductivity and heat capacity) the heat can flow quickly or slowly increasing the pyrogen temperature until ignition occurs.

4.3 Flight Dynamic Analysis

The FDA subteam was responsible for three analysis tasks. These tasks included creating a model that can be used to estimate the rocket’s attitude dynamics, as well as its trajectory, evaluating the aerodynamic forces on the vehicle, and evaluating the forces and moments on the different fins designed for the rocket.

4.3.1 FDA Analysis Task 1: Vehicle Dynamics and Performance Model

Using the established methodology that outlines the mechanics of the system as described in Section 3.3.1, an analysis code was developed in MATLAB. For ease of development, the analysis was broken up into several subroutines that activated in response to specific time or state triggers. Figure 96 The Three Distinct Stages of the Code below breaks down the discrete stages in the code, as well as the triggers that cause the process to switch from one stage to the other.

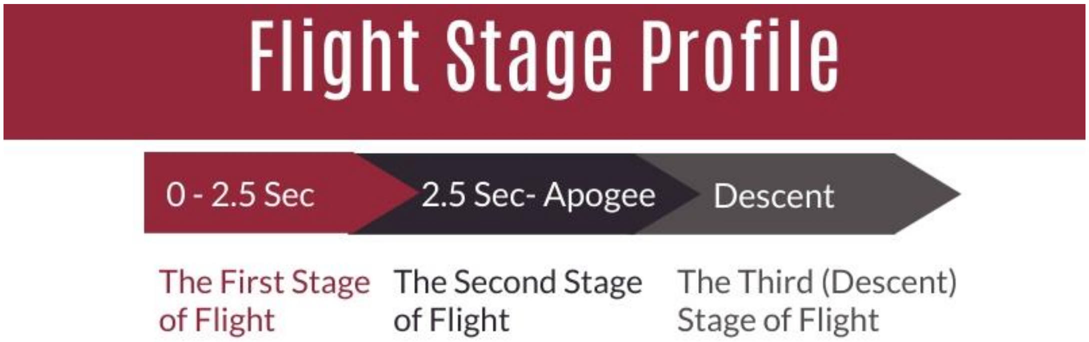

Flight Stage Profile
0 - 2.5 Sec
2.5 Sec- Apogee
Descent
The First Stage of Flight
The Second Stage of Flight
The Third (Descent) Stage of Flight

Figure 96 The Three Distinct Stages of the Code

The codes required for this task are listed in Table 19:

Table 19 Code Required for This Task

Code Name	Code Function
Main_Code.m	Main code that initializes and runs the subfunctions that simulate the flight

Point_Mass_ascent_Stage_1.m	Subroutine that simulates the flight during Stage 1
Point_Mass_ascent_Stage_1.m	Subroutine that simulates the flight from the end of Stage 1 to apogee
Body2Inertial.m	Subroutine that converts the body-fixed coordinates of the rocket to the Earth Inertial Frame
Inertial2Body.m	Subroutine that converts the Earth Inertial Frame to the body-fixed coordinates of the rocket
Atmo.m	Calculates the density of air at each altitude
Autorotation_Function	Modified Version of the autorotation solver from 4.1.2. Runs as a function that requires input of altitude to initialize

The main body of the code (See Appendix E) shows the initialization of the relevant parameters. In order to ensure the code executes properly, a small artificial starting velocity is considered. This allows the code to begin smoothly. Next, the rocket mass characteristics at each stage step are specified. This allows for cross-checking the values found through the simulation, as well as changing the projectile's mass after the separation of Stage 1. The thrust profiles of each motor were created and inserted into the state simulator.

The initial objectives of this task included an attitude solution to be included in the primary rocket solution as well. However, due to time constraints, this task was not completed, but the principal methodology employed, that of the quaternion reference frame definition, is still included in the code. The quaternion is still used for reference frame transformations and the attitude angles form the basis of the respective reference frames, so these values remain in the solver.

This solver only considers the translational motion of the rocket and treats the attitude as a constant value, that is that the vehicle long axis is always aligned with the velocity vector. The initial angles specified by the launch rail are assumed to be constant and are evaluated as the “true attitude” for converting the thrust generated by the rocket into the inertial earth frame.

After specifying the initial attitude of the rocket on the launch rail, as well as all previous parameters, the quaternion and subsequent coordinate frame transformation matrix are defined, and the system solver is ready to be employed.

Appendix E shows the main code that is utilized for the ascent stage of the rocket which is broken down into the stages shown in the flight stage profile in Figure 96 The Three Distinct Stages of the Code. The first stage lasts for the length of the burn time of the first motor, in this case, 2.5 seconds. After 2.5 seconds have elapsed, the solver switches to the Stage 2 solver and simulates both the stage separation and the motor burnout. In this case, the timespan is 4.7 seconds from separation to burnout. One second for the separation process, as outlined in Section 4.2.3, and 3.7 seconds for the motor burnout. After burnout, the same protocol is used, but the motor thrust contribution is reduced to zero and the rocket flies until it reaches apogee. At apogee, the code executes the descent solver specified in 4.1.2, “Autorotation_Function”.

With the main execution code being well defined, next is the specific subroutines that run the simulation. The first subroutine was the “Point_mass_ascent_Stage_1.m” file which handled the flight of the rocket during the first stage, in this case for the first 2.5 seconds (see Appendix 7.5.2). It required the inputs of the burn time of the first stage, the specific impulse, or ISP, of the first stage motor, and the thrust profile of the first stage motor. This subroutine first defines the drag coefficient as well as the surface area of the rocket that is experiencing the drag. Next, the quaternion is defined. Due to the lack of an attitude solver, the attitude was assumed constant, so the quaternion values, and subsequently the rotation matrix between reference frames, are unchanging. The air density is calculated as a function of the altitude, but the gravity was assumed constant due to the relatively low apogee value, so there is not a significant reason to consider the gravity gradient change.

Using the mechanics outlined in the FDA Task 1 Methodology, as described in Section 3.3.1, the subroutine evaluates the time derivatives of the position and velocity states. The subroutine uses the thrust profile provided by the PTSS subteam for the first motor and interpolates the values to find an approximate thrust at each time in the simulation. The time derivative of the position is expressed as the velocity at each point; resulting in an easily defined derivative. Next, the accelerations in the body-fixed X, Y, and Z directions are specified as described in Section 3.3.1. The Y and Z directions only experience acceleration due to the force of gravity, while the X direction experiences acceleration due to thrust, drag, and gravity.

$$Force_{Wind} = S\rho V^2 \tag{76}$$

S represents the surface area of the rocket that is being impacted by the wind, ρ represents the air density, and V represents the velocity of the wind, a random value. The value of surface area resolved by the solver is a value given by the ARS team. For the solver, the wind is initially defined as a random unit vector in the inertial frame by the main code. This direction is assumed as the consistent wind direction, and any wind force contributions will occur in this direction. The wind direction is then passed into the “Point_Mass_ascent_Stage_1” and “Point_Mass_ascent_Stage_2”. In those solvers, the wind direction is given a random magnitude and blows for a span of one second in each solver. The wind can be specified for specific directions and strengths, but the solver is intended to be a general solver that works for any wind direction or magnitude. The force calculation is added to Equations 18 through 20 based on the direction specific contributions of the wind.

The “Point_Mass_ascent_Stage_1” subroutine is called using the ode45 solver in MATLAB, which evaluates the described derivatives to update the state matrix. Once this subroutine is complete, the main code calls the “Point_Mass_ascent_Stage_2” subroutine. The main operational part of this code is the same, but with some additions. This subroutine includes a 1 second halt where the first stage is separated from the second, and then ignition begins and the rocket thrusts upwards again. This is to account for the necessary delay caused by the separation mechanism and was designed to protect the lower stage from damage caused by rocket exhaust. The subroutine then operates identically to the Stage 1 subroutine, except now using the specific Stage 2 motor specifications. Once burnout is achieved, the thrust term is set to zero, and the flight is simulated until apogee. Once apogee is reached, the autorotation solver specified in 4.1.2 “Autorotation_System_Model” is utilized to simulate the descent. For this purpose, the Solver is reconfigured as a function and the state values are utilized by “Main_Code”. A composite of the total state is then created and used to generate the trajectory plots.

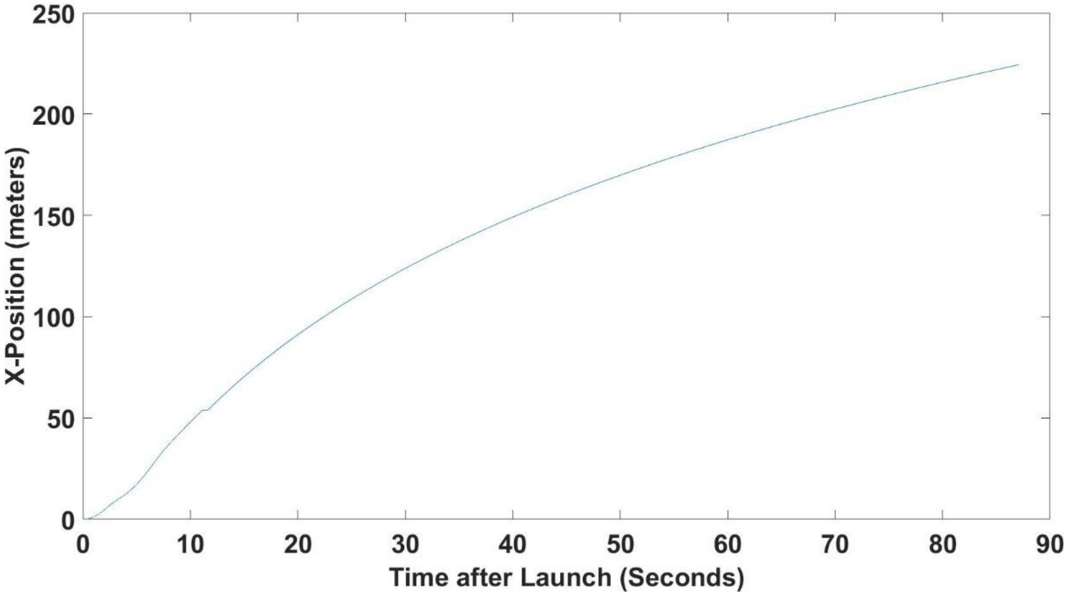


Figure 97 X-Direction Displacement During Flight

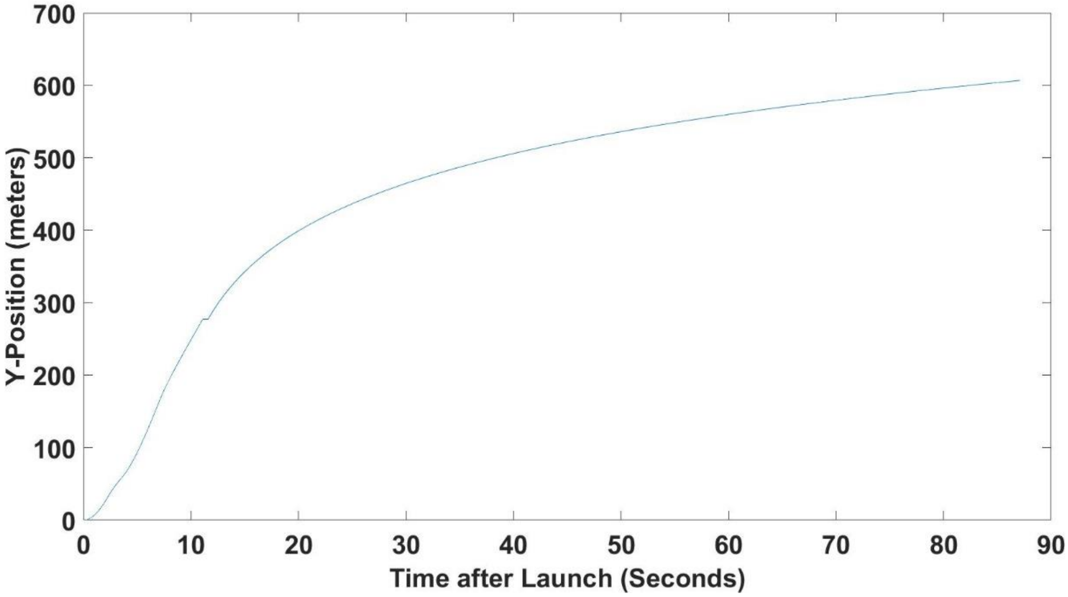


Figure 98 Y-Direction displacement during flight

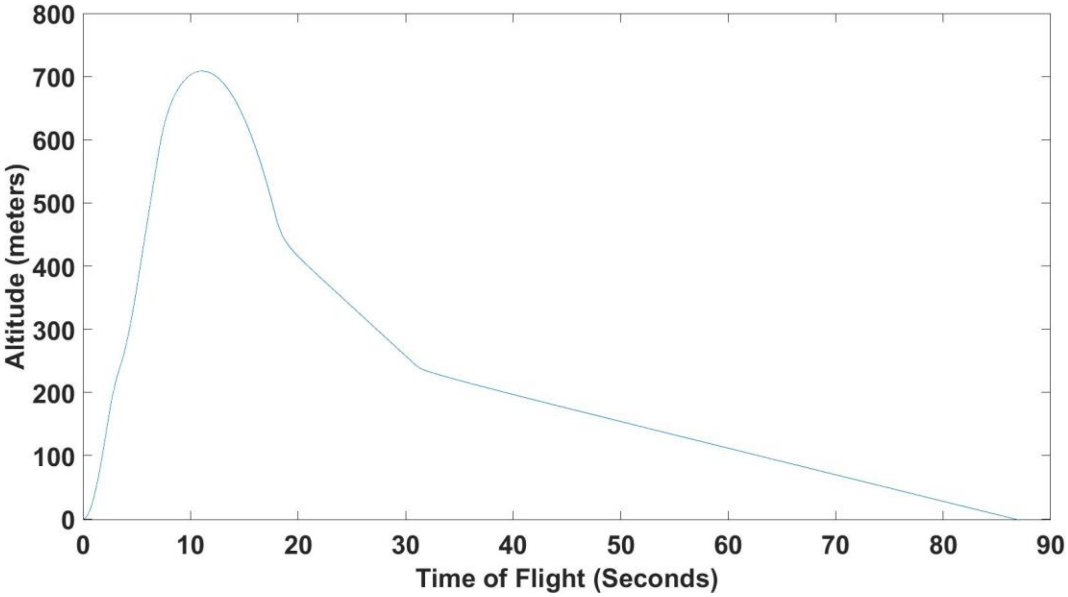


Figure 99 Altitude during flight

Figure 97 and Figure 98 show the horizontal displacements during the flight of the X and Y inertial directions, respectively. Figure 99 shows the altitude during flight. In these plots, some triggers are visible due to the immediate change in velocity. For instance, apogee is reached at an altitude of 714.8 meters, and the descent solver accounts for a free fall time from approximately 11 sec. to 20 sec. At 20 sec. the drogue parachute is deployed which slows the descent, and at approximately 30 sec. the autorotation device is deployed which slows the descent even further. As expected, the displacements continue to grow, but are decelerated due to drag during the descent.

Further, a 3D plot of this flight path shows the estimated landing site for the rocket after launch, and the images below show multiple perspectives of the same launch, in order to make the behavior clearer than a single image. Figure 100 and Figure 102 show the flight from an elevated perspective while Figure 101 shows a side profile of the altitude and X-direction solely. Note the axes on the graph, the Y-direction and altitude motion are both far greater than that of the X direction. Further, there is no explicit consideration of the horizontal displacement in the Autorotation solver, so the drag induced by the blades and the drogue parachute as they pertain to the horizontal displacement is not considered.

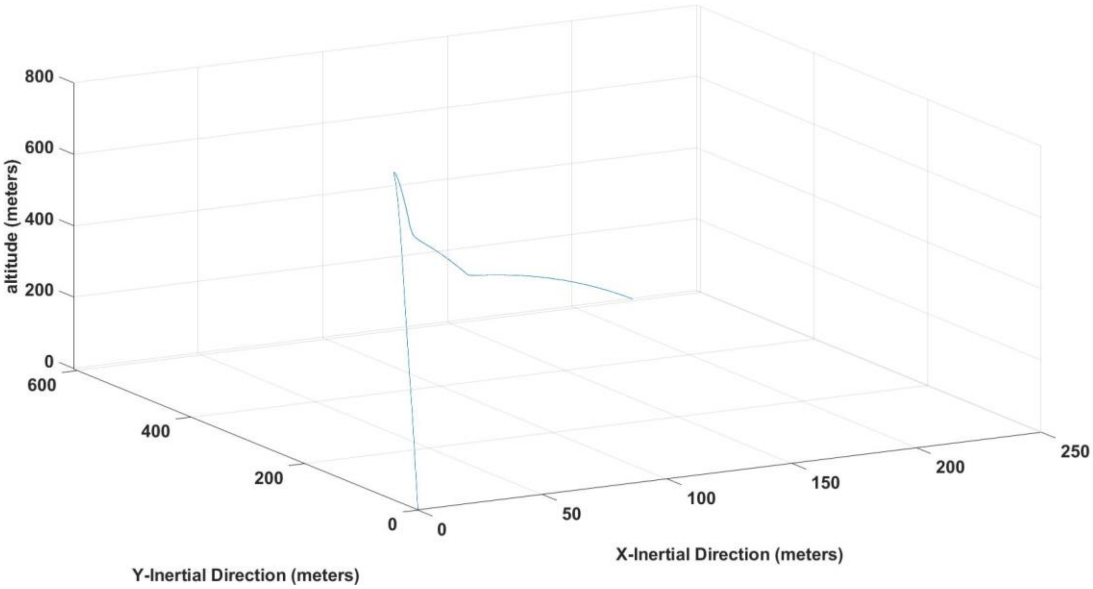


Figure 100 Elevated Perspective of the Flight

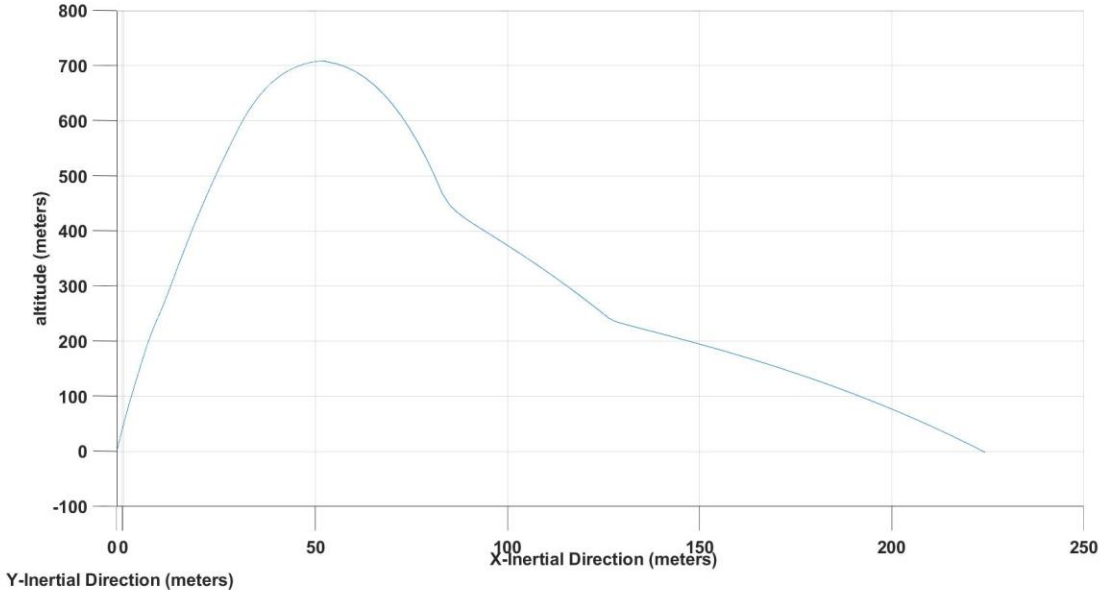


Figure 101 Perspective showing purely the motion in the X direction and altitude

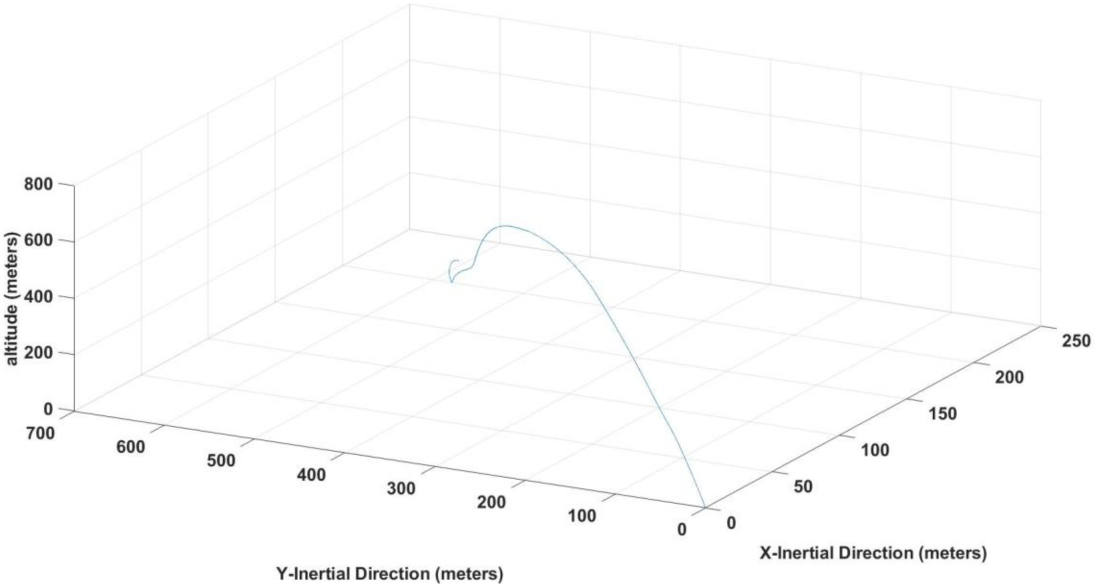


Figure 102 Elevated perspective of the flight from alternate perspective

It should be noted that the functions "Point_mass_ascent_Stage_1.m" and "Point_mass_ascent_Stage_2.m" (Appendix E, 7.5.2 and 7.5.3) could be combined into a single analysis subroutine, but for the convenience of a code analysis they were separated. This measure made it easier to distinguish the processes for error evaluation and parameter definition. In order to combine them, the team simply would make the time-related distinctions between the two stages within the code, and input the values of burn time, ISP, and thrust profile for both motors.

Among these primary subroutines, there are also some minor subroutines that update certain variables. The "atmo" solver finds the air density based on the altitude of the reading. The "Body2Inertial" and "Inertial2Body" codes convert the states from the two coordinate frames in the direction that the names imply. These codes can be seen in Appendix E, sections 7.5.6, 7.5.4, and 7.5.5, respectively.

Figures 101 through 103 show the full flight characteristics of the rocket, but some of the granular details from the initial stage are lost due to this combination. Figure 103 shows the rocket's vertical velocity up to the point of apogee.

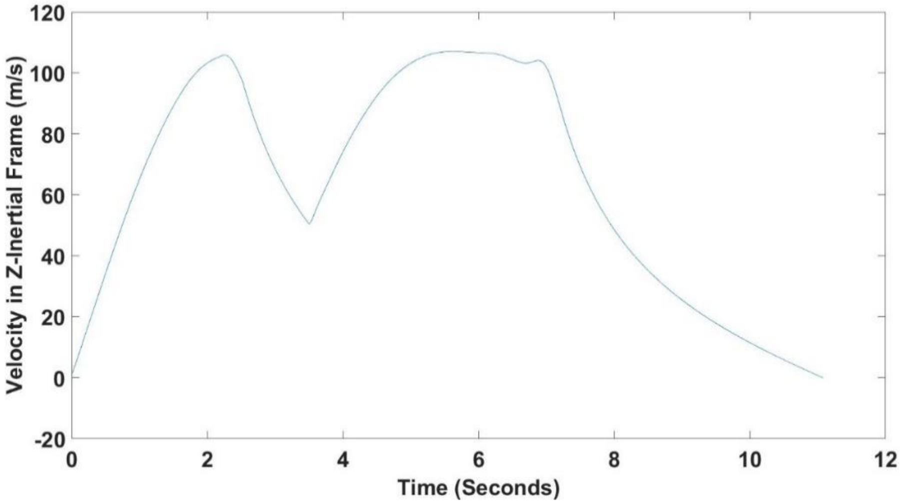


Figure 103 Vertical Velocity component until apogee

Figure 103 shows the way that the vertical velocity varies during ascent. For the first 2.5 seconds there is a clear velocity increase, but the acceleration decreases since the motor has reduced thrust towards the end of its burn. From 2.5 to 3.5 seconds there is no thrust component, so the rocket decelerates due to drag and gravity. At 3.5 seconds, though, the second stage finishes detaching from the first, and the second stage fires, increasing the velocity until around seven seconds where the thrust weakens and eventually ends due to burnout. For the rest of the flight, the rocket decelerates because of gravity and drag, until apogee is reached.

According to this simulation, the rocket reaches an altitude above 2000 ft, which is in excess of the 1500 ft project goal. Although the attitude solver is incomplete, the translational model is still sufficient for guiding design decisions. A result that did not meet the required altitude threshold would lead to re-design choices for reducing mass, changing motors, or altering the stage separation method to reduce the inter-stage time.

4.3.2 FDA Analysis Task 2: Vehicle Aerodynamic Loads – Simulation

The rocket geometry was set up in Ansys Fluent as described in Section 3.3.2. With this set up, the aerodynamic loads applied on the rocket at various points in its flight were found. Due to time constraints, the rocket was only analyzed at a flight speed of 100 m/s. The rocket was analyzed at angles of attacks of 0°, 5°, 10°, 15°, 20°, and 25°. These angles were chosen as they represent the range of angles that the rocket would experience during its flight. they would be the most common angles the rocket would be at during flight. The method described in Section 3.3.2 was used to simulate the range of AOAs in the system. Figure 105 shows the moments applied on the rocket about its vertical axis. This moment, which induces spin on the rocket, was caused by the cambered fins used on the rocket. The types of fins used, and their selection process are described in Section 4.3.3.

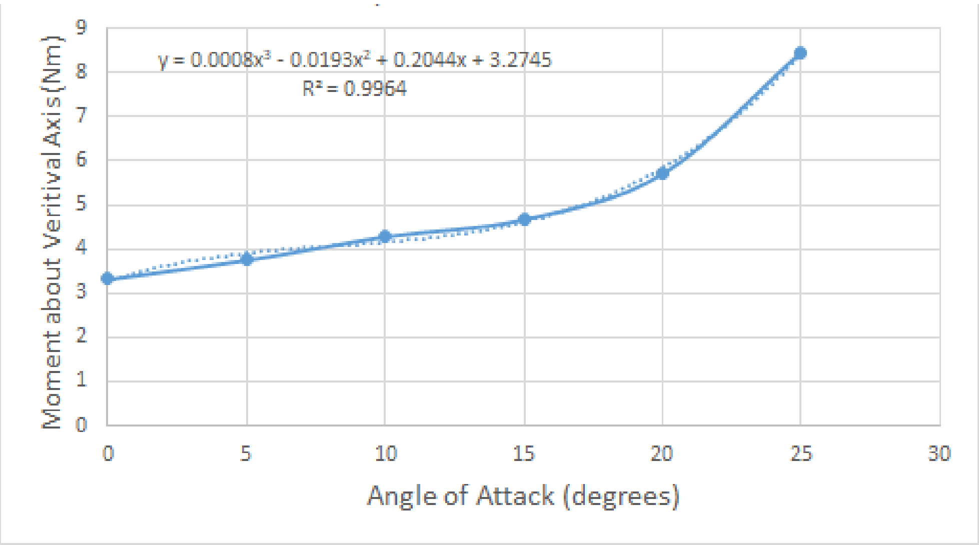


Figure 104 Moment about Vertical Axis vs. Angle of attack

Figure 104 depicts a steady increase in moments. This is because increasing the angle of attack causes the fins to generate a higher amount of lift force [72], which therefore translates into an increased torque and moment. If analyses at different velocities were conducted, it would be is expected that the moment would also increase with the increasing velocities. This reasoning behind this is based in the lift equation [73], which shows that the lift increases exponentially with rising velocity [73]. Since the fins line the side of the rocket, these lifts act as moments around the rocket's vertical axes.

The drag on the rocket was also evaluated at the different AOAs. These values were calculated to be normal to the direction of the flow, and represented the drag induced by the flow on the rocket. The plot of the drag versus the AOA can be seen below in Figure 105. The drag significantly increases as the AOA increases because at higher AOAs more of the rocket is exposed to the oncoming flow raising its surface area and therefore raising the drag force.

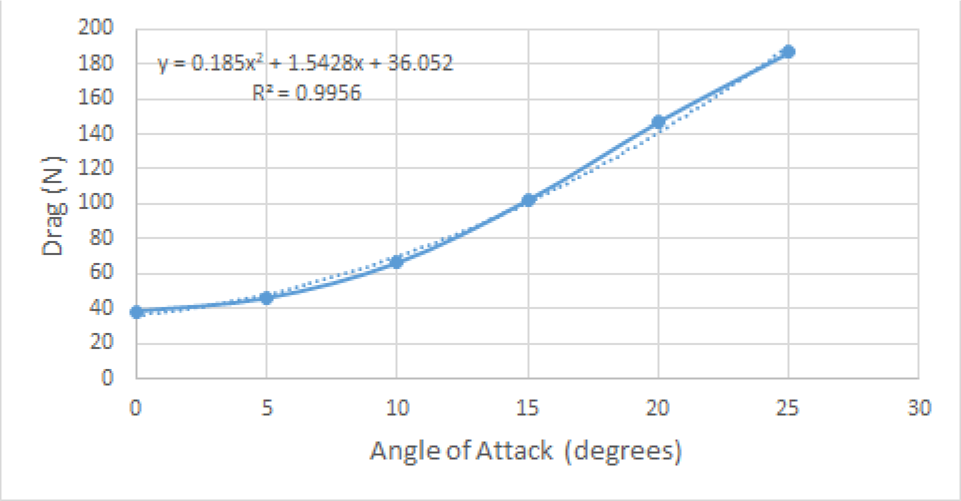


Figure 105 Full Rocket Drag vs Angle of Attack

Dynamic pressure contours were also generated as a way to visualize the flow, and how the flow moves around the rocket at different points in its flight. Figure 106 and Figure 107 below highlight the differences in the extremes of the calculated angles of flow; at 0 degrees and 25 degrees. The rest of the dynamic pressure contours for the full rocket body can be seen in Appendix F.

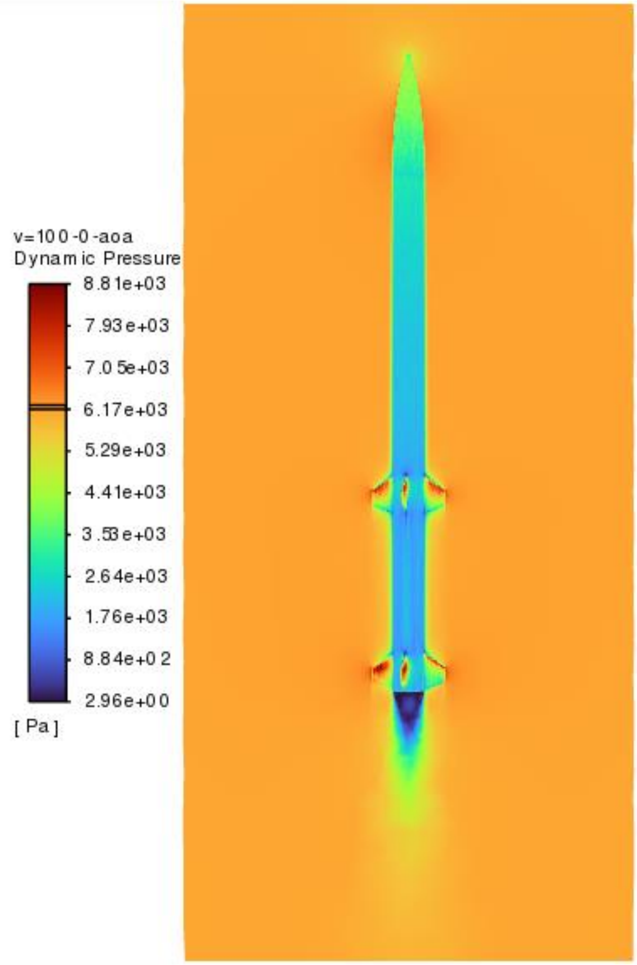


Figure 106 Dynamic Pressure at 100 m/s and 0 degrees

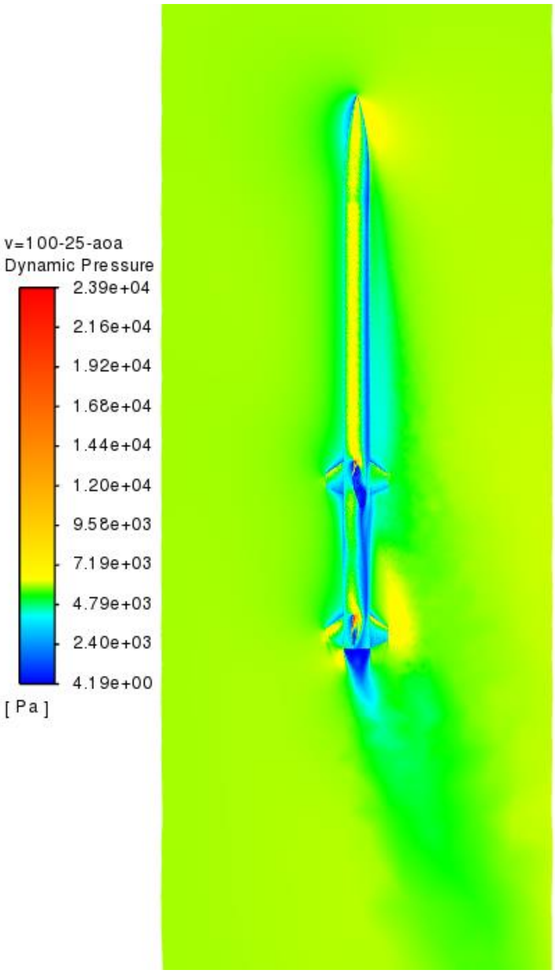
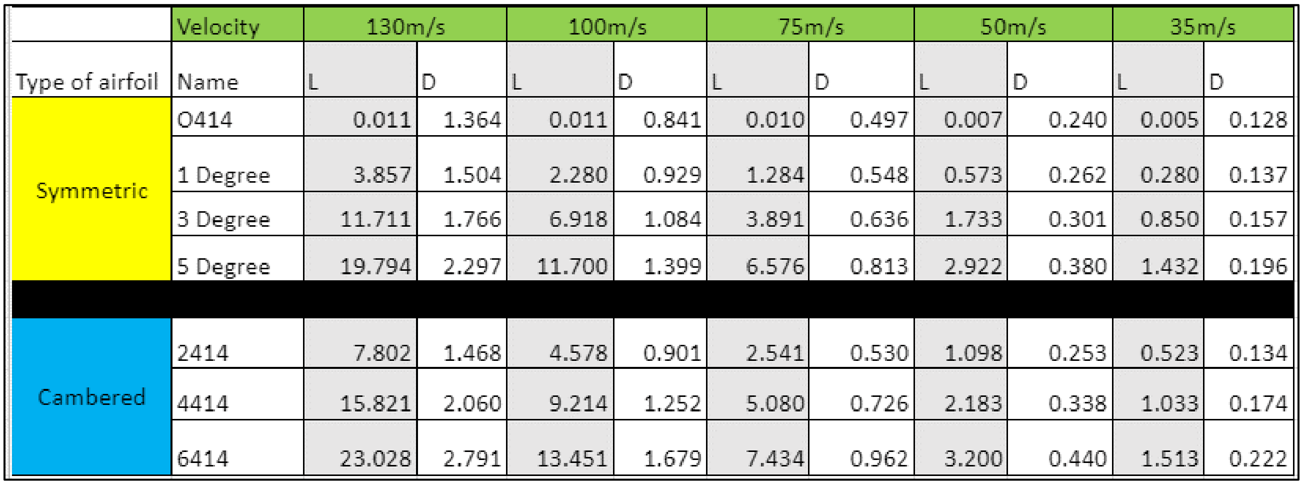


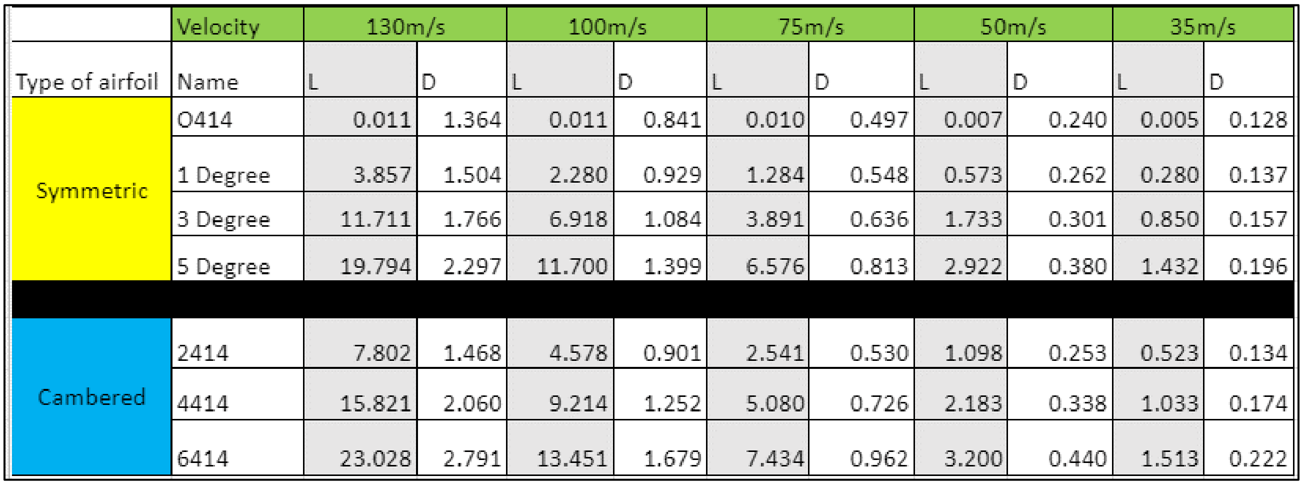
Figure 107 Dynamic Pressure at 100 m/s at 25 degrees

4.3.3 FDA Analysis Task 3: Stabilization Fin Aerodynamic Loads

Using the set up described in Section 3.3.3, seven different fins were analyzed at various airspeeds ranging from 30m/s to 135m/s. All these fins shared the same geometry with exception to their camber percentage and angle of attack. Fins with NACA 2414, 4414, and 6414, were analyzed to see the effect of different camber percentages on the aerodynamic loads. The fins with the NACA 0414 fins were analyzed at AOAs of 0°, 1°, 3°, and 5° to see the effect the AOA has on the aerodynamic loads. The lift and drag data gathered from each of the fins can be seen in Table 20 Fin Lift and Drag Results below. The data was used to make plots of lift and drag versus velocity; one such plot can be seen in Figure 108 and the rest can be observed in Appendix H



	Velocity	130m/s		100m/s		75m/s		50m/s		35m/s	
Type of airfoil	Name	L	D	L	D	L	D	L	D	L	D
Symmetric	0414	0.011	1.364	0.011	0.841	0.010	0.497	0.007	0.240	0.005	0.128
	1 Degree	3.857	1.504	2.280	0.929	1.284	0.548	0.573	0.262	0.280	0.137
	3 Degree	11.711	1.766	6.918	1.084	3.891	0.636	1.733	0.301	0.850	0.157
	5 Degree	19.794	2.297	11.700	1.399	6.576	0.813	2.922	0.380	1.432	0.196
Cambered	2414	7.802	1.468	4.578	0.901	2.541	0.530	1.098	0.253	0.523	0.134
	4414	15.821	2.060	9.214	1.252	5.080	0.726	2.183	0.338	1.033	0.174
	6414	23.028	2.791	13.451	1.679	7.434	0.962	3.200	0.440	1.513	0.222



	Velocity	130m/s		100m/s		75m/s		50m/s		35m/s	
Type of airfoil	Name	L	D	L	D	L	D	L	D	L	D
Symmetric	0414	0.011	1.364	0.011	0.841	0.010	0.497	0.007	0.240	0.005	0.128
	1 Degree	3.857	1.504	2.280	0.929	1.284	0.548	0.573	0.262	0.280	0.137
	3 Degree	11.711	1.766	6.918	1.084	3.891	0.636	1.733	0.301	0.850	0.157
	5 Degree	19.794	2.297	11.700	1.399	6.576	0.813	2.922	0.380	1.432	0.196
Cambered	2414	7.802	1.468	4.578	0.901	2.541	0.530	1.098	0.253	0.523	0.134
	4414	15.821	2.060	9.214	1.252	5.080	0.726	2.183	0.338	1.033	0.174
	6414	23.028	2.791	13.451	1.679	7.434	0.962	3.200	0.440	1.513	0.222

Table 20 Fin Lift and Drag Results

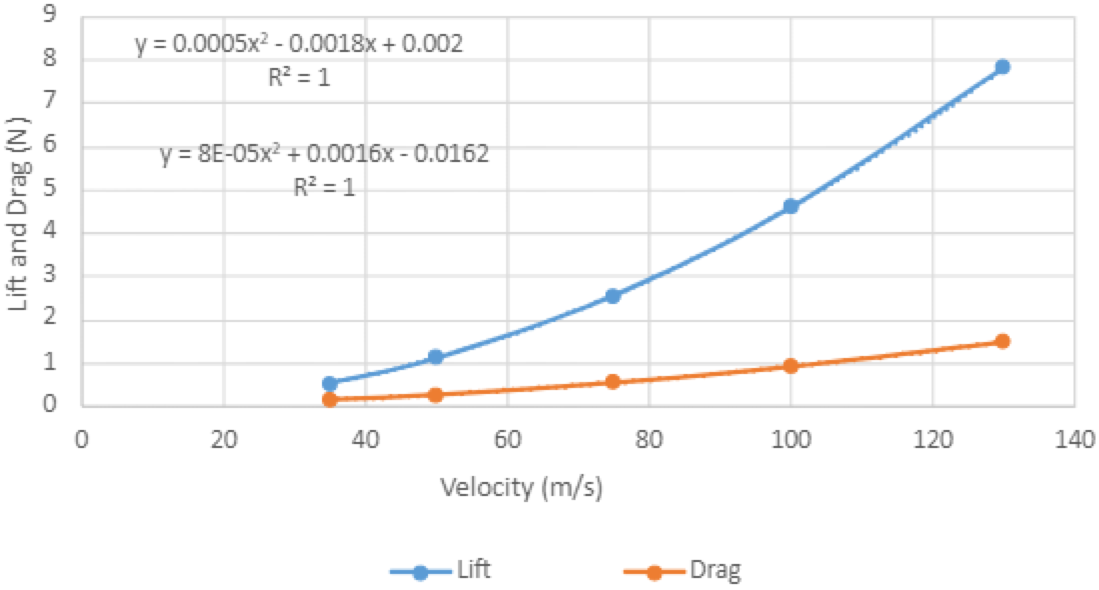


Figure 108: Lift and Drag vs Velocity for 2414 Fin

These results aided in finding the ideal fin by showing which fins generated significant lift forces, while maintaining a relatively low drag force. Additionally, the data above, allowed the lift and drag to be represented as functions of velocity. This allowed for the creation of a function in MATLAB, which used the velocity profile from the dynamic model in FDA Analysis Task 1, the lift and drag versus velocity functions, and the rocket's moment of inertia to create spin rate versus time plots for each fin. Figure 42 in Section 3.3.3 shows the process of translating lift generated by the fins into spin rate. A plot of the spin rate of the 4414 fin versus time can be seen in Figure 109 below.

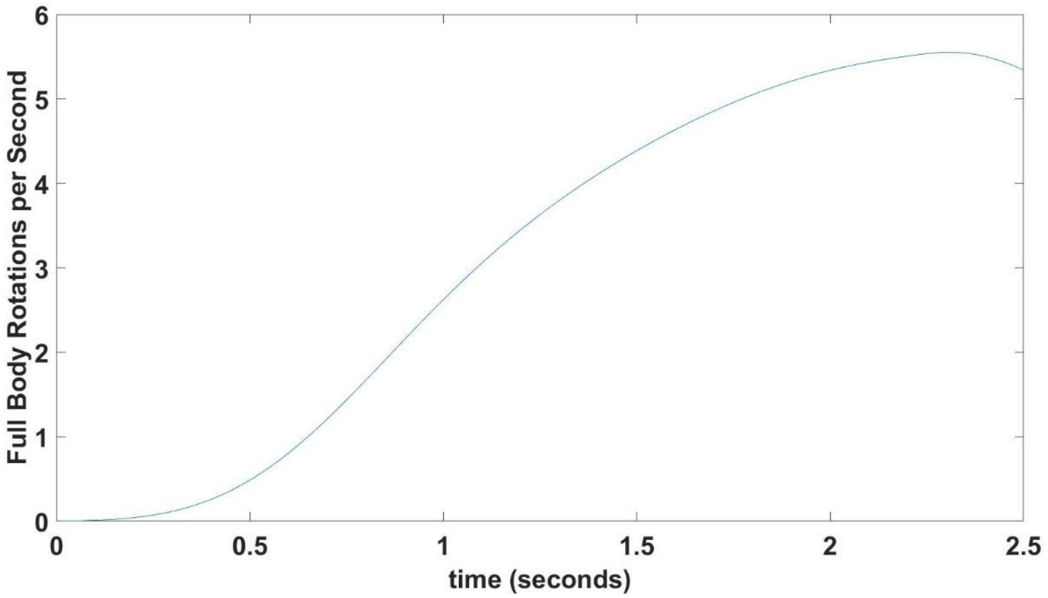


Figure 109: Spin Rate versus Time, NACA 4414

With the spin rate functions, along with the lift and drag functions, the best fin for the project was found. The fin determined to be best for our project was the fin with the NACA 4414 cambered airfoil. The 4414 fin was selected because of its ability to generate a high lift of 15.82N, which translated to a max spin rate of 5.68 rotations/s while keeping a low drag of 2.06N at the maximum observed velocity. Also, the cambered 4414 fins were only needed on the first stage of the rocket, as the second stage does not need the additional stability that the spin offers, due to it already being at a high velocity. Because of this, the second stage of the rocket has fins with NACA 0414 airfoils. This combination of fins was used in the full rocket aerodynamic load analysis, presented in Section 4.3.2.

Contours of dynamic pressure were also evaluated for the two fins types that were used on the final rocket design. The dynamic pressure distributions were then provided to the ARS subteam in order to perform a fin stress analysis presented in Section 4.2.4. Below are the aforementioned contours of dynamic pressure for the fins with the NACA 0414 and 4414. These plots correspond to a flight velocity of 35 m/s.

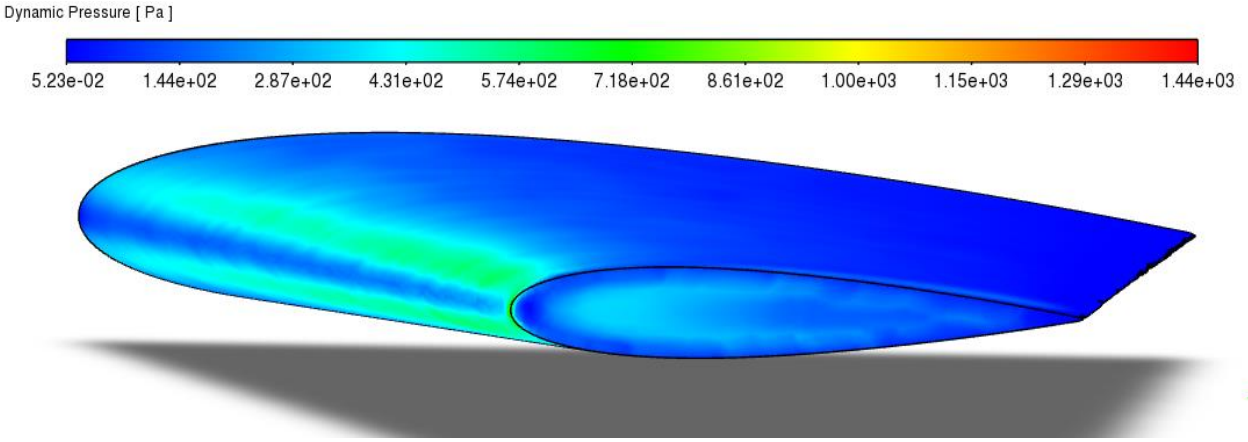


Figure 110: Dynamic Pressure on "Top" Surface of NACA 0414

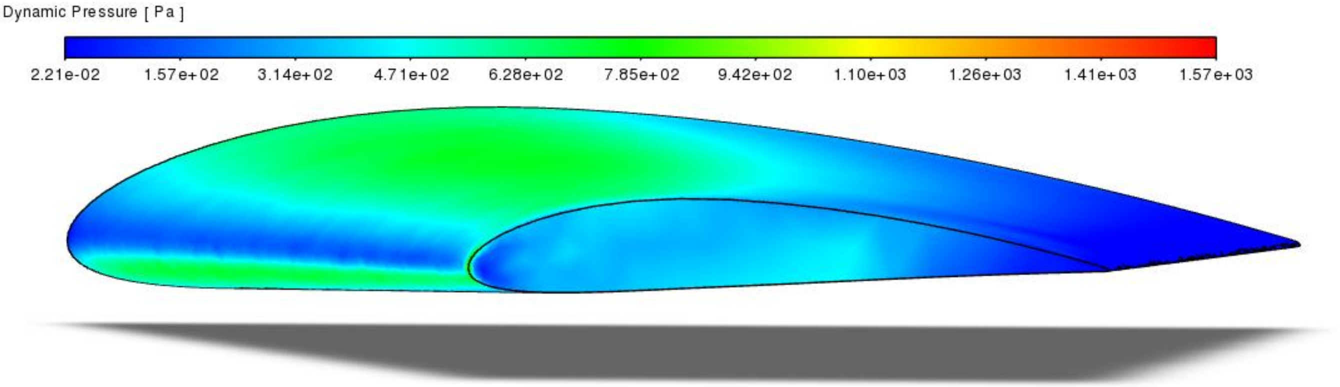


Figure 111: Dynamic Pressure on "Top" Surface of NACA 4414

5 Summary, Recommendations, Broader Impacts

This section presents the broader impacts and take-aways from each subteam. Ultimately this paper discussed the design and analysis of various innovative components of a high-powered model rocket (HPMR). The HPMR designed was a multi-stage vehicle incorporating an innovative recovery system within each stage. The first stage was separated using a complex airbrake system that was modeled and prototyped using SOLIDWORKS and then 3D printed. The second stage was designed to use a unique autorotation system as its primary method of recovery. This autorotation system was deployed from within the airframe of the HPMR and was modeled and analyzed using software such as Ansys Fluent, SOLIDWORKS, and MATLAB. Using Ansys Fluent, the net forces and moments on the autorotation system's blades were modeled over a variety of flight conditions. This produced a function that could be used to model the rocket's descent velocity. The autorotation descent model was then incorporated into the general model for the entire HPMR. The model was based on baseline dynamics for the ascent stage and the autorotation model during descent. This modeling method allows for easy comparison of design choices by outputting the position, velocity, and performance of the HPMR. The results of this analysis were used to determine whether the HPMR could perform as expected and be recovered without damage. For the HPMR's ignition system, COMSOL was used to simulate the ignition of an electrical match combining electrical and heat transfer calculations. The team also made use of the Cantera tool suite for MATLAB to analyze the chemical reaction that occurs in the first and second stage rocket engines. Furthermore, the pressure and temperature data found from this analysis was used to model the temperature and velocity distributions inside the rocket engine, using COMSOL. Another aspect of the project was the consideration two different types of fins, cambered and canted. Ansys Fluent was used to analyze which version of the fins would produce the most ideal spin rate over different cambers and angles of attack. The results from Fluent were put into our HPMR simulation code, to make sure the rocket would still meet the mission requirements with the various fins. The aerodynamic loads on the full rocket body, with the selected fins attached, were then analyzed in Fluent as well. The fins are also unique in that they are unified with their respective airframes as part of a fin can. Ansys was used to determine where the points of highest stress are on the fins. The results for each of these areas of analysis are presented throughout the entirety of this report.

5.1 Airframe and Recovery System: Summary and Recommendations

Throughout the course of this project the ARS subteam was responsible for determining the integrity and reliability of the recovery systems and structural components of the high-powered model rocket. The ARS subteam was assigned four analysis tasks to accomplish this. The first analysis task was focused on the structural integrity of the airframe and critical components within the rocket. The ARS team was able to ultimately determine whether the material for these components would yield due to the pressures of black powder ignition. The ARS team's analysis proved to be successful, and they were able to determine that the airframe and critical components would hold structurally under these conditions.

Due to the COVID-19 pandemic, the ARS subteam was unable to perform an ejection test to prove that the analysis in Ansys was correct. Therefore, the ARS team recommends that future rocketry MQPs, who may be unable to perform tests on their components, evaluate multiple sets of airframes. By doing this the team will be able to determine which airframe is the most reliable option and will be least likely to yield under the pressure conditions of the black powder ignition.

The second ARS analysis task centered upon the autorotation blade model. This task was essential in concluding that the rocket would land at a safe speed while descending under autorotation. The autorotation system is a passive design that relies solely on aerodynamic forces such as lift and drag to work effectively. For future MQP teams interested in this idea, the team recommends performing tests using a wind tunnel or a drop test in order to ensure that the autorotation system will spin up and generate lift as expected. By performing physical tests the next MQP team may be able to pinpoint certain inaccuracies or evaluate just how reliable the performance of the autorotation system may be. Due to the COVID-19 pandemic the team was unable to build or test the autorotation system. The ARS team recommends taking great care in designing these systems in order to ensure accuracy and reliability of the model.

The third ARS task successfully simulated the aerodynamic forces on a rotating airfoil for a limited number of angles of attack. Combined with the results of ARS Task 2, this would have let the ARS team construct the optimal autorotation recovery system to slow the rocket's descent. Unfortunately, due to the COVID-19 pandemic, the team was unable to meet in person throughout the duration of the year, therefore the rocket could never be constructed physically. With more time, the ARS subteam would have increased the number of Angle of Attack cases, along with testing different airfoils to find which would be most suitable for the autorotation system.

The fourth ARS task was aimed to determine if the fins on the fin cans would yield due to the stresses of flight. The task successfully simulated the stressed on the fins and it was determined that the fins would withstand flight. Not only would they withstand the flight they would do so with an extreme margin. Due to the COVID-19 pandemic, the team did not build the fin cans, therefore, they were never put to a flight test. Because the factor of safety on the fins was so high, the ARS team recommends investigating reducing cross-sectional area and weight of the fin cans.

5.2 Propulsion Thermal Separation Systems: Summary and Recommendations

The PTSS subteam was tasked with designing and analyzing separation systems for the rocket, selecting rocket motors, and analyzing their performance, and investigating ignition systems for a solid rocket motor.

To analyze the rocket motors Cantera's chemical equilibrium solver to model the reaction of the solid propellant inside the rocket motor and determine the motor performance. This analysis was completed thoroughly, with an iterative process that aimed to determine the set of parameters that matched the manufacturer reported motor performance. While there were some major limitations to the analysis, such as not being able to model the bonding agent in the rocket motor and assuming constant mass flow results, the results are all realistic and within reasonable ranges to the reported motor characteristics. The COMSOL analysis was also very thorough, with many iterations and tweaks to ensure realistic results. A space to expand on this study would be to model the system at different times, by varying the thickness of the unburnt propellant layer. This would be able to provide results that vary with time and propellant consumption, which would better represent the reality of the situation.

The third PTSS analysis task was focused on finding an air brake and a spring which would produce enough drag to separate the two stages quickly before second stage ignition, while inducing minimal forces on Stage 2 in the process. This task innovated on existing stability issues found in staging events as defined in section 2.2.2. The design idea culminated in a full-scale prototype being manufactured with limited lab testing on a sample spring. For future MQP teams interested in this idea, the structural integrity of the air brake bay frame and air brake arms would need to be analyzed to ensure they could withstand the forces produced by the spring and by the aerodynamic loads. More generally, a spring analysis to verify analytical calculations would be an excellent addition. This was planned to be completed but due to time constraints it could not be carried out. Due to the COVID-19 pandemic the team was unable to test this system in flight, and only a 3D printed prototype could be manufactured. The PTSS team recommends focusing on refining the design of the system and investigating different materials in which the system could be manufactured from.

The fourth PTSS analysis task focused on simulating the time-dependent temperature profile of an electric match. Utilizing electric heating meant defining both electrical and heat transfer domains and conditions that had a clear interdependence. The main issue this simulation faced was controlling the temperature in the nichrome bridge wire. In this analysis, a piecewise function was used to limit the thermal conductivity of the nichrome material after it reached its melting point. This is only one way of approaching this issue and there is room for more accurate modeling to be applied. Another major issue with complications from COVID-19 was that the initial plan of designing an arc ignitor changed direction over the course of the project. Future teams can still use the research done on arc ignitors for a possible ignition technique. The PTSS team recommends running physical tests to confirm the accuracy of this model and for general data needed to estimate ignition times. An arc ignitor is still a viable option, especially for a single motor stage that does not rely heavily on ignition timing. More research can be done in making arc ignitor modules smaller and more compact to possibly fit onto a second stage motor as well.

5.3 Flight Dynamic Analysis: Summary, Recommendations and Broader Impacts

In this project, the FDA team was primarily tasked with ensuring the design choices were sufficient for meeting the proposal requirements as well as ensuring stability in the rocket during flight. To achieve these goals, the FDA team completed the three analysis tasks outlined previously: designing a robust simulation of the rocket, evaluating the aerodynamic loads on the rocket body and fins, and determining the most appropriate fins for spin stabilization of the rocket.

The simulation was very effective for verifying design choices and providing an expected landing site for the rocket at launch. It was also useful for providing accurate values for the ARS team and PTSS team to test the results of their systems. The accurate data allowed for validation of the altitude used by the autorotation system as well as the maximum velocity experienced by the stage separation system, to ensure it would effectively separate. The Attitude model was not completed due to time constraints, but the outline for the methods that would be used is shown in the methodology section for the FDA subteam.

The second analysis task, finding the aerodynamic loads on the full rocket, although successful, was hindered by time constraints. Simulations were run at several different angles of attack, but the only velocity examined was 100 m/s. With more time the team could have ran the analyses for a variety of velocities, developing a better understanding of the loads effecting the rocket during flight. Additionally, due to the COVID-19 pandemic, the rocket was not actually constructed, preventing a comparison between experimental results and our simulated ones.

The last FDA analysis task was simulating the aerodynamic loads on a series of fins and then determining which fin would be best for the project. This task proved to be very successful, the team was able to finish all of their goals for this task. A future team could expand on this task by finding the spin stability data and seeing just how effective or ineffective spinning the rocket is at stabilizing it. Also, due to COVID-19 the team could not construct or test the fins, so none of the simulated fin data could be compared with experimental results.

This MQP experience has been very successful in many aspects, but there were certainly some areas that could be improved upon. Firstly, the COVID-19 pandemic was very impactful in the FDA subteam's ability to collaborate. The FDA subteam recommends that future teams design the coding algorithm for the main simulation model together as a team, and perhaps even delegate different protocols to different members. This recommendation is due to the fact that the model wasn't completed on time, and that the sole member that focused on the topic encountered difficulties in combining the models. The FDA subteam believes that including more people would have been beneficial to solve problems like this. The FDA subteam also recommends physical testing of the fins in order to examine the results in person and add to the educational experience.

Designing a model to simulate the landing region of a rocket based on known starting conditions has broader impacts with respect to real world applications. A known landing area allows for recovery teams to predict final locations and prepare for rocket recapture. When there is a known desired final location, the model can be configured to aim towards that final location. This is very useful for any mission where far-away targets want to be reached by a rocket, such as when the landing site and the objective zone are separated by a great distance. While spin stabilization results were not verified experimentally, the team still recognizes that there is a benefit to using such a system. Hobbyists cannot create guided rocket systems, so having a method to stabilize the rocket without active controls is essential for flight.

5.4 Overall Project Broader Impacts

The goal of this MQP was to investigate multiple innovative components of a HPMR with specific focus on novel ignition, advanced staging, and innovative recovery systems. While designing, analyzing, and testing these components, the team came across multiple social, economic, environmental, and societal impacts. As a team, understanding these impacts was essential to ensure successful and meaningful execution of the project.

The design, test, and integration of rockets is a well-established are of engineering. From government agencies to amateur rocket enthusiasts and events, to academic applications and competitions, the social impacts of designing rockets are quite significant. Not only does this social construct cross various applications, but it also crosses various age groups. The rocketry community consists of engineers, scientists, students, and hobbyists of all ages. This means that children or adults who may have no experience in the study of aerospace engineering, are still actively finding ways to explore the complexities surrounding rocket design. It is common for parents with or without an education in aerospace engineering to be found leading their children around at model rocketry launches and events. Thus, aerospace engineering has no bounds, with an accepting community open to all people ready and willing to learn. For hobby enthusiasts, model rocketry can be a great way to meet new people, join new groups, and network with the community. While working, the team engaged with this community and learned from a variety of online sources from enthusiasts, engineers, and scientists alike.

From a societal perspective, government agencies such as NASA and the U.S. military are directly involved with the construction and testing of large military and commercial rockets to ensure the safety of citizens. For example, the U.S. military uses guided missiles in combat applications. These types of missiles, however, are only appropriate for military purposes. In addition, the FAA sets height restrictions for the flight of model rockets to ensure the safety of airplane travelers and any currently flying vehicles. Thus, an FAA waiver is needed for any amateur rocketry launches.

To accommodate the economic impacts of model rocketry, the team worked to design a reliable and reusable HPMR. Due to COVID-19 the team was unable to test these design constraints and ensure the successful recovery of the rocket, however, via simulation and analysis the team was able to verify that the rocket would have been recoverable. By designing the rocket this way the team was able to save money for future MQP teams by providing reusable parts and an educational model to learn from. This idea can also be seen in industry, specifically with the SpaceX Falcon 9 reusable booster. By designing a reusable system, Space X has expanded the Aerospace industry by opening new jobs and areas of research.

The environmental impacts of rocketry must be considered. The chemicals contained within the motors of HPMR can be harmful to the environment if not handled correctly. Typically, HPMR launches are held on farmers' fields, therefore it is essential to protect the plots for future farming seasons. To address field contamination, fire-proof protective blankets are placed over launch pads. In addition, the chemicals from the exhaust of rocket motors such as ammonium perchlorate can be dangerous in large amounts for humans and animals to breathe in. However, due to the rocket launches being outside, this risk can be easily contained. In the case a rocket lands within a tree there are concerns of pollution. This can be mitigated by using the necessary recovery equipment to ensure the rocket lands at a reasonable speed within the bounds of the launch site.

6 References

[1] Lunar. (2009). “What are model rockets”.
Retrieved from http://www.lunar.org/docs/handbook/modelrockets.shtml

[2] National Association of Rocketry. (n.d.) “High power rocketry”.
Retrieved from https://www.nar.org/high-power-rocketry-info/

[3] Newton, M. (n.d.) “Rocket anatomy 101”.
Retrieved from https://www.nar.org/wp-content/uploads/2014/03/NAR-Rocketry-Basics.pdf

[4] Cavender, D. (n.d.) “NASA high powered video series counterpart documents”. Retrieved from https://www.nasa.gov/sites/default/files/atoms/files/sl_video_instruction_book.pdf

[5] Scott, J. (2003). “Rocket nose cones and altitude”.
Retrieved from http://www.aerospaceweb.org/question/aerodynamics/q0151.shtml

[6] Ramblin' Rocket Club Georgia Tech. (n.d.) “Model rocket aerodynamics”.
Retrieved from http://rocket.gtorg.gatech.edu/files/slides/Model_Rocket_Aerodynamics.pdf

[7] Crowell, G. (n.d.) “The descriptive geometry of nose cones”. Retrieved From http://servidor.demec.ufpr.br/CFD/bibliografia/aerodinamica/Crowell_1996.pdf

[8] Van Milligan, A. (2013). “Drag of nose cones”.
Retrieved from https://www.apogeerockets.com/education/downloads/Newsletter346.pdf

[9] “Carbon fiber vs kevlar vs fiberglass - which one is right for YOU?” KYLE.ENGINEERS (Director). (2016, Jul 31,).[Video/DVD]

[10] Jarvis, J. (2009). “The jarvis illustrated guide to carbon fiber construction”.
Retrieved from https://www.raketenmodellbau.org/repository/archive/167792?view=true

[11] Moore, J. (2012). “Nasa engineers test rotor reentry”. Retrieved from https://www.aopa.org/news-and-media/all-news/2012/november/13/nasa-engineers-test-rotor-reentry

[12] WPI University Student Launch Initiative. (a). “USLI Critical Design Review”
Report to NASA. Jan 2019, Report was required for high powered rocketry competition

[13] WPI University Student Launch Initiative. (b). “USLI Critical Design Review”
Report to NASA. Jan 2020. Report was required for high powered rocketry competition

[14] WPI Battle of the Rockets Goatbusters mars rover team 1040 “Critical Design Review” 2017-2018. Report was required for high powered rocketry competition

[15] Federal Aviation Administration. (2008). “Requirements for Amateur Rocket Activities”. Retrieved from https://www.federalregister.gov/documents/2008/12/04/E8-28703/ requirements-for-amateur-rocket-activities

[16] Marker, J. (2018). “How to calculate lift for rotor blades.” Retrieved from https://sciencing.com/calculate-friction-sleeve-bearing-7802059.html

[17] Barber, T. (n.d.) “Model rocket helicopter (gyrocopter) duration”. Retrieved from https://www.nar.org/wp-content/uploads/2014/05/ Helicopter-Duration-Barber.pdf

[18] Amato, N., Huaman, Z., Hyland, A., Koslow, J., Pickunka, J., Procaccini, J., . . . Tappen, J. (2020). “Design, analysis, and test of a high-powered model rocket”. WPI Major Qualifying Project (MQP). Report JB3-2001, Advisor: J. Blandino, 2020 Retrieved from https://web.wpi.edu/Pubs/E-project/Available/E-project-040720-011451 /unrestricted/Design,Analysis,_and_Test_of_a_High-Powered_Model_Rocket.pdf

[19] Malone, J., & Edwards, A. (2009). “NASA successfully tests parachute for ares rocket”. Retrieved from https://www.nasa.gov/mission_pages/constellation/ares/drogue_test.html

[20] “Orion’s parachute system”. (2018). Retrieved from https://www.nasa.gov/sites/default/files/atoms/files/orion_parachutes.pdf

[21] Apogee Rockets. (a). (n.d.) “Drogue parachutes”. Retrieved from https://www.apogeerockets.com/BuildingSupplies/Parachutes/ DrogueParachutes#:~:text=Drogue%20parachutes%20are%20used%20for,highest %20altitude%20of%20the%20flight).&text=This%20is%20a%2015%22%20canopy, pilot)%20chute%20to%20pull..

[22] Rocket Arium. (n.d.) “Nylon parachutes”. Retrieved from https://www.rocketarium.com/Build/Parachutes/Nylon#:~:text=Nylon% 20parachutes%20are%20the%20best,made%20from%20rip-stop%20nylon.

[23] Apogee Rockets. (2017a). “Johnny star commander”. Retrieved from https://www.apogeerockets.com/education/downloads/Newsletter447.pdf

[24] Milligan, T. (2009). “How to pick the correct size streamer”. Retrieved from https://www.apogeerockets.com/education/downloads/Newsletter244.pdf

[25] Linder, C. (2020). “Why plasma lighters are better than butane”. Retrieved from https://www.popularmechanics.com/technology/gadgets/a31133313/how-plasma-lighters-work/

[26] Whelan, M., & DeLair, R. (2016). “Arc lamps”. Retrieved from http://edisontechcenter.org/ArcLamps.html

[27] GreatScott!. (2017). “DIY arc lighter”.

Retrieved from https://www.youtube.com/watch?v=lk_1lzMiUVc

[28] Perfect Flite. (n.d.) “Making low current rocket motor igniters”.
Retrieved from http://www.perfectflite.com/Downloads/Making%20igniters.pdf

[29] Milligan, T. (2008). “Model rocket design and construction” (3e ed.) Apogee.
Retrieved from https://www.apogeerockets.com/downloads/PDFs/MRDC_sample.pdf

[30] NASA. (n.d) “Saturn V News Reference”. 1968.
Retrieved from https://history.nasa.gov/SP-4206/notes.html

[31] Longren DR. (n.d) “Stage separation dynamics of spin-stabilized rockets”.
Journal of Spacecraft and Rockets. 1970;7(4).
Retrieved from https://arc.aiaa.org/doi/10.2514/3.29959

[32] Blue Origin. (n.d.) “New shepard”.
Retrieved from https://www.blueorigin.com/new-shepard/

[33] May, G. (2012). “What's in a rocket motor”.
Retrieved from https://www.nar.org/wp-content/uploads/2014/03/NAR-Rocketry-Basics.pdf

[34] Box, S., Bishop, C., & Hunt, H. (2010). “A stochastic six-degree-of-freedom flight simulator for passively controlled high power rockets”. Journal of Aerospace Engineering, Retrieved from https://www.researchgate.net/publication/245307313_A_Stochastic_Six-Degree-of-Freedom_Flight_Simulator_for_Passively_Controlled_High_Power_Rockets

[35] Curtis, H. (2005). “Orbital mechanics for engineering students” (3e ed.). Amsterdam: Elsevier Butterworth Heinemann.:

[36] Benson, T. (2014b). “Practical rocketry”.
Retrieved from https://www.grc.nasa.gov/www/k-12/rocket/TRCRocket/practical_rocketry.html

[37] Benson, T. (2014a). “Examples of controls”.
Retrieved from https://www.grc.nasa.gov/WWW/K-12/rocket/rktcontrl.html

[38] Gasmire, C. (n.d.) “Model rocket spin stabilization explained”. Retrieved from https://themodelrocket.com/model-rocket-spin-stabilization-explained/#:~:text=Spin%20stabilization%20works%20by%20spinning,that%20come%20along%20with%20it

[39] Madcow Rocketry. (n.d.) “5” G12 Airframe”.
Retrieved from https://www.madcowrocketry.com/5-g12-airframe/

[40] Apogee Rockets. (2017b). “What is the best fin shape for a model rocket?”
Retrieved from https://apogeerockets.com/education/downloads/Newsletter442.pdf

[41] Mandelli, D., Rabiti, C., Cogliati, J. J., Kinoshita, R. A., Alfonsi, A., & Sen, R. S. (2015).

"RAVEN user manual". United States: doi:10.2172/1235208
Retrieved from https://www.osti.gov/servlets/purl/1235208

[42] Reference Design. (n.d) "KL25-AGMP01 User Guide". 2015.
Retrieved from https://www.nxp.com/docs/en/user-guide/RD-KL25-AGMP01-UG.pdf

[43] Helidirect. (n.d.) "Align 780mm carbon fiber blades".
Retrieved from https://www.helidirect.com/products/align-780-carbon-fiber-blades

[44] Amato, N., Huaman, Z., Hyland, A., Koslow, J., Pickunka, J., Procaccini, J., . . . Tappen, J. (2020). "*Design, analysis, and test of a high-powered model rocket".* WPI Major Qualifying Project (MQP). Report JB3-2001, Advisor: J. Blandino, 2020 Retrieved from https://web.wpi.edu/Pubs/E-project/Available/E-project-040720-011451/unrestricted/Design,_Analysis,_and_Test_of_a_High-Powered_Model_Rocket.pdf

[45] Coleman, D. (n.d.) "GoPro HERO6 black vs HERO5 black". Retrieved from https://havecamerawilltravel.com/gopro/gopro-hero6-black-vs-hero5-black/#:~:text=The%20Frame%20is%20included%20in,battery%20and%20memory%20card%20installed

[46] Amazon. (n.d.) "Soft plush among us plush among us game plush toy with music kawaii stuffed doll cute red small among us plushie (1pc, brown)". Retrieved from https://www.amazon.com/Plush-Among-Kawaii-Stuffed-Plushie/dp/B08LSHNWP4/ref=asc_df_B08LSHNWP4/?tag=hyprod20&linkCode=df0&hvadid=475873706534&hvpos=&hvnetw=g&hvrand=7318299430492741589&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=1018405&hvtargid=pla-1210826379592&psc=1

[47] Amazon. (n.d.) "Plexus-Plastic-Cleaner-Protectant". Retrieved from:
https://www.amazon.com/Plexus-Plastic-Cleaner-Protectant-13-Ounce/dp/B00092CKN

[48] Steck, G. (2010). "The best ways to cut fiberglass panels."
Retrieved from https://www.doityourself.com/stry/how-to-cut-fiberglass-paneling

[49] WeeTect. (n.d.) "Polycarbonate bending: The complete guide". Retrieved from https://www.weetect.com/polycarbonate-bending/#:~:text=Doing%20so%20at%20least%20along,bent%20to%20the%20desired%20angl

[50] METRA Launch. (n.d.) "Metra Rocket Club". [cited 2021Apr12].
Retrieved from: https://metrarocketclub.org/thrust-to-weight-charts/

[51] "ThrustCurve Hobby Rocket Motor Data AeroTech K1000T". (n.d.) • ThrustCurve. [cited 2021 Apr12]. Retrieved from:
https://www.thrustcurve.org/motors/AeroTech/K1000T/

[52] "ThrustCurve Hobby Rocket Motor Data Ceasaroni K661". (n.d.) • ThrustCurve. [cited 2021 Apr12]. Retrieved from:

https://www.thrustcurve.org/motors/Cesaroni/2430K661-P/

[53] Saiki, T. (2018). “High voltage pulse generation based on relaxed self-excited oscillation using electrostatic induction in external capacitors” Retrieved from https://arxiv.org/ftp/arxiv/papers/1806/1806.08905.pdf

[54] Siouris, G. M. (2011). “Missile guidance and control systems”. New York, NY: Springer.

[55] ANSYS fluent. (n.d.) “12.0 Theory guide - 4.5.1 k-w standard – Model”. (2009, January 23). Retrieved from https://www.afs.enea.it/project/neptunius/docs/fluent/html/th/node66.html

[56] Collins, D. (2019). “Mechanical properties of materials: Stress and strain”. Retrieved from https://www.linearmotiontips.com/mechanical-properties-of-materials-stress-and-strain/

[57] Permabond. (n.d.) “Strongest adhesives for bonding metal, glass & plastic”.
Retrieved from https://www.permabond.com/resource-center/strongest adhesive/#:~:text=
Let%20me%20clarify%20%E2%80%93%20single%20component,
(5100%E2%80%93%206000%20psi).

[58] Matweb. (1990). “Plywood data sheet”. Retrieved from
http://www.matweb.com/search/datasheet_print.aspx?matguid=bd6620450973496ea2578c
283e9fb807

[59] Apogee Rockets. (n.d.) (c). “Small nylon shear pins - 20 pack”. Retrieved from
https://www.apogeerockets.com/Building-Supplies/Misc-Hardware/Nylon-Shear-Pins-20-pack

[60] Rocketry Calculator. (2016). “BP estimator”. Retrieved from https://rocketrycalculator.com/
rocketry-calculator/bp-estimator/?__cf_chl_jschl_tk__=2fd7859118874523118e7d65459e9f
0405f4c546-1618195870-0-AedtPRqt80mpLAuFcZ7XFONAIFBISLKTFGIqJh_NaM4ADT
e9_2wBVkDR6vgKJr4p9dRvaWbeR5-A5h-WTDc-U0PnpU-bArY1ZGQlxNJoRq370yaVHB
myPuv9EjWLnl0ZTw3OKMuAi2u2RtaXq4C_swOxBXH5kRc0dqz08-bxTPdV6TUGML-AD
CVhw-wF3AxvkQOPJTqg7oqLBqa1PgndvjnYoKT--UjvknyI3QUHfQggzu92SICSqu045FUx
K-SR4TlzI_dlx7i6e3dNX-wnP4geScCPLIMNyyNhUpb16miRtewn02QwZL3K1OjQUqSIigK
EowC5ZHiSPuhq2EskCf8cDKxThXH_-aRFxxOyLelJdxaYPnAXNlAwBWbFcjlqMAE6B7tu
QY_1ULdUrU-o7g1Ud7IJc4zfz4g37lXwIhiw

[61] Apogee Rockets. (n.d.) (b). “Large 4-40 nylon shear pins - 20 pack”. Retrieved from
https://www.apogeerockets.com/Building-Supplies/Misc-Hardware/Large-4-40-Nylon-Shear
-Pins-20-pack

[62] Azo Materials. (n.d.) “E-glass fibre datasheet”. Retrieved from
https://www.azom.com/properties.aspx?ArticleID=764

[63] Ansys Fluent. (n.d.) N.A., “ANSYS FLUENT User’s Guide,” ANSYS, 2012.
Retrieved from https://www.afs.enea.it/project/neptunius/docs/fluent/html/ug/main_pre.htm

[64] Tomblin, J., Mckenna, J., Ng, Y., & Raju, K. S. (2001). “Advanced general aviation transport experiments B-basis design allowables for wet layup / field repair fiber reinforced composite material systems 7781 glass fabric / MGS 418” Retrieved from https://agate.niar.wichita.edu/Materials/WP3.3-033051-114.pdf

[65] Cantera. (n.d.) “MATLAB tutorial”.
Retrieved from https://cantera.org/tutorials/matlab-tutorial.html

[66] Snyder, C. (2020). “Chemical equilibrium with applications”.
Retrieved from https://cearun.grc.nasa.gov/ThermoBuild/

[67] Cambridge University. (n.d.) “Motor assembly shown with specified motor hardware & reload kit”. (2020). Cambridge University Press. doi:10.1017/9781108612951.009 Retrieved from https://search.datacite.org/works/10.1017/9781108612951.009

[68] The RCS Store. (2021). “75mm medusa nozzle”. Retrieved from https://www.rocketmotorparts.com/75mm_Medusa_Nozzle/p1577809_7763077.aspx

[69] Zhang, J., Zhi, S. and Sun, B. (n.d.) “Estimation of thermophysical properties of solid propellants based on particle packing model”. Sci. China Technol. Sci. 56, 3055–3069 (2013). Retrieved from https://doi.org/10.1007/s11431-013-5368-1

[70] Gorski, N. 14 Feb. 2018. “How To Make Electric Matches”. Retrieved from https://www.skylighter.com/blogs/how-to-make-fireworks/electric-matches-for-your-july-4thdisplay#:~:text=Electric%20matches%20typically%20consist%20of,ignite%20at%20relatively%20low%20temperatures.%22

[71] “Boron Potassium Nitrate (BKNO3)”. (n.d.) Material Safety Data Sheet. Pacific Scientific Energetic Materials Company. Product Number 14151. Retrieved from https://psemc.com/

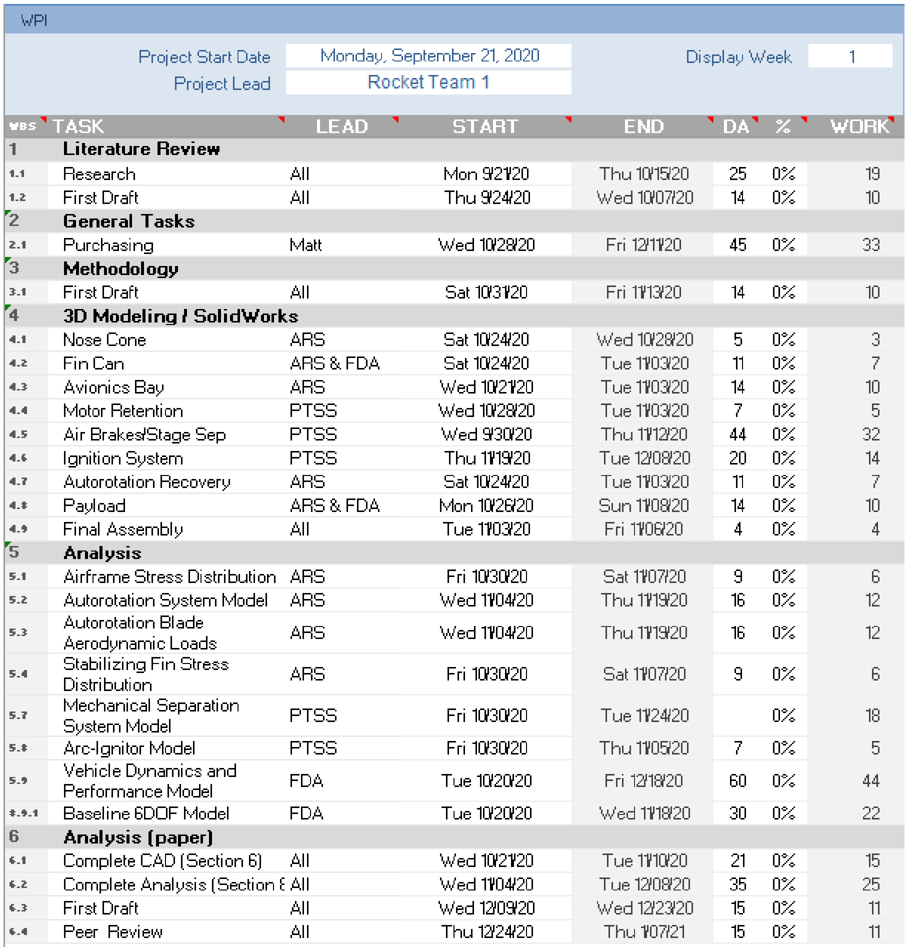
[72] Hall N, editor. (n.d.) “Velocity Effects on Aerodynamic Forces”. NASA. NASA; 2018. Retrieved from: https://www.grc.nasa.gov/www/k-12/airplane/vel.html

[73] Hall N. (n.d.) “Inclination Effects on Lift. NASA” . NASA; 2018. Retrieved from: https://www.grc.nasa.gov/www/k-12/airplane/incline.html

7 Appendices

7.1 Appendix A Gantt Chart

For the purposes of clarity, this chart only shows the tasks for the first week of each term.



WPI

Project Start Date: Monday, September 21, 2020

Project Lead: Rocket Team 1

Display Week: 1

WBS	TASK	LEAD	START	END	DA	%	WORK
1	**Literature Review**						
1.1	Research	All	Mon 9/21/20	Thu 10/15/20	25	0%	19
1.2	First Draft	All	Thu 9/24/20	Wed 10/07/20	14	0%	10
2	**General Tasks**						
2.1	Purchasing	Matt	Wed 10/28/20	Fri 12/11/20	45	0%	33
3	**Methodology**						
3.1	First Draft	All	Sat 10/31/20	Fri 11/13/20	14	0%	10
4	**3D Modeling / SolidWorks**						
4.1	Nose Cone	ARS	Sat 10/24/20	Wed 10/28/20	5	0%	3
4.2	Fin Can	ARS & FDA	Sat 10/24/20	Tue 11/03/20	11	0%	7
4.3	Avionics Bay	ARS	Wed 10/21/20	Tue 11/03/20	14	0%	10
4.4	Motor Retention	PTSS	Wed 10/28/20	Tue 11/03/20	7	0%	5
4.5	Air Brakes/Stage Sep	PTSS	Wed 9/30/20	Thu 11/12/20	44	0%	32
4.6	Ignition System	PTSS	Thu 11/19/20	Tue 12/08/20	20	0%	14
4.7	Autorotation Recovery	ARS	Sat 10/24/20	Tue 11/03/20	11	0%	7
4.8	Payload	ARS & FDA	Mon 10/26/20	Sun 11/08/20	14	0%	10
4.9	Final Assembly	All	Tue 11/03/20	Fri 11/06/20	4	0%	4
5	**Analysis**						
5.1	Airframe Stress Distribution	ARS	Fri 10/30/20	Sat 11/07/20	9	0%	6
5.2	Autorotation System Model	ARS	Wed 11/04/20	Thu 11/19/20	16	0%	12
5.3	Autorotation Blade Aerodynamic Loads	ARS	Wed 11/04/20	Thu 11/19/20	16	0%	12
5.4	Stabilizing Fin Stress Distribution	ARS	Fri 10/30/20	Sat 11/07/20	9	0%	6
5.7	Mechanical Separation System Model	PTSS	Fri 10/30/20	Tue 11/24/20		0%	18
5.8	Arc-Ignitor Model	PTSS	Fri 10/30/20	Thu 11/05/20	7	0%	5
5.9	Vehicle Dynamics and Performance Model	FDA	Tue 10/20/20	Fri 12/18/20	60	0%	44
5.9.1	Baseline 6DOF Model	FDA	Tue 10/20/20	Wed 11/18/20	30	0%	22
6	**Analysis (paper)**						
6.1	Complete CAD (Section 6)	All	Wed 10/21/20	Tue 11/10/20	21	0%	15
6.2	Complete Analysis (Section 8	All	Wed 11/04/20	Tue 12/08/20	35	0%	25
6.3	First Draft	All	Wed 12/09/20	Wed 12/23/20	15	0%	11
6.4	Peer Review	All	Thu 12/24/20	Thu 1/07/21	15	0%	11

7.2 Appendix B MATLAB Code for Autorotation Model

7.2.1 Autorotation System Model Rocket Team 1 2020-2021

```
clc;
clear all;
close all;
```

7.2.2 Defining Initial Rocket Parameters

```
%A comercially available blade was selected for the Autorotation model. The
%following initial parameters of the model are as follow.

n = 3;                 %Specified number of blades
b = 0.780;             %Span of the blade in meters (m)
c = 0.064;             %Chord length of the blade in meters (m)
psi = deg2rad(5);      %Pitch angle (radians)

%The following parameters are constant due to the natural laws of physics

g = 9.81;              %Acceleration due to gravity in meters/sec (m/s)
rho = 1.255;           %Density of air in kilograms per meter cubed (kg/m^3)
mu  = 1.81*10^-5;      %Viscosity of air in kilograms per meter seconds (kg/ms)

m = convmass(7.5,'lbm','kg'); %overall mass of the craft
D = 0.609; % 2 feet
A = pi*D^2/4;
%I = (convmass(322.45,'lbm','kg')/144) * unitsratio('m', 'ft') * unitsratio('m', 'ft'); %moment
of
I = 20; %momen of inertia of the blade system
```

7.2.3 Data Acquired from XFLR5

```
%Looking at ranges
%of reynolds number within a reasonable Mach region and using XFLR5 to
%obtain the CL and CD data at High, Med, Low, and stalling AOA points for
%our specified airfoil.

% To be changed later to current data

% Reynolds number is higher than 100000
drag_data = load("data/drag_modified.mat").drag_out;
lift_data = load("data/lift_modified.mat").lift_out;

% For calculating the critical angle of attack
Cl_stall=readmatrix("data/NACA_0012_Cl_stall");
Cd_stall=readmatrix("data/NACA_0012_Cd_stall");

C_d_chute = 0.75; %Coefficient of drag of the drogue parachute
```

7.2.4 HPMR Flight Events

```
% Times
t_0 = 0;     %s starting time
t_f = 100;   %s final time
t_chute = 7;    %s time of drogue chute deployment
t_ar = 20;   %s time of autorotation
```

```
dt = 0.01;    %s time step
dx = 1e-2;    %meters blade length step

tspan = t_0: dt: t_f; % total time span
```

7.2.5 Initial conditions

```
%Decent at apogaee

v_0 = 0;    %Velocity at apogee is 0
h_0 = 747.2;  %Initial height during descent will be equal to apogee
```

7.2.6 Variable and Equation List

omega = angular velocity of autorotation system x = arbitrary distance along autorotation blade v_rel = relative wind velocity v_y   = vertical velocity v_rel = @(omega, x, v_y) sqrt((omega*x)^2 + (v_y)^2);

```
% r_e   = Reynold's number calculation
% rho   = air density
% c     = chord length
% r_e = @(v_rel) (rho*v_rel*c) / mu;

% phi   = relative wind angle
% phi = @(omega, x, v_y) atan2(v_y, omega * x);

% psi   = angle of pitch
% alpha = angle of attack
% alpha = @(phi, psi) phi - psi;

% d_A   = differential area of the blade
% dx    = differential length
% d_A = @(dx) c * dx;

% d_L   = lift force
% d_D   = drag force
% C_l   = coefficient of lift
% C_d   = coefficient of drag
% C_l = calcC_l(r_e, alpha, Cd_data_high, Cd_data_med, Cd_data_low);
% C_d = calcC_d(r_e, alpha, Cd_data_high, Cd_data_med, Cd_data_low);

% d_L = @(x, C_l2, v_rel2) 0.5 * rho * d_A(x) * C_l2 * v_rel2^2;
% d_D = @(x, C_d2, v_rel2) 0.5 * rho * d_A(x) * C_d2 * v_rel2^2;

% d_L(0, 0, 0);
```

7.2.7 Execute

```
% Initial values
omega_0 = 0; % 0 rad/s
x_0 = h_0; % apogee
v_0 = 0; % 0 m/s at apogee
acc_0 = 0; % 0 m/s^2 at apogee


tic;  %Begining to time the execution of the ODE

%ODE45 runs over a set time and calculates the matrix kinematics over tspan
%Kinematics = (height, velocity, angular velocity, acceleration)
%calc_kinematics is a function executed by the ODE which calculates the
%kindematics

opts = odeset('RelTol' , 1e-2, 'AbsTol', 1e-4);  %reduces the tolerance of the ODE
[t, kinematics] = ode45(@(t, kinematics) calc_kinematics(t, kinematics, dt, rho, C_d_chute, A, m,
n, ...
    g, I, b, t_chute, t_ar, ...
    psi, dx, mu, c, lift_data, drag_data, ...
    Cl_stall, Cd_stall), ...
    tspan, [h_0; v_0; omega_0; acc_0], opts);

toc; %End of time to execute ODE

figure(1);
plot(t, kinematics(:, 1));
xlabel("Time (s)");
ylabel("Rocket Height (m)");
title("Time vs. rocket Height");
grid on;

figure(2);
plot(t, kinematics(:, 2));
xlabel("Time (s)");
ylabel("Rocket Speed (m/s)");
title("Time (s) vs. Rocket Speed");
grid on;

figure(3);
plot(t, kinematics(:, 3));
xlabel("Time (s)");
ylabel("Angular Speed (rad/s)");
title("Time (s) vs. Angular Speed (rad/s)");
grid on;

figure(4);
plot(t, gradient(kinematics(:, 4)) * 10);
xlabel("Time (s)");
ylabel("Acceleration (m/s^2)");
title("Time (s) vs. Acceleration (m/s^2)");
```

```
grid on;

% The purpose of this function is to calculate the kinematics of the craft
% over time

function [kinematics_out] = calc_kinematics(t, kinematics, dt, rho, C_d_chute, A, m, n, ...
    g, I, b, t_chute, t_ar, ...
    psi, dx, mu, c, lift_data, drag_data, ...
    Cl_stall, Cd_stall)

    omega = kinematics(3);  %Angular velocity (rad/s)
    v = kinematics(2);      %Velocity (m/s)
    h = kinematics(1);      %Height (m)


    if t < t_chute
%         fprintf("FREEFALL MODE\n");

        % only gravity is acting on the rocket
        a = -g;
        torque = 0;
    elseif t < t_ar
%         fprintf("CHUTE MODE\n");
        F_d_chute = 0.5 * rho * C_d_chute * A * v^2; %Force of deployment of the drogue parachute

        % Only the chute and gravity are acting on the rocket
        a = (F_d_chute / m) - g;
        torque = 0;
    else
%         fprintf("AUTOROTATION MODE\n");
        x_test = 0:dx:b;
        opts = odeset('RelTol',1e-4,'AbsTol',1e-6);

        % Blade Model ODE. Produces a Matrix called forces over the span of
        % the blade
        % Executes a function blade_forces which calculates the forces in
        % the y-direction and torque

        [x, forces] = ode45(@(x, forces) blade_forces(x, forces, omega, v, psi, rho, mu, c, ...
            lift_data, drag_data, ...
            Cl_stall, Cd_stall), x_test, [0; 0], opts);

        torque = sum(forces(:, 1));
        F_blade = sum(forces(:, 2));

% figure(1);
% grid on;
% plot(x, forces(:, 1));
% xlabel("Blade Location (m)");
% ylabel("torque (N/m)");
% title("Blade Location vs. Torque");
%
% figure(2);
% plot(x, forces(:, 2));
% xlabel("Blade Location (s)");
% ylabel("Force Y (N)");
% title("Blade Location (m) vs. Force Y");

        % gravity + AR
        a = ((n * F_blade) / m) - g;
    end

%     fprintf("This is t: %d\n", t);
%     %fprintf("This is dt: %d\n", dt);
%     fprintf("This is omega: %d\n\n", omega);
%
%     fprintf("This is acc: %d\n", a);
%     fprintf("This is velocity: %d\n", v);
%     fprintf("This is height: %d\n", h);
%    fprintf("---------------\n");

    kinematics_out = [v; a; torque/I; a];
end

% This function calculates the blade forces at a particular time step

function [forces_out] = blade_forces(x, forces, omega, v_y, psi, rho, mu, c, ...
    lift_data, drag_data, ...
    Cl_stall, Cd_stall)

v_rel = calcv_rel(omega, x, v_y); % Calculates relative velocity
phi = calcphi(omega, x, v_y);     % Calculates angle of relative wing velocity
alpha = calcalpha(phi, psi);      % Calculates angle of attack

r_e = calcr_e(rho, v_rel, c, mu); % Calculates the reynolds number


[C_l, C_d] = calc_coefficients(r_e, alpha, phi, lift_data, drag_data, Cl_stall, Cd_stall);

d_A = c;

d_L = calcd_L(rho, d_A, C_l, v_rel);
d_D = calcd_D(rho, d_A, C_d, v_rel);

torque = x*(-d_D*cos(alpha)+d_L*sin(alpha));
d_Fy = (d_L*cos(alpha)+d_D*sin(alpha));

forces_out = [torque; d_Fy];
end
```

```
Elapsed time is 18.207604 seconds.
```

7.2.8 calc_coefficients.m

```
function [C_l, C_d] = calc_coefficients(r_e, alpha, phi, Cl_data, Cd_data,
Cl_stall, Cd_stall)

    alpha_in = alpha; % comes in as RADIANS

    rounded_re = round(r_e, -4);
    if rounded_re>1e6
       rounded_re = 1e6;
    end



    col = find(Cl_data(1,:)==rounded_re); %Finds the column that contains the
Cd and Cl values
    cl_values = Cl_data(2:32, col);
    cd_values = Cd_data(2:32, col);

    alphas = linspace(-5,25,32);
    [max_c max_index] = max(cl_values);

%     fprintf("r_e in: %d", r_e);
%     fprintf("rounded_re: %d", rounded_re);
%     fprintf("alpha in: %d", alpha);
%     fprintf("max_index: %d \n\n\n", max_index);

    max_alpha = alphas(max_index);

    alpha = (-5:25)';



    if alpha_in > deg2rad(max_alpha)
```

```
        alpha_in = rad2deg(alpha_in);
        C_l = interp1((19:60)', Cl_stall(1:42,1), alpha_in, 'spline');
        C_d = interp1((19:60)', Cd_stall(1:42,1), alpha_in, 'spline');
    else
        alpha_in = rad2deg(alpha_in);
%         fprintf("deg: %d", alpha_in);
        C_l = interp1(alpha, cl_values, alpha_in, 'spline');
        C_d = interp1(alpha, cd_values, alpha_in, 'spline');
    end
    %     plot(alpha, c_values);
end
```

7.2.9 calcalpha.m

```
function [alpha] = calcalpha(phi, psi)
%ALPHA Summary of this function goes here
%   Detailed explanation goes here
    alpha = phi - psi;
end
```

7.2.10 calcd_D.m

```
function [d_D] = calcd_D(rho, d_A, C_d, v_rel)
%D_L Summary of this function goes here
%   Detailed explanation goes here
    d_D = 0.5 * rho * d_A * C_d * v_rel^2;
end
```

7.2.11 calcd_L.m

```
function [d_L] = calcd_L(rho, d_A, C_l, v_rel)
%D_L Summary of this function goes here
%   Detailed explanation goes here
    d_L = 0.5 * rho * d_A * C_l * v_rel^2;
end
```

7.2.12 calcphi.m

```
function [phi] = calcphi(omega, x, v_y)
%PHI Summary of this function goes here
%   Detailed explanation goes here
        if omega == 0
            phi = 0.5  * pi;
        else
            phi = abs(atan(v_y / (omega * x))); %the absolute value was added
because
            %a negative normal angle is positive for AOA
        end
end
```

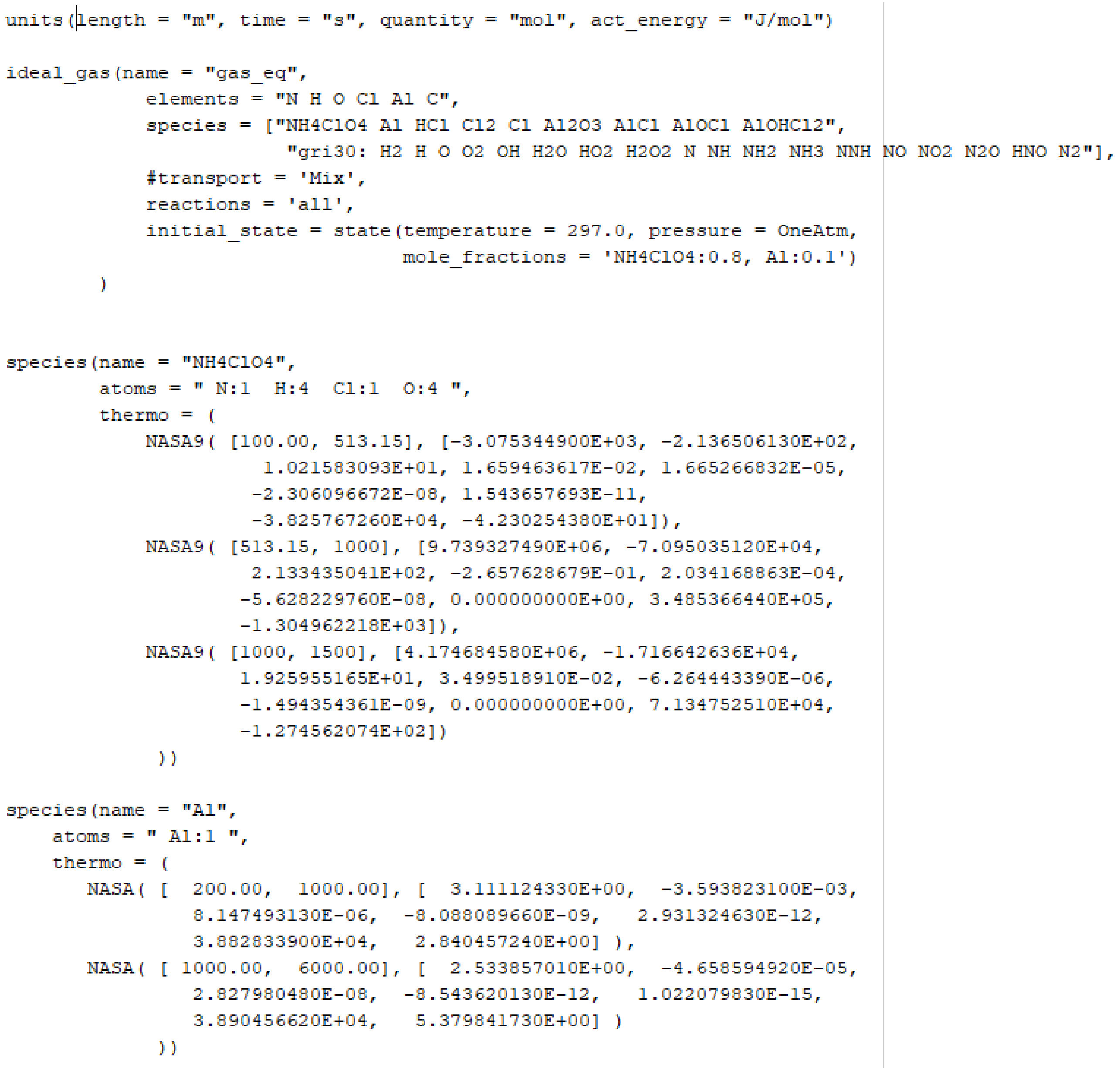
7.2.13 calcr_e.m

```
function [r_e] = calcr_e(rho, v_rel, c, mu)
%R_E Summary of this function goes here
%   Detailed explanation goes here
    r_e = (rho * abs(v_rel) * c) / mu;
end
```

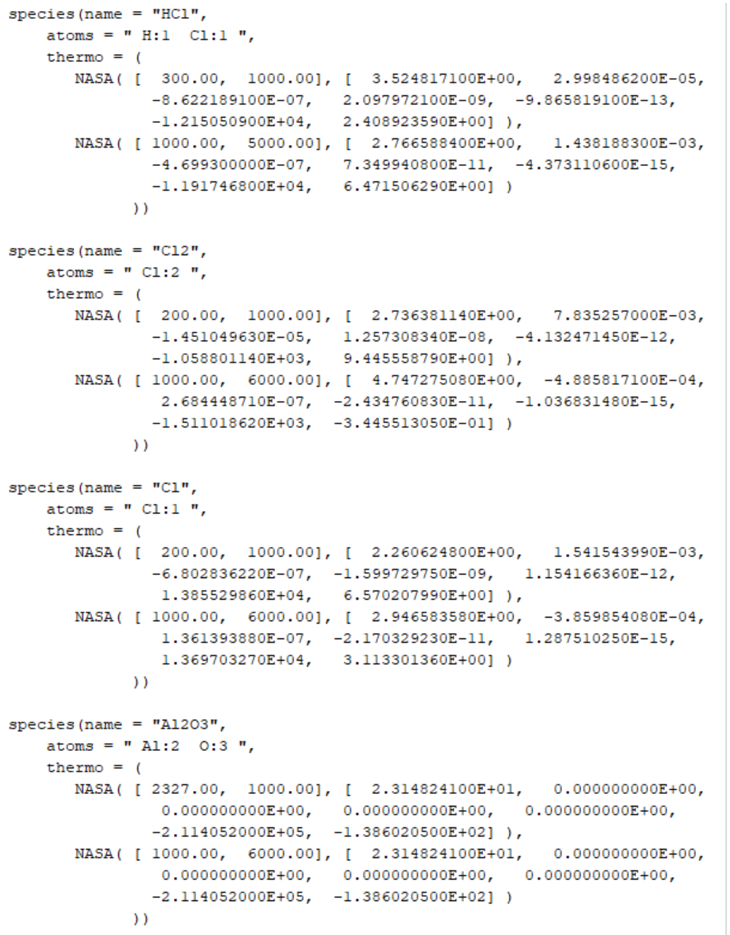
7.2.14 calcv_rel.m

```
function [v_rel] = calcv_rel(omega, x, v_y)
%V_REL Summary of this function goes here
%   Detailed explanation goes here
    v_rel = sqrt((omega*x)^2 + (v_y)^2);
end
```

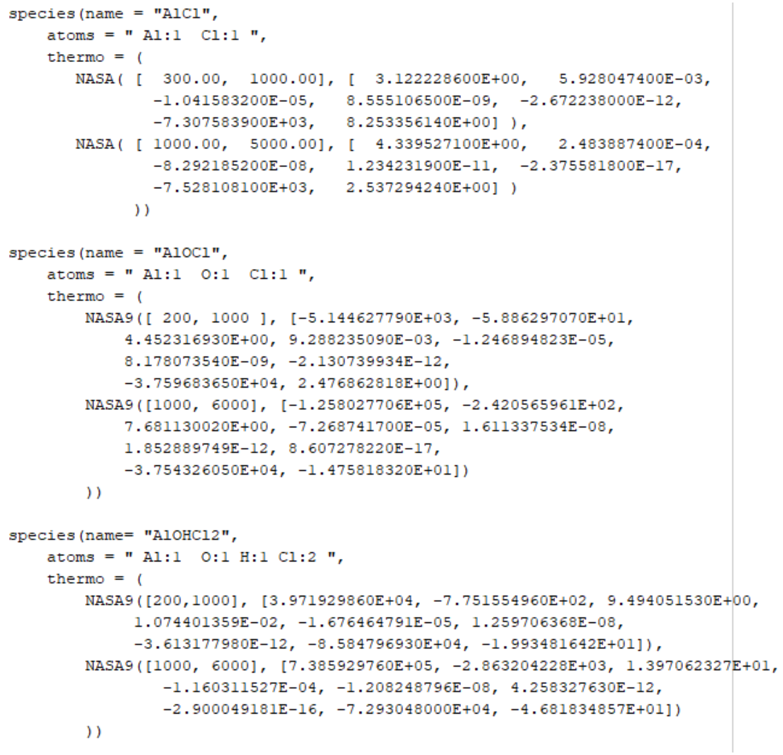
7.3 Appendix C Data File for Cantera Equilibrium Analysis



```
units(length = "m", time = "s", quantity = "mol", act_energy = "J/mol")

ideal_gas(name = "gas_eq",
            elements = "N H O Cl Al C",
            species = ["NH4ClO4 Al HCl Cl2 Cl Al2O3 AlCl AlOCl AlOHCl2",
                        "gri30: H2 H O O2 OH H2O HO2 H2O2 N NH NH2 NH3 NNH NO NO2 N2O HNO N2"],
            #transport = 'Mix',
            reactions = 'all',
            initial_state = state(temperature = 297.0, pressure = OneAtm,
                                  mole_fractions = 'NH4ClO4:0.8, Al:0.1')
        )


species(name = "NH4ClO4",
        atoms = " N:1  H:4  Cl:1  O:4 ",
        thermo = (
            NASA9( [100.00, 513.15], [-3.075344900E+03, -2.136506130E+02,
                      1.021583093E+01, 1.659463617E-02, 1.665266832E-05,
                     -2.306096672E-08, 1.543657693E-11,
                     -3.825767260E+04, -4.230254380E+01]),
            NASA9( [513.15, 1000], [9.739327490E+06, -7.095035120E+04,
                     2.133435041E+02, -2.657628679E-01, 2.034168863E-04,
                    -5.628229760E-08, 0.000000000E+00, 3.485366440E+05,
                    -1.304962218E+03]),
            NASA9( [1000, 1500], [4.174684580E+06, -1.716642636E+04,
                    1.925955165E+01, 3.499518910E-02, -6.264443390E-06,
                    -1.494354361E-09, 0.000000000E+00, 7.134752510E+04,
                    -1.274562074E+02])
             ))

species(name = "Al",
    atoms = " Al:1 ",
    thermo = (
       NASA( [  200.00,  1000.00], [  3.111124330E+00,  -3.593823100E-03,
                8.147493130E-06,  -8.088089660E-09,   2.931324630E-12,
                3.882833900E+04,   2.840457240E+00] ),
       NASA( [ 1000.00,  6000.00], [  2.533857010E+00,  -4.658594920E-05,
                2.827980480E-08,  -8.543620130E-12,   1.022079830E-15,
                3.890456620E+04,   5.379841730E+00] )
             ))
```

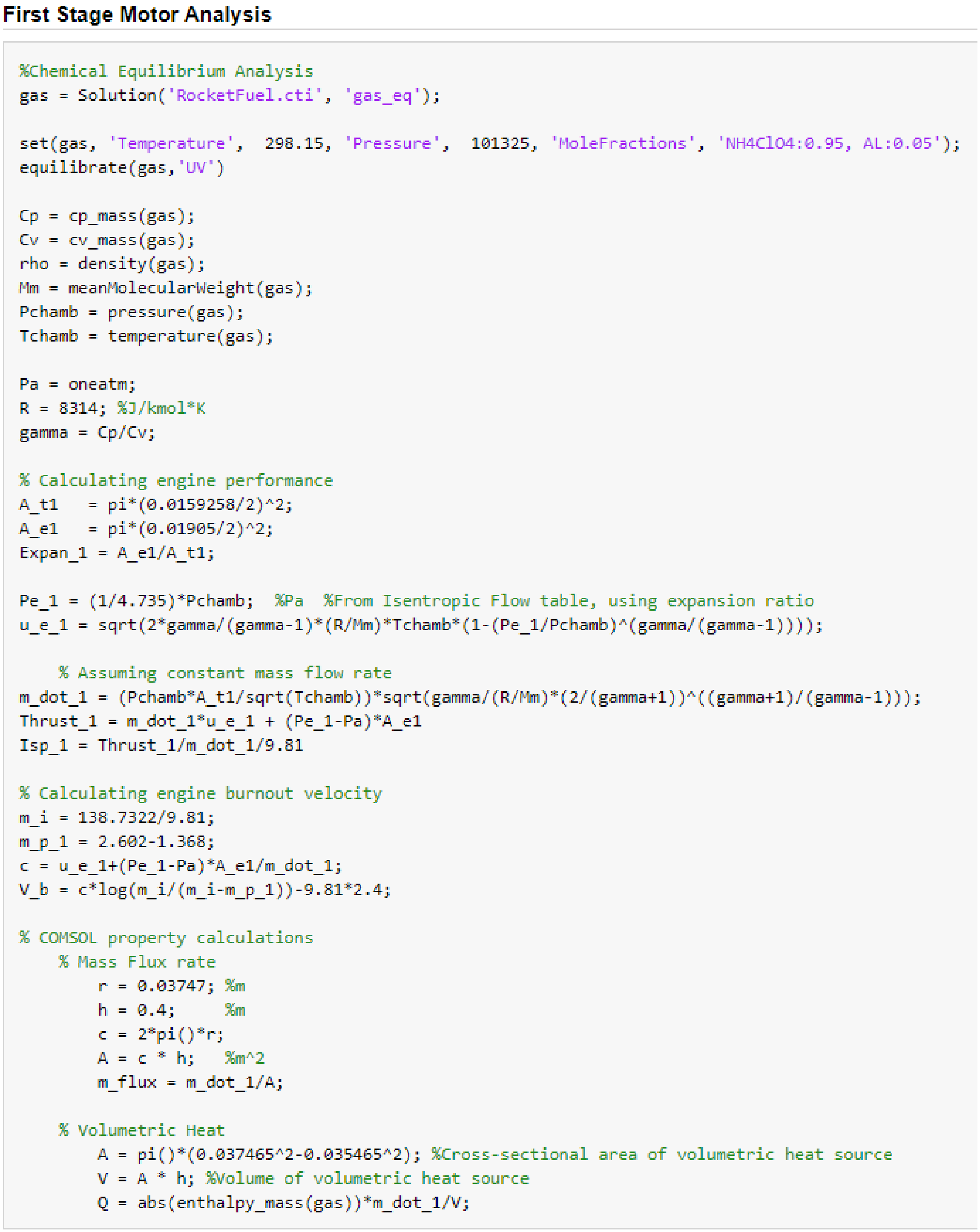


```
species(name = "HCl",
    atoms = " H:1  Cl:1 ",
    thermo = (
       NASA( [  300.00,  1000.00], [  3.524817100E+00,   2.998486200E-05,
               -8.622189100E-07,   2.097972100E-09,  -9.865819100E-13,
               -1.215050900E+04,   2.408923590E+00] ),
       NASA( [ 1000.00,  5000.00], [  2.766588400E+00,   1.438188300E-03,
               -4.699300000E-07,   7.349940800E-11,  -4.373110600E-15,
               -1.191746800E+04,   6.471506290E+00] )
             ))

species(name = "Cl2",
    atoms = " Cl:2 ",
    thermo = (
       NASA( [  200.00,  1000.00], [  2.736381140E+00,   7.835257000E-03,
               -1.451049630E-05,   1.257308340E-08,  -4.132471450E-12,
               -1.058801140E+03,   9.445558790E+00] ),
       NASA( [ 1000.00,  6000.00], [  4.747275080E+00,  -4.885817100E-04,
                2.684448710E-07,  -2.434760830E-11,  -1.036831480E-15,
               -1.511018620E+03,  -3.445513050E-01] )
             ))

species(name = "Cl",
    atoms = " Cl:1 ",
    thermo = (
       NASA( [  200.00,  1000.00], [  2.260624800E+00,   1.541543990E-03,
               -6.802836220E-07,  -1.599729750E-09,   1.154166360E-12,
                1.385529860E+04,   6.570207990E+00] ),
       NASA( [ 1000.00,  6000.00], [  2.946583580E+00,  -3.859854080E-04,
                1.361393880E-07,  -2.170329230E-11,   1.287510250E-15,
                1.369703270E+04,   3.113301360E+00] )
             ))

species(name = "Al2O3",
    atoms = " Al:2  O:3 ",
    thermo = (
       NASA( [ 2327.00,  1000.00], [  2.314824100E+01,   0.000000000E+00,
                0.000000000E+00,   0.000000000E+00,   0.000000000E+00,
               -2.114052000E+05,  -1.386020500E+02] ),
       NASA( [ 1000.00,  6000.00], [  2.314824100E+01,   0.000000000E+00,
                0.000000000E+00,   0.000000000E+00,   0.000000000E+00,
               -2.114052000E+05,  -1.386020500E+02] )
             ))
```



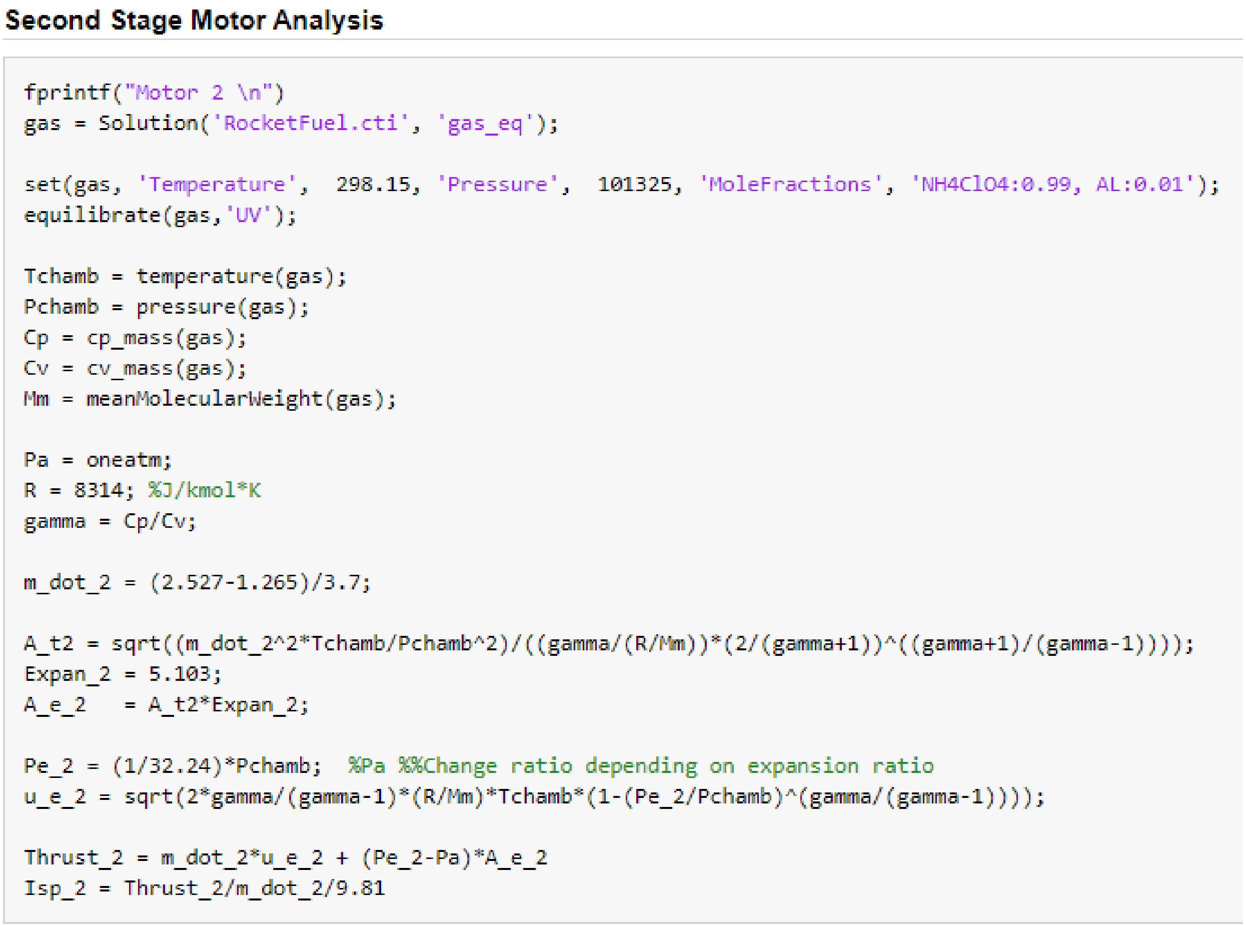
```
species(name = "AlCl",
    atoms = " Al:1  Cl:1 ",
    thermo = (
       NASA( [  300.00,  1000.00], [  3.122228600E+00,   5.928047400E-03,
               -1.041583200E-05,   8.555106500E-09,  -2.672238000E-12,
               -7.307583900E+03,   8.253356140E+00] ),
       NASA( [ 1000.00,  5000.00], [  4.339527100E+00,   2.483887400E-04,
               -8.292185200E-08,   1.234231900E-11,  -2.375581800E-17,
               -7.528108100E+03,   2.537294240E+00] )
             ))

species(name = "AlOCl",
    atoms = " Al:1  O:1  Cl:1 ",
    thermo = (
        NASA9([ 200, 1000 ], [-5.144627790E+03, -5.886297070E+01,
            4.452316930E+00, 9.288235090E-03, -1.246894823E-05,
            8.178073540E-09, -2.130739934E-12,
            -3.759683650E+04, 2.476862818E+00]),
        NASA9([1000, 6000], [-1.258027706E+05, -2.420565961E+02,
            7.681130020E+00, -7.268741700E-05, 1.611337534E-08,
            1.852889749E-12, 8.607278220E-17,
            -3.754326050E+04, -1.475818320E+01])
        ))

species(name= "AlOHCl2",
    atoms = " Al:1  O:1 H:1 Cl:2 ",
    thermo = (
        NASA9([200,1000], [3.971929860E+04, -7.751554960E+02, 9.494051530E+00,
             1.074401359E-02, -1.676464791E-05, 1.259706368E-08,
             -3.613177980E-12, -8.584796930E+04, -1.993481642E+01]),
        NASA9([1000, 6000], [7.385929760E+05, -2.863204228E+03, 1.397062327E+01,
                -1.160311527E-04, -1.208248796E-08, 4.258327630E-12,
                -2.900049181E-16, -7.293048000E+04, -4.681834857E+01])
        ))
```

7.4 Appendix D Cantera and Motor Analysis MATLAB Code



First Stage Motor Analysis

```
%Chemical Equilibrium Analysis
gas = Solution('RocketFuel.cti', 'gas_eq');

set(gas, 'Temperature',  298.15, 'Pressure',  101325, 'MoleFractions', 'NH4ClO4:0.95, AL:0.05');
equilibrate(gas,'UV')

Cp = cp_mass(gas);
Cv = cv_mass(gas);
rho = density(gas);
Mm = meanMolecularWeight(gas);
Pchamb = pressure(gas);
Tchamb = temperature(gas);

Pa = oneatm;
R = 8314; %J/kmol*K
gamma = Cp/Cv;

% Calculating engine performance
A_t1   = pi*(0.0159258/2)^2;
A_e1   = pi*(0.01905/2)^2;
Expan_1 = A_e1/A_t1;

Pe_1 = (1/4.735)*Pchamb;  %Pa  %From Isentropic Flow table, using expansion ratio
u_e_1 = sqrt(2*gamma/(gamma-1)*(R/Mm)*Tchamb*(1-(Pe_1/Pchamb)^(gamma/(gamma-1))));

    % Assuming constant mass flow rate
m_dot_1 = (Pchamb*A_t1/sqrt(Tchamb))*sqrt(gamma/(R/Mm)*(2/(gamma+1))^((gamma+1)/(gamma-1)));
Thrust_1 = m_dot_1*u_e_1 + (Pe_1-Pa)*A_e1
Isp_1 = Thrust_1/m_dot_1/9.81

% Calculating engine burnout velocity
m_i = 138.7322/9.81;
m_p_1 = 2.602-1.368;
c = u_e_1+(Pe_1-Pa)*A_e1/m_dot_1;
V_b = c*log(m_i/(m_i-m_p_1))-9.81*2.4;

% COMSOL property calculations
    % Mass Flux rate
        r = 0.03747; %m
        h = 0.4;     %m
        c = 2*pi()*r;
        A = c * h;   %m^2
        m_flux = m_dot_1/A;

    % Volumetric Heat
        A = pi()*(0.037465^2-0.035465^2); %Cross-sectional area of volumetric heat source
        V = A * h; %Volume of volumetric heat source
        Q = abs(enthalpy_mass(gas))*m_dot_1/V;
```

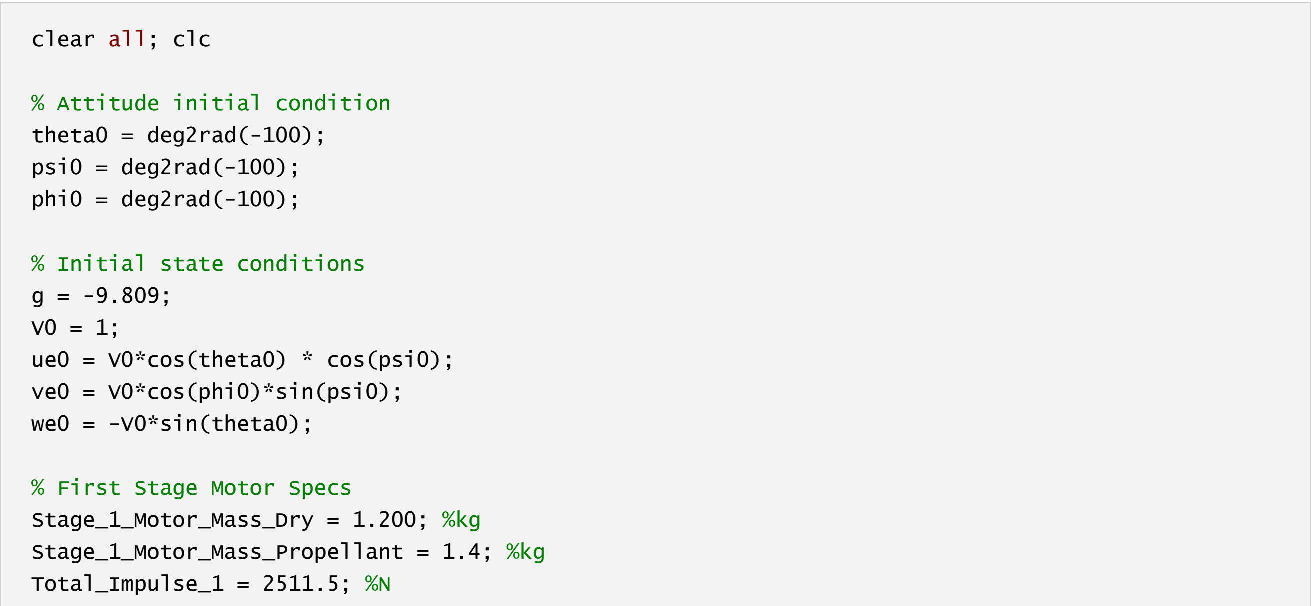


Second Stage Motor Analysis

```
fprintf("Motor 2 \n")
gas = Solution('RocketFuel.cti', 'gas_eq');

set(gas, 'Temperature',  298.15, 'Pressure',  101325, 'MoleFractions', 'NH4ClO4:0.99, AL:0.01');
equilibrate(gas,'UV');

Tchamb = temperature(gas);
Pchamb = pressure(gas);
Cp = cp_mass(gas);
Cv = cv_mass(gas);
Mm = meanMolecularWeight(gas);

Pa = oneatm;
R = 8314; %J/kmol*K
gamma = Cp/Cv;

m_dot_2 = (2.527-1.265)/3.7;

A_t2 = sqrt((m_dot_2^2*Tchamb/Pchamb^2)/((gamma/(R/Mm))*(2/(gamma+1))^((gamma+1)/(gamma-1))));
Expan_2 = 5.103;
A_e_2    = A_t2*Expan_2;

Pe_2 = (1/32.24)*Pchamb;  %Pa %%Change ratio depending on expansion ratio
u_e_2 = sqrt(2*gamma/(gamma-1)*(R/Mm)*Tchamb*(1-(Pe_2/Pchamb)^(gamma/(gamma-1))));

Thrust_2 = m_dot_2*u_e_2 + (Pe_2-Pa)*A_e_2
Isp_2 = Thrust_2/m_dot_2/9.81
```

7.5 Appendix E FDA Task 1 Codes

7.5.1 20’-21’ Rocket team 1 Main Analysis code for FDA Translational Solver



```
clear all; clc

% Attitude initial condition
theta0 = deg2rad(-100);
psi0 = deg2rad(-100);
phi0 = deg2rad(-100);

% Initial state conditions
g = -9.809;
V0 = 1;
ue0 = V0*cos(theta0) * cos(psi0);
ve0 = V0*cos(phi0)*sin(psi0);
we0 = -V0*sin(theta0);

% First Stage Motor Specs
Stage_1_Motor_Mass_Dry = 1.200; %kg
Stage_1_Motor_Mass_Propellant = 1.4; %kg
Total_Impulse_1 = 2511.5; %N
```

```
Burn_time_1 = 2.5;
mdot_1 = Stage_1_Motor_Mass_Propellant / Burn_time_1;
ISP_1 = Total_Impulse_1 / (Stage_1_Motor_Mass_Propellant * abs(g)); %2.5 Second burn-time

% Thrust profile of the stage 1 Motor - Utilized in "Point_Mass_ascent_Stage_1"
Thrust_Profile_1 =    [0.0040 895.149; 0.015 1119.762; 0.025 1093.337 ;0.095 1096.640 ;0.200
1109.853; 0.300 1116.459
   0.400 1123.065 ; 0.500 1132.975 ; 0.600 1139.581 ; 0.700 1136.278 ; 0.800 1136.278 ; 0.900
1136.278 ; 1.000 1139.581
   1.100 1132.975 ; 1.200 1129.672 ; 1.300 1126.369 ; 1.400 1119.762 ; 1.500 1109.853 ; 1.600
1096.640 ;  1.700 1063.609
   1.800 1017.365 ; 1.900 971.121 ; 2.000 914.968 ; 2.100 868.724 ; 2.180 865.421 ; 2.200
878.634 ; 2.218 858.815 ; 2.269 670.536
   2.300 578.048 ; 2.332 445.923 ; 2.356 336.920 ; 2.389 224.613 ; 2.436 105.7 ; 2.500 0.0];

% Second Stage Motor Specs
Stage_2_Motor_Mass_Dry = 2.528-1.182; %kilograms
Stage_2_Motor_Mass_Propellant = 1.182; %kilograms
Total_Impulse_2 = 2430.4; %Newtons
Burn_time_2 = 3.7; % Seconds
mdot_2 = Stage_2_Motor_Mass_Propellant / Burn_time_2;
ISP_2 = Total_Impulse_2 / (Stage_2_Motor_Mass_Propellant * abs(g)); %3.7 Second burn-time

% Thrust profile of the stage 2 Motor - Utilized in "Point_Mass_ascent_Stage_2"
Thrust_Profile_2 = [   0.041 588.591 ; 0.073 659.371 ; 0.122 635.157 ; 0.225 634.226 ; 0.679
713.388
   1.039 751.572 ; 1.241 758.091 ; 1.832 727.357 ; 2.298 687.311 ; 2.729 667.753 ; 3.195 645.402
   3.367 670.547 ; 3.584 182.538 ; 3.672 55.879 ; 3.72 18.626 ; 3.798 0.0];

% Total Rocket Mass
Rocket_Wet_Mass = 138.7322/abs(g); %kg, value provided by Team
Rocket_Dry_Mass = Rocket_Wet_Mass - Stage_2_Motor_Mass_Propellant -
Stage_1_Motor_Mass_Propellant; % Dry mass of both stages
Stage_1_Dry = Rocket_Wet_Mass - Stage_1_Motor_Mass_Propellant; % Mass after the fuel is spent,
but they are still together
Stage_1_Dry_Sep = Rocket_Wet_Mass - 67.192304/abs(g); % Dry mass after the first stage
separation, initializing flight mass of the Stage 2 Protocol
Stage_2_Dry = Rocket_Wet_Mass - 67.192304/abs(g) - Stage_2_Motor_Mass_Propellant; % Final dry
mass of Stage 2 after firing is complete

%Initializing time-trackers
dt = 0.01; % Time steps that the states will be recorded by
time = 0; % Setting the initial time to zero

% Random wind direction (magnitude will be defined in the solver)
Random_Wind = randn(3,1);
Wind_Direction = Random_Wind/norm(Random_Wind);

% Quaternion Definition
A0 =    (sin(psi0/2)*sin(theta0/2)*cos(psi0/2)) - (cos(psi0/2)*cos(theta0/2)*sin(phi0/2));
B0 = -1*(cos(psi0/2)*sin(theta0/2)*cos(phi0/2)) - (sin(psi0/2)*cos(theta0/2)*sin(phi0/2));
C0 = -1*(sin(psi0/2)*cos(theta0/2)*cos(phi0/2)) + (cos(psi0/2)*sin(theta0/2)*sin(phi0/2));
D0 = -1*(cos(psi0/2)*cos(theta0/2)*cos(phi0/2)) - (sin(psi0/2)*sin(theta0/2)*sin(phi0/2));
```

```
theta = theta0;psi = psi0; phi = phi0; V = V0;ue = ue0;ve = ve0;we = we0;
A = A0;B = B0;C = C0;D = D0;
Cm = [A^2-B^2-C^2+D^2,2*(A*B-C*D),2*(A*C+B*D)
    2*(A*B+C*D),-1*A^2+B^2-C^2+D^2,2*(B*C-A*D)
    2*(A*C-B*D),2*(B*C+A*D),-1*A^2-B^2+C^2+D^2];
% The "Cm" term represents the transformation DCM from the body fixed to
% inertial coords. To move opposite, take the transpose of Cm
V0 = Cm'*[ue0,ve0,we0]';
u0 = V0(1); v0 = V0(2); w0 = V0(3);
alpha0 = deg2rad(2);
y0 = [0 0 0 u0 v0 w0 Rocket_Wet_Mass alpha0];

% Define the initial state values of the system

Position = [y0(1),y0(2),y0(3)];
VelocityVector = [y0(4),y0(5),y0(6)];
i = 1; % Loop tracker initial value
State_Inertial = Body2Inertial([Position';VelocityVector'],Cm);
State_Storage_Inertial = [];
State_Storage = [];
m = Rocket_Wet_Mass;
t1=0;


%Full Stage Model
while State_Inertial(6) >=0 % While The Vertical Velocity (Z-Direction Earth Frame) is positive
    State_Initial = [Position';VelocityVector';m];
    % Call the initial Solver for the flight of the first motor
    if time <=Burn_time_1
    [t1,y1] = ode45(@(t,y)
Point_Mass_ascent_Stage_1(t,y,Burn_time_1,ISP_1,Thrust_Profile_1,Wind_Direction), [time,
time+dt], State_Initial);
    % Call the second solver for the separation sequence and the flight of the second motor
    elseif time<=(Burn_time_1 + 1)
    [t1,y1] = ode45(@(t,y)
Point_Mass_ascent_Stage_2(t,y,Burn_time_1,ISP_2,Burn_time_2,Thrust_Profile_2,Wind_Direction),
[time, time+dt], State_Initial);
    % This solver uses the second solver, but analyzes the separation time
    % (1 Second as provided by PTSS Mechanical Separation System)
    else
    [t1,y1] = ode45(@(t,y)
Point_Mass_ascent_Stage_2(t,y,Burn_time_1,ISP_2,Burn_time_2,Thrust_Profile_2,Wind_Direction),
[time, time+dt], State_Initial);
    end
    % Update the state with the values that any of the above solvers give
    UpdatedState = y1(end,:);
    Position = UpdatedState(1:3);
    VelocityVector = UpdatedState(4:6);
    if (time<=Burn_time_1)
    m = UpdatedState(7);
    elseif (time>=Burn_time_1) andand (time<=Burn_time_1 + 1)
    m = Stage_1_Dry_Sep;
    else
    m = UpdatedState(7);
    end


    % Updating the values for the next loop and storing some for plotting
    time = time + dt;
    timesteps(i) = time; % Stores the time vector for plotting
    State_Inertial = Body2Inertial(UpdatedState,Cm); % Converts the Body fixed coordinates to the
inertial frame
    State_Storage(i,:) = UpdatedState; % Stores the body fixed coordinates State
    State_Storage_Inertial(i,:) = State_Inertial; % Stores the Intertial Frame State
    altitude(i) = State_Inertial(3); % Stores the altitude at each timestep

i = i+1; % Updates the storage protocol to allow for easy storage

end
% Descent Code implementation
[timestamps,State_sto] = Autorotation_Function(altitude(end));
for j=1:length(timestamps)
timestamps_normalized(j) = timesteps(end) + timestamps(j);
end

% Combine the states from the Main Solver and the Descent Solver
State_sto_transpose = State_sto';
Combined_Time = [timesteps timestamps_normalized];
Combined_altitude = [altitude State_sto_transpose(1,:)];
% Stores the values of position for the descent stage
Horizontal_Priming_IV = [State_Storage_Inertial(end,1),State_Storage_Inertial(end,2),...
State_Storage_Inertial(end,4),State_Storage_Inertial(end,5)];
[timerino,staterino] = ode45(@(t,y) Descent_Horizontal(t,y,m),timestamps_normalized,
Horizontal_Priming_IV');

Horizontal_Translation_descent = staterino;

%Combine the states from the ascent stage and the descent stage
Combined_X_State = [State_Storage_Inertial(:,1); Horizontal_Translation_descent(:,1)];
Combined_Y_State = [State_Storage_Inertial(:,2); Horizontal_Translation_descent(:,2)];
Combined_Z_State = Combined_altitude;

Combined_Position = [Combined_X_State,Combined_Y_State,Combined_Z_State'];

% %% Plots
% %Graph of the entire flight time from start to landing
% figure ()
% plot3(Combined_Position(:,1),Combined_Position(:,2),Combined_Position(:,3))
% title('Graph of Rocket Flight (Inertial Reference Frame)')
% zlabel('altitude (meters)')
% xlabel('X-Inertial Direction (meters)')
% ylabel('Y-Inertial Direction (meters)')
% grid on
% set(gca,'FontSize',20,'fontWeight','bold')
% set(findall(gcf,'type','text'),'FontSize',20,'fontWeight','bold')
%
% %Graph of the flight up to apogee
% figure ()
% plot3(State_Storage_Inertial(:,1),State_Storage_Inertial(:,2),State_Storage_Inertial(:,3))
% title('Graph of Rocket Flight (up to apogee)')
% zlabel('altitude (meters)')
% xlabel('X-Inertial Direction (meters)')
% ylabel('Y-Inertial Direction (meters)')
% grid on
%
% % Graph solely showing the X - body velocity
% figure ()
% plot(timesteps, State_Storage(:,4))
% title('Graph of Velocity in X-Body direction during ascent')
% ylabel('Velocity (m/s)')
% xlabel('Time (seconds)')
% set(gca,'FontSize',36,'fontWeight','bold')
% set(findall(gcf,'type','text'),'FontSize',36,'fontWeight','bold')
%
% % Graph solely showing the Z - inertial velocity
% figure ()
% plot(timesteps, State_Storage_Inertial(:,6))
% title('Graph of Velocity in Z-Inertial')
% ylabel('Velocity in Z-Inertial Frame')
% xlabel('Time (Seconds)')
% set(gca,'FontSize',36,'fontWeight','bold')
% set(findall(gcf,'type','text'),'FontSize',36,'fontWeight','bold')
%
% % Graph of the Altitude with time
% figure ()
% plot(Combined_Time, Combined_altitude)
% ylabel('Altitude (meters)')
% xlabel('Time of Flight (Seconds)')
% title('Altitude of Rocket during Flight')
% set(gca,'FontSize',30,'fontWeight','bold')
% set(findall(gcf,'type','text'),'FontSize',30,'fontWeight','bold')
%
% % Graph of the Absolute Velocity during Flight
% figure ()
% plot(timesteps, sqrt(State_Storage_Inertial(:,6).^2 + State_Storage_Inertial(:,5).^2 +
State_Storage_Inertial(:,4).^2))
% title('Total Velocity During Flight')
% xlabel('Time (seconds)')
% ylabel('Velocity (m/s)')
%
% % Graph of the mass of the rocket during flight
% figure ()
% plot(timesteps, State_Storage(:,7))
% title('Mass of Rocket During Flight')
% xlabel('Time after Launch (Seconds)')
% ylabel('Mass (kg)')
%
% % Graph of the X - inertial position during flight
% figure()
% plot(Combined_Time, Combined_Position(:,1))
% title('X-Direction Position During Flight')
% xlabel('Time after Launch (Seconds)')
% ylabel('X-Position (meters)')
% set(gca,'FontSize',30,'fontWeight','bold')
% set(findall(gcf,'type','text'),'FontSize',30,'fontWeight','bold')
%
% % Graph of the Y - inertial position during flight
% figure()
% plot(Combined_Time, Combined_Position(:,2))
% title('Y-Direction Position During Flight')
% xlabel('Time after Launch (Seconds)')
% ylabel('Y-Position (meters)')
% set(gca,'FontSize',30,'fontWeight','bold')
% set(findall(gcf,'type','text'),'FontSize',30,'fontWeight','bold')
%
```

7.5.2 Point_Mass_ascent_Stage_1

```
function [dydt,y_Inertial] =
Point_Mass_ascent_Stage_1(t,y1,Burn_time_1,ISP_1,Thrust_Profile,Wind_Direction)

Cd = 0.4258; % Solution found by FDA Task 2
S = ((20/2)/(12*3.28))^2*pi;  % Surface area given by ARS Frontal
g = -9.809;
dydt = zeros(7,1);
SA = 2*pi*((5.5/2)/(12*3.28))*((77.4/2)/(12*3.28)); % Side surface area given by ARS
% Create stagnant quaternion values for a stagnant DCM
theta0 = deg2rad(-100);
psi0 = deg2rad(-100);
phi0 = deg2rad(-100);
A0 =    (sin(psi0/2)*sin(theta0/2)*cos(psi0/2)) - (cos(psi0/2)*cos(theta0/2)*sin(phi0/2));
B0 = -1*(cos(psi0/2)*sin(theta0/2)*cos(phi0/2)) - (sin(psi0/2)*cos(theta0/2)*sin(phi0/2));
C0 = -1*(sin(psi0/2)*cos(theta0/2)*cos(phi0/2)) + (cos(psi0/2)*sin(theta0/2)*sin(phi0/2));
D0 = -1*(cos(psi0/2)*cos(theta0/2)*cos(phi0/2)) - (sin(psi0/2)*sin(theta0/2)*sin(phi0/2));
A = A0;B = B0;C = C0;D = D0;
%Define a stagnant DCM for the point mass system
Cm = [A^2-B^2-C^2+D^2,2*(A*B-C*D),2*(A*C+B*D)
    2*(A*B+C*D),-1*A^2+B^2-C^2+D^2,2*(B*C-A*D)
    2*(A*C-B*D),2*(B*C+A*D),-1*A^2-B^2+C^2+D^2];
y_altitude_check = Body2Inertial(y1,Cm);
[~,rho] = Atmo(y_altitude_check(3));

% Define the strength of the wind, assumed that the wind is always blowing
% positively in the pre-defined direction
Wind_Strength = abs(randn) * 4; % m/s
Wind = Wind_Direction*Wind_Strength;
Wind_Body = Cm'*Wind;
Wind_Force = 0.4*SA*rho*norm(Wind_Body)*norm(Wind_Body);
Wind_Specific_Force = Wind_Force * Wind_Body / norm(Wind_Body);


if t <= Burn_time_1
    T = interp1(Thrust_Profile(:,1), Thrust_Profile(:,2), t, 'spline');
    mdot = T / (abs(g)*ISP_1);
else
    T = 0;
    mdot = 0;
end

if (t>=1) andand (t<=2)
    Wind_Contribution(1) = Wind_Specific_Force(1);
    Wind_Contribution(2) = Wind_Specific_Force(2);
    Wind_Contribution(3) = Wind_Specific_Force(3);
else
    Wind_Contribution(1) = 0;  % Use S because this is frontal area
    Wind_Contribution(2) = 0;
    Wind_Contribution(3) = 0;
end


g_body = Cm'*[0,0,g]';
dydt(1) = y1(4);     % u, velocity in X-direction
dydt(2) = y1(5);     % v, velocity in Y-direction
dydt(3) = y1(6);     % w, velocity in Z-direction
dydt(4) = (T - Cd*.5*rho*y1(4)*abs(y1(4))*S + Wind_Contribution(1))/y1(7) + g_body(1); % Xba,
acceleration in body Y-direction
dydt(5) = Wind_Contribution(2) + g_body(2);       % Yba, acceleration in body Y-direction
dydt(6) = Wind_Contribution(3) + g_body(3);      % Zba, acceleration in body Z-direction
dydt(7) = -mdot;        % mdot, mass flowrate



end
```

7.5.3 Point_Mass_ascent_stage_2

```
function [dydt,y_Inertial] =
Point_Mass_ascent_Stage_2(t,y1,Burn_time_1,ISP_2,Burn_time_2,Thrust_Profile,Wind_Direction)


Cd = 0.4258; % Solution found by FDA Task 2
S = ((20/2)/(12*3.28))^2*pi;  % Surface area given by ARS Frontal
g = -9.809;
dydt = zeros(7,1);
SA = 2*pi*((5.5/2)/(12*3.28))*((77.4/2)/(12*3.28)); % Side surface area given by ARS
% Create stagnant quaternion values for a stagnant DCM
theta0 = deg2rad(-100);
psi0 = deg2rad(-100);
phi0 = deg2rad(-100);
A0 =    (sin(psi0/2)*sin(theta0/2)*cos(psi0/2)) - (cos(psi0/2)*cos(theta0/2)*sin(phi0/2));
B0 = -1*(cos(psi0/2)*sin(theta0/2)*cos(phi0/2)) - (sin(psi0/2)*cos(theta0/2)*sin(phi0/2));
C0 = -1*(sin(psi0/2)*cos(theta0/2)*cos(phi0/2)) + (cos(psi0/2)*sin(theta0/2)*sin(phi0/2));
D0 = -1*(cos(psi0/2)*cos(theta0/2)*cos(phi0/2)) - (sin(psi0/2)*sin(theta0/2)*sin(phi0/2));
```

```
A = A0;B = B0;C = C0;D = D0;
%Define a stagnant DCM for the point mass system
Cm = [A^2-B^2-C^2+D^2,2*(A*B-C*D),2*(A*C+B*D)
    2*(A*B+C*D),-1*A^2+B^2-C^2+D^2,2*(B*C-A*D)
    2*(A*C-B*D),2*(B*C+A*D),-1*A^2-B^2+C^2+D^2];



y_altitude_check = Body2Inertial(y1,Cm);
[~,rho] = Atmo(y_altitude_check(3));
Sep_Time =  1; % Indicates a stage separation time of 1 second

% Define the strength of the wind, assumed that the wind is always blowing
% positively in the pre-defined direction

Wind_Strength = abs(randn) * 4; % m/s
Wind = Wind_Direction*Wind_Strength;
Wind_Body = Cm'*Wind;
Wind_Force = 0.4*SA*rho*norm(Wind_Body)*norm(Wind_Body); %Newtons
Wind_Specific_Force = Wind_Force * Wind_Body / norm(Wind_Body);

% What's going on with the "g" values here? Not super consistent or good,
% must fix
if (Burn_time_1<=t) andand (t<=(Burn_time_1 + Sep_Time)) % No thrust or mdot during stage
separation
   T = 0;
   mdot = 0;
elseif (Burn_time_1 + Sep_Time<=t) andand (t<= Burn_time_1 + Sep_Time + Burn_time_2)
    T = interp1(Thrust_Profile(:,1), Thrust_Profile(:,2), (t-(Burn_time_1+Sep_Time)),
'spline'); % t-2.5 accounts for the time of the first stage
    mdot = T / (abs(g)*ISP_2);
else
    T = 0;
    mdot = 0;
end

if (t>=Burn_time_1 + Sep_Time + 1) andand (t<=Burn_time_1 + Sep_Time + 3)
    Wind_Contribution(1) = Wind_Specific_Force(1);
    Wind_Contribution(2) = Wind_Specific_Force(2);
    Wind_Contribution(3) = Wind_Specific_Force(3);
else
    Wind_Contribution(1) = 0;
    Wind_Contribution(2) = 0;
    Wind_Contribution(3) = 0;
end

g_body = Cm'*[0,0,g]';
dydt(1) = y1(4);     % u, velocity in X-direction
dydt(2) = y1(5);     % v, velocity in Y-direction
dydt(3) = y1(6);     % w, velocity in Z-direction
dydt(4) = (T - Cd*.5*rho*y1(4)*abs(y1(4))*S + Wind_Contribution(1))/y1(7) + g_body(1);
dydt(5) = Wind_Contribution(2) + g_body(2);       % Yba, acceleration in body Y-direction
dydt(6) = Wind_Contribution(3) + g_body(3);      % Zba, acceleration in body Z-direction
dydt(7) = -mdot;        % mdot, mass flowrate
```

```
y_Inertial = Body2Inertial(y1,Cm);

end
```

7.5.4 Body2Inertial.m

```
function [y_Inertial] = Body2Inertial(y,Cm)
%The default state of Cm as I defined it converts from body to earth
y_position_body = [y(1),y(2),y(3)]';
y_velocity_body = [y(4),y(5),y(6)]';

y_Inertial_position = Cm*y_position_body;
y_Inertial_velocity = Cm*y_velocity_body;
y_Inertial = [y_Inertial_position',y_Inertial_velocity'];

end
```

7.5.5 Inertial2Body.m

```
function [y_Body] = Inertial2Body(y,Cm)
%The default state of Cm as I defined it converts from body to earth
y_position_inertial = [y(1),y(2),y(3)]';
y_velocity_inertial = [y(4),y(5),y(6)]';

y_Body_position = Cm'*y_position_inertial;
y_Body_velocity = Cm'*y_velocity_inertial;
y_Body = [y_Body_position',y_Body_velocity'];

end
```

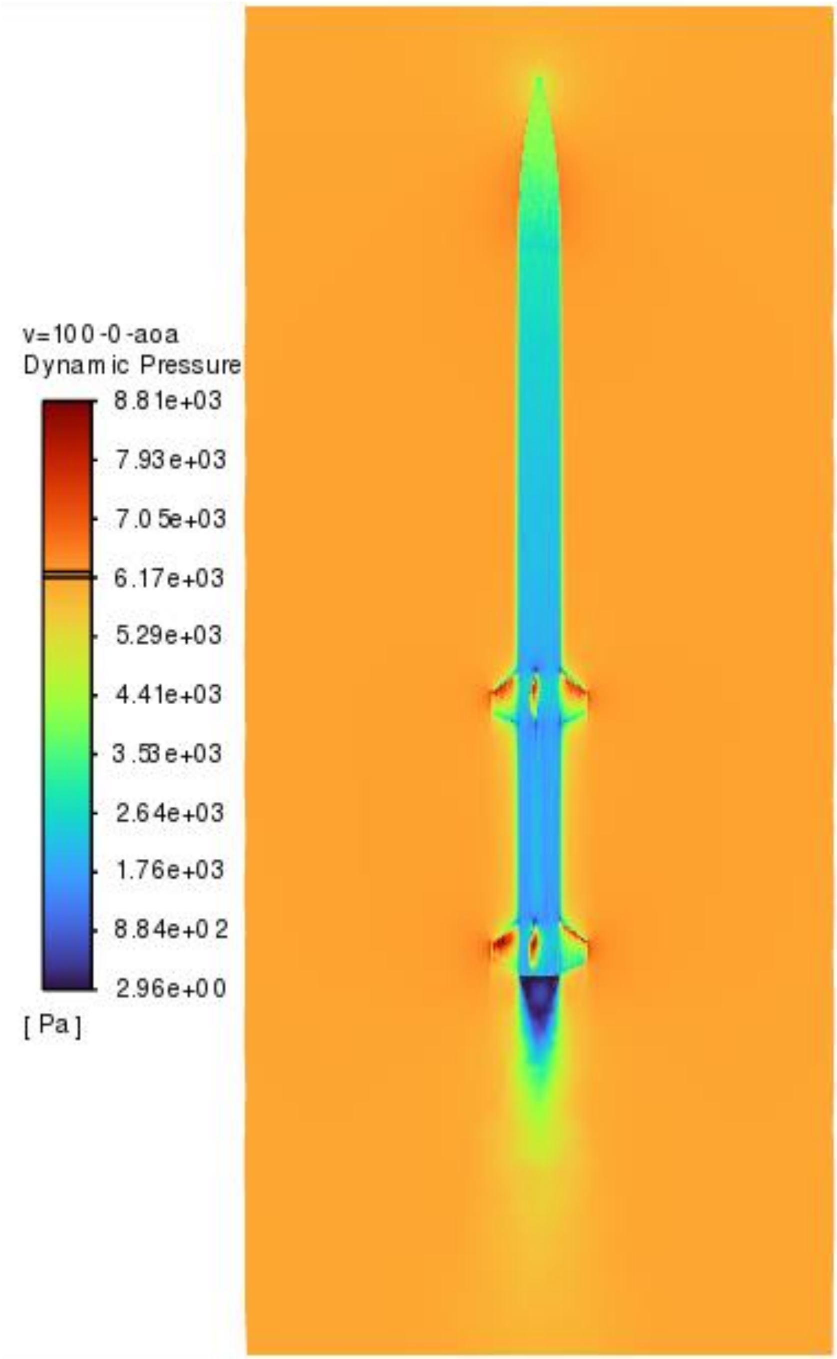
7.5.6 Atmo.m

```
function [tau,rho] = Atmo(h)
%Determines the pressure as a function of altitude provided
%various constants. IN TERMS OF METERS! NOT KM
rho_sl=1.225; %Standard Density at Sea Level (kg/m^3)
P_sl=101.32; % Standard Pressure at Sea Level (kPa)
tau_sl=288.16; %Standard Temperature at Sea Level (K)
g=9.8; %Gravity (m/s^2)
a1=-6.5*10^-3; %Temperature Gradient (K/m)
R=287; %Specific Gas Constant (J/(kg*K))
P_11=22.346; %Pressure at h=11 km
rho_11=0.3642; %Density at h=11 km
tau_11 =216.66;  %Temperature at h=11 k (used as gradient)
```

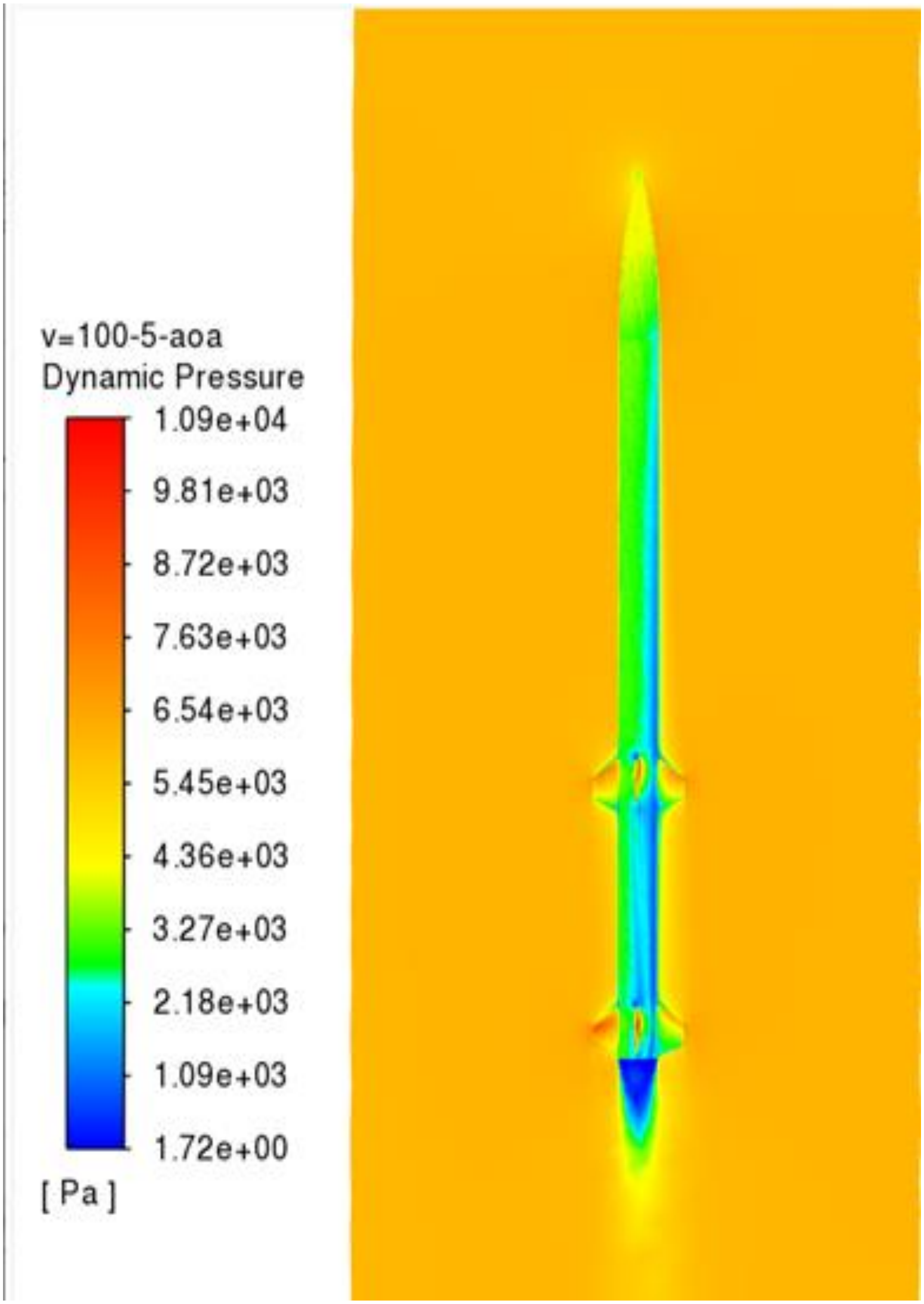
```
if h<11000

    tau = tau_sl+a1*h; %to obtain temperature as a function of altitude 0-11 km
    rho = rho_sl*((tau/tau_sl)^(-1-g/(a1*R))); %main outcome

else
    rho = rho_11*exp((-g*(h-11000))/(R*tau_11));
    tau=tau_11;
end
end

%This code can be updated to provide a 3x1 vector/matrix with temperature,
%pressure and density all based on altitude
```

7.6 Appendix F Dynamic Pressure Contours for Full Rocket Body

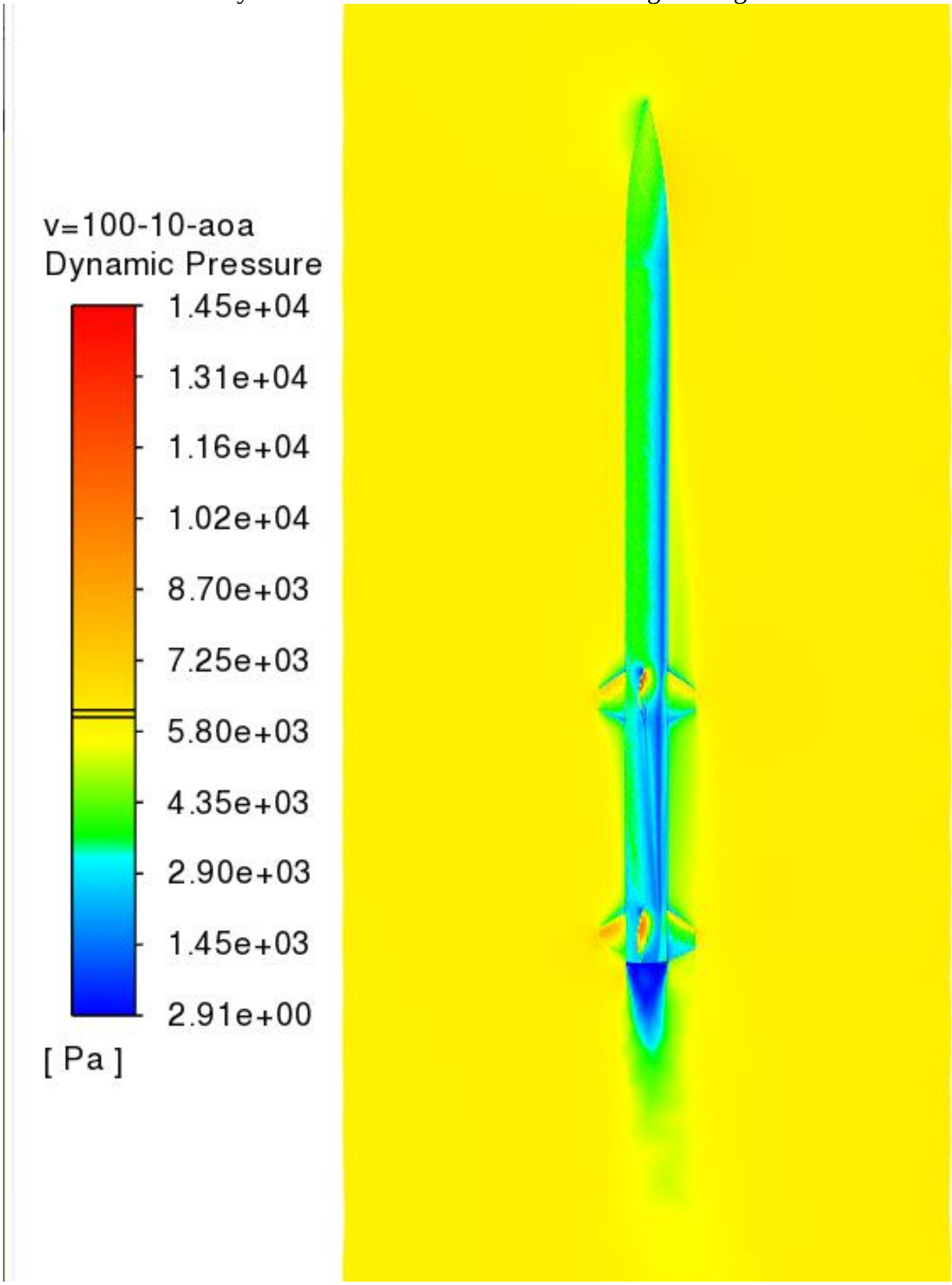
Contours of Dynamic Pressure at 100 m/s and 0 degree Angle of Attack



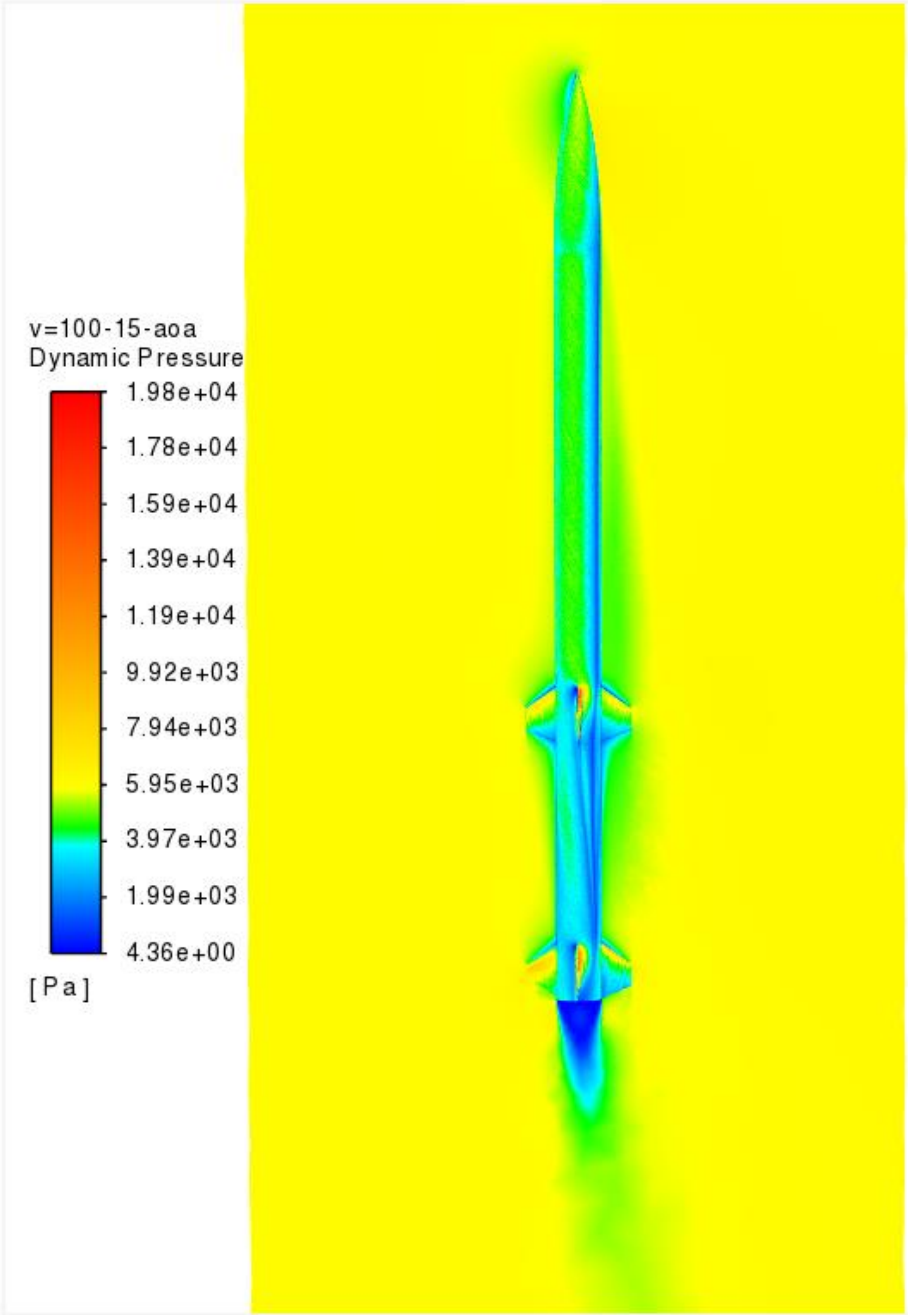
Contours of Dynamic Pressure at 100 m/s and 5 degree Angle of Attack



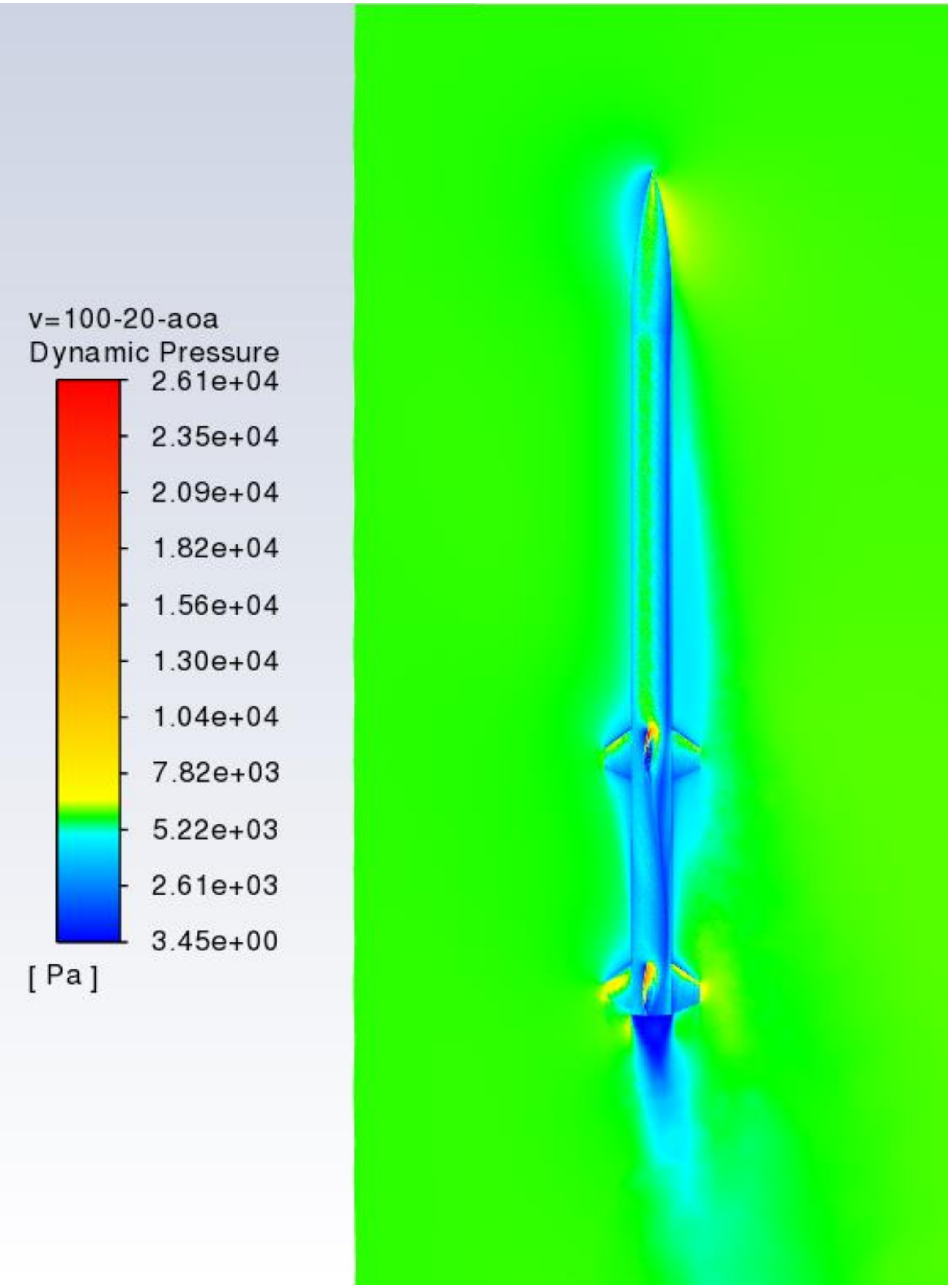
Contours of Dynamic Pressure at 100 m/s and 10 degree Angle of Attack



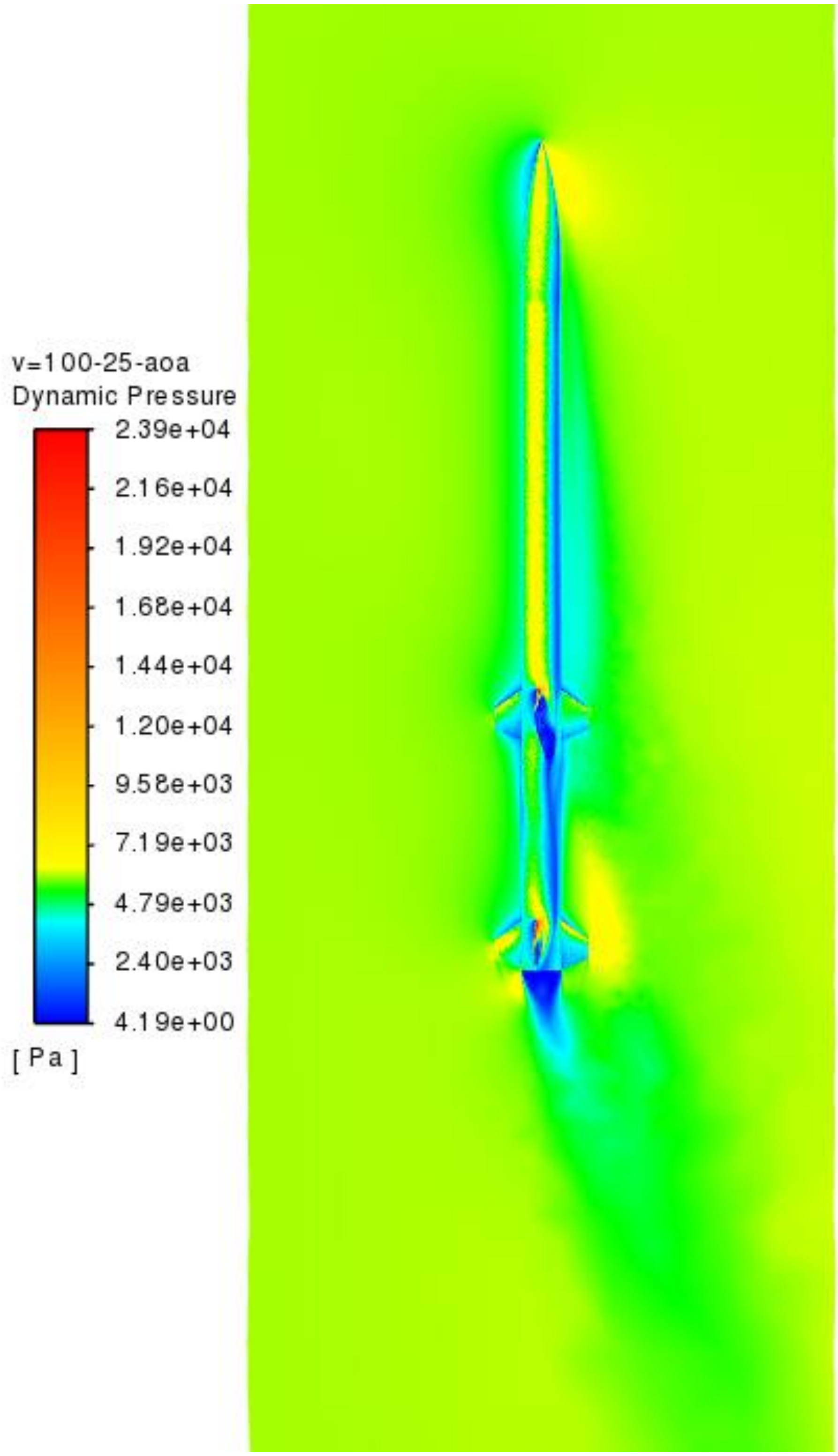
Contours of Dynamic Pressure at 100 m/s and 15 degree Angle of Attack



Contours of Dynamic Pressure at 100 m/s and 20 degree Angle of Attack

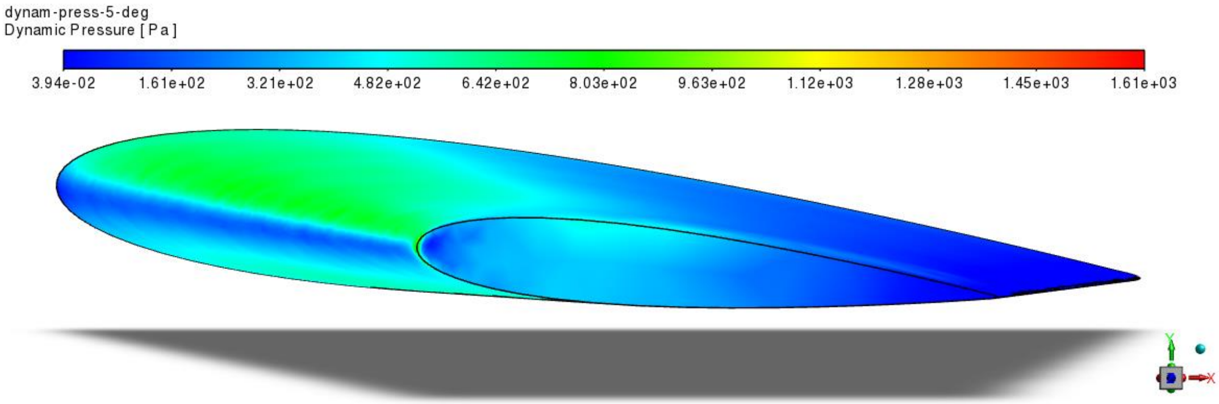


Contours of Dynamic Pressure at 100 m/s and 25 degree Angle of Attack

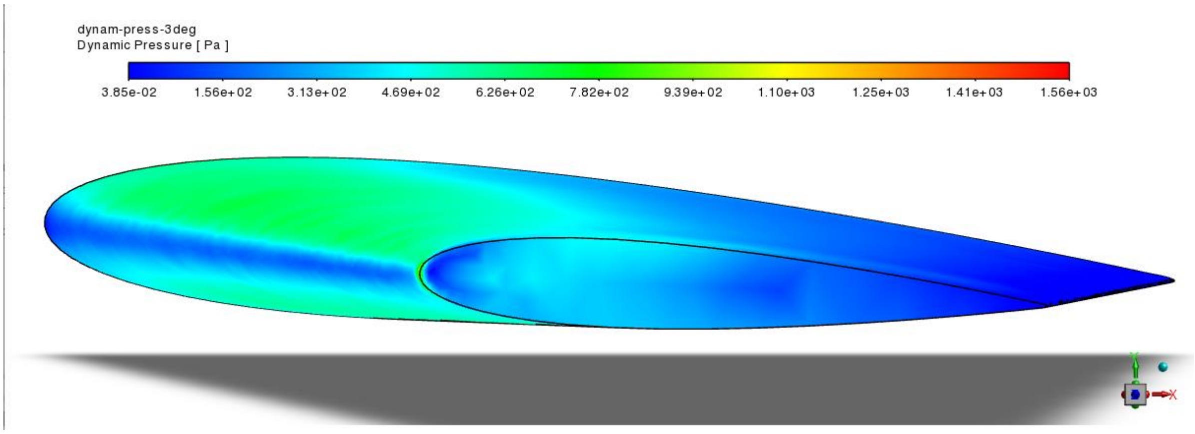


7.7 Appendix G Dynamic Pressure Contorts for Each Fin Case

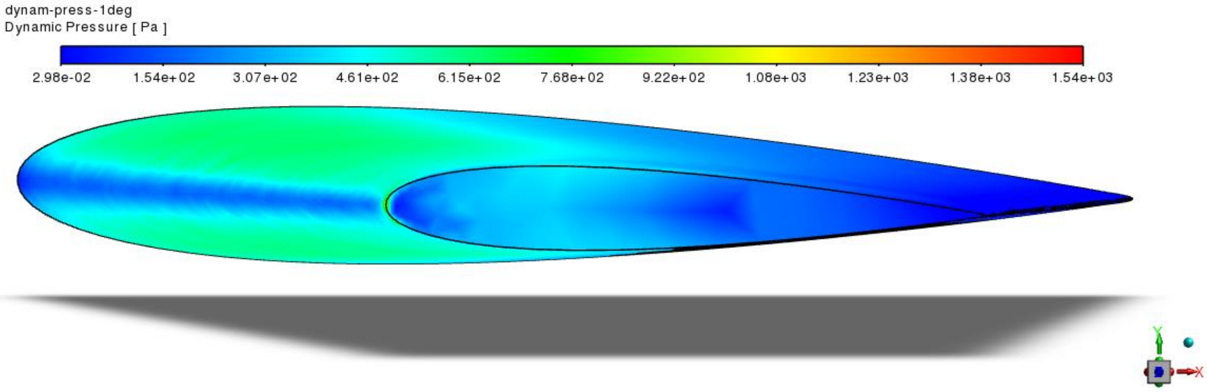
Contours of Dynamic Pressure at 35 m/s for the 0414 Fin at 5 degree Angle of Attack



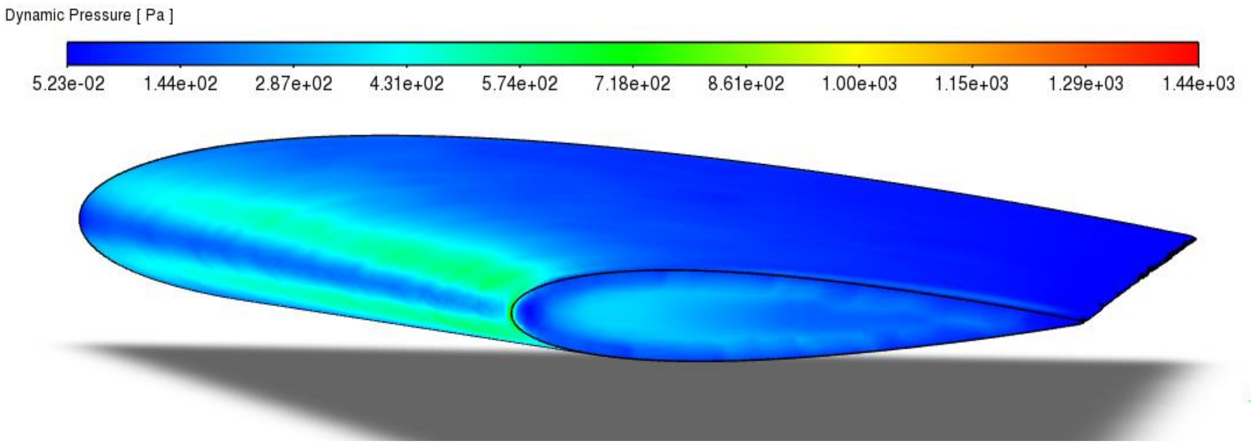
Contours of Dynamic Pressure at 35 m/s for the 0414 Fin at 3 degree Angle of Attack



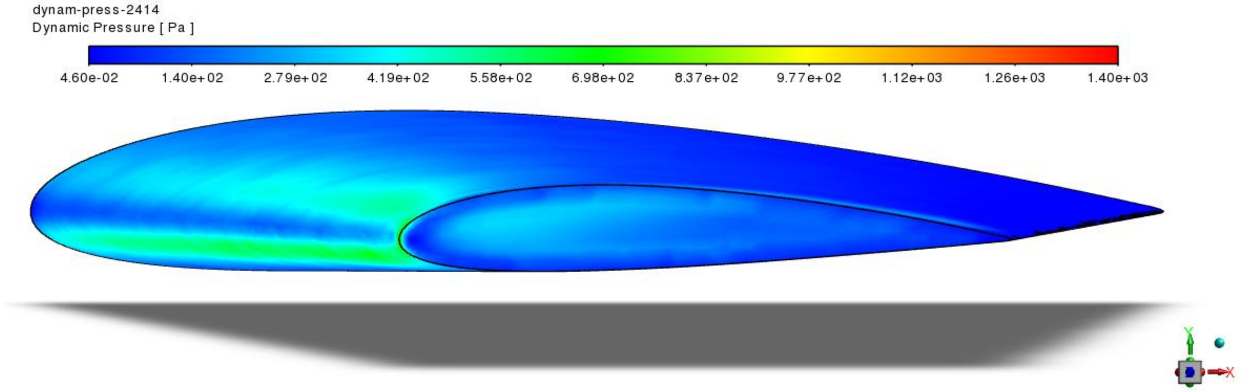
Contours of Dynamic Pressure at 35 m/s for the 0414 Fin at 1 degree Angle of Attack



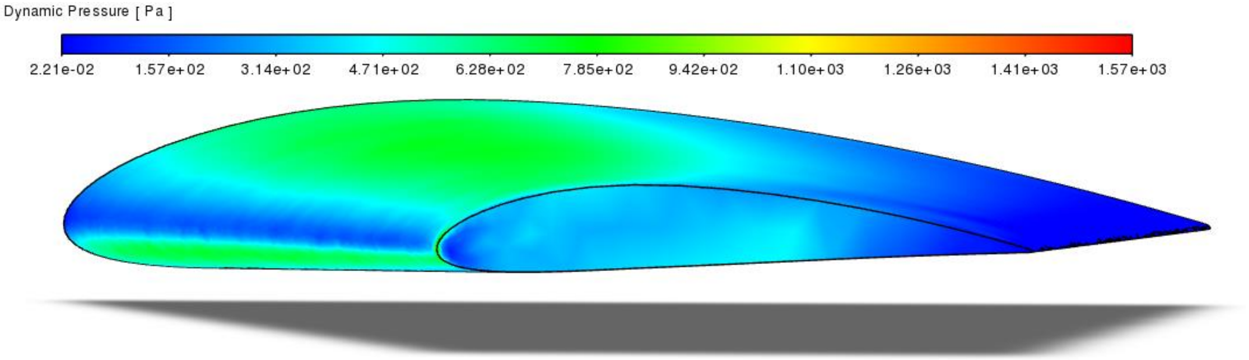
Contours of Dynamic Pressure at 35 m/s for the 0414 Fin at 0 degree Angle of Attack



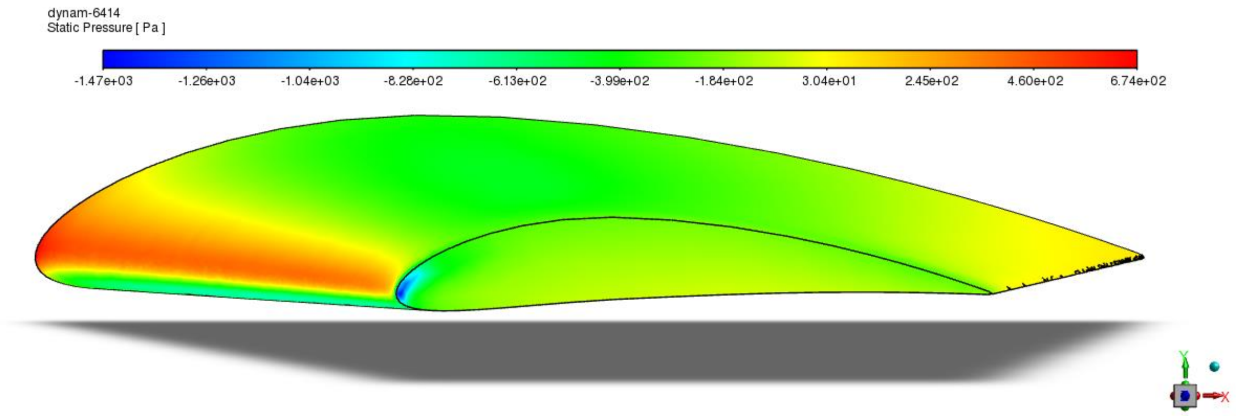
Contours of Dynamic Pressure at 35 m/s for the 2414 Fin



Contours of Dynamic Pressure at 35 m/s for the 4414 Fin

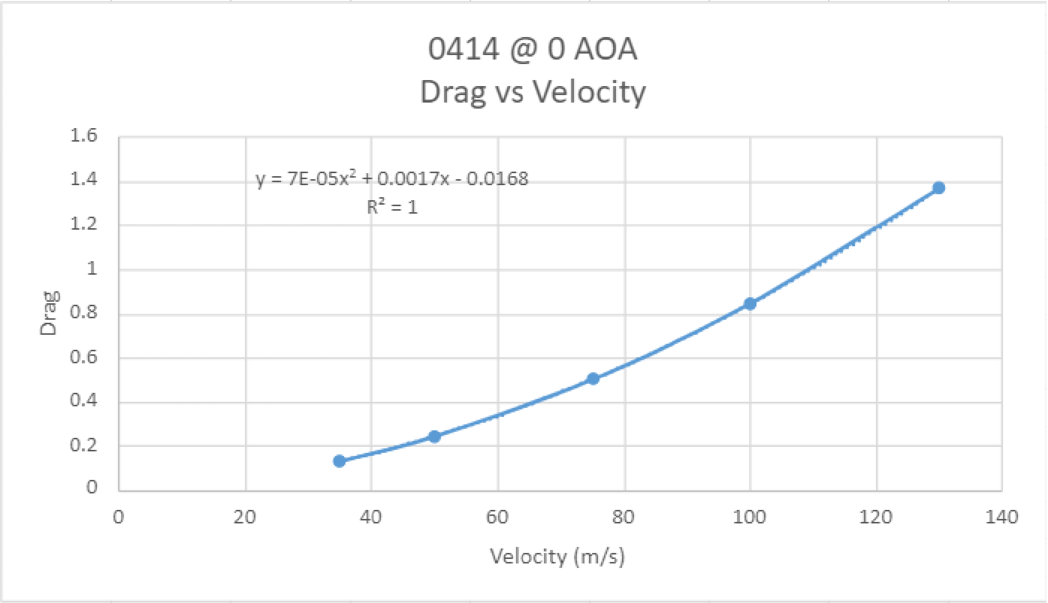


Contours of Dynamic Pressure at 35 m/s for the 6414 Fin

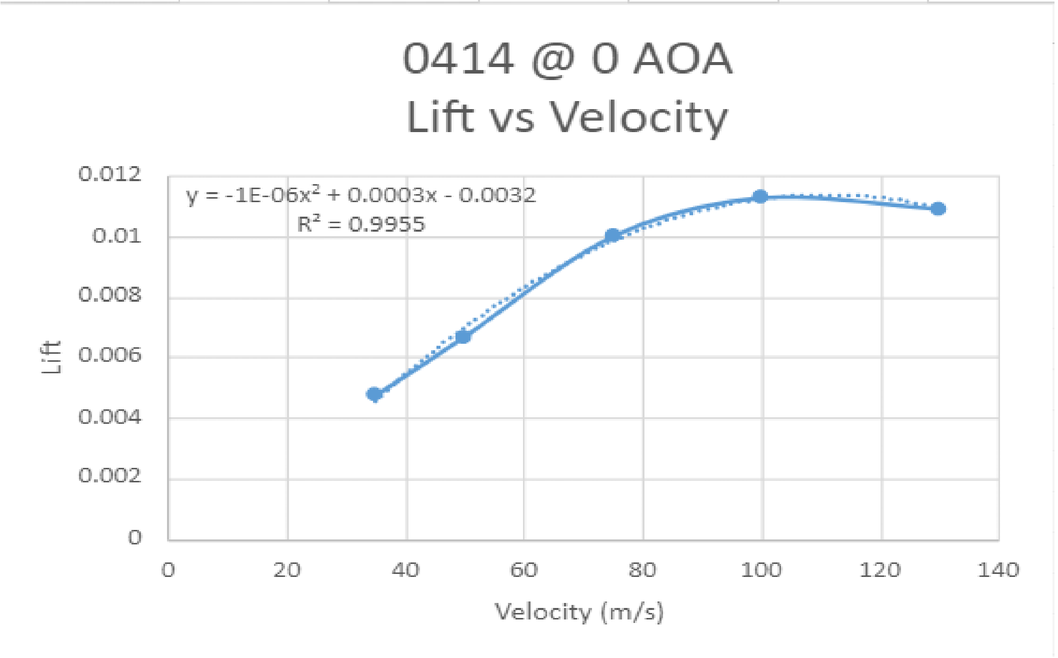


7.8 Appendix H Lift and Drag vs Velocity for Each Fin Case

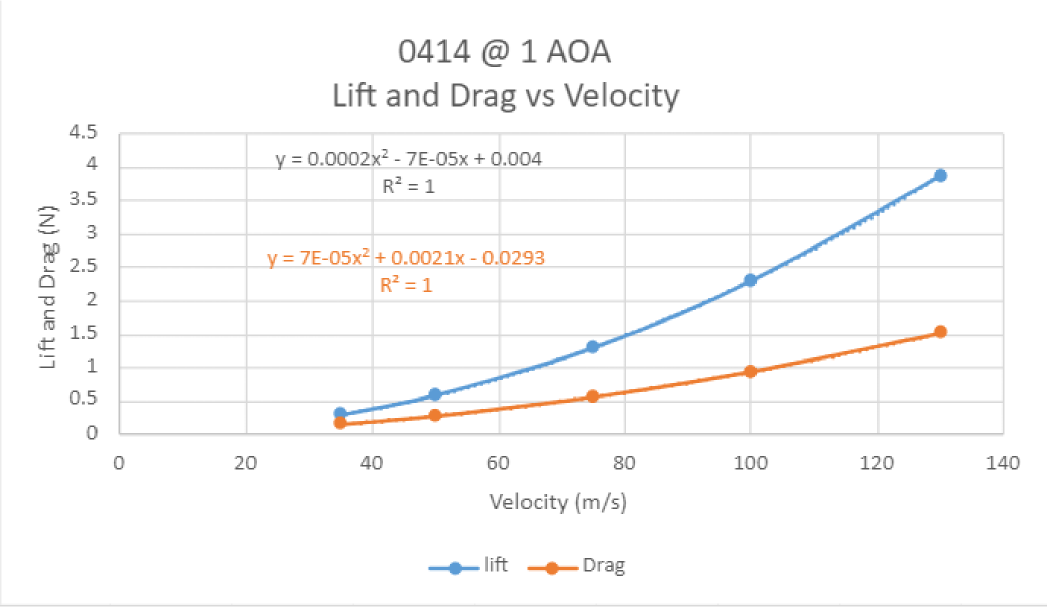
Drag vs Velocity for the 0414 Fin at 0 degree AOA



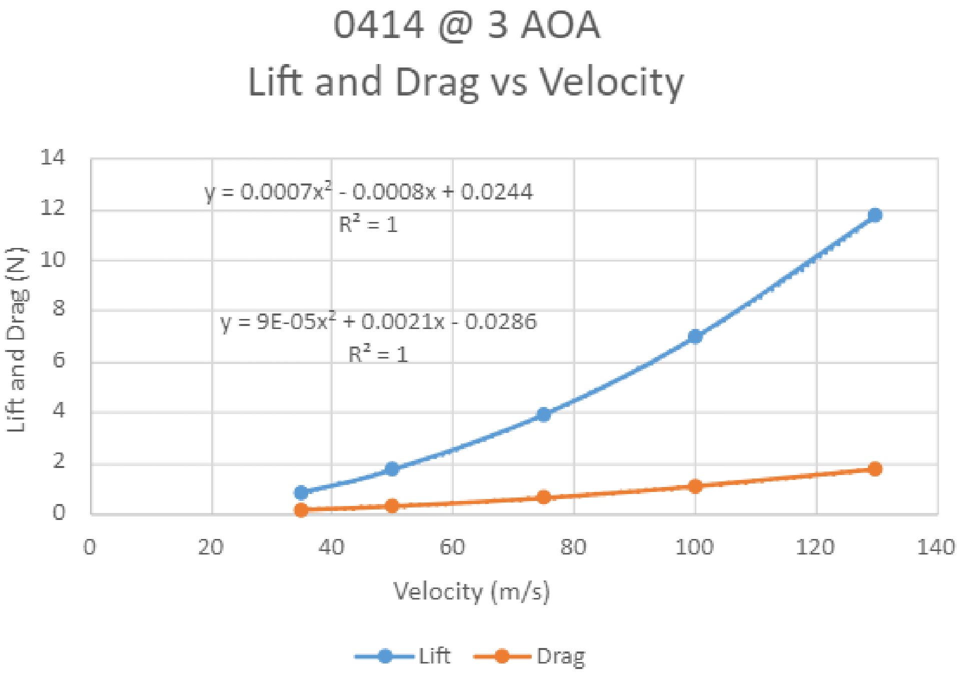
Lift vs Velocity for the 0414 Fin at 0 degree AOA



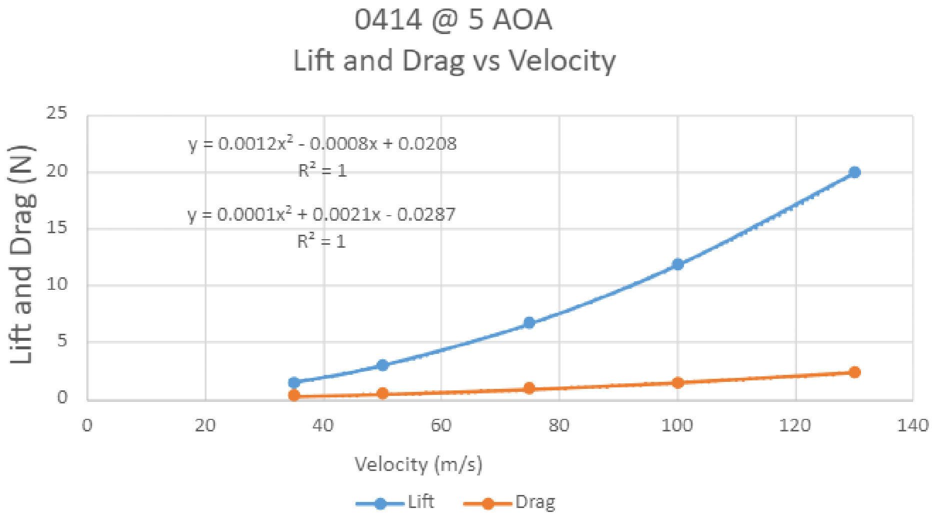
Lift and Drag vs Velocity for the 0414 Fin at 1 degree AOA



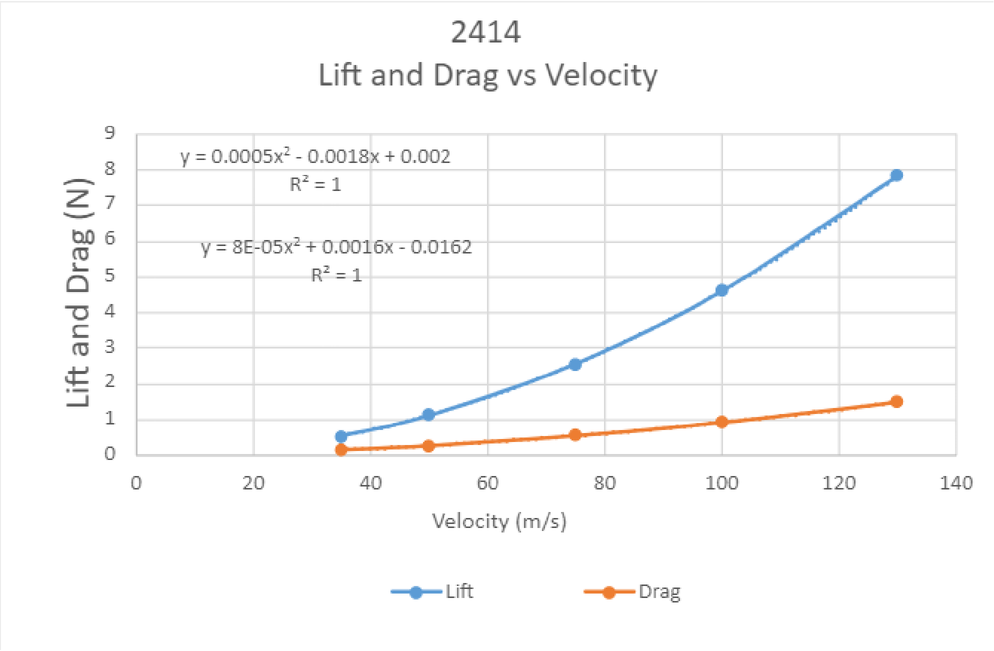
Lift and Drag vs Velocity for the 0414 Fin at 3 degree AOA



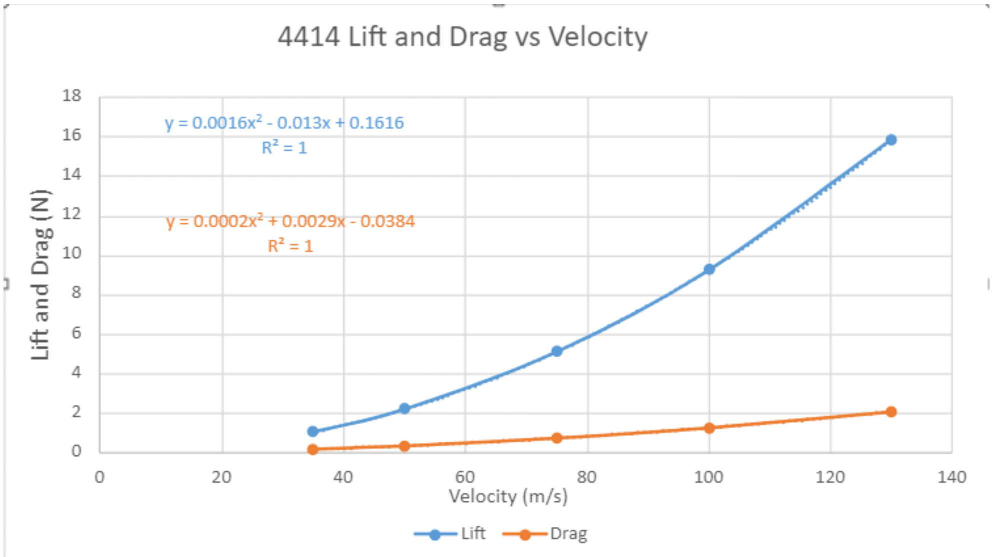
Lift and Drag vs Velocity for the 0414 Fin at 5 degree AOA



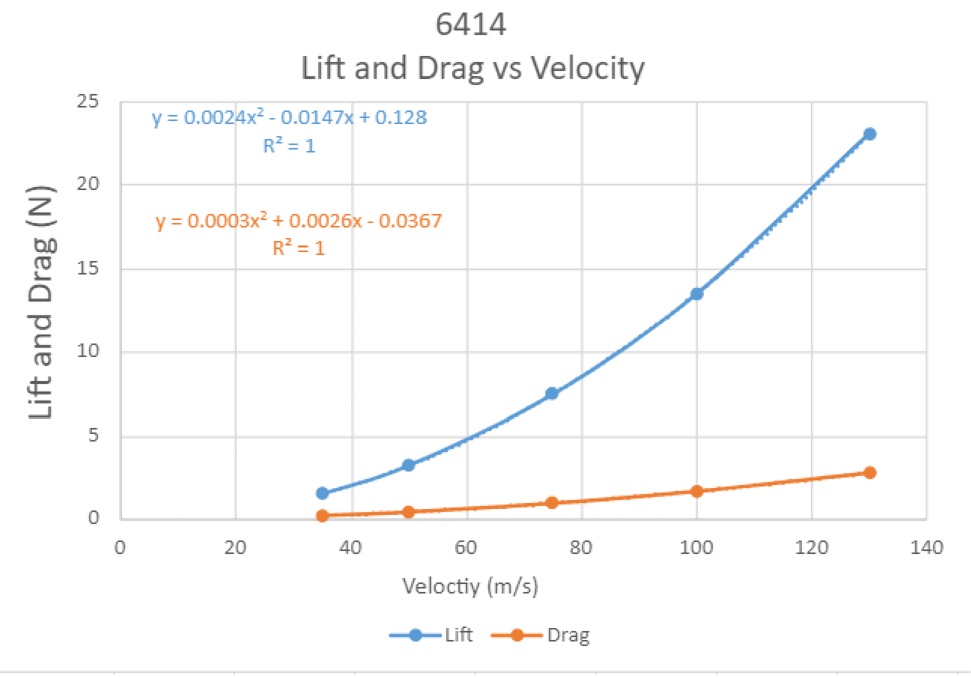
Lift and Drag vs Velocity for the 2414 Fin



Lift and Drag vs Velocity for the 4414 Fin

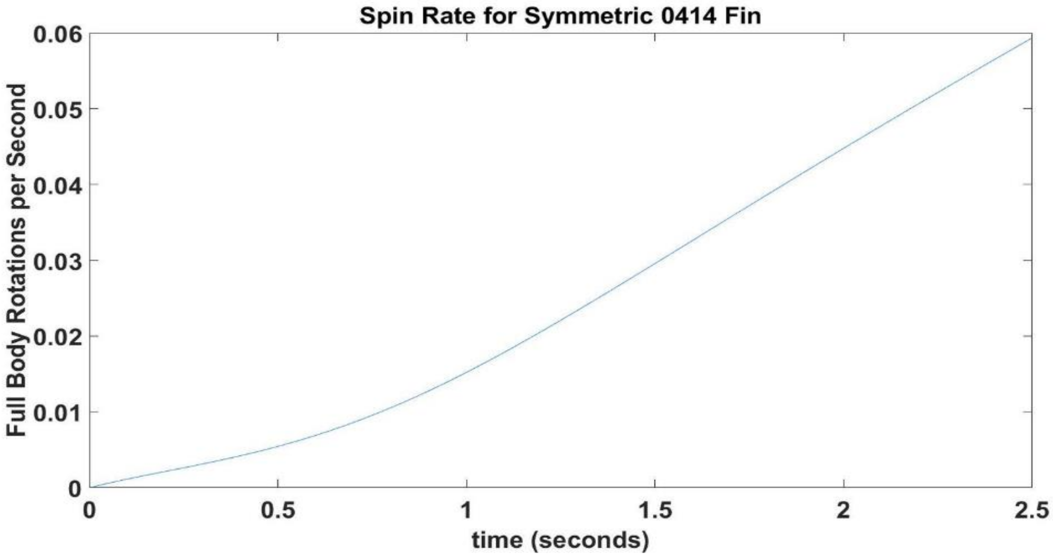


Lift and Drag vs Velocity for the 6414 Fin

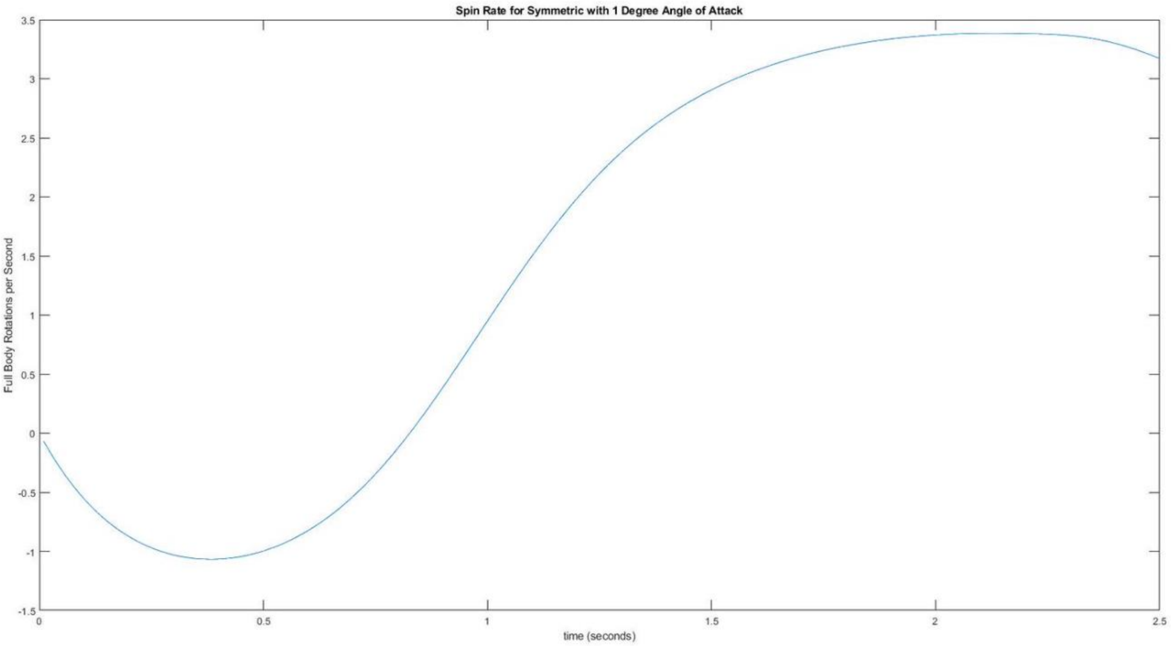


7.9 Appendix I Spin vs Time for Each Fin Case

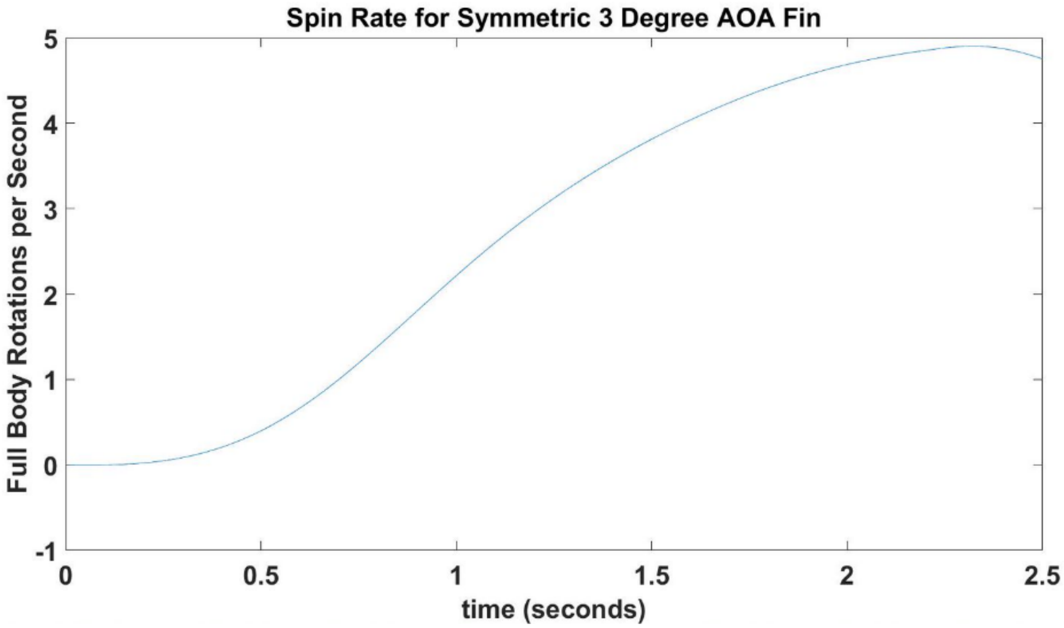
Rotations vs Time for the 0414 Fin at 0 degree AOA



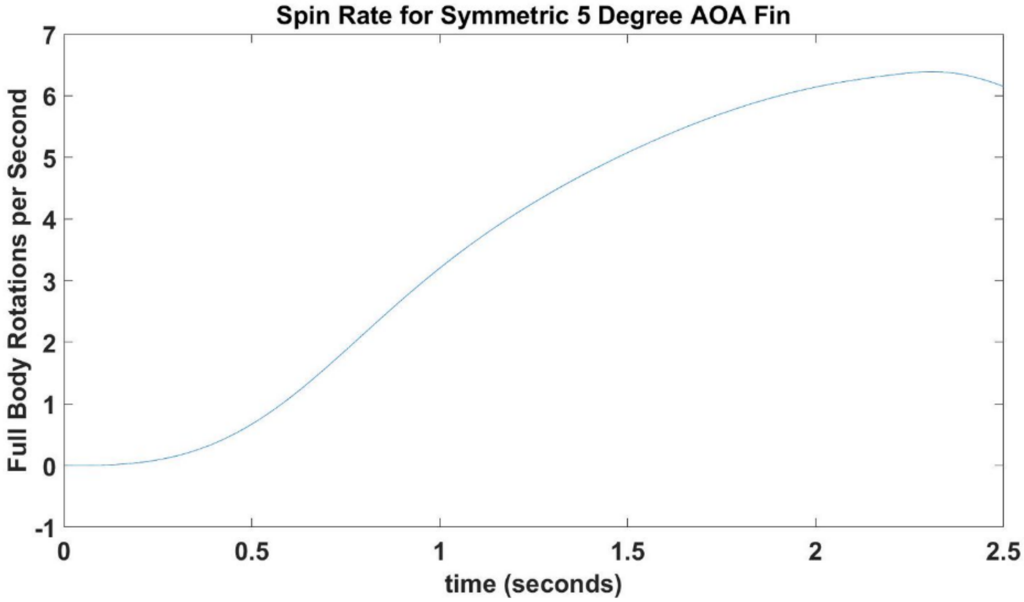
Rotations vs Time for the 0414 Fin at 1 degree AOA



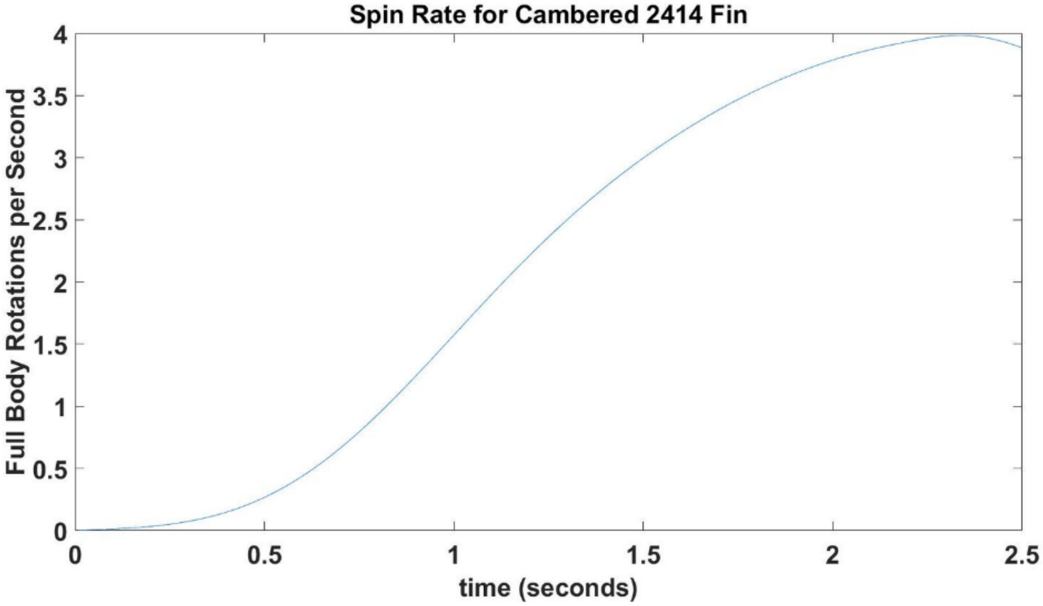
Rotations vs Time for the 0414 Fin at 3 degree AOA



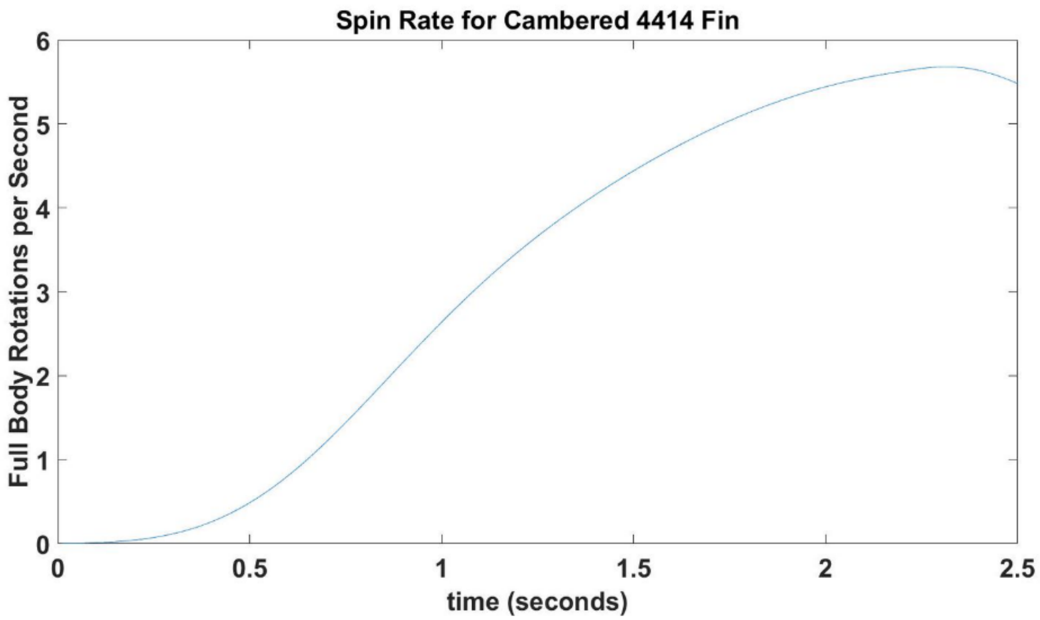
Rotations vs Time for the 0414 Fin at 5 degree AOA



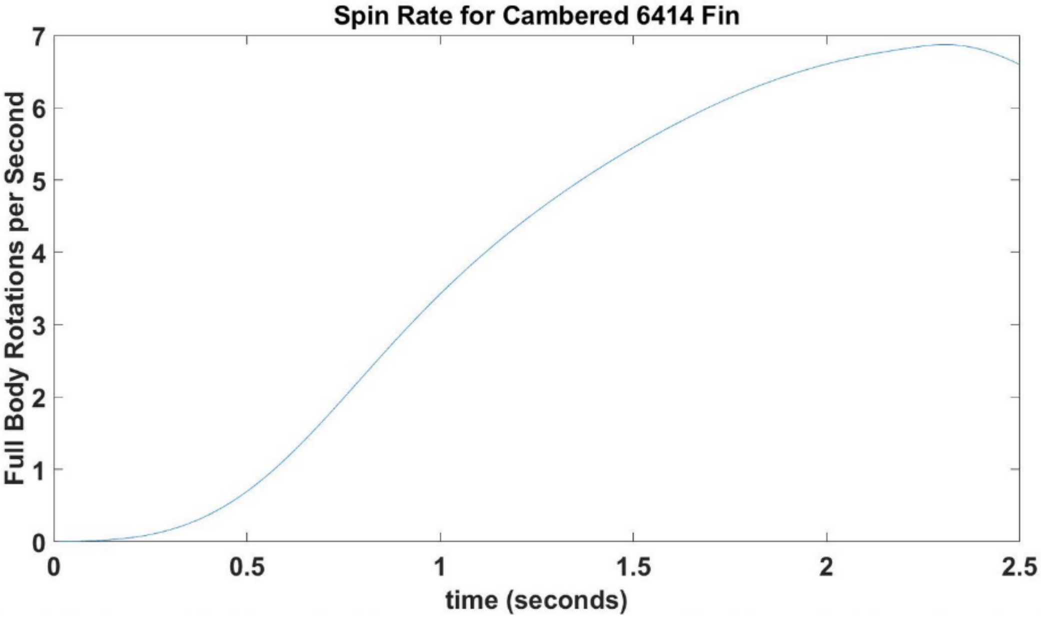
Rotations vs Time for the 2414 Fin



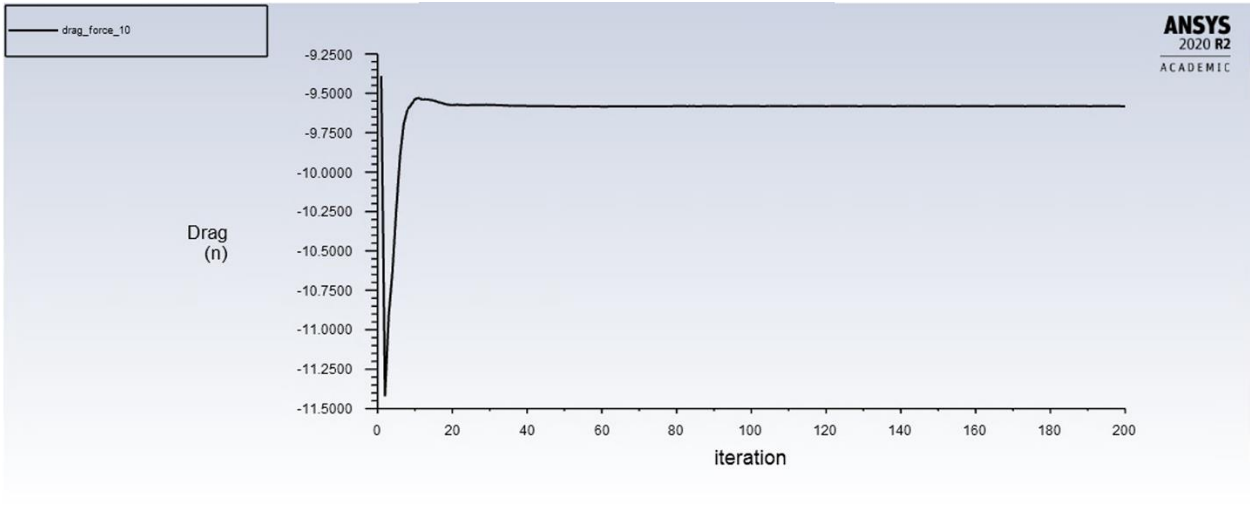
Rotations vs Time for the 4414 Fin



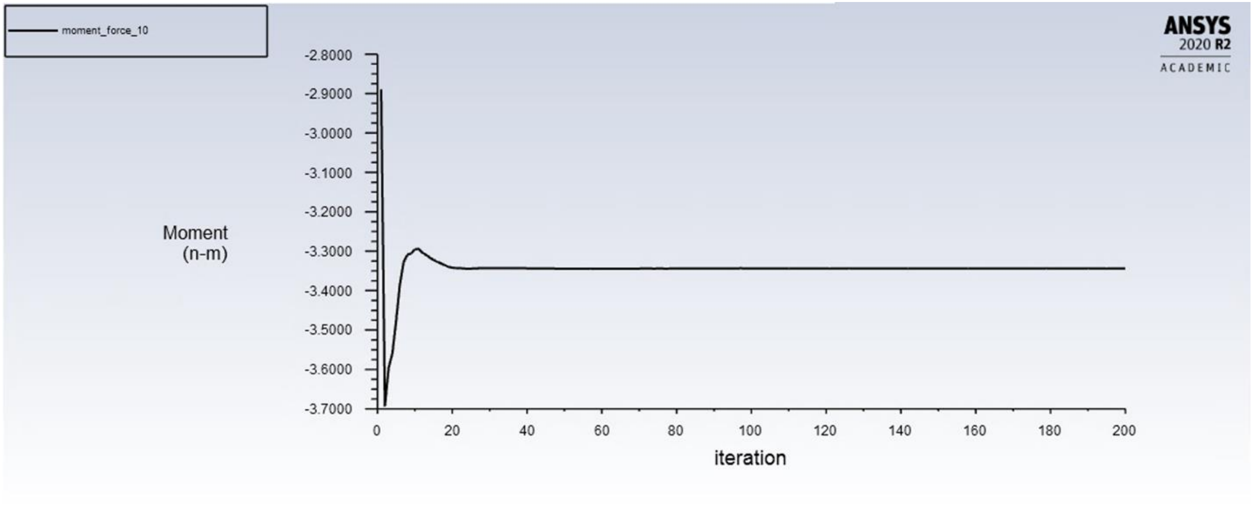
Rotations vs Time for the 6414 Fin



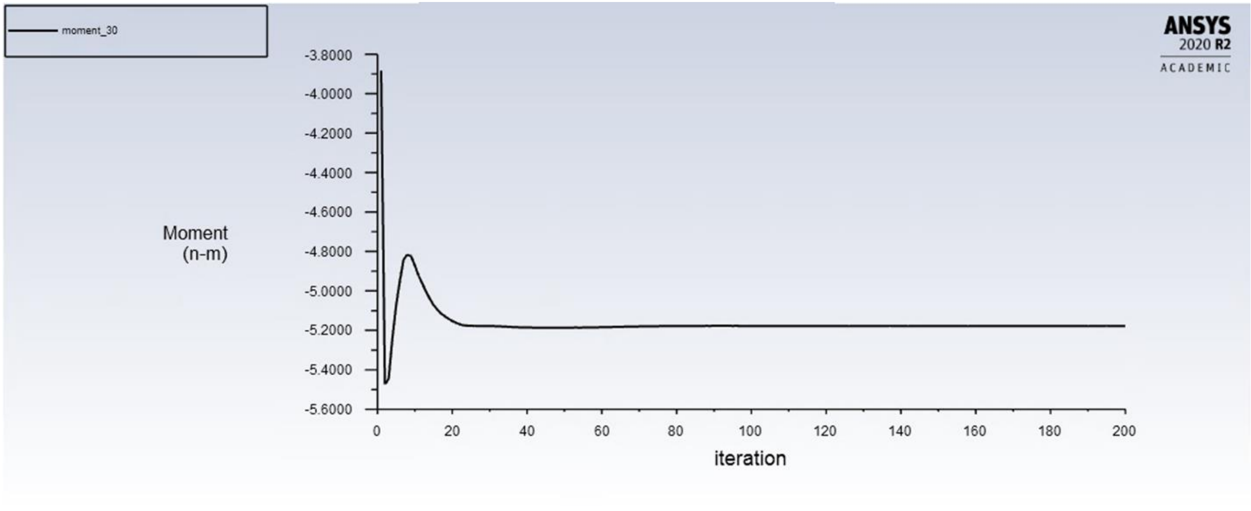
7.10 Appendix J Air Brake Moment and Drag Force Fluent Graphs



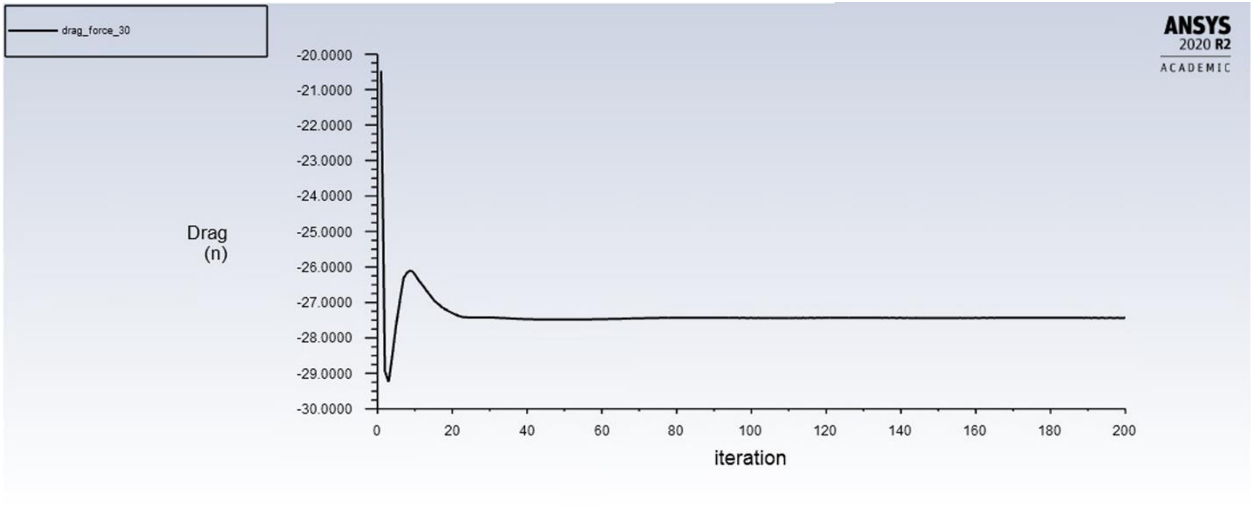
10° Air Brake Variant Drag Force Graph



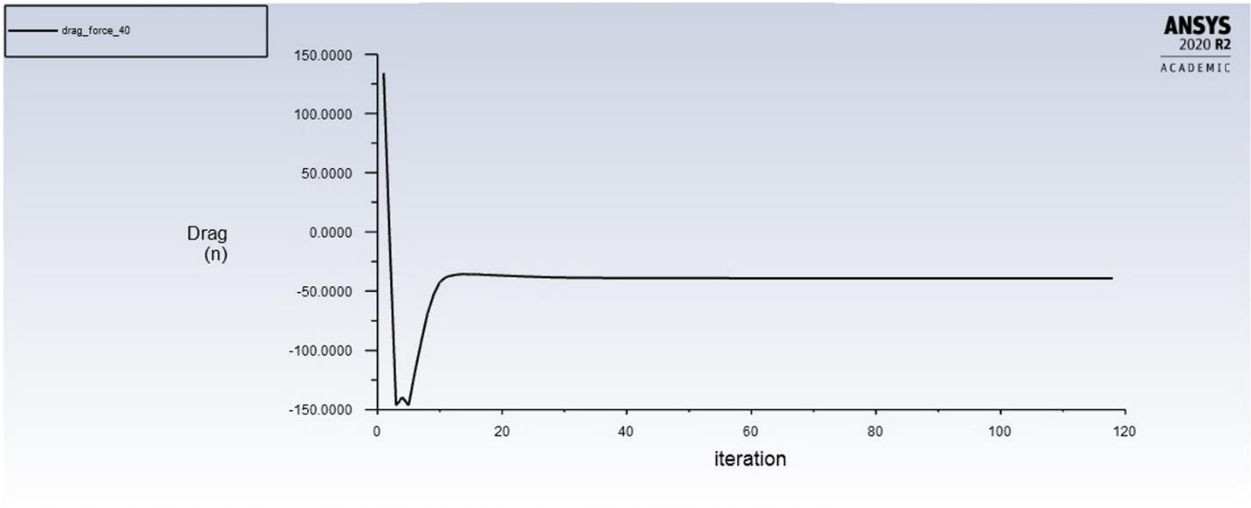
10° Air Brake Variant Moment Graph



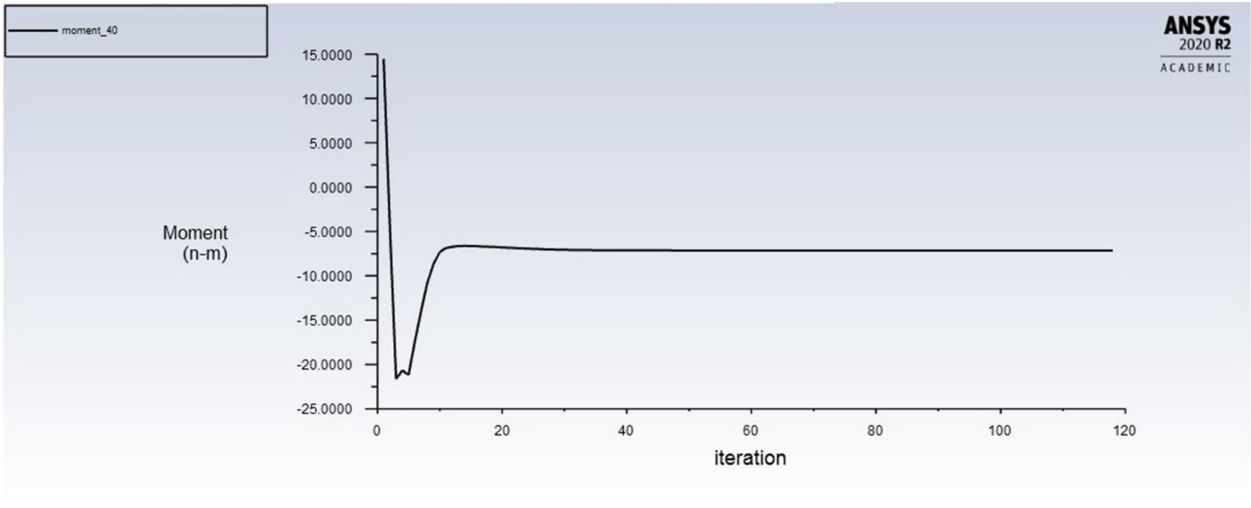
30° Air Brake Variant Moment Graph



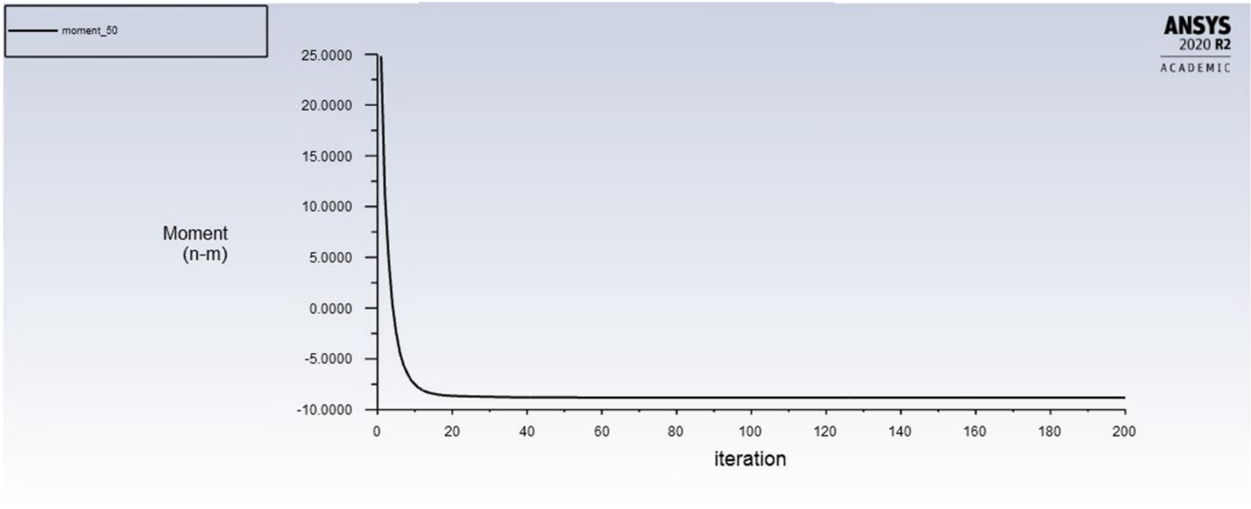
30° Air Brake Variant Drag Force Graph



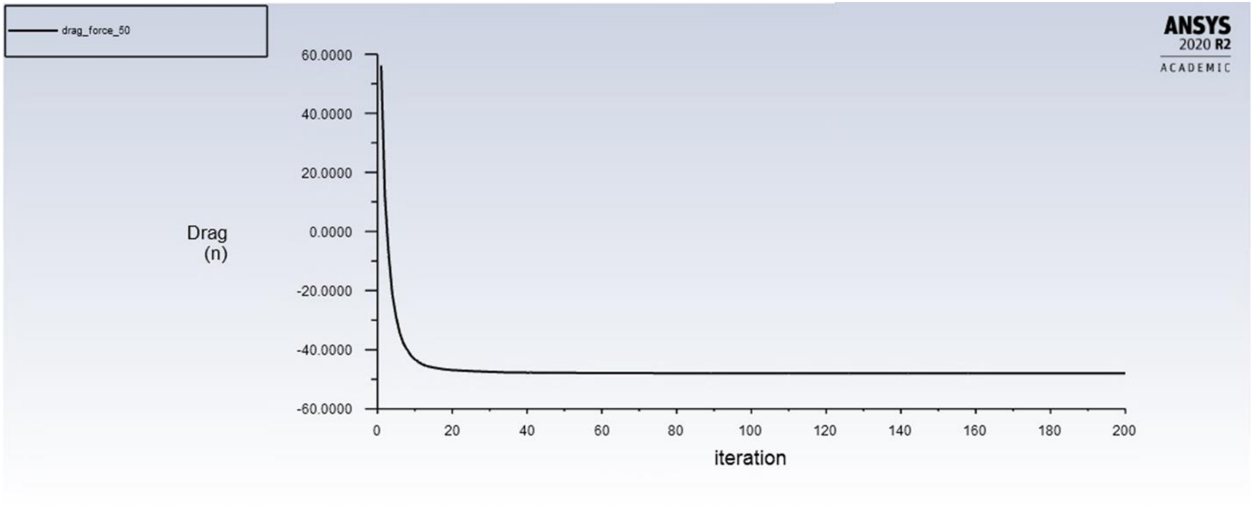
40° Air Brake Variant Drag Force Graph



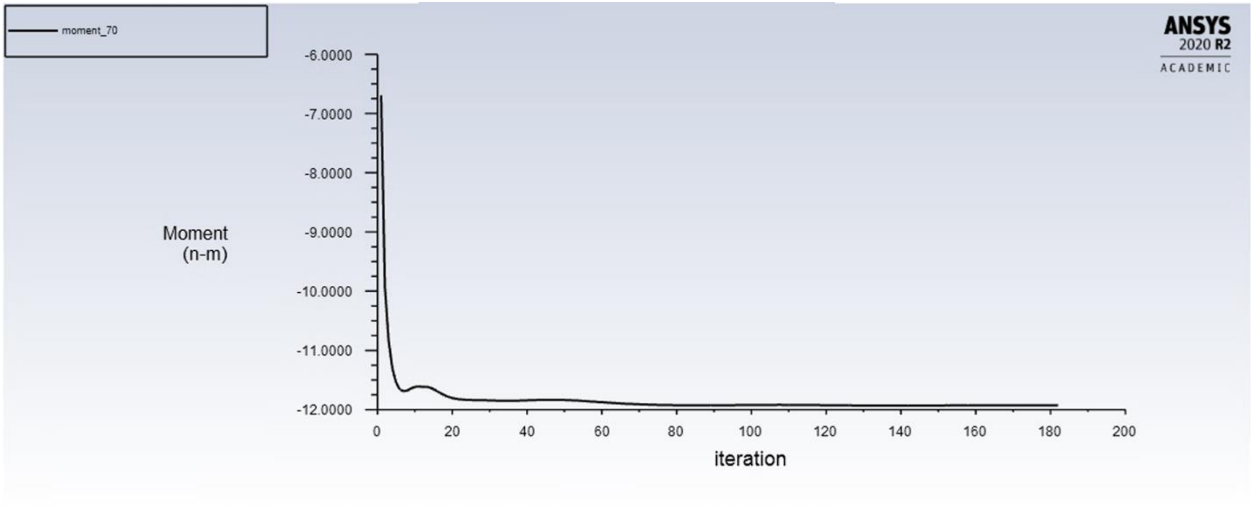
40° Air Brake Variant Moment Graph



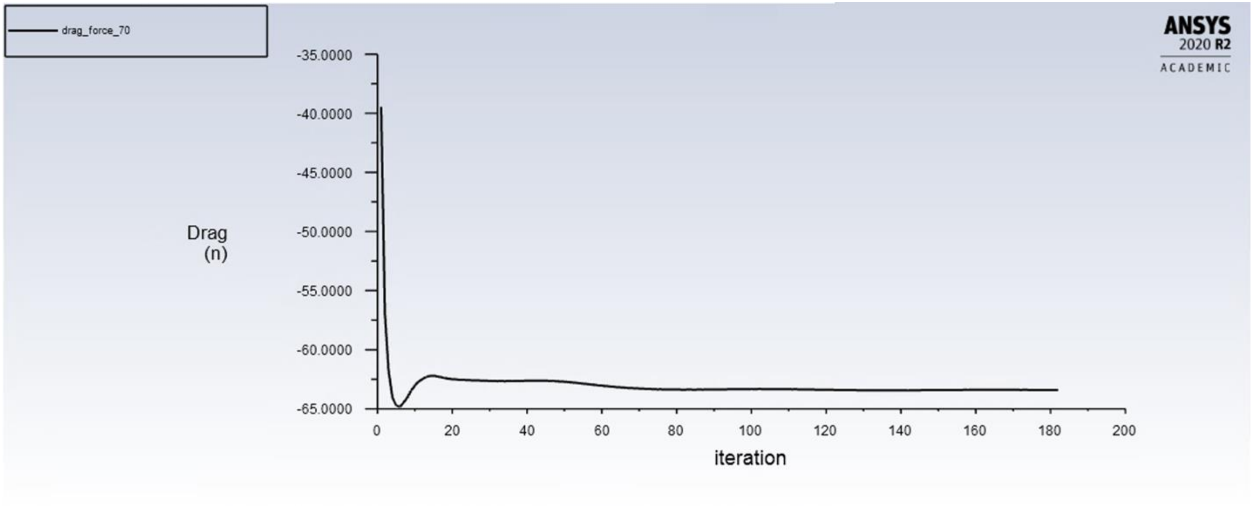
50° Air Brake Variant Moment Graph



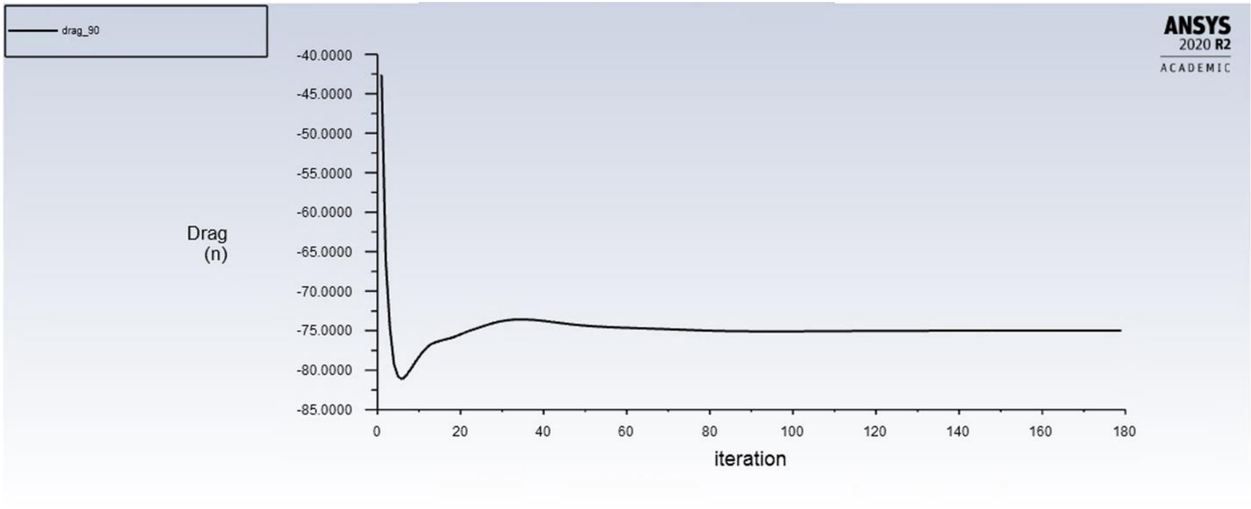
50° Air Brake Variant Drag Force Graph



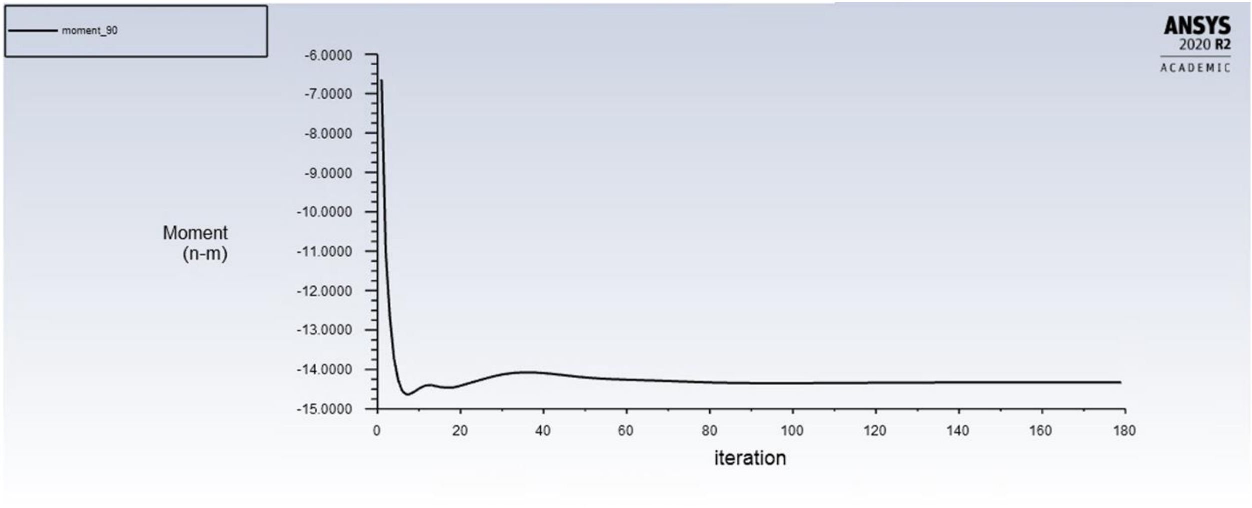
70° Air Brake Variant Moment Graph



70° Air Brake Variant Drag Force Graph



90° Air Brake Variant Drag Force Graph



90° Air Brake Variant Moment Graph