

AIAA - Design Build Fly 2024-2025

By

Gregory Beaudoin
Hamza Khan
Matthew Matta
Bradley Moore
Kyle Pierson
Patrick Rohleder
Will Russo
Ryan Webster



SUBMITTED AS PART OF THE CAPSTONE PROJECT

DEPARTMENT OF MECHANICAL ENGINEERING

COLLEGE OF ENGINEERING

UNIVERSITY OF MASSACHUSETTS LOWELL

17-DECEMBER-2024

Signatures of Authors:

Gregory Beaudoin

Kyle Pierson

Hamza Khan

Patrick Rohleder

Matthew Matta

William Russo

Bradley Moore

Ryan Webster

Capstone Advisor: David Willis

Associate Professor, Associate Chair for Undergraduate Studies

Department of Mechanical and Industrial Engineering

1) Abstract

The American Institute for Aeronautics and Astronautics (AIAA) hosts an annual competition called Design/Build/Fly (DBF). Universities from many countries around the world participate by designing aircraft for the competition. At the start of the Fall semester in September, AIAA released the year's specific project theme as well as a list of design constraints and rules that every team was responsible for following. Teams spent the year designing, building, and flying a radio-controlled aircraft that performs the theme's missions during the competitions fly-off in April. The 2024-2025 iteration of the DBF competition was hosted by Raytheon and the fly-off will take place in Tucson, Arizona. The project for the 2024-2025 iteration was the X-1 supersonic flight program. This represents the controlled air launching of experimental aircraft by utilizing a carrier plane that has a higher flight ceiling to carry up the experimental aircraft before it can efficiently take off. The first mission was the delivery flight. This modelled the carrier plane flying on its own without the X-1 test vehicle or bottle fuel tanks. The second mission was the captive carry flight where the X-1 test vehicle, representing the experimental aircraft, was attached beneath the carrier plane along with the bottles beneath the wing, representing the fuel tanks. The third mission was the launch flight where the carrier plane released the X-1 test vehicle mid-flight where it autonomously flies down and lands within a target zone.

The design process started with the team examining historical data from past DBF teams which set some expectations for possible design choices such as the speed and weight of the aircraft. The capstone team was split up into four smaller, sub teams: Aerodynamics, Electronics & Propulsion, Structures, and Manufacturing. Each sub team was responsible for completing their respective elements of the plane while frequently collaborating to successfully reach a final design for the Fall 2024 semester.

2) Acknowledgments

The team would like to thank Professor David Willis for helping guide the team through design and fabrication of the aircraft. The team would also like to thank the Lawrence Lin Makerspace staff for providing resources and space for the team to build and test the aircraft components.

3) Table of Contents

1. Abstract	2
2. Acknowledgements	3
3. Table of Contents	3
3.1 List of Figures	4
3.2 List of Tables	5
4. Introduction	6
4.1 Project Motivation	6
4.2 Background Information	6
4.3 Literature Review	7
5. Project Logistics	8
5.1 Team Organization	8
5.2 Project Schedule	9
5.3 Safety	10
5.4 Interdisciplinary Collaboration	10
6. Design Methodology	11
6.1 Discussion of Procedures and Methods	11
6.2 Design Considerations and Constraints	12
6.3 Design Alternatives and Decision-Making Process	13
6.4 Design Iteration and Refinement	17
6.5 Application of Codes and Standards	20
6.6 Professional and Ethical Responsibilities	21
6.7 Analysis Methods	22
6.7.1 Aerodynamics	22
6.7.2 Electronics and Propulsion	25
6.7.3 Structures	26
6.8 Evaluation of Design through Testing Procedures	27
6.8.1 Electronics and Propulsion	27
6.8.2 Structures	29
7. Design Solution and Final Specification	30
7.1 CAD Drawings of Full Aircraft	30
7.2 Codes and Standards	30
7.3 Critical Calculations and Analyses	31
7.3.1 Aerodynamics	31
7.3.2 Electronics and Propulsion	34
7.3.3 Structures	35
7.4 Final Design Solution	38

8. Results.....	39
8.1 Aerodynamics.....	39
8.2 Electronics and Propulsion.....	48
8.3 Structures	50
9. Risk Assessment	54
10. Detailed Cost Analysis.....	56
11. Broader Impacts.....	60
12. Summary and Conclusion	61
13. Recommendation for Further Study.....	61
14. Bibliography	63
15. Appendices	64
Appendix A – Complete Drawings.....	64
Appendix B – Final Landing Gear Design Solutions.....	72
Appendix C – Gantt Chart	73
Appendix D – Programming Codes	74
Appendix E – Spreadsheets for Calculations, Data, and Project Tracking.....	112
Appendix F – ME Departmental Supply and Resource Request Forms	115
Appendix G– Honors-Add On.....	117

3.1 List of Figures

<i>Figure 5.1. Breakdown of the Primary Sub-Team Structure</i>	<i>9</i>
<i>Figure 5.2. DBF Sub-Team Organization Chart.....</i>	<i>9</i>
<i>Figure 5.3. Project Timeline.....</i>	<i>10</i>
<i>Figure 6.1. Design/Build/Fly Fall 2024 Design Flowchart</i>	<i>12</i>
<i>Figure 6.2. Design Iteration Examples for the Front Landing Gear</i>	<i>18</i>
<i>Figure 6.3. Design Iteration Examples for the Rear Landing Gear</i>	<i>18</i>
<i>Figure 6.4. Spar Connector Iteration 1</i>	<i>19</i>
<i>Figure 6.5. Spar Connector Iteration 2</i>	<i>19</i>
<i>Figure 6.6. Spar Connector Iteration 3</i>	<i>19</i>
<i>Figure 6.7. Spar Connector Iteration 4</i>	<i>20</i>
<i>Figure 6.8. Preliminary Aerodynamic Analysis: COL vs Velocity</i>	<i>23</i>
<i>Figure 6.9. Preliminary Aerodynamic Analysis: Drag vs Velocity</i>	<i>24</i>
<i>Figure 6.10. Preliminary Aerodynamic Analysis: Lift to Drag Ratio vs Velocity</i>	<i>24</i>
<i>Figure 6.11. Effective Power vs Velocity.....</i>	<i>25</i>
<i>Figure 6.12. Thrust Test Stand Setup.....</i>	<i>28</i>
<i>Figure 6.13. Servo Test Setup.....</i>	<i>29</i>
<i>Figure 6.14. Beam Bending Test Setup.....</i>	<i>29</i>
<i>Figure 7.1. CAD Drawing of Final Aircraft Prototype</i>	<i>30</i>
<i>Figure 7.2 C (L) vs (a) polar of Clark-Y airfoil configurations</i>	<i>33</i>
<i>Figure 7.3. Replication of Beam Bending Test Conditions in ANSYS.....</i>	<i>36</i>
<i>Figure 7.4. Final Design Build</i>	<i>38</i>
<i>Figure 8.1. 3D Aero Analysis of aircraft in XFLR5 (Streamline visualization at 8° AOA).....</i>	<i>40</i>
<i>Figure 8.2. C_(M) vs. α (Mission 1)</i>	<i>41</i>
<i>Figure 8.3. C_(M) vs. α (Mission 2 & 3)</i>	<i>41</i>
<i>Figure 8.4. Mission 1 Root Locus (Longitudinal Stability)</i>	<i>45</i>
<i>Figure 8.5. Mission 1 Root Locus (Lateral Stability)</i>	<i>45</i>

<i>Figure 8.6. Mission 2 & 3 Root Locus (Longitudinal Stability)</i>	46
<i>Figure 8.7. Mission 2 & 3 Root Locus (Lateral Stability)</i>	46
<i>Figure 8.8. Thrust Testing Results</i>	48
<i>Figure 8.9. Servo Testing Results</i>	49
<i>Figure 8.10. Ribbed Wing “Wing-Tip” Test Constraints</i>	52
<i>Figure 8.11. Wing-Tip Test Total Deformation</i>	52
<i>Figure 8.12. Wing-Tip Test Equivalent (Von-Mises) Stress</i>	52
<i>Figure 8.13. Front Landing Gear ANSYS Simulation Results</i>	53
<i>Figure 8.14. Rear Landing Gear ANSYS Simulation Results</i>	54
<i>Figure 10.1 Percent Cost for Capstone Group</i>	57
<i>Figure 10.2 Cost of Materials for Capstone Group</i>	58
<i>Figure 10.3 Percent Cost of Engineering Firm</i>	58
<i>Figure 10.4 Cost of Manufacturing for Engineering Firm</i>	59
<i>Figure 10.5 Cost of Labor for Engineering Firm</i>	59
<i>Figure 10.6 Cost of Materials for Engineering Firm</i>	60

3.2 List of Tables

<i>Table 6.1. Considerations and Constraints</i>	13
<i>Table 6.2. Decision Matrix for Wing Material</i>	13
<i>Table 6.3. Decision Matrix for Fuselage Material</i>	14
<i>Table 6.4. Decision Matrix for Wing Type</i>	14
<i>Table 6.5. Decision Matrix for Tail Type</i>	14
<i>Table 6.6. Decision Matrix for Landing Gear Type</i>	15
<i>Table 6.7. Decision Matrix for Fuel Container Shape</i>	15
<i>Table 6.8. Decision Matrix for Mission Priority</i>	15
<i>Table 6.9. Decision Matrix for Motor Type</i>	16
<i>Table 6.10. Preliminary Aircraft Design Dimensions for the Wing and Fuselage</i>	16
<i>Table 6.11. Preliminary Aircraft Design Dimensions for the Tail</i>	16
<i>Table 7.1. Codes and Standards</i>	31
<i>Table 7.2. Preliminary Stall and Takeoff Speed for Missions 1-3</i>	32
<i>Table 7.3. Estimated Load on Control Surfaces and Required Servo Torque</i>	35
<i>Table 8.1. Comparison of Mission 1 Takeoff and Stall Speed Results</i>	39
<i>Table 8.2. Comparison of Mission 2 & 3 Takeoff and Stall Speed Results</i>	40
<i>Table 8.3. Mission 1 Static Stability Analysis Results</i>	42
<i>Table 8.4. Mission 2 & 3 Static Stability Analysis Results</i>	42
<i>Table 8.5. Aircraft Inertia Tensor (Mission 1)</i>	43
<i>Table 8.6. Aircraft Inertia Tensor (Mission 2 & 3)</i>	44
<i>Table 8.7. Mission 1 Dynamic Stability (Longitudinal Modes/Eigenvalues)</i>	47
<i>Table 8.8. Mission 1 Dynamic Stability (Lateral Modes/Eigenvalues)</i>	47
<i>Table 8.9. Mission 2 & 3 Dynamic Stability (Longitudinal Modes/Eigenvalues)</i>	47
<i>Table 8.10. Mission 2 & 3 Dynamic Stability (Lateral Modes/Eigenvalues)</i>	47
<i>Table 8.11. Servo Distribution Chart</i>	50
<i>Table 8.12. Comparison of ANSYS Simulation and Beam Bending Testing for Wing Spar</i>	50
<i>Table 9.1. Risk Assessment Matrix</i>	56
<i>Table 11.1 Broader Impacts and Considerations</i>	60

4) Introduction

4.1 Project Motivation

This project represents the University of Massachusetts Lowell's design and construction of an aircraft for the American Institute of Aeronautics and Astronautics (AIAA) Design-Build-Fly 2024-2025 competition. The aircraft was designed to complete a ground mission followed by 3 flight missions. The team's goal was to create an aircraft that remained competitive for all 3 flight missions, with a focus on optimizing for mission 2 (M2). Largely through mechanical testing and simulation analysis, the first prototype was fabricated to meet the mission goals and requirements.

4.2 Background Information

The specific design consideration for the 2024-2025 iteration of the AIAA DBF competition was to simulate an X-1 Supersonic Flight Program. This, in the aviation industry, represents the controlled test of experimental aircraft. This is typically done by utilizing a larger aircraft with a higher flight ceiling that can carry the experimental aircraft up to its desired altitude before "air launching." This is necessary for these aircraft to perform within their flight envelope by taking off at the proper altitude and velocity for which they are best suited.

This test procedure is simulated during the third mission (M3) within the competition where an X-1 glider is released from underneath the aircraft and must glide down and land within a target zone. The carrier aircraft must also simulate the large fuel tanks commonly found under the wings of these real-life aircraft by utilizing at least 2, 16 fluid ounce bottles mounted externally under the wings.

The designed aircraft is a single motor, high wing, taildragger, with a conventional empennage. A single motor was chosen so that the loading from the motor would be located along the vertical spar, inside the fuselage. With the proper battery and motor combination, this configuration would be sufficient to produce enough thrust to carry the 15lb aircraft through the



missions. The high-wing design was chosen due to its reliability and ease of manufacturability. This choice allowed the wing to be a singular component with an uninterrupted spar, mounted onto the top of the fuselage which would provide the necessary strength required to hold the bottles under the wing. The taildragger landing gear configuration was chosen to allow space underneath the aircraft for the X-1 glider to be mounted while providing sufficient stability during takeoff and landing. A conventional tail was chosen, similar to the high wing design, for its ease of manufacturability and reliability.

4.3 Literature Review

The aircraft was designed and fabricated utilizing knowledge obtained from the following courses taken at the University of Massachusetts Lowell:

1. MECH.2010 – Computer Aided Design: SolidWorks was utilized to create a CAD model of the aircraft and its components.
2. MECH.2020 – Manufacturing Laboratory: Manufacturing techniques were learned in this course that influenced the specific techniques utilized for our aircraft.
3. MECH.2960 – Material Science for Engineers: Key takeaways from this course relate to our material selection for different components of the aircraft by looking at material properties and behaviors.
4. MECH.3020 – Instrumentation and Measurement Laboratory: Servo motors were tested with an oscilloscope utilizing knowledge and experience taken from this course.
5. MECH.3110 – Applied Strength of Materials: The loading conditions on the aircraft during flight, takeoff, and landing were analyzed using ANSYS, a software used commonly during this course.
6. MECH.3210 – Kinematics of Mechanisms: Servos and push rods for control surfaces (ailerons/flaps/elevator/rudder) were built using four-bar linkage knowledge from this course.
7. MECH.3610 – Mathematical Methods for Mechanical Engineers: This course introduces in greater depth, and almost exclusively utilizes MATLAB, a software that was utilized to perform certain aspects for the analysis of the aircraft.

8. MECH.3810 – Fluid Mechanics: Concepts such as lift, drag, boundary layers, and turbulent vs laminar flow was covered in this course which directly play into the aerodynamic analysis of the aircraft.
9. MECH.5830 – Advanced Aerodynamics: Concepts from Fluid Mechanics are further developed alongside new concepts like airfoil shapes, wing design, and stability.
10. MECH.4730 – Design Theory and Constraints: The design of experiments performed in this project were based on materials studied in the course taught by Professor Moysenko of ME department.
11. MECH.4510 – Dynamics Systems Analysis: Modeled different force patterns commonly seen in dynamic systems to predict system behavior.
12. ENGN.2180 – Introduction to Aerospace Lab: Knowledge regarding aerodynamic analysis was utilized.
13. IENG.4140 – Engineering Economics: The economics analysis used for materials selection for this capstone project was studied in the course taught by Professor Mingles.

5) Project Logistics

5.1 Team Organization

The Capstone team consisted of three primary sub-teams: Aerodynamics, Structures, and Electronics/Propulsion, as shown in Figure 5.1. Other sub-teams included the X-1 Glider Design and Manufacturing as seen in Figure 5.2. This structure was defined based on the various necessities for the aircraft, as well as to best utilize the strengths of the engineers on the Capstone team. A top-level team oversaw the general design of the aircraft in its entirety while serving their sub-team separately. All sub-teams were responsible for progress in their respective teams by allocating, facilitating, and managing tasks amongst all individuals within each team. Communications were held on a near-daily basis in common interactions as well as an online messaging and collaboration platform. In using these communications, as well as holding weekly meetings, it was ensured that all members of the team were able to express their feedback and other ideas. A breakdown of the team structure is shown below in Figures 5.1 and 5.2.

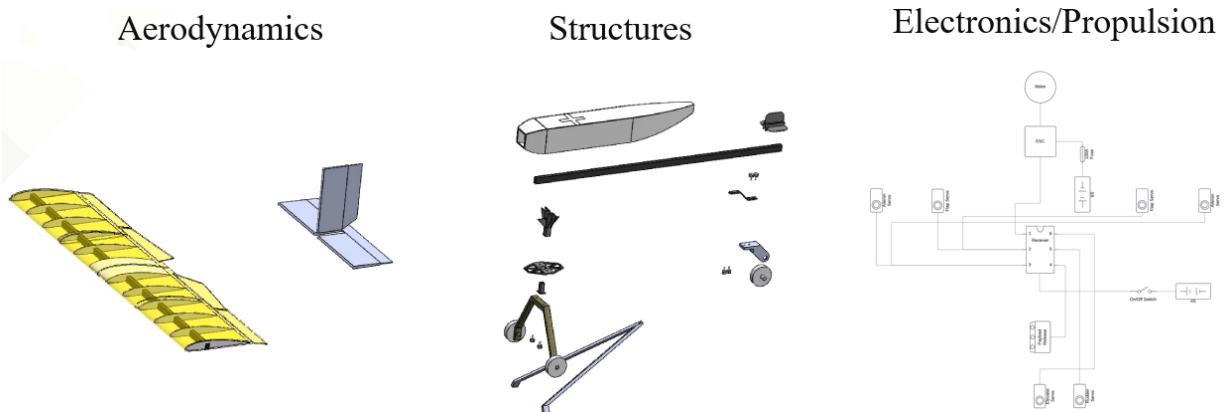


Figure 5.1. Breakdown of the primary sub-team structure

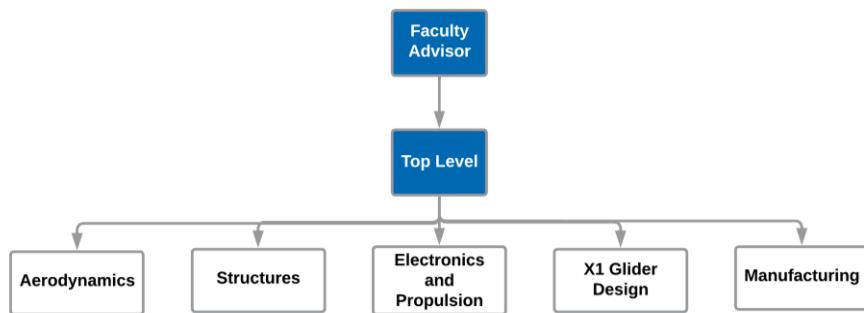


Figure 5.2. DBF Sub-Team Organization Chart

5.2 Project Schedule

The project timeline is displayed using the Gantt Chart in Figure 5.3. This timeline is planned to be completed over a two-semester period, which allowed the team to complete a multitude of tasks while still staying on track to meet all deadlines. Full team meetings were held on a weekly basis. Slideshows were prepared by the team for each meeting to organize progress and hold the team accountable. As previously mentioned, individual sub-teams met periodically to progress on tasks specific to their team.

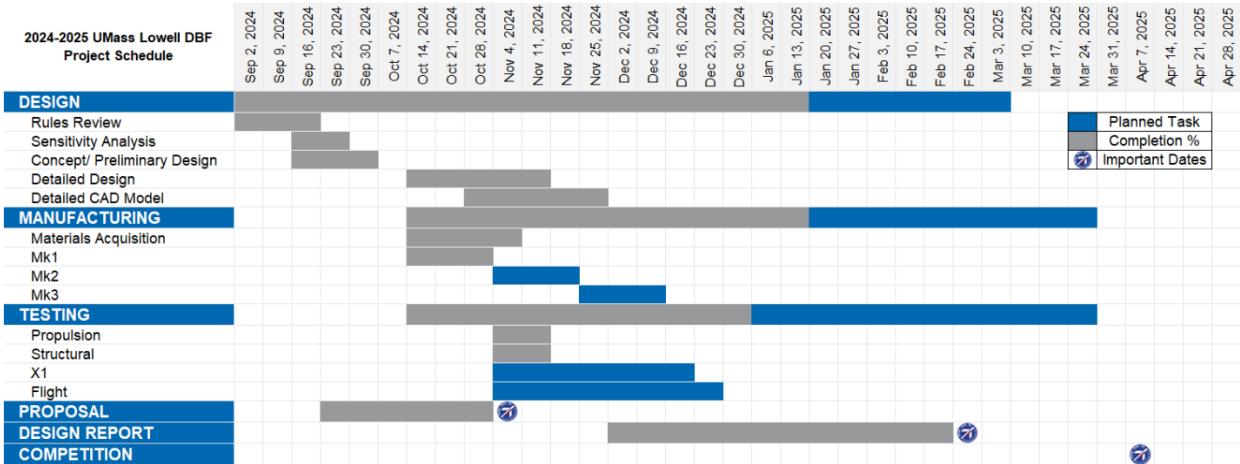


Figure 5.3. Project Timeline

5.3 Safety

Safety procedures pertaining to all manufacturing and testing of the aircraft prototype were paramount. While working on this project in the Lawrence Lin Makerspace, it was crucial that any engineer who worked on manufacturing and testing followed the guidelines laid out by the General Safety Training. In doing so, proper PPE was utilized, and no tool was to be used without knowledge of using said tool, especially including all laser cutters and 3D Printers. Safety of all users outside of the Capstone Team, including the general public, was also accommodated by practicing safe testing procedures when in public areas.

5.4 Interdisciplinary Collaboration

The aircraft design was performed solely by Mechanical Engineering students, and all electric components were tested, designed, and analyzed by the engineering students on the Electronics and Propulsion team. All electrical engineering tasks did not require assistance from any students or faculty in that discipline. Interdisciplinary collaboration was utilized while performing all design, manufacturing, assembly, and testing of the aircraft of the Lawrence Lin Makerspace. Any tools, equipment, and space used in this area required collaboration with the Lawrence Lin Makerspace staff. Contact with the UMass Lowell Student Government Association (SGA) was

also necessary while preparing the team's Bill of Materials and ordering parts to complete assembly of the aircraft.

6 Design Methodology

6.1 Discussion of Procedures and Methods

Approaching the design process, which is visualized in the flowchart in Figure 6.1, was laid with the understanding that the team needed to design an RC Aircraft that met mission requirements as described by the AIAA. Opportunities for design components began with background research of well-performing teams at the AIAA Design-Build-Fly Competitions. Recording historical data such as wingspan, overall aircraft weight, and cruise velocity allowed the team to improve upon and incorporate these key aspects into the team's design. As the team performed this research, design considerations and constraints were developed as key requirements for the aircraft, some of which were laid out by AIAA, which included minimizing weight, and maximizing lift, for example, while also considering public health, safety, and welfare.

Decision matrices were also developed to evaluate the feasibility of which materials to use as well as the shape and configuration of various components on the aircraft. In utilizing these decision matrices, the team was put in a position to make rational decisions for these various components which included cost, ease of manufacturing, durability, and weight. Team discussions regarding these matrices also allowed the team to utilize an iterative design process and decide between alternative designs. Upon the completion of these decision matrices, the use of a competition scoring sensitivity analysis using MATLAB allowed the team to develop a high-quality solution while still meeting design requirements. Each sub-team then proceeded to perform analyses utilizing structural, aerodynamics, and electronics simulations through various software such as ANSYS, xflr5, and e-Calc to support the selection of the various components as well as make design changes as problems arose. As these analyses were performed, several assumptions were made, which included the maximum forces applied, types of forces acting on the structure, and implementing isotropic versus anisotropic material properties as it pertained to

refining the design while also considering safety factors. Additional assumptions included the approximation of point masses in xflr5 simulations, and the torque outputs of the motors used.

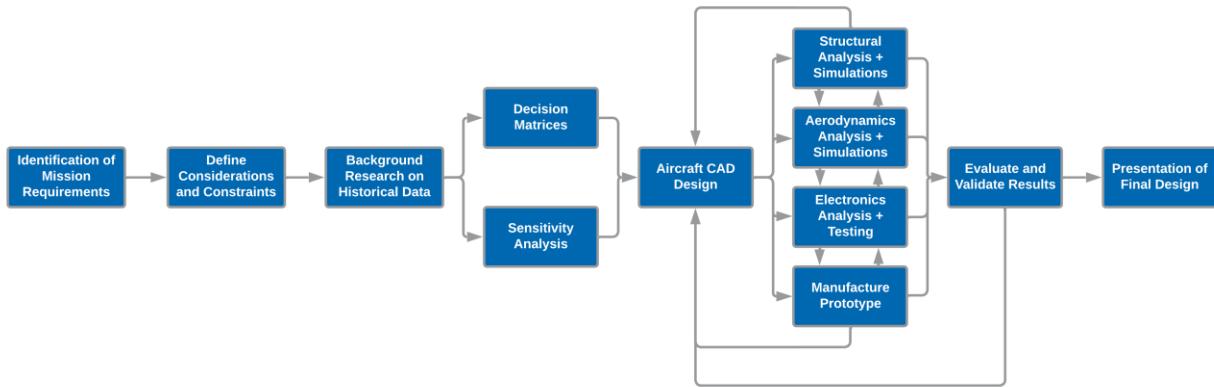


Figure 6.1. Design/Build/Fly Fall 2024 Design Flowchart

6.2 Design Considerations and Constraints

Amongst the Design Considerations and Constraints shown in Table 6.1, it was important that the team considered public health, safety, and welfare while also considering societal factors. Safety is certainly a consideration of the highest order, which is why the team is only choosing to obtain commercially produced propellers in the aircraft to prevent injury from those who assemble the aircraft as well as judges, spectators, and others involved in the Design-Build-Fly competition. The team considered the use of reusable and recyclable bottles for the fuel tanks to promote environmental sustainability. Finally, by considering different batteries in the listed materials in Table 6.1, the team also ensured that the environment was considered and that any battery being used would be disposed of properly.

Table 6.1. Considerations and Constraints

Consideration	Constraint(s)
Size	Maximum Aircraft Wingspan is 6 feet
Structure/Strength	Must withstand 2.5 g (0.005 lbs.) of load at each wing tip
Geometry	No less than 0.25 in. between any part of the fuselage, wings, outer surface, and X1 test vehicle wings.
Fuel Tank	Must be a commercially available beverage bottle with a minimum capacity of 16 fluid ounces
	Minimum two external fuel tanks involved for Missions 2 and 3
Safety	Fuel Tank Bottle must not be made of glass
	Commercially Produced Propellers and Blades
Weight	X1 Test Vehicle: maximum of 0.55 lbs.
	Take-off Gross Weight with Payload of aircraft must be less than 55 lbs.
Motor	Propeller Driven and Electric Powered
	Commercial Brushed or Brushless Electric
Battery	One pack connected to propulsion system
	NiCad/NiMH or Lithium only
	Total stored energy cannot exceed 100 Watt-hours
	Maximum current rating is 100 amps

6.3 Design Alternatives and Decision-Making Process

The design that was chosen for this year's AIAA Design, Build, Fly competition was developed in different stages. Decision matrices were used to develop preliminary characteristics of the aircraft such as material, shape, and type of different components. The decision matrices used during this project are provided in Tables 6.2 - 6.9.

Table 6.2. Decision Matrix for Wing Material

	Criteria 1	Criteria 2	Criteria 3	Criteria 4	Criteria 5	
DESCRIPTION	Cost	Ease of Manufacturability	Weight	Durability	Recoverability	
WEIGHT	10%	30%	30%	20%	10%	100%
OPTIONS	Criteria 1 SCORES	Criteria 2 SCORES	Criteria 3 SCORES	Criteria 4 SCORES	Criteria 5 SCORES	
Foam (Polystyrene, Styrofoam)	10.00	9.00	10.00	5.33	6.00	8.37
Plywood	6.00	8.00	4.00	5.00	2.67	5.47
Balsa Wood	3.00	9.00	8.00	5.00	2.33	6.63
CFRP (carbon fiber-reinforced polymer)	2.00	4.00	7.00	7.33	6.33	5.60
Acrylic	10.00	7.00	6.00	6.67	6.00	6.83
3D Print (PLA)	3.00	8.00	5.00	7.00	7.33	6.33
Carbon Fiber Fabric	2.00	5.00	9.00	8.33	10.00	7.07

Table 6.3. Decision Matrix for Fuselage Material

	Criteria 1	Criteria 2	Criteria 3	Criteria 4	Criteria 5	
DESCRIPTION	Cost	Ease of Manufacturability	Weight	Durability	Recoverability	
WEIGHT	10%	30%	35%	20%	5%	100%
OPTIONS	Criteria 1 SCORES	Criteria 2 SCORES	Criteria 3 SCORES	Criteria 4 SCORES	Criteria 5 SCORES	
6061-T6 Aluminum	2.00	4.00	4.00	8.67	6.67	4.87
CFRP	2.00	3.00	7.33	8.67	7.33	5.77
3D Printed	6.00	7.00	4.00	5.33	5.00	5.42
Plywood	7.00	6.00	4.50	4.67	5.00	5.26
Balsa Wood	8.00	7.00	7.33	3.33	2.67	6.27
Acrylic	9.00	7.67	7.33	5.33	5.00	7.08
Foam + Fiberglass	10.00	8.67	9.33	6.67	7.33	8.57

Table 6.4. Decision Matrix for Wing Type

	Criteria 1	Criteria 2	Criteria 3	Criteria 4	Criteria 5	
DESCRIPTION	Cost	Ease of Manufacturing	Weight	Durability	Control and Stability	
WEIGHT	10%	20%	20%	15%	35%	100%
OPTIONS	Criteria 1 SCORES	Criteria 2 SCORES	Criteria 3 SCORES	Criteria 4 SCORES	Criteria 5 SCORES	
High						0
Rectangular Conventional	8.67	10.00	8.00	8.33	10.00	9.22
Tapered Wing (Front)	6.00	7.33	8.33	7.33	8.33	7.75
Tapered Wing (rear)	5.67	7.00	8.33	7.33	8.00	7.53
Trapezoidal Wing	6.00	5.00	8.33	6.33	7.67	6.90
Elliptical Wing	5.33	5.00	7.33	7.67	8.67	7.18

Table 6.5. Decision Matrix for Tail Type

	Criteria 1	Criteria 2	Criteria 3	Criteria 4	Criteria 5	
DESCRIPTION	Cost	Ease of Manufacturing	Weight	Durability	Control and Stability	
WEIGHT	5%	40%	30%	10%	15%	100%
OPTIONS	Criteria 1 SCORES	Criteria 2 SCORES	Criteria 3 SCORES	Criteria 4 SCORES	Criteria 5 SCORES	
Conventional	7.00	9.00	8.00	7.00	9.00	8.40
T-Tail	6.00	7.00	7.00	5.00	8.00	6.90
Boom Tail	6.00	7.00	6.00	7.00	9.00	6.95
V-Tail	8.00	9.00	9.00	9.00	5.00	8.35

Table 6.6. Decision Matrix for Landing Gear Type

	Criteria 1	Criteria 2	Criteria 3	Criteria 4	
DESCRIPTION	Cost	Ease of Manufacturing	Weight	Durability	
WEIGHT	20%	30%	30%	20%	100%
OPTIONS	Criteria 1 SCORES	Criteria 2 SCORES	Criteria 3 SCORES	Criteria 4 SCORES	
Tricycle	8.33	7.33	8.33	7.00	7.77
Bow Tail Dragger	7.00	7.00	6.33	5.33	6.47
Quadricycle	5.00	5.67	4.33	5.00	5.00

Table 6.7. Decision Matrix for Fuel Container Shape

	Criteria 1	Criteria 2	Criteria 3	Criteria 4	Criteria 5	
DESCRIPTION	Fluid Capacity	Ease of Attachment to Pylons	Cost/Availability	Aerodynamics	Structural Integrity	
WEIGHT	30%	15%	10%	30%	15%	100%
OPTIONS	Criteria 1 SCORES	Criteria 2 SCORES	Criteria 3 SCORES	Criteria 4 SCORES	Criteria 5 SCORES	
Tall/Slim	4.33	7.33	6.67	9.33	6.67	6.87
Short/Wide	5.00	5.33	6.00	2.67	5.00	4.45
Medium/Rounded	5.67	5.33	6.33	5.00	6.33	5.58
Rectangular Shape	5.67	6.33	2.33	3.00	6.67	4.78
Tapered	5.33	4.33	3.33	6.67	4.33	5.23

Table 6.8. Decision Matrix for mission priority

	Criteria 1	Criteria 2	Criteria 3	
DESCRIPTION	Weight	# of Laps	Score	
WEIGHT	20%	30%	50%	100%
OPTIONS	Criteria 1 SCORES	Criteria 2 SCORES	Criteria 3 SCORES	
Mission 1	7.00	2.67	3.00	3.70
Mission 2	8.33	7.33	8.00	7.87
Mission 3	9.00	8.00	8.33	8.37

Table 6.9. Decision Matrix for Motor Type

	Criteria 1	Criteria 2	Criteria 3	Criteria 4	Criteria 5	
DESCRIPTION	Cost	Ease of Installation	Weight	Power	Durability	
WEIGHT	15%	30%	25%	20%	10%	100%
OPTIONS	Criteria 1 SCORES	Criteria 2 SCORES	Criteria 3 SCORES	Criteria 4 SCORES	Criteria 5 SCORES	
Out Runner	8.00	7.00	7.00	6.00	5.00	6.75
In Runner	5.00	4.00	5.00	6.00	7.00	5.10

A few initial prototypes were developed with this information to gain familiarity with different manufacturing techniques without as much emphasis on the specific dimensions of the design. This was done to identify potential weak points and challenges early on and quickly correct them. Once the aircraft characteristics were more thoroughly determined, a flight-ready prototype was developed to observe behavior in different flight conditions. After creating the decision matrices, the design was further developed by using historical data gathered from teams that scored highly at previous competitions and checking those numbers against the constraints provided for this year's competition by AIAA. Based on the mission requirements, it was determined that the team's overall score would be maximized by minimizing lap time and maximizing the total payload carried during flight. In combination with the historical data that was analyzed, the preliminary dimensions were chosen based on these two flight characteristics. A summary of the initial design dimensions is provided below in Tables 6.10 and 6.11.

Table 6.10. Preliminary Aircraft Design Dimensions for the Wing and Fuselage

Wings		Fuselage	
Wingspan (in.)	70	Shape	Multiple (stick)
Chord Length (in.)	13.68	Length (in)	52.5
Wing Area (in ²)	957.6	Width (in)	5
Airfoil Shape	Clark-Y	Height (in)	5
Wing Cube Loading	14.00		
Dihedral (degrees)	1.70		

Table 6.11. Preliminary Aircraft Design Dimensions for the Tail

Tail	
Horizontal Stabilizer Shape	Rectangle
Horizontal Stabilizer Width (in)	21.82
Horizontal Stabilizer Area (in ²)	199.00
Vertical Stabilizer Shape	Rectangle
Vertical Stabilizer Height (in)	9.49
Vertical Stabilizer Area (in ²)	71.86

Based on historical data, an approximate range of 15-20 pounds was chosen as the ideal takeoff weight of the aircraft. The lower end of this range was selected as it would allow a heavier payload to be carried during flight. Based on this weight, a wing cube loading analysis was performed to determine the wing dimensions, as this is a commonly used performance metric for designing RC aircraft. Equation 1 shown below was used to conduct the wing cube loading analysis [1].

$$WCL = \frac{\text{Takeoff Weight (oz)}}{\left(\frac{\text{Wing Area (in}^2\text{)}}{144}\right)^{1.5}} \quad 1$$

For this year's AIAA competition, the maximum allowable wingspan for the aircraft is 72 inches. Noting this constraint, the wingspan for the team's design was set at 70 inches to give some margin for potential manufacturing errors. This left the chord length and the takeoff weight as the only varying quantities in the wing cube loading analysis. A target value was reached by plugging in takeoff weight and wing area values from previous high-performing teams and averaging them. Once this value was determined, the ideal wing area could be calculated and used to find the chord length of the wing that would match the wing cube loading value of the team's design with the target value from the historical date. These were tested until a combination yielded a wing cube loading value that was as close to the target value as possible. A summary of the wing cube loading analysis is shown in Table E-1.

6.4 Design Iteration and Refinement

Once the initial dimensions of the aircraft were determined, finite element analysis simulations were conducted to test how the different components of the aircraft would hold up under different loading conditions. Different safety factors were used based on impact to aircraft weight and consequences of component failure to the success of the project. If the simulations resulted in stress values that did not line up with the desired safety factors, adjustments to the dimensions were made accordingly and testing was redone. A comparison of the initial and final designs for the front and rear landing gear components is shown in Figures 6.2 and 6.3 below.

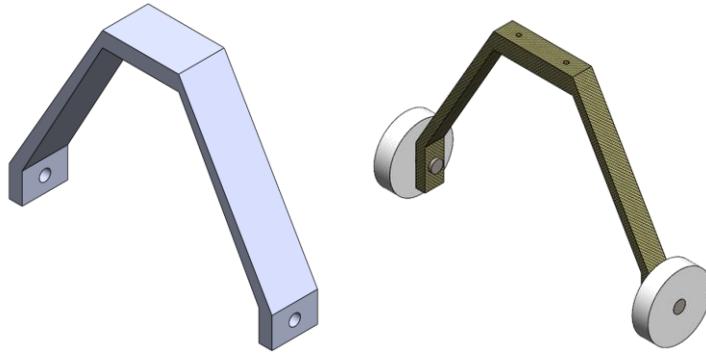


Figure 6.2. Design Iteration Examples for the Front Landing Gear

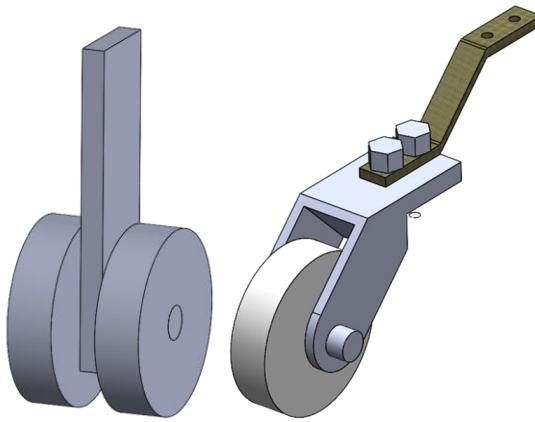


Figure 6.3. Design Iteration Examples for the Rear Landing Gear

The first iteration of the wing featured a skeleton with a wooden horizontal spar, and laser-cut cardboard ribs. The exterior of the wing was made from sheets of foam board with kerf cuts that allowed it to fold around the skeleton in the airfoil's shape. The fuselage was also made from cardboard, which was laser cut from a single sheet before being folded into shape. Due to the material choice of the fuselage, an important aspect of the aircraft's design was utilizing a rigid interior skeleton made of carbon fiber spars, with custom 3D printed hardware for mounting the wing, empennage, and landing gear to the load-bearing skeleton. The component for connecting the horizontal and vertical spars together and its CAD iterations are shown below in Figures 6.4, 6.5, 6.6, and 6.7.

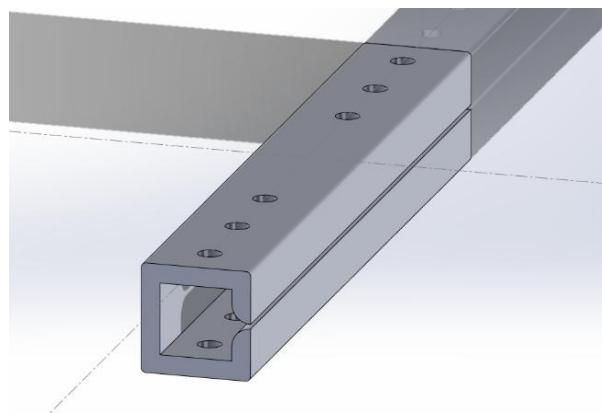


Figure 6.4. Spar Connector Iteration 1

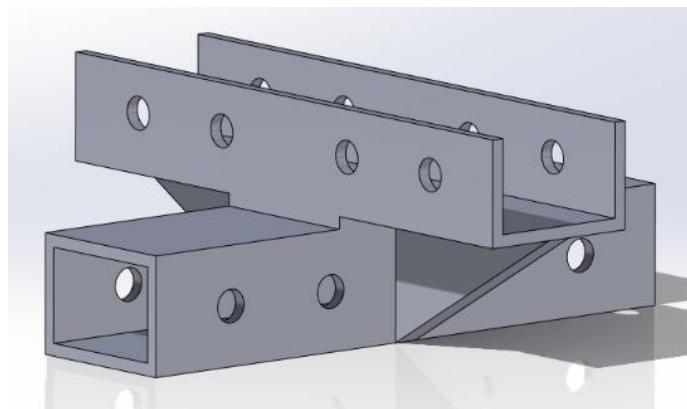


Figure 6.5. Spar Connector Iteration 2

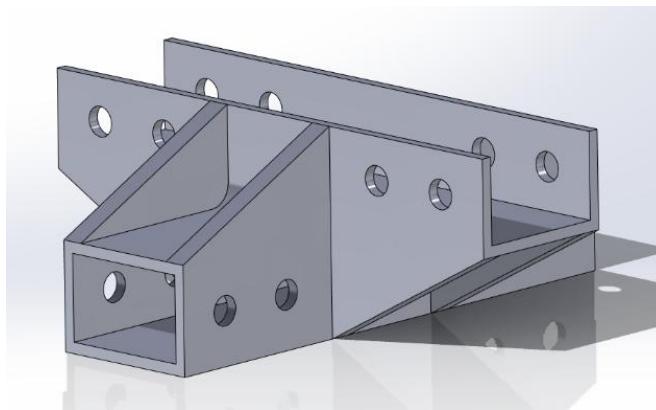


Figure 6.6. Spar Connector Iteration 3

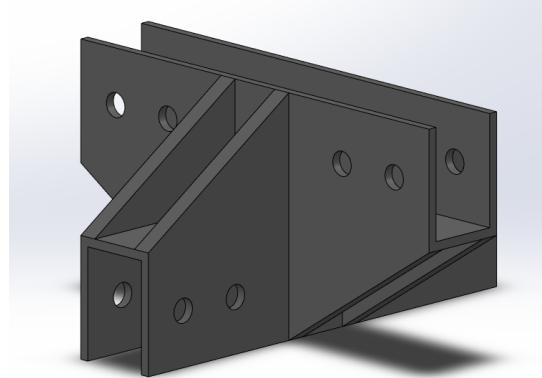


Figure 6.7. Spar Connector Iteration 4

The spar connector's initial purpose was to simply be able to join the two halves of the horizontal spar within the wing together as shown in Figure 6.4. However, it was modified to be able to connect the horizontal spar to the vertical spar as well. After the part was designated for this purpose, it was further modified with extra ribs, and thicker walls to make sure it would be strong enough. The final spar connector as shown in Figure 6.7 was 3D printed from low-density PLA. If the part is identified as a point of structural failure upon testing the aircraft, changing the material selection to carbon fiber reinforced PETG may be a viable method of refinement.

During the manufacturing process of the initial prototype, several other points of interest were identified where improvements could be made in future iterations. By the end of the Fall 2024 semester, the project remains in its initial iteration. However, the proposed improvements for the next iteration are explained in greater detail in Section 13.

6.5 Application of Codes and Standards

Development of the aircraft required the consulting of codes and standard to ensure that the design is safe and effective. The Federal Aviation Administration (FAA) 14 CFR Part 107 was a crucial standard for the design of a “small unmanned aircraft system” [2]. Subpart B in this standard lists crucial operating rules which include condition of safe operation, operation in certain airspace, and hazardous operation. One of the critical pieces of information regard the Remote Pilot Certification, which include the stipulations to be eligible for flight of the aircraft, which include the passing of an initial aeronautical knowledge test. 14 CFR Part 107 was a

crucial standard to consider as it related to the operation and safety of the aircraft. These rules as well as several others were a crucial role in aiding the design such that weight limitations were considered, and that the controller was designed such that the pilot could have as smooth operation of the aircraft as possible.

ASTM standards were also crucial in aiding in the design of the aircraft as it pertains to testing, design, and assembly. ASTM F2910 – 22 was a crucial standard to consider, as it outlines the design and assembly of a small aircraft [3]. When performing structural testing, it was critical to note that any aircraft structures must not fail at 1.5 times the load limit, which was considered when performing Ansys Structural Testing. This standard also considered design of propellers, propulsion, the landing gear, and other aircraft components.

Considering standards such as the Federal Communications Commission (FCC) and Underwriters Laboratories (UL) were crucial when refining the electronics for the aircraft. FCC Part 15 Subpart C indicated the various fundamental frequencies in which the controller must operate, notably stating a maximum frequency range of 24075-24175 MHz, and for legacy systems operating at lower frequencies at 54-72 MHz. As it pertained to servo motor testing, which was operated by a battery, it was ensured that all testing was performed using a new battery, per the same subpart. Total power input must also not exceed 100 milliwatts [4].

6.6 Professional and Ethical Responsibilities

The engineering team recognized that there were professional responsibilities when making critical design decisions. Most critically, the team ensured to discuss potential risks that would occur if the aircraft failed to ensure public safety. When performing structural analyses using Finite Element software, for example, it was critical that the team used a safety factor at minimum of 1.5 for all loaded structures. It was also the team's professional responsibility to validate data upon the conclusion of all tests and analyses to ensure that data was being reported accurately. These validations also justified several aerodynamic, electronics, and structural design decisions.

Due to the nature of the project, there were also some ethical considerations regarding the project. Namely, the competition is hosted by a Department of Defense contractor, Raytheon, and so the idea behind the project lends a hand to developing systems for national defense. With drone warfare moving to the front of the list of emerging technologies in this field, it is important to recognize that the designs for this year are simulating this style of launching a drone while in the air. Along with the ethics pertaining to the applications of the project, there are also the ethics behind the team's decisions. The team decided to prioritize the safety of the individuals in and around the areas in which the team acted. This was critical during the electronics and propulsion team's motor test.

While considering these responsibilities, the team also considered the global, economic, environmental, and societal implications of these solutions. The team considered the global impacts of the aircraft design while understanding that some aspects of the aircraft design were developed based on the work of previous AIAA Design-Build-Fly teams, such as Wing Cube Loading. When selecting materials for the aircraft components, the team ensured to design with cost-effective materials, such as cardboard for the wing ribs, to meet the Capstone Team's \$800 budget. Also, specific batteries for the electronics as well as the fuel tank bottles were considered to reduce the environmental risk in the event that they are not disposed of properly.

6.7 Analysis Methods

6.7.1 Aerodynamics

Alongside gathering general dimensions for the aircraft such as a 70-inch wingspan and a M2/M3 total weight of 15 pounds, a preliminary aerodynamic analysis was performed. Due to the maximum wingspan allowed being 6 feet across, or 72 inches, the team decided to go with 70 inches for the wingspan to stay under the limit. This would save time during the competition during the technical inspection stage, as the aircraft would pass with regard to this dimension easily and not be prevented from continuing in the competition. From historical data collected on prior successful DBF teams, a target of 14 for wing-cubed-loading was set, which is further explained in Appendix D. This parameter further helped to dimension the wing by setting a

chord length of about 13.68 inches. With these important dimensions chosen, Figure 6.8 details a brief analysis using MATLAB to calculate important flight characteristics such as the coefficient of lift, the coefficient of drag, the lift to drag ratio, the thrust required, and the power required. These parameters were calculated as a function of velocity in increments of 2 ft/s, ranging from 0 to 200 ft/s.

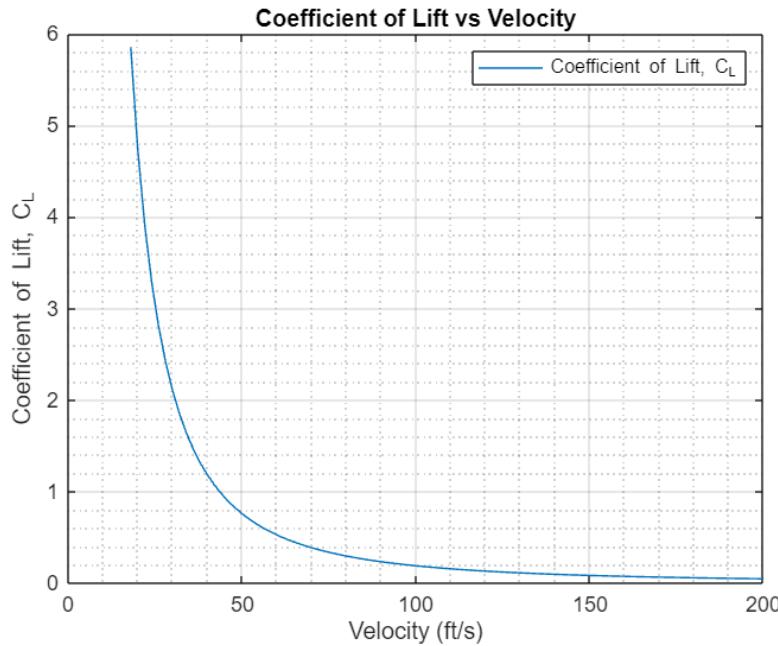


Figure 6.8. Preliminary Aerodynamic Analysis: COL vs Velocity

Figure 6.8 shows that the aircraft requires a lower coefficient of lift is required to produce the same amount of thrust as velocity increases, assuming the angle of attack is fixed. This was expected and is backed up mathematically by the equation for lift:

$$L = C_L \frac{\rho}{2} V^2 S$$

2

The preliminary aerodynamic analysis also observed the relationship between the different types of drag and velocity. Figure 6.9 below displays how induced drag and parasitic drag are affected by velocity, and this helps to determine an expected flight speed envelope for the aircraft.

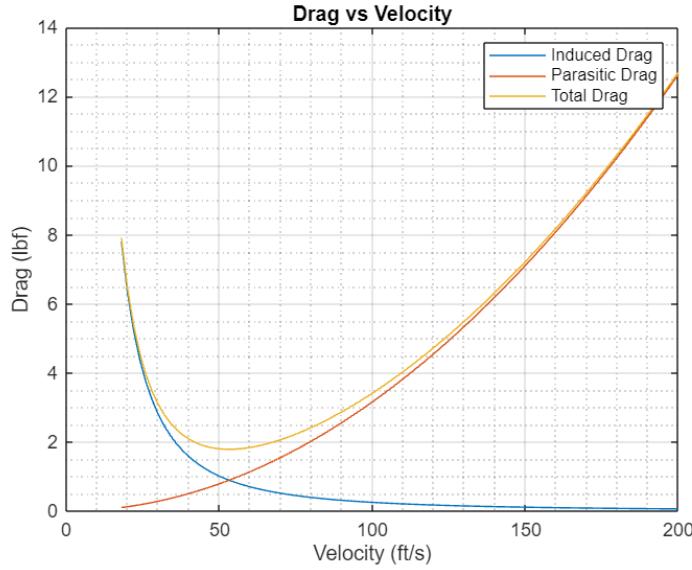


Figure 6.9. Preliminary Aerodynamic Analysis: Drag vs Velocity

Due to the scoring for M2 and M3, speed is a factor that the Capstone team needed to design around. Looking at the expected region of minimum total drag and considering the team's initial goal was a target of approximately 80 ft/s, the aircraft would likely be flying within the region of minimum drag. Pushing the aircraft to fly faster, closer to 120 ft/s, would increase the drag forces on the aircraft by nearly a factor of 2. This further backed up the team's initial decision to remain at approximately 80 ft/s for cruise velocity.

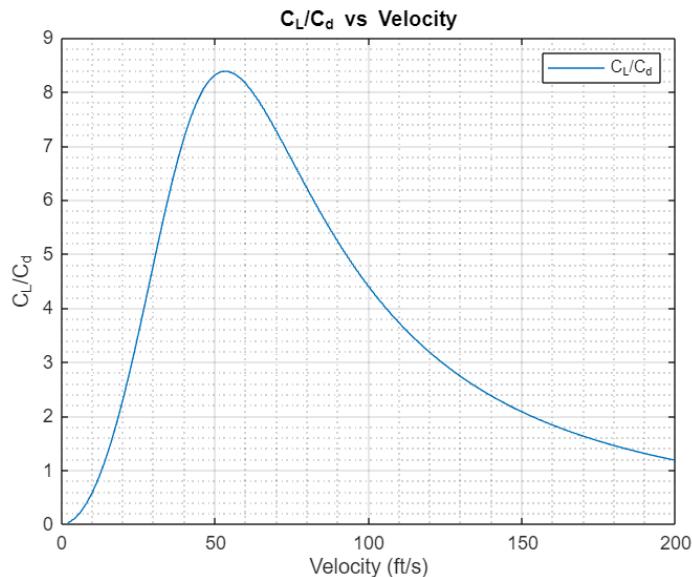


Figure 6.10. Preliminary Aerodynamic Analysis: Lift to Drag Ratio vs Velocity

An important flight characteristic that was observed was the lift-to-drag ratio and how it changed with velocity as seen in Figure 6.10. The peak lift-to-drag ratio resided approximately at the same velocity as the minimum total drag. This was expected and shows that despite the aircraft not flying at the peak lift-to-drag velocity, it will still produce approximately six times more lift than drag.

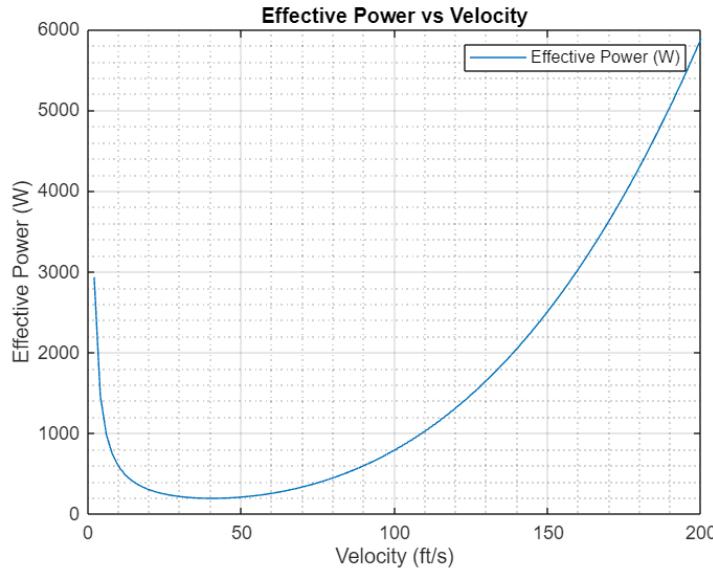


Figure 6.11. Effective Power vs Velocity

Above is an important hand-off from the aerodynamics team to the electronics and propulsion team. Figure 6.11 displays the relationship between velocity and the effective power required for the motor to pull the aircraft while in air. This relationship was observed with the aircraft weight expected during M2/M3 to give the electronics and propulsion team some helpful information regarding choosing a battery and motor.

6.7.2 Electronics and Propulsion

Using the aerodynamic analysis done previously, the electronics team used a program called eCalc to obtain a selection of recommended propulsion setups consisting of propellers, motors, and batteries. These recommended setups were then compared to the current inventory available to the team from previous semesters of DBF. A selection was made that met the criteria detailed by the team's preliminary design. The selected motor was an eFlite 52 590 KV paired with a

Turnigy 3300 mAh 6S battery and a propeller size of 8x12. This selection was integrated into the electrical system with an 80-amp ESC which was chosen to fall below the competition limitation of 100-amp continuous current draw.

The team performed testing on the chosen propulsion setup to obtain thrust and battery capacity data. From the results of the test, the efficacy of the propeller motor combination and the capacity of the battery were determined.

Servo testing was performed to obtain data on the force limits of each available servo size. Loading conditions in the test were determined by calculations made based on the forces the aircraft's control surfaces would feel during flight. From the results of the servo testing, proper servo sizes were determined.

6.7.3 Structures

The structures team was responsible for creating the CAD models for the aircraft and analyzing the behavior of the aircraft under different loading conditions. Using the dimensions that were defined in previous stages of the design process, models for each aircraft component were created with appropriate materials assigned. Additional models were created for subassemblies of the front and rear landing gear, a subassembly of the load-bearing components, and a full assembly of the aircraft. During this process, a few different prototypes were created to supplement the CAD models and test out different manufacturing techniques.

Once CAD models were created, ANSYS simulations were conducted on load-bearing components of the aircraft. These simulations were used to predict how the aircraft structure would hold up during flight and landing. The first simulation on the wings modelled the wing tip test performed at a pre-flight check for the competition. This was done by constraining the ends of the wing spar, similarly to roller supports, and applying a 100 lbf downward force in the middle of the wing. This model was validated by conducting a beam-bending test in the Makerspace with a 2.50" x 20.25" x 0.50" aluminum beam and comparing the maximum deflection. Since the actual wing spar had different dimensions and material properties than the

beam that was used for this validation, the simulation was first set up with the same dimensions and material properties as the actual beam that was tested for the deflection comparison.

The expected deflection of the beam was calculated based on the geometric and material properties of the beam for further validation. The calculated deflection, which is representative of the beam deflection with two simple supports, is shown in Equation 3. The results of these deflection comparisons are provided in Section 8.

$$\delta = \frac{F * L^3}{48 * E * I}$$

3

Simulations were also conducted on the front and rear landing gear, as those components are intended to absorb the majority of the impact force during landing. The team decided that the validation of the wing model was sufficient to prove the team's ability to properly model the landing gear as well. Per Standard 14 CFR 25.303, all components of an aircraft that are flight tested should be able to withstand 1.5 or more times the loads applied to it. For the landing gear specifically, it was decided that the front gear should be built to a safety factor of at least 10 and the rear landing gear should be built to a safety factor of at least 3. This was done for a few different reasons. As previously mentioned, the landing gear, specifically the front landing gear will absorb the majority of the force during landing, so any damage during the missions would cause significant issues for the rest of the aircraft. Additionally, the landing gear was modeled as an isotropic material even though carbon fiber is generally anisotropic due to the orientation of the fibers. The method of manufacturing the landing gear resulted in fibers that had varying orientations, so the decision to model the landing gear as isotropic was made under the assumption that it would be more conservative and a more accurate simulation on average.

6.8 Evaluation of Design through Testing Procedures

6.8.1 Electronics and Propulsion

To evaluate the motor, propeller and battery combination, a static thrust test was performed utilizing a thrust stand, two calibrated 20Kg load cells, and an AC ammeter. The test consisted of

two runs at the target thrust value calculated during the preliminary aerodynamic analysis adjusted using factors of safety of 1.5 and 1.2 and logging the data from the load cells using a Python code which can be seen in Appendix D-5. The logged mv/V data from the load cells was integrated using calibration data in order to transform the data into thrust readings. These readings were plotted to visualize the values of thrust the propulsion setup would obtain at various current inputs. A visual presentation of the thrust test setup can be seen in Figure 6.12.

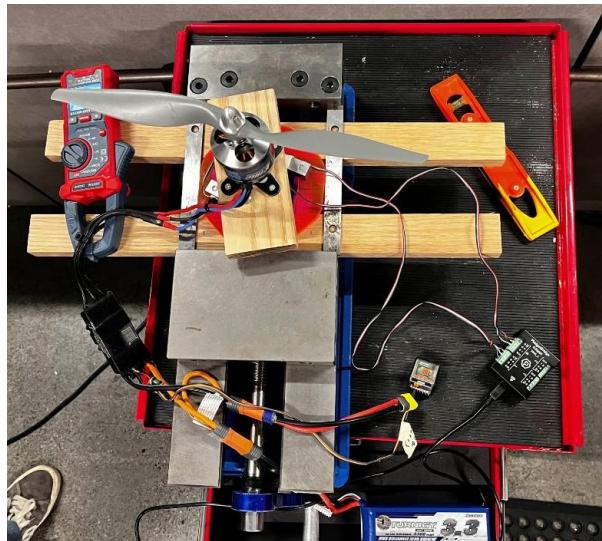


Figure 6.12. Thrust Test Stand Setup

Testing of the servo motors was performed to determine what size servos were necessary to move the control surfaces and obtain the electrical current data to operate the servo in order to size the control surface battery. The servo test utilized an Arduino to spin the servo motor from 0-90 degrees while the estimated weight each control surface would experience was hooked on to the servo arm. An oscilloscope was connected on either side of a 1-ohm resistor to measure the electrical current the servo motor would pull throughout its cycle. These tests were performed for servos sized 9g, 25g, and 43g and used weights calculated in section 7.3.2. A photograph of the servo test can be seen in Figure 6.13.

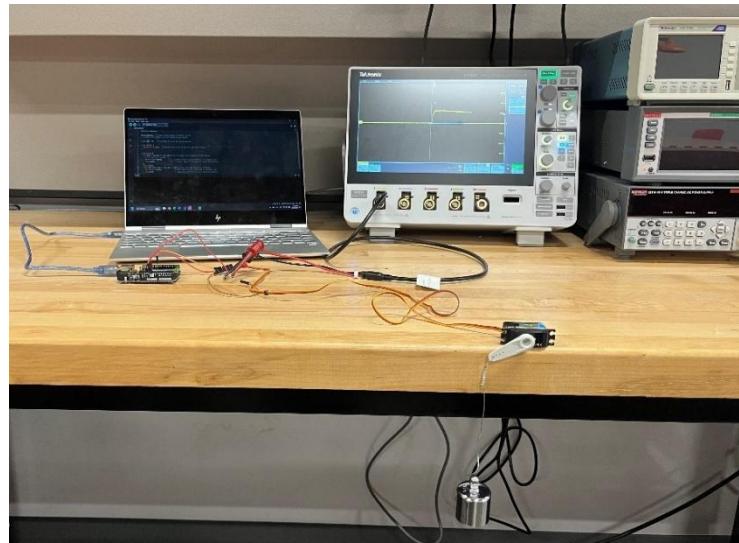


Figure 6.13. Servo Test Setup

6.8.2 Structures

The structures team conducted a test to estimate the accuracy of ANSYS when simulating beam bending as seen in section 6.7.2. The test consisted of bending an aluminum beam by using a known force in the center of the beam and comparing the results to ANSYS. The test apparatus can be seen in Figure 6.14 and consisted of a force gauge connected to a threaded rod that allowed consistent force application. The beam is supported by a rod on each side that can be adjusted based on the length of the beam being tested. A displacement sensor was also placed under the lower rectangular prism that connects the force gauge to the threaded rod so the displacement could be recorded and compared to ANSYS.

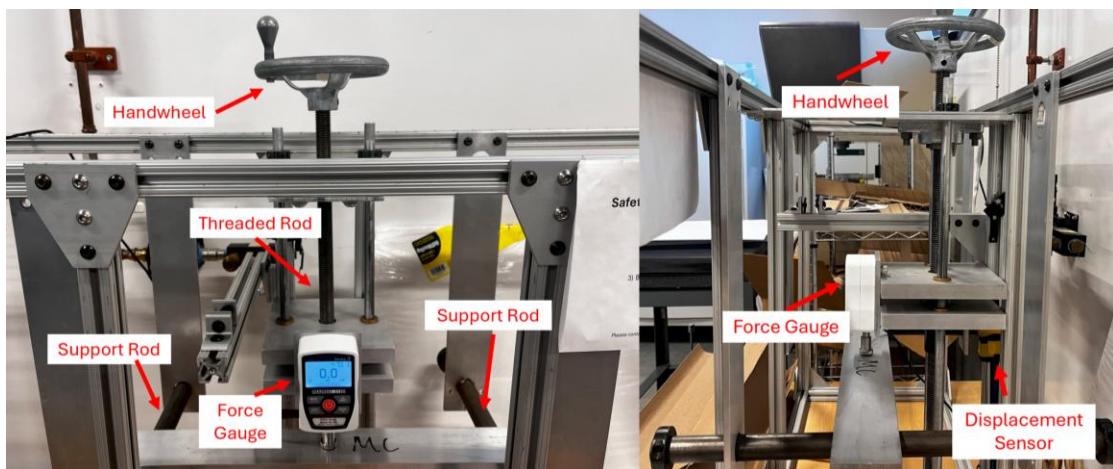


Figure 6.14. Beam Bending Test Setup

7 Design Solution and Final Specifications:

7.1 CAD Drawing of Full Aircraft

A CAD drawing of the final prototype of the aircraft can be seen in Figure 7.1. The length of the wing was 70 in. with the length of the fuselage being 36.02 in. It should be noted that the height of the landing gear is 12.23 in. off the ground, which shows that the landing gear would be able to be placed under the fuselage. The length from the nosecone to the end of the rear landing gear is 57.49in which shows that there is also enough space to fit the glider under the fuselage.

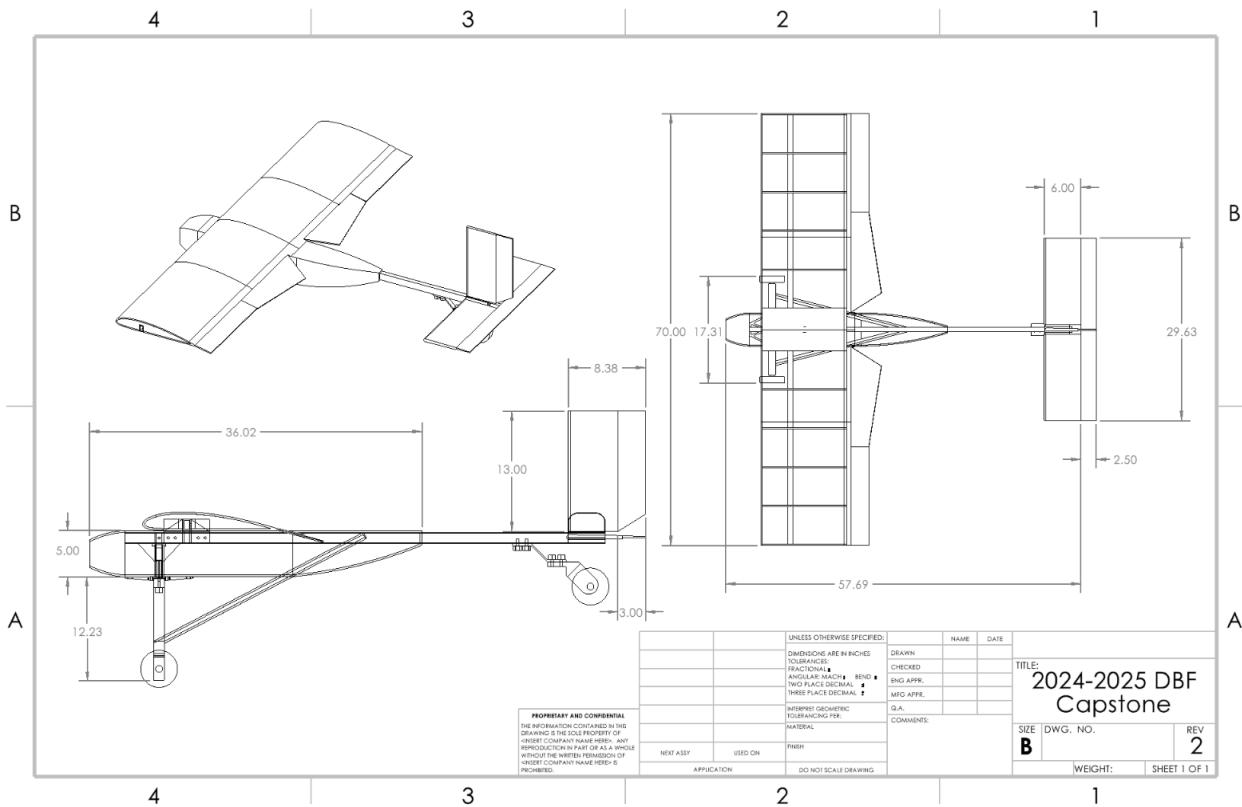


Figure 7.1. CAD Drawing of Final Aircraft Prototype

7.2 Codes and Standards

There are many codes and standards that need to be followed in order to build and operate an aircraft safely. In Table 7.1, the codes and standards the team followed are within FAA

guidelines as well as any other codes that are believed to be pertinent to the team can be seen. The most important regulations to learn and follow were the ones pertaining to the rules created by the FAA around flying RC drones and aircraft. The FAA has evolved these guidelines greatly in the past 10 years as hobby RC flight has increased in prevalence and popularity so staying on top of these regulations is important to understand what needs to be done in order to fly the aircraft legally.

Table 7.1. Codes and Standards

Code/Standard	What Does the Code or Standard Say
Part 107 [1]	<ul style="list-style-type: none"> • License for flying aircraft over 250g • Aircraft must be registered with FAA • Aircraft must have a transponder if not flown in a FRIA
Remote ID Information [5]	If not flying in FRIA site, aircraft must broadcast: <ul style="list-style-type: none"> • Drone ID (Serial Number) • Drone location and altitude • Drone Velocity • Takeoff Location and elevation • Time Mark
14 CFR § 25.303 Factor of Safety [6]	Federal Airworthiness Regulation Part 25.303 says that a safety factor of 1.5 must be applied to the prescribed limit loads which are considered external loads on the structure
ASTM F593 [7]	This specification covers the requirements for stainless steel bolts, hex cap screws, and studs 0.25 to 1.50 in., inclusive, in nominal diameter in a number of alloys in common use and intended for service applications requiring general corrosion resistance

7.3 Critical Calculations and Analyses

7.3.1 Aerodynamics

Concluding the preliminary sizing and aerodynamic assessment of the aircraft, an extensive aerodynamic analysis was performed to determine the takeoff performance, selected wing airfoil, control surface sizes, and stability characteristics. To approximate the takeoff speed of the aircraft, the stall speed was initially calculated using Equation 4: [8]

$$V_S = \sqrt{\frac{2W}{\rho S C_{L_{MAX}}}}$$

4

The stall speed computed from preliminary analysis assumed the wing maximum lift coefficient ($C_{L_{MAX}}$) to be equal to 1. This assumption was made to provide a reasonable estimate for the wing C_L prior to the eventual 2D airfoil selection and 3D wing definition in XFLR5. As such, the stall speed was then equal to the velocity of the aircraft where C_L equaled 1. Once the stall speed had been calculated, the takeoff speed was then determined for propeller-driven aircraft using Equation 5 [8]. Results for preliminary stall and takeoff speed can be found in Table 7.2.

$$V_{\text{Takeoff}} = 1.15 V_S$$

5

Table 7.2. Preliminary Stall and Takeoff Speed for Missions 1-3

Mission	Aircraft Weight (lbs.)	V_{Stall} (ft/s)	$C_L @ V_{\text{Stall}}$	V_{Takeoff} (ft/s)	$C_L @ V_{\text{Takeoff}}$
Mission 1	7.44	30.72	1.0	35.333	0.755
Mission 2 & 3	15.00	43.58	1.0	50.121	0.755

The preliminary stall and takeoff speed data provided a basis for airfoil selection by numerically quantifying the required C_L for the aircraft to maintain the fixed lift condition for takeoff. To ensure this, 2D direct foil analyses were performed in XFLR5 to analyze if selected foils could produce C_L values in excess of the stall and takeoff C_L found during preliminary analysis. Another consideration in eventual airfoil selection was ease of manufacturability. Construction of the 3D wing geometry for the aircraft had intended on utilizing folded foam board for the construction process. The implication of this was that the selected airfoil profile needed to be simple, largely having a flat bottom contour in comparison to a greater curved design. Previous UMass Lowell DBF teams utilizing identical wing manufacturing techniques had successful implementation of the Clark-Y airfoil both in manufacturing and flight. Based on the combination of the C_L values required from preliminary analysis and the team defined manufacturing objectives, the Clark-Y airfoil was ultimately selected due to its success in fulfilling both requirements.

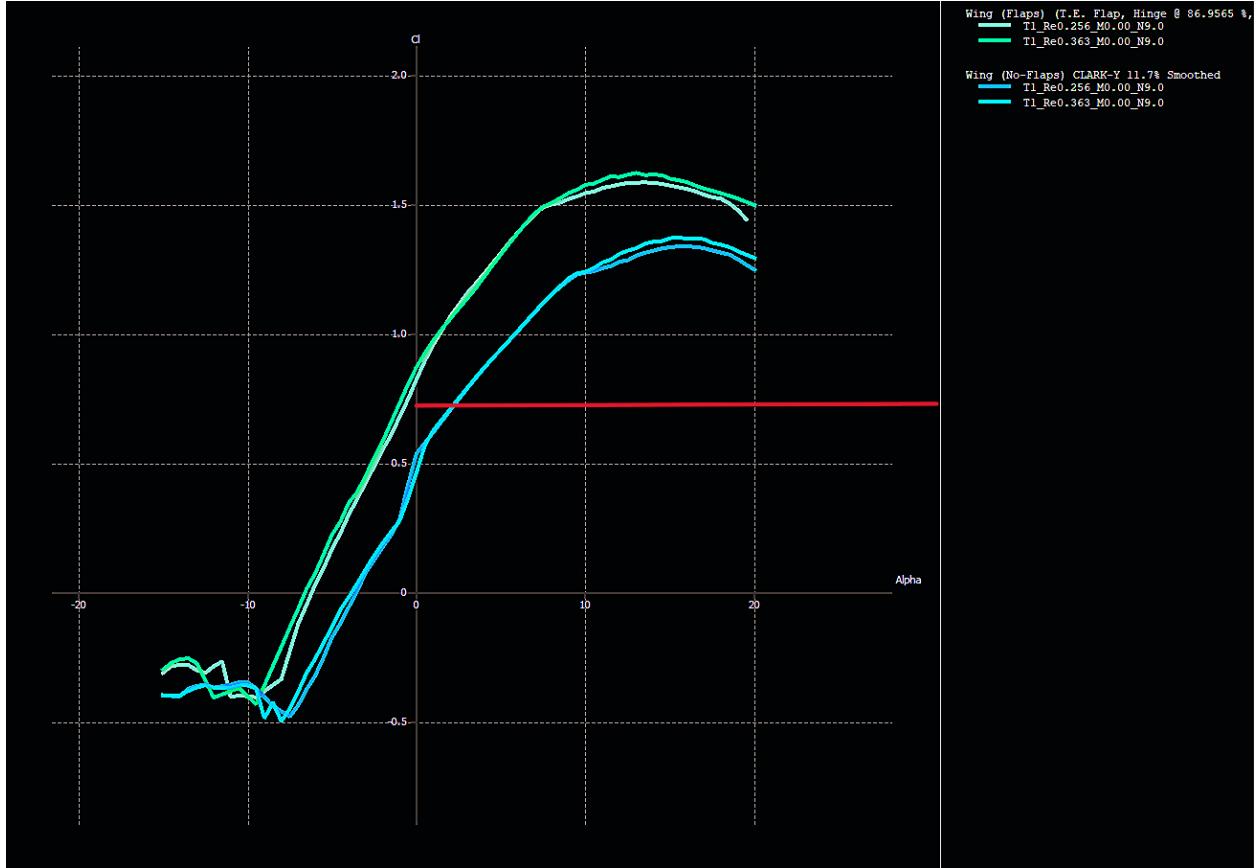


Figure 7.2. C_L vs (α) polar of Clark-Y airfoil configurations. C_L @ $V_{Takeoff}$ is indicated by red line.

Completion of the 2D foil analysis was followed by defining the geometry and overall size of the aircraft's empennage (tail structure) and primary control surfaces. This served the function of setting up the aircraft to undergo 3D aerodynamic analysis in XFLR5 while also providing geometric dimensions for manufacturing of the prototype.

The empennage of the aircraft consists of the vertical and horizontal stabilizer, along with its accompanying control surfaces. Appropriate empennage size is critical to aircraft performance, as this ensures that the tail structure is capable of generating sufficient lift and trim forces for the aircraft during flight modes such as takeoff and cruise. The vertical and horizontal stabilizers were sized by utilizing tail volume coefficients for general aviation, single engine aircraft as

outlined in Raymer [9]. The selected tail volume coefficients were 0.0443 and 0.672 for the vertical and horizontal stabilizer, respectively. The surface area of both the vertical and horizontal stabilizers were determined by rearranging Equation 6. Numerical analysis detailing empennage sizing may be found in Appendix D.

$$c_V = \frac{l_v \cdot S_v}{S_w \cdot b_w} \quad 6$$

$$c_H = \frac{l_H \cdot S_H}{S_w \cdot c_{MAC}}$$

Control surface dimensions for the aircraft were based on reported sizes from historical data and previous UML DBF groups. The main parameters of interest were the control surface sizes as percentage of the main wing, horizontal and vertical stabilizers chord length and span. Detailed calculations for control surface sizing can be found in Appendix D, with tabulated values seen in Table E-2.

7.3.2 Electronics and Propulsion

Electrical limitations to the total energy stored and total current were imposed through the 2024-2025 competition rules. To determine allowable battery sizes, Equations 7 and 8 were provided through the AIAA-DBF competition rules:

$$100A \geq W_{hours} = \frac{\text{Capacity}}{1000} (A) * V \quad 7$$

$$100A \geq I (A) = \frac{\text{Capacity}}{1000} (A) * C_{rating} \quad 8$$

With the limitations imposed to battery storage, battery life becomes an important factor to the design of the aircraft. To calculate the estimated time of complete battery consumption, Equation 9 was used.

$$T(\text{minutes}) = \frac{\frac{\text{Capacity}}{1000} (\text{A})}{I (\text{A})} * 60$$

9

The control surfaces of the aircraft are controlled by servo motors, the sizes of the servo motors must be selected to support the in-flight loads and maneuvering. Calculations were made to find the approximate forces felt by each control surface during the estimated maximum loading condition. Estimated torque requirements were determined using the calculated forces. The force and torque equations can be seen in Equations 10 and 11, respectively.

$$F = \frac{1}{2} \rho v^2 A c_d \sin(\theta) \quad 10$$

Where ρ is the air density, v is the velocity of the air, A is the control surface area, c_d is the drag coefficient, and θ is the deflection angle of the control surface.

$$T = rF \quad 11$$

Where r is the length of the servo arm, and F is the estimated force calculated in Equation 10. The estimated wind load and torque requirements can be seen in Table 7.3.

Table 7.3. Estimated Load on Control Surfaces and Required Servo Torque

	Estimated Load and Required Torque: Max Speed	
	Wind Load (ozf.)	Torque Requirement (oz.-in)
Aileron Up, Down	12.71	9.09
Flap Up, Down	23.54	16.83
Elevator Up, Down	19.70	12.51
Rudder Left, Right	9.54	6.05

7.3.3 Structures

As discussed in Section 6.7.3, the team validated the beam bending test setup using equations for deformation, as this was the simplest quantity to measure from the provided setup. The Finite

Element Analysis that was performed was validated by this beam bending test setup. Displayed in Figure 7.3 is the equivalent setup captured in an ANSYS simulation.

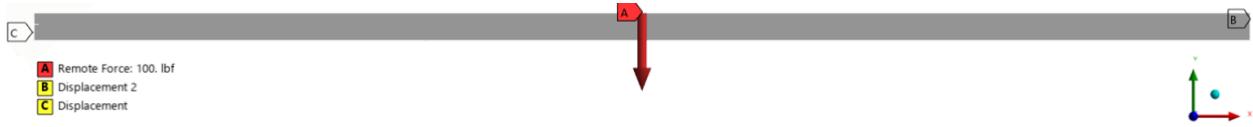


Figure 7.3. Replication of Beam Bending Test Conditions in ANSYS

As shown in Figure 7.3, the two displacements capture the behavior of the beam bending test in the actual testing apparatus, restricting movement in the y and z axes while allowing movement in the x-axis. Equation 12 shown below determines the theoretical deflection of a beam fixed by two simple supports when a force is applied in the middle [10]. This equation is dependent on the material properties and geometry of the beam.

$$\delta = \frac{F * L^3}{48 * E * I} = \frac{F * L^3}{48 * E * \left(\frac{1}{12} b h^3\right)} = \frac{F * L^3}{4 * E * b h^3} \quad 12$$

$$\delta = \frac{(100 \text{ lbf}) * (20.25 \text{ in.})^3}{4 * \left(9906077 \frac{\text{lbf}}{\text{in.}^2}\right) * (2.5 \text{ in.})(0.5 \text{ in.})^3} = 0.0671 \text{ in.}$$

Another important calculation for the ANSYS simulations was the impact force experienced by the landing gear. This was initially modeled as an impulse force with an angle of attack of 3.5° . However, the angle changed over time with the progression of analyses performed by the aerodynamics sub-team. A critical calculation was the impact force being applied to the landing gear components. As a result, the team simplified this process by using a conservative force estimate of 10 G's (150 lbf) and 6 G's (60 lbf) on the front and rear landing gear, respectively.

The equation used for impulse force is shown in Equation 13 below.

$$mV_y = F_y \Delta t \quad 13$$

$$F_y = \frac{m_y V}{\Delta t}$$

$$F_y = \frac{(6.80 \text{ kg}) \left(\left(30.48 \frac{\text{m}}{\text{s}}\right) \sin(3.5^\circ) \right)}{0.025 \text{ s}} = 506.12 \text{ N} = 113.88 \text{ lbf}$$

Safety factor calculations were also used to gauge if the dimensions were appropriate for each landing gear component. These were calculated by comparing the maximum equivalent stress experienced by the components in the simulations to the ultimate stress of Carbon Fiber composite materials. The calculated safety factors of the front and rear landing gear components are shown in Equation 14 below [11].

$$SF = \frac{\sigma_{\text{ultimate}}}{\sigma_{\text{allowable}}} \quad 14$$

$$SF_{\text{front}} = \frac{\sigma_{\text{ultimate}}}{\sigma_{\text{allowable}}}$$

$$SF_{\text{front}} = \frac{139961 \text{ psi}}{13329 \text{ psi}}$$

$$\mathbf{SF_{\text{front}} = 10.5}$$

$$SF_{\text{rear}} = \frac{\sigma_{\text{ultimate}}}{\sigma_{\text{allowable}}}$$

$$SF_{\text{rear}} = \frac{139961 \text{ psi}}{41342 \text{ psi}}$$

$$\mathbf{SF_{\text{rear}} = 3.38}$$

Wing Cube Loading analysis was also performed to get an estimate of the ideal wing area based on the target weight of 15 pounds determined by analyzing historical data from previous high-performing reports. The calculation for this is provided below using Equation 1.

$$WCL = \frac{\text{Takeoff Weight (oz)}}{\left(\frac{\text{Wing Area (in}^2\text{)}}{144}\right)^{1.5}}$$

$$WCL = \frac{240 \text{ oz}}{\left(\frac{957.6 \text{ in}^2}{144}\right)^{1.5}}$$

$$\mathbf{WCL = 14.00}$$

7.4 Final Design Solution



Figure 7.4. Final Design Build

The final design for the Fall 2024 DBF Capstone team shown in Figure 7.4 was an aircraft that implemented a rectangular high wing, conventional empennage, with a singular motor, and taildragger landing gear configuration. The wingspan was set to be 70 inches across to stay under the 6ft maximum set by the DBF rules this year. The plane had an M1 weight of approximately 7.5 pounds, without the X-1 test vehicle and the fuel tank bottles. Construction of the wing was done by folding foam board around laser-cut cardboard ribs. The wing was attached using large rubber bands to the top of the fuselage – a common method used for RC aircraft – which provided a strong connection between the vertical and horizontal spars. The two spars were made of carbon fiber hockey sticks which were a lightweight, high-strength solution allowing the bottles (fuel tanks) to be mounted underneath the wing securely without causing too much deflection in the wing.

8 Results

8.1 Aerodynamics

Three-dimensional aerodynamic analysis of the aircraft was performed to gather more information about the Mission 1-3 takeoff performance characteristics. This was namely done in an effort to determine the angle of attack (α) necessary for the aircraft at takeoff as well as verify if the stall and takeoff velocities of the aircraft largely agreed with the preliminary aerodynamic analysis. To model the aircraft's takeoff condition more appropriately in XFLR5, the analyzed geometry included the aircraft's primary control surfaces. This was done in order to model control surface deflection at takeoff, notably for the trailing-edge flaps and elevator as seen in Figure 8.1. The flaps and elevator were modeled with a 20° upward and 15° degree downward deflection, respectively. The selected angles were based on the anticipated maximum deflection for the control surfaces, which were consistent with prior UML DBF aircraft. Since preliminary analysis did not initially consider the surface area from the wings' control surfaces, the total wing surface area in the preliminary analysis was updated and re-analyzed to enable a more accurate comparison of the preliminary and simulation results. A Type 2 fixed lift solver setting was implemented in XFLR5 for all 3D takeoff analyses run. This was done in order to obtain a more accurate comparison of simulation results to the preliminary analysis which also utilized the same fixed lift assumption. Results of the 3D aerodynamic analysis for the stall and takeoff velocities, associated lift coefficients and α , are detailed in Tables 8.1 and 8.2 for Missions 1-3

Table 8.1. Comparison of Mission 1 Takeoff and Stall Speed Results

Analysis Method	Wing Surface Area (in ²)	Aircraft Weight (lbs)	V _{Stall} (ft/s)	C _L @ V _{Stall}	V _{Takeoff} (ft/s)	C _L @ V _{Takeoff}	α (°)
Preliminary Analysis	1101.24	7.44	28.65	1.00	32.95	0.755	—
XFLR5			29.11	0.966	34.42	0.691	8
Percent Error (%)			1.59	3.51	4.47	9.26	

Table 8.2. Comparison of Mission 2 & 3 Takeoff and Stall Speed Results

Analysis Method	Wing Surface Area (in ²)	Aircraft Weight (lbs)	V _{Stall} (ft/s)	C _L @ V _{Stall}	V _{Takeoff} (ft/s)	C _L @ V _{Takeoff}	α (°)
Preliminary Analysis	1101.24	15.00	40.66	1.00	46.76	0.756	—
XFLR5			41.55	0.956	48.88	0.748	8
Percent Error (%)			2.18	4.60	4.53	1.07	

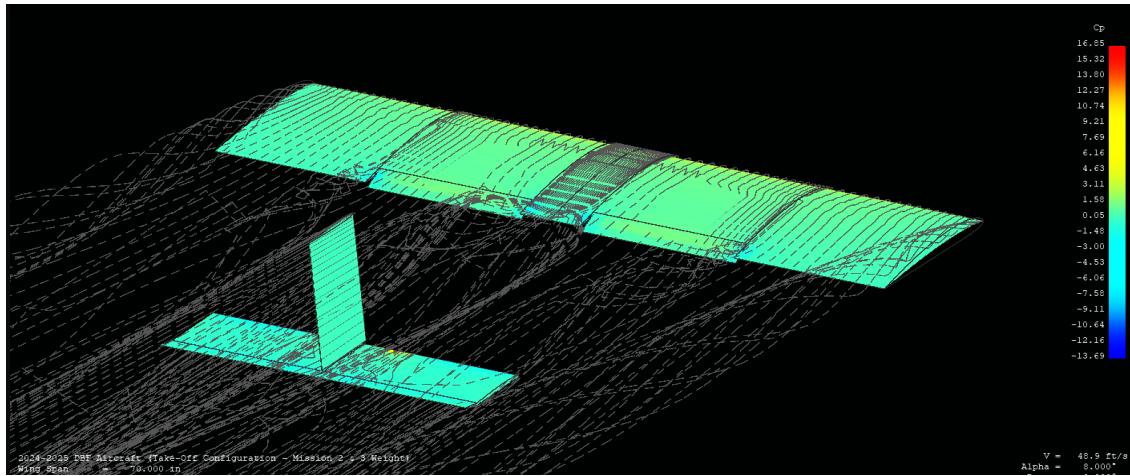


Figure 8.1. 3D Aero Analysis of aircraft in XFLR5 (Streamline visualization at $\alpha = 8^\circ$)

Based on the results of the 3D aerodynamic analysis performed in XFLR5, an α set to 8 degrees would be sufficient for the aircraft design at takeoff. This was validated by observing the low percent error between the compared values for the stall and takeoff velocities, as well as the associated lift coefficients between analysis methods which suggest the simulation and preliminary analysis results were in agreement with one another.

Static and dynamic stability analyses were performed for the aircraft in the cruise configuration during Missions 1-3. Favorable stability characteristics ensure that the natural tendency for the aircraft is to return to a state of equilibrium or attitude after induced perturbations. Since the majority of the aircraft's flight time is spent at cruise, ensuring both positive static and dynamic stability for the aircraft at this flight mode was sought in the stability evaluation. Static stability analysis began with ensuring prerequisite stability for the aircraft. This is the condition that the slope of the pitching moment coefficient (C_m) vs α curve of the aircraft at cruise is negative. The

implication of this is that when the aircraft undergoes pitch-up and pitch-down behavior in flight after a perturbation, the C_m is able to restore the aircraft to its pre-perturbed state. In general, for increasingly positive α of the aircraft, the C_m should become increasingly more negative while increasingly negative α should produce increasingly more positive C_m . As seen in Figures 8.2 and 8.3, the aircraft satisfies this prerequisite stability case at level flight across Missions 1-3.

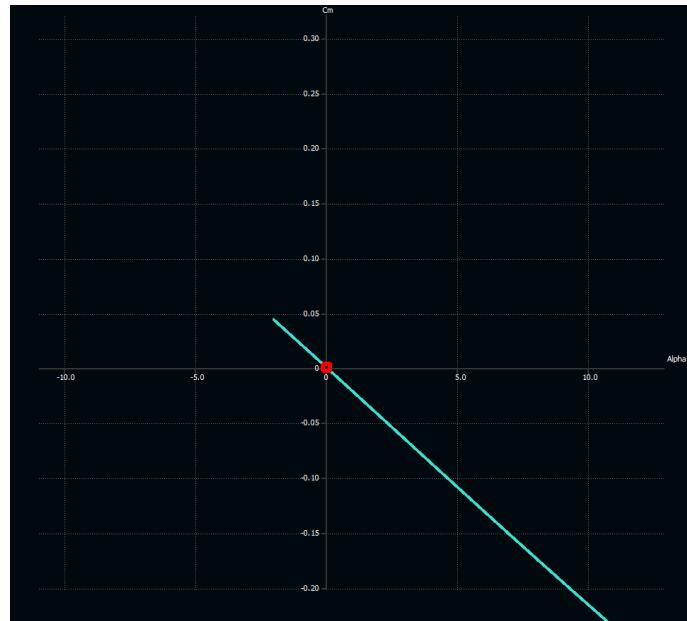


Figure 8.2. C_m vs α (Mission 1)

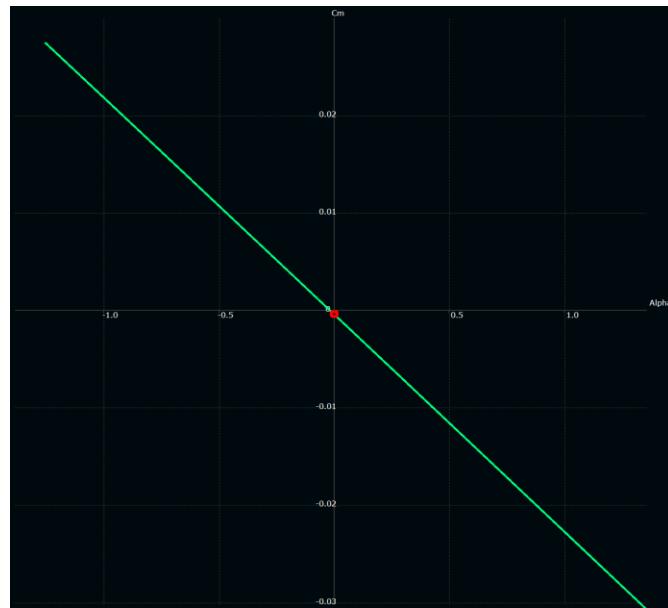


Figure 8.3. C_m vs α (Mission 2 & 3)

Static stability analysis continued, determining the trim angle of the horizontal stabilizer for the aircraft in order to produce a C_m equal to 0 at α equals zero degrees. A C_m of zero at level flight meant the aircraft could fly without constant or repetitive actuation of the flight control surfaces by the pilot in order to maintain a zero degree α . To obtain this C_m at cruise, the angle of the horizontal stabilizer was iterated upon until the value of the C_m for the aircraft was near zero. It was found that by deflecting the entire horizontal stabilizer to -0.325 degrees, sufficient trim forces could be exerted to obtain a near zero C_m . The trim velocity required to maintain the near zero C_m was also found with the static stability analysis, which defined the approximate cruise velocities (V_{Cruise}) for balanced Mission 1-3 flight. Regarding the V_{Cruise} from the Mission 1 static stability analysis, seen in Table 8.3, the value for the cruise velocity while attainable, could be increased by retrimming the aircraft to be statically stable at a flight velocity closer to the V_{Cruise} range estimated from the preliminary analysis. The V_{Cruise} from the Mission 2-3 static stability analysis as seen in Table 8.4, is largely considered to be acceptable for the aircraft given the initial V_{Cruise} estimates defined in the preliminary analysis, between $80 - 120 \frac{ft}{s}$ at the Mission 2-3 aircraft weight.

Table 8.3. Mission 1 Static Stability Analysis Results

$C_m @ \alpha = 0^\circ$	0.00041
Horizontal Stabilizer Trim Angle	-0.325°
V_{Cruise}	63.2

Table 8.4. Mission 2 & 3 Static Stability Analysis Results

$C_m @ \alpha = 0^\circ$	-0.00042
Horizontal Stabilizer Trim Angle	-0.325°
V_{Cruise}	90.2

Dynamic stability for the aircraft refers to the aircraft's behavior/response over time following a perturbation, which is largely oscillatory. The dynamic stability analysis for the aircraft was performed to ensure that the various aerodynamic oscillatory modes the aircraft underwent were both sufficiently damped and low frequency across both the longitudinal and lateral axes. To perform the dynamic stability case in XFLR5, inertia tensors were defined for the aircraft. This involved the addition of point mass representations of critical aircraft components across Mission 1-3 such as the electronics, landing gear, fuel tanks, and X-1 Glider. Tables 8.5 and 8.6 below, outline the inertia tensors for the aircraft across Missions 1-3.

Table 8.5. Aircraft Inertia Tensor (Mission 1)

Mass (lbs)	X (in)	Y (in)	Z (in)	Component Name
0.500	6.000	0.000	0.000	Fuselage
1.500	-7.000	0.000	0.00	Motor
1.500	-6.000	0.000	0.00	Battery
0.650	46.000	0.000	0.000	Combined Tail Mass
0.375	21.000	0.000	0.000	Spar Mass
1.500	-2.500	0.000	0.000	Front Landing Gear
0.125	46.000	0.000	0.000	Rear Landing Gear
0.250	7.250	0.000	0.000	Propeller
0.290	-6.250	0.000	0.000	ESC
0.083	46.50	-3.000	0.500	Elevator Servo
0.083	46.50	-0.500	2.000	Rudder Servo
0.083	3.000	8.750	-0.500	Right Flap Servo
0.083	3.000	26.250	-0.500	Right Aileron Servo
0.083	3.000	-26.250	-0.500	Left Flap Servo
0.083	3.000	-8.750	-0.500	Left Aileron Servo
0.251	-7.000	0.000	0.000	Motor Mount Box
7.44				Total Mass (lbs)

Table 8.6. Aircraft Inertia Tensor (Mission 2 & 3)

Mass (lbs)	X (in)	Y (in)	Z (in)	Component Name
0.500	6.000	0.000	0.000	Fuselage
0.550	3.000	0.000	-0.250	X-1 Glider
3.500	3.375	17.500	0.00	Right Fuel Tank
3.500	3.375	-17.500	0.00	Left Fuel Tank
1.500	-7.000	0.000	0.00	Motor
1.500	-6.000	0.000	0.00	Battery
0.650	46.000	0.000	0.000	Combined Tail Mass
0.375	21.000	0.000	0.000	Spar Mass
1.500	-2.500	0.000	0.000	Front Landing Gear
0.125	46.000	0.000	0.000	Rear Landing Gear
0.250	7.250	0.000	0.000	Propeller
0.290	-6.250	0.000	0.000	ESC
0.083	46.50	-3.000	0.500	Elevator Servo
0.083	46.50	-0.500	2.000	Rudder Servo
0.083	3.000	8.750	-0.500	Right Flap Servo
0.083	3.000	26.250	-0.500	Right Aileron Servo
0.083	3.000	-26.250	-0.500	Left Flap Servo
0.083	3.000	-8.750	-0.500	Left Aileron Servo
0.251	-7.000	0.000	0.000	Motor Mount Box
15.000				Total Mass (lbs)

Running the dynamic stability analysis, results were obtained for the oscillatory modes or eigenvalues of the aircraft using a Root Locus plot for Missions 1-3 as seen in Figures 8.4-8.7. As seen in Tables 8.7-8.10, numerical results of the dynamic stability analysis indicate that nearly all of the oscillatory modes exhibited positive dynamic stability with eigenvalues lying on the left-hand plane of the Root Locus plot. These modes were accompanied with sufficient damping and low frequency of oscillation both longitudinally and laterally apart from the spiral mode. While the spiral mode is considered unstable for the aircraft in Missions 1-3, by lying on the right-hand plane of the Root Locus Plot, the low frequency and magnitude characteristic of the mode means it is largely correctable by the pilot through careful observation and slight rudder adjustment.

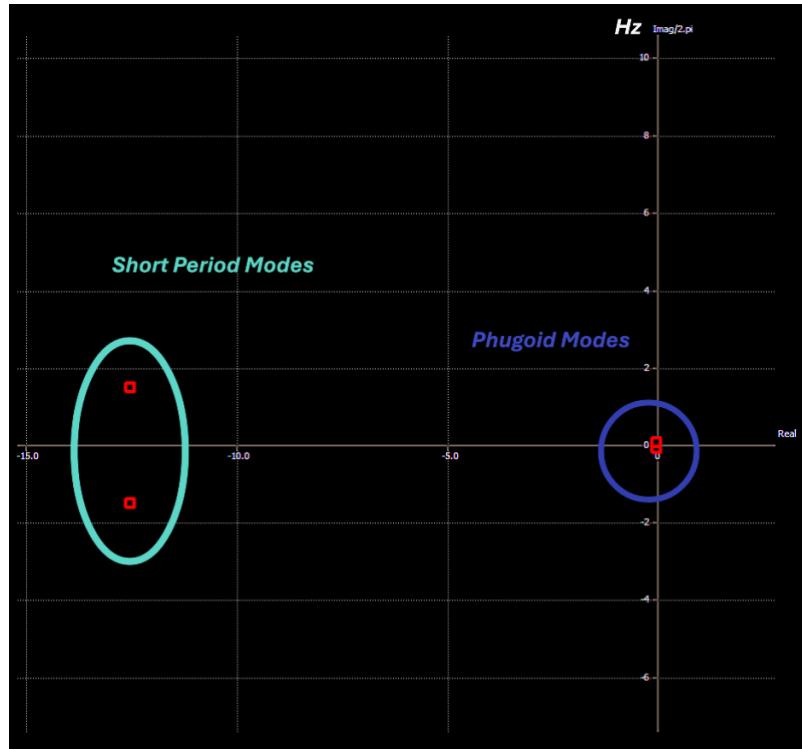


Figure 8.4. Mission 1 Root Locus (Longitudinal Stability)

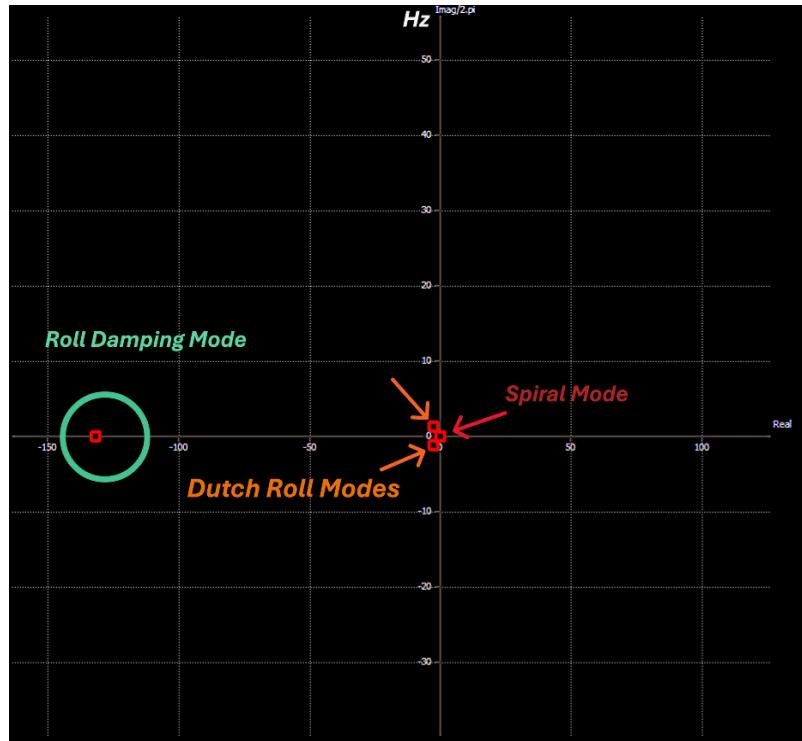


Figure 8.5. Mission 1 Root Locus (Lateral Stability)

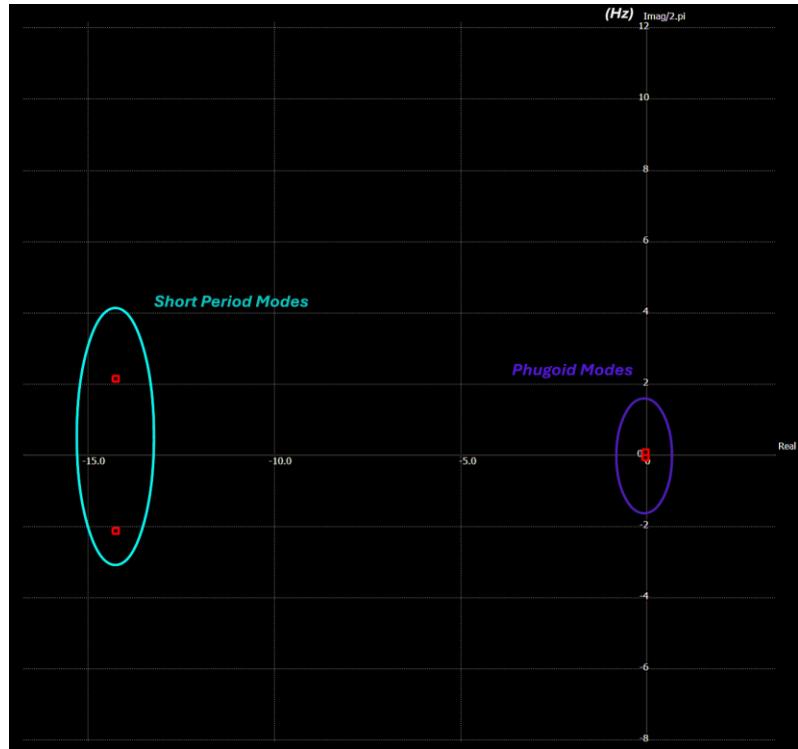


Figure 8.6. Mission 2 & 3 Root Locus (Longitudinal Stability)

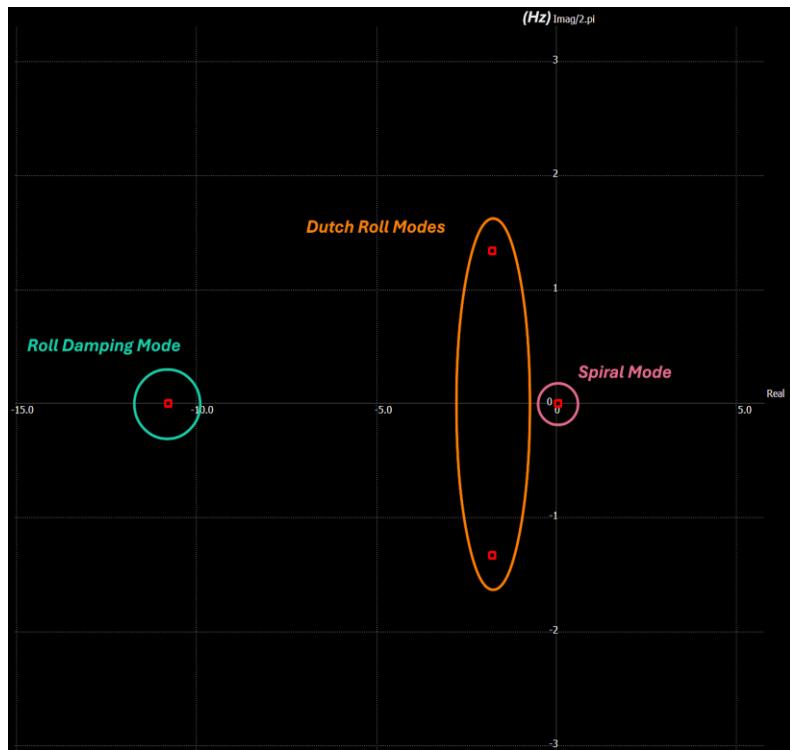


Figure 8.7. Mission 2 & 3 Root Locus (Lateral Stability)

Table 8.7. Mission 1 Dynamic Stability (Longitudinal Modes/Eigenvalues)

Longitudinal Modes	Mode Values (Eigenvalues)
Short Period Mode #1	-12.51737 - 9.34024i
Short Period Mode #2	-12.51737 + 9.34024i
Phugoid Mode #1	-0.02756 - 0.50426i
Phugoid Mode #2	-0.02756 + 0.50426i

Table 8.8. Mission 1 Dynamic Stability (Lateral Modes/Eigenvalues)

Lateral Modes	Mode Values (Eigenvalues)
Roll Damping Mode	-131.55239 + 0.00000i
Dutch Roll Mode #1	-2.42973 - 7.83115i
Dutch Roll Mode #2	-2.42973 + 7.83115i
Spiral Mode	0.07630 + 0.00000i

Table 8.9. Mission 2 & 3 Dynamic Stability (Longitudinal Modes/Eigenvalues)

Longitudinal Modes	Mode Values (Eigenvalues)
Short Period Mode #1	-14.23096 - 13.35896i
Short Period Mode #2	-14.23096 + 13.35896i
Phugoid Mode #1	-0.01829 - 0.40945i
Phugoid Mode #2	-0.01829 + 0.40945i

Table 8.10 Mission 2 & 3 Dynamic Stability (Lateral Modes/Eigenvalues)

Lateral Modes	Mode Values (Eigenvalues)
Roll Damping Mode	-10.75947 + 0.00000i
Dutch Roll Mode #1	-1.76313 - 8.37973i
Dutch Roll Mode #2	-1.76313 + 8.37973i
Spiral Mode	0.05188 + 0.00000i

8.2 Electronics & Propulsion:

Using the resultant data from the thrust test, the team was able to determine the efficacy of the chosen propulsion setup as well as the estimated life of the battery.

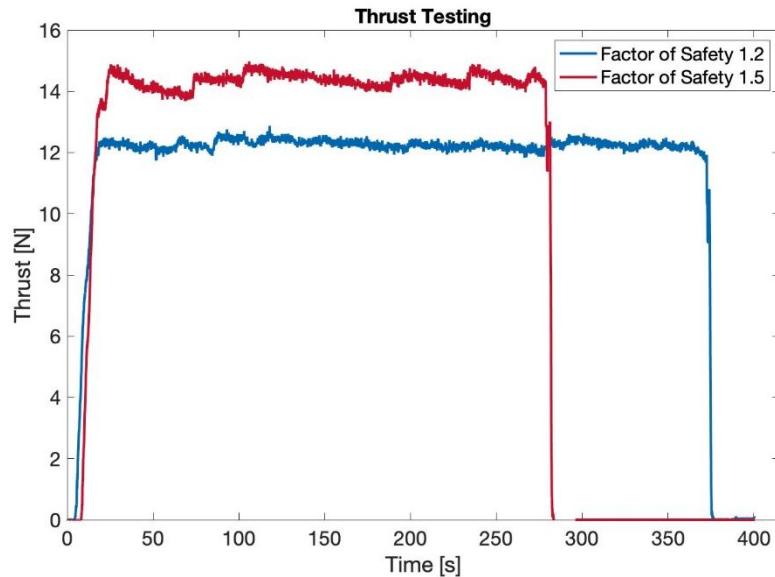


Figure 8.8. Thrust Testing Results

Through this testing it was determined the motor, battery, and propeller combination will produce sufficient thrust, however, the time that the aircraft can remain in flight with the current battery setup is limited. In Figure 8.1, it is shown at a thrust safety factor of 1.2, the goal of a 6-minute flight time off a single charge of the battery is reached but does not provide enough extra battery life for takeoff, landing, and unforeseen flight conditions. The team would like our flight time to be lengthened as well as increase the thrust factor of safety up to 1.5, to do this the electronics team has recommended a lower required value of thrust to achieve the design specifications. A suggested approach to acquire this value is through lowering the overall weight of the aircraft.

The resultant data from the servo test allowed for the selection of properly sized motors for each control surface. The selection of the motor for each surface depended on two criteria; the torque provided from the motor is greater than what is necessary to move the control surface and the weight of the motor. The servo test provided the necessary information to make this selection as

well as additional electrical data to assist in the selection of the secondary control battery. Figure 8.9 shows the resultant graphs from the servo test.

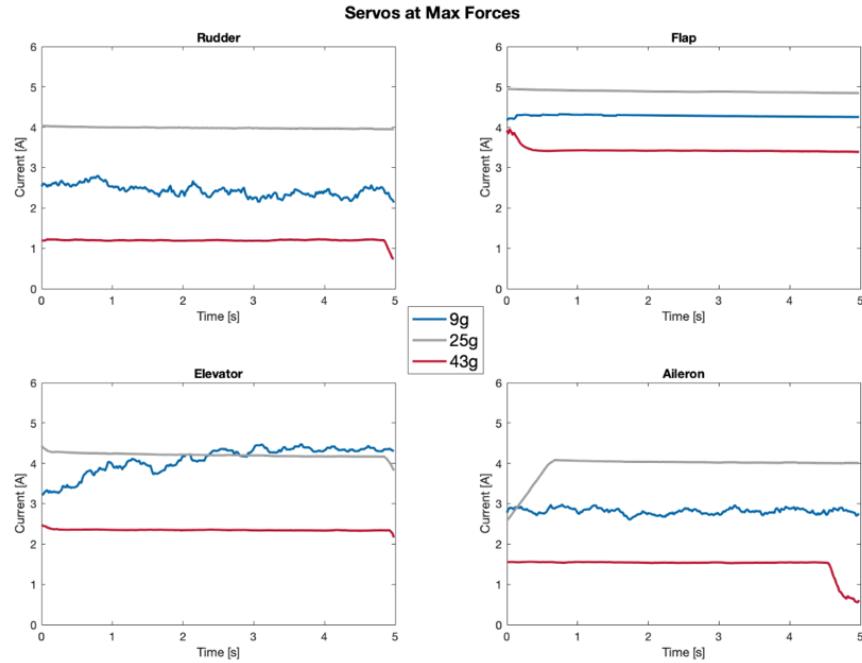


Figure 8.9. Servo Testing Results

Using the data and information gathered through the test, the selections of each control surface servo motor and the control battery were made. To optimize the weight of the electrical system, the lightest servos that could provide enough torque to fully move each of the control surfaces were selected. The servo distribution and required battery size for the selection can be seen in Table 8.11.

Table 8.11. Servo Distribution Chart

Servo Distribution						
Weight (g)	Weight (lbs)	Placement	Spec Sheet Stall Torque (oz*in)	Estimated Torque Req (oz*in)	Operating Voltage	Max Operational Current
43	0.0948	L,R Flap	44.52	36.20	6	3.42
25	0.0551	L,R Aileron	29.16	19.55	6	4.03
43	0.0948	Elevator	44.52	26.90	6	2.34
9	0.0198	Rudder	20.86	13.02	6	2.38
Total Servo Weight		Max Total Operational Current		Required Battery Size		
(g)	(lbs)	(A)		(mAh)		
188	0.4145	12.168		3.380		

Using the required battery size calculated from the servo selection, a 1200 mAh 2 cell battery was selected for the control battery. This battery supplies more than the required storage necessary for the control electronics, has an exceptionally low weight, and is part of the team's existing inventory from a previous semester, making it the recommended choice.

8.3 Structures

Upon completion of the beam bending test, it was determined that the aluminum beam deflected at approximately one-sixteenth of an inch, and is compared to hand calculations and simulation results in Table 8.12.

Table 8.12. Comparison of ANSYS Simulation and Beam Bending Testing for Wing Spar

	FEA	Hand Calc
Expected Result (in.)	0.0563	0.0671
Measured Result (in.)	0.0625	0.0625
% error	10.97%	6.800%

This comparison yielded a 10.97% discrepancy with the FEA, and 6.800% as a result of the formula in Equation 12, which was deemed acceptable as there were likely imperfections in the actual beam as well as lack of measurement with a precise tool. While Aluminum 6061 material properties were utilized in the FEA and hand calculations, these true properties were difficult to account for in the real-life model, as that model may have different properties based on differences in manufacturing or wear. A ruler was also used to measure the deflection as opposed to a more precise tool such as a dial indicator. Considering that the deflection is on a small scale,

this error was considered negligible. Based on the results of this test setup validation, it was deemed that the ANSYS simulations performed using carbon fiber properties was accurate.

There is also an observable difference between the value obtained from the formula and the Finite Element Analysis (FEA), with a percent difference of 17.5% between both values, which is likely due to the set of boundary conditions. Table 8.5 shows that the hand calculations predict a higher value while the FEA predicts a lower value than what was measured in the beam bending test. In the ANSYS model, the beam was restricted from movement in the y and z directions, but free to move in the x-direction, per the coordinate system in Figure 8.4. Based on these constraints, it is likely that the ANSYS model did not reflect the simple supports that are assumed in Equation 12. Simple supports suggest that there is free rotation at both ends. Recreating the same beam with free rotation at the ends in ANSYS likely yields a more similar result when utilizing Equation 12. This lower deflection is likely explained by the fact that the beam was unable to rotate per the set constraints in the ANSYS model.

Based on the results of this validation it was assumed that the ANSYS simulation for the wing spar would still remain accurate enough when changing the dimensions and material properties to match the actual wing spar used on the aircraft that it could still be used to effectively analyze wing behavior during flight. The accuracy of the simulation with aluminum provided confidence that the same could be achieved for carbon fiber. Plus, it is more difficult to measure the deflection of carbon fiber with the provided setup due to its higher strength-to-weight ratio. Upon completion of this beam bending test, an ANSYS simulation was performed on the expected final design of the wing, which included approximate material properties and constraints based on the AIAA's wing-tip test, completed at each yearly competition. Results of the ANSYS simulation are shown below in Figures 8.10, 8.11, and 8.12.

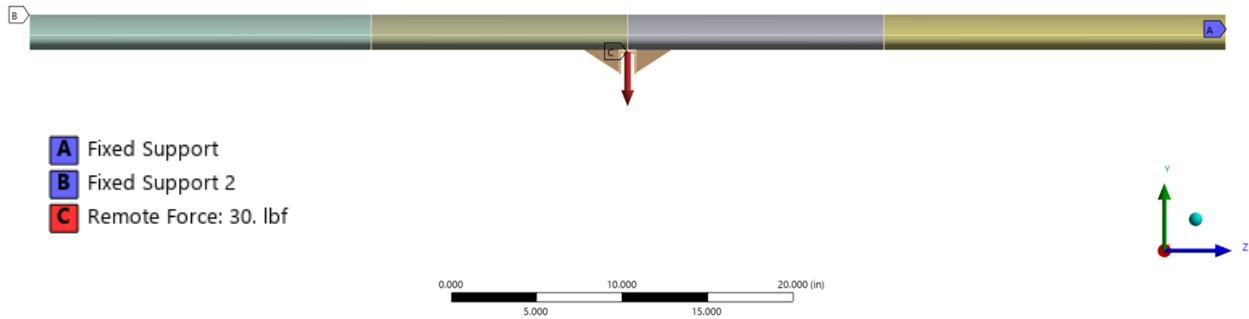


Figure 8.10. Ribbed Wing “Wing-Tip” Test Constraints

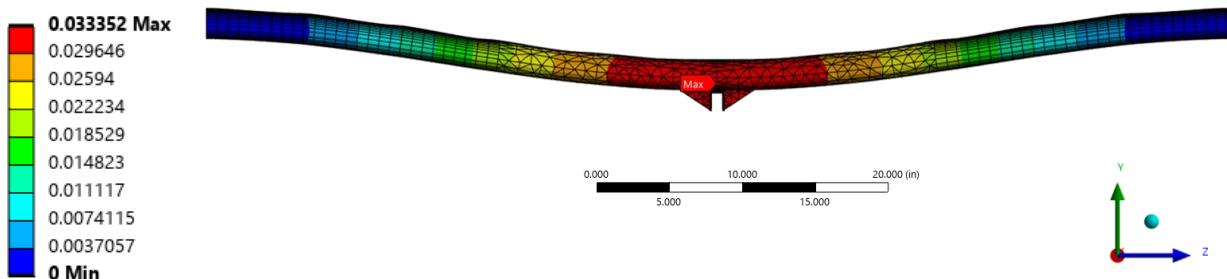


Figure 8.11. Wing-Tip Test Total Deformation

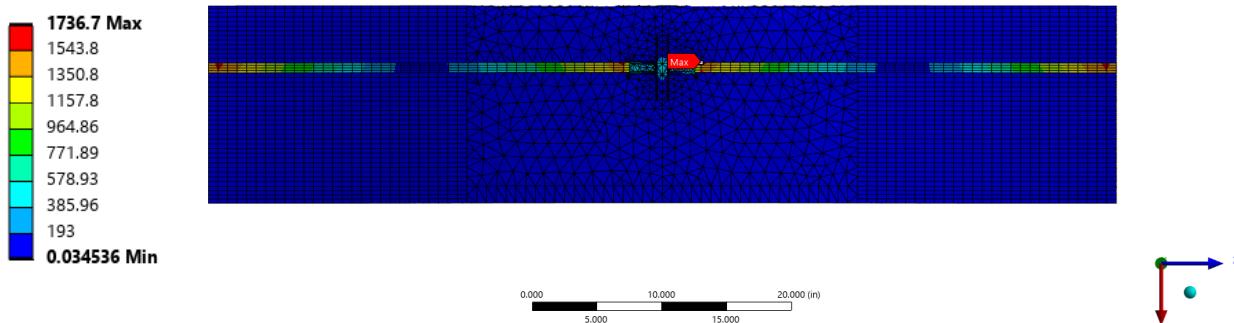


Figure 8.12. Wing-Tip Test Equivalent (Von-Mises) Stress

As a result of the beam bending test setup validation, the team proceeded with the AIAA’s standard wing tip test, which is set to the same constraints as shown in Figure 6.14, where both ends of the wing are fixed and as a load pulled from the center. 30 pounds was the selected load as an approximate 2 Gs based on the overall aircraft weight of 15 lbf. In this ANSYS simulation, the material properties for each of the components of the wings are set. From the bottom view of the wing, shown in Figure 8.5, the major bearing component of this wing is the spar, and is placed along the wing to resist shear and bending forces, and holds up fairly well based on a simulated deflection of 0.033 inches. These results match the predicted outcomes considering a

high strength-to-weight ratio of the carbon fiber wing spars, and the validation using the beam bending test reinforced the team's confidence to accurately model these simulations in ANSYS.

Target safety factors of 10 and 3 were chosen for the front and landing gear, respectively. Per Standard 14 CFR § 25.303 Factor of Safety, load-bearing components of aircraft must have a safety factor of at least 1.5 [5]. The team decided that due to the importance of the landing gear to the overall durability of the aircraft, it was acceptable to exceed this value because the benefits of ensuring the landing gear would remain intact outweighed the drawback of a marginal weight increase to the aircraft. To approach these target safety factors, geometry was altered slightly in between tests on the ANSYS simulations until the values were deemed acceptable by the team.

Images of the simulation results are shown in Figures 8.13 and 8.14.

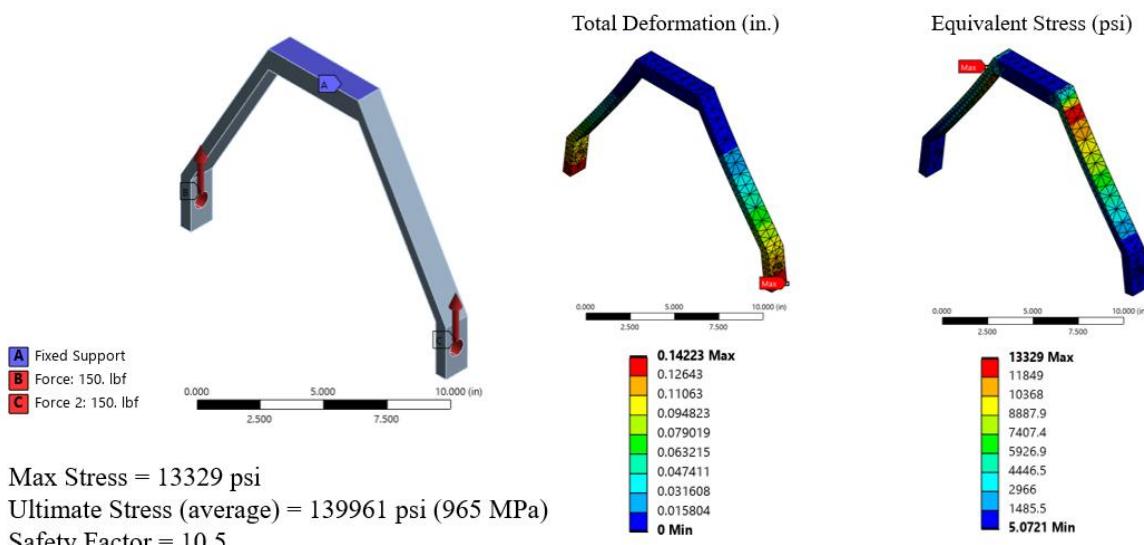


Figure 8.13. Front Landing Gear ANSYS Simulation Results

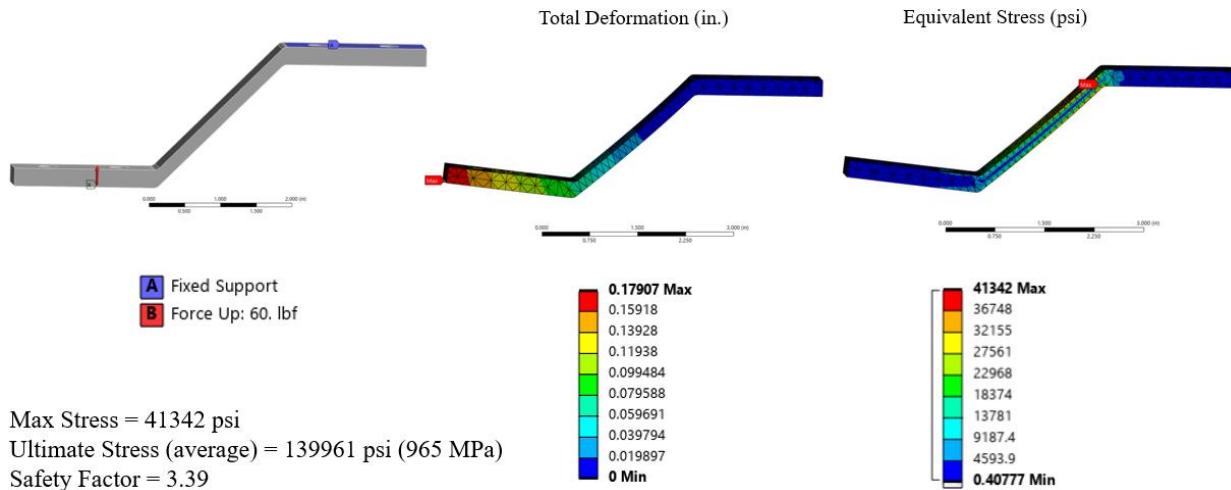


Figure 8.14. Rear Landing Gear ANSYS Simulation Results

The team accepted the deformation values output by the ANSYS simulation under the assumption that having some flex would allow the landing gear to better absorb and distribute the impact force. Additionally, these values did not pose a concern with damaging the X-1 glider stored underneath the fuselage because the dimensions chosen for the landing gear left plenty of additional room for some flex in the design. Looking at the deformation, the goal of the landing gear was to have some deformation to allow the impact force to be absorbed by the landing gear more effectively. Stress values were also considered acceptable based on the ultimate stress of carbon fiber and the strong correlation with the target safety factors that were chosen.

9 Risk Assessment

Risks associated with this project were vital to consider when developing the design. The biggest risk associated with the aircraft is the safety of spectators when flying. The aircraft is large enough that if it were to experience an uncontrolled landing, it could potentially cause serious injuries to spectators. More specifically, this is a concern for the aircraft's load-bearing components. If the spars experience damage during pre-flight testing or execution of each of the flight missions, the pilot would lose the ability to control the plane during flight. Additionally, if the landing gear were to fail, the pilot would lose the ability to safely land the aircraft, which poses risks to both the spectators and the aircraft itself. This risk has been addressed by over-designing the dimensions of each component of the aircraft and performing the associated

iterations of simulations to meet or exceed the required safety factors per the codes and standards addressed in Sections 6 and 7.

Another risk associated with the aircraft is electrical failure. If the electronics are damaged or are unable to keep the aircraft's flight pattern stable, this would put the operator and spectators at risk of being injured. This risk has been accounted for by scoping electronic equipment that meets the required power demands of the designed aircraft to generate enough thrust and activate each control surface during the maximum loading conditions as well as practicing safe handling and storage protocols for the chosen LiPo batteries. If the electronics were to catch on fire during flight, there is potential for significant environmental damage if the aircraft were to get caught in a tree for example. This may be mitigated by performing all flights in an area where there is a low concentration of trees.

Risks associated with the testing and manufacturing of the aircraft were also important to consider. Taking precautionary measures, such as wearing the proper PPE and doing the proper training for different testing and manufacturing apparatuses was done to mitigate the risk of injury. Another important note is that due to the risks associated with flying RC aircraft, there are certifications required to do so. Since the team was not able to acquire the necessary certifications, no flight testing was conducted due to the potential legal implications. To assess the severity and likelihood of each of these risks, a risk assessment matrix was created. As shown below in Table 9.1, several risks fall in the yellow area of the matrix. The team deemed this as acceptable because although the severities are very high, the rates of occurrence are very low due to the control methods for each risk.

Table 9.1. Risk Assessment Matrix

		Significance					
		Very Insignificant (A)	Insignificant (B)	Moderate (C)	Significant (D)	Very Significant (E)	
Rate of Occurrence	Very Unlikely (1)			1. Aircraft failure due to pre-flight checks		1. Aircraft damage due to landing gear failure 2. Environmental damage caused by the aircraft crashing	
	Unlikely (2)				1. Injuries from manufacturing the plane	1. Injuries to spectators due to structural failure 2. Injuries to spectators due to electronic failure	
	Moderate (3)						
	Likely (4)						
	Very Likely (5)						

10 Detailed Cost Analysis

The 3 main factors in the cost analysis are the labor cost, materials, and manufacturing of producing the aircraft. In order to estimate the time spent on the project, 3 hours per credit was used which means that on average, about 9 hours per person per week was spent on this project as Capstone is a 3-credit class. The semester is 15 weeks and there are 8 people in Capstone which means that about 1,080 hours were spent on this project over the course of the semester. An additional 60 hours were spent by a student in the Design Build Fly club coding, designing, and building a test apparatus for the team's propellor. If the Capstone group was working at a company to build the aircraft it could be assumed that the engineers would be paid at \$40 an hour. It should also be assumed that the engineers would have healthcare and benefits which would cost about 60% of their salary, or an additional \$24 per hour. Assuming the engineers

would work the same amount of hours as the Capstone group, the total cost of engineers would be \$69,120. The student who worked on the test apparatus for the propeller would be treated as a consultant as they are not part of the Capstone group. The price of a consultant is estimated to be about \$150 per hour, which would be \$9,000 in total. That brings the total cost of labor to \$78,120, Figure 10.5.

The cost of materials and manufacturing can be calculated by looking at the bill of materials, Table E-3, and resources used at the Makerspace, Table E-4. Assuming that an engineering firm would have to buy the materials and equipment that the Capstone group already had or was donated, those costs would have to be added to the total cost. If the cost of the donated or reused materials is counted as part of the total cost of materials, the cost goes from \$343.45, Figures 10.1 and 10.2, to \$1,530.18, Figure 10.6. The Makerspace provides students resources for free so the cost of the Makerspace would have to be estimated and also added on to the total cost. It was estimated that the cost of the Makerspace laser cutter or 3D printer would be about \$35 per hour, and they were used for about 71 hours which brings the cost of manufacturing to \$2,485, Figure 10.4. By combining the cost of the 3 factors, the total cost of the aircraft comes out to \$82,125.18, Figure 10.3.

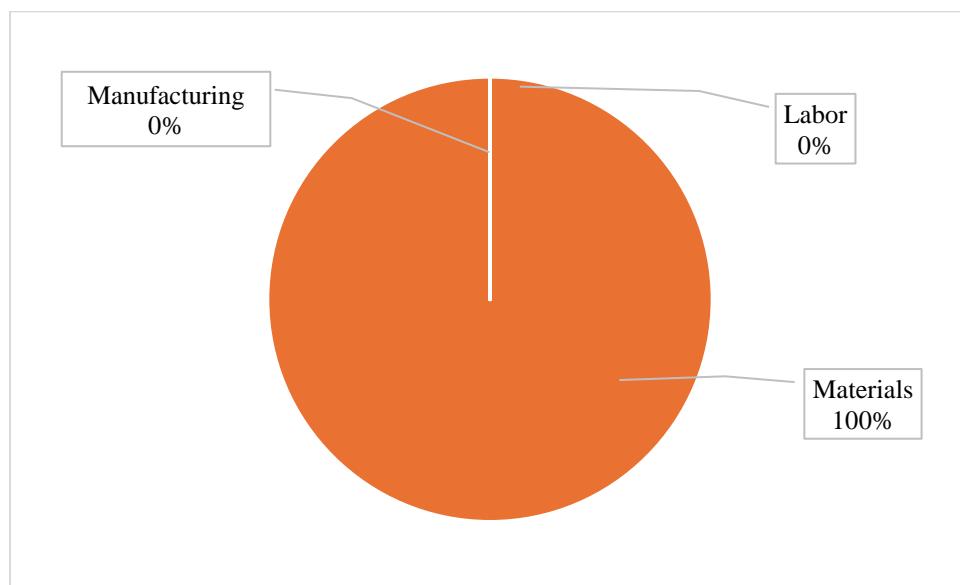


Figure 10.1 Percent Cost for Capstone Group

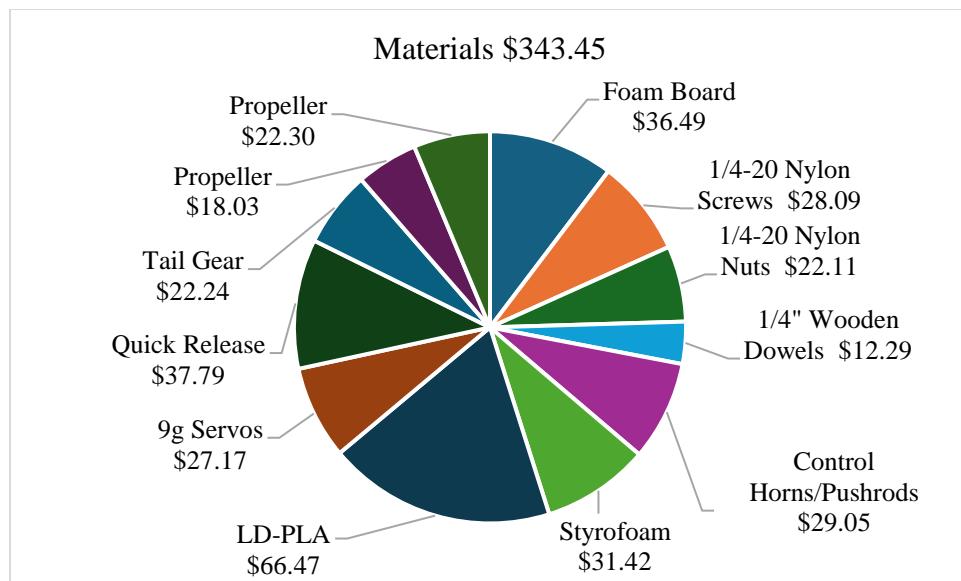


Figure 10.2 Cost of Materials for Capstone Group

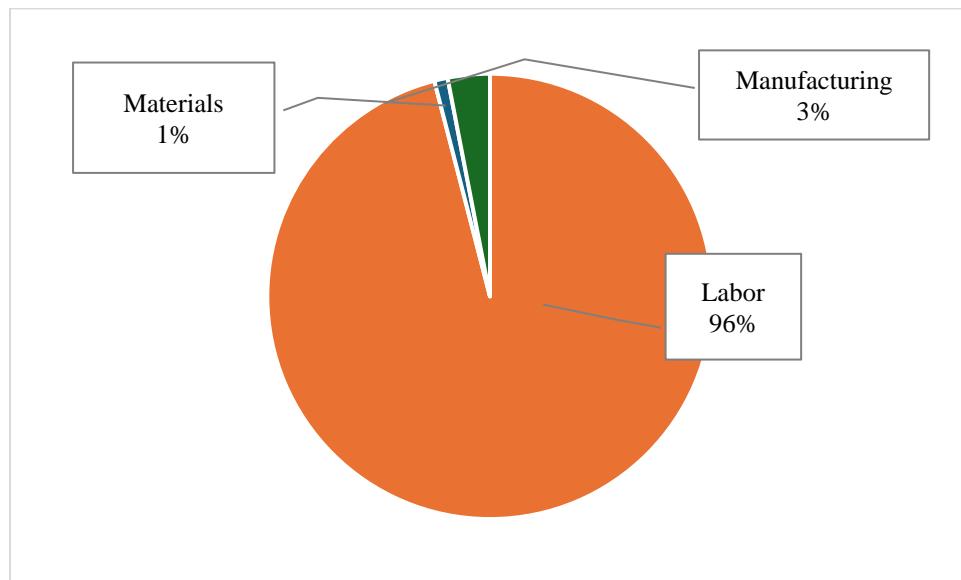


Figure 10.3 Percent Cost of Engineering Firm

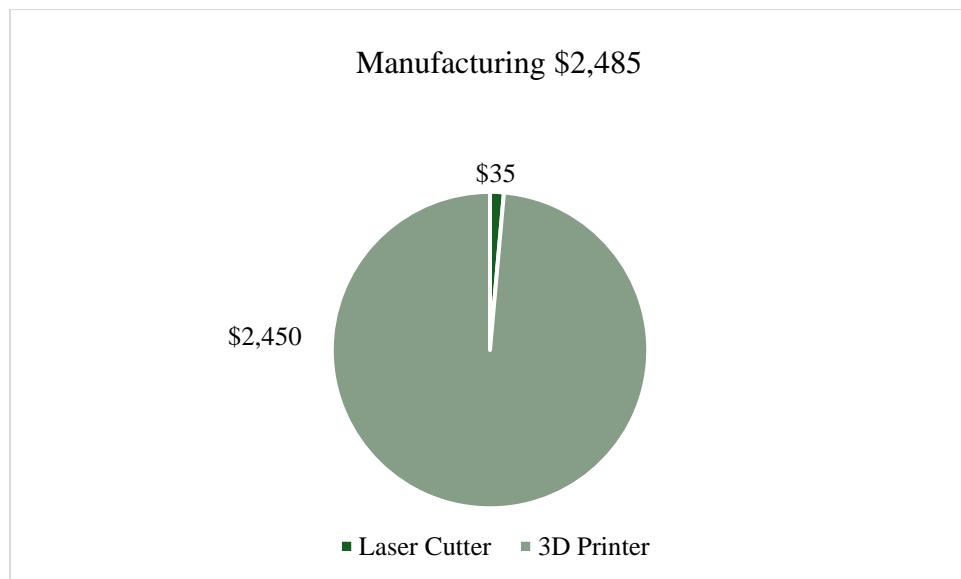


Figure 10.4 Cost of Manufacturing for Engineering Firm

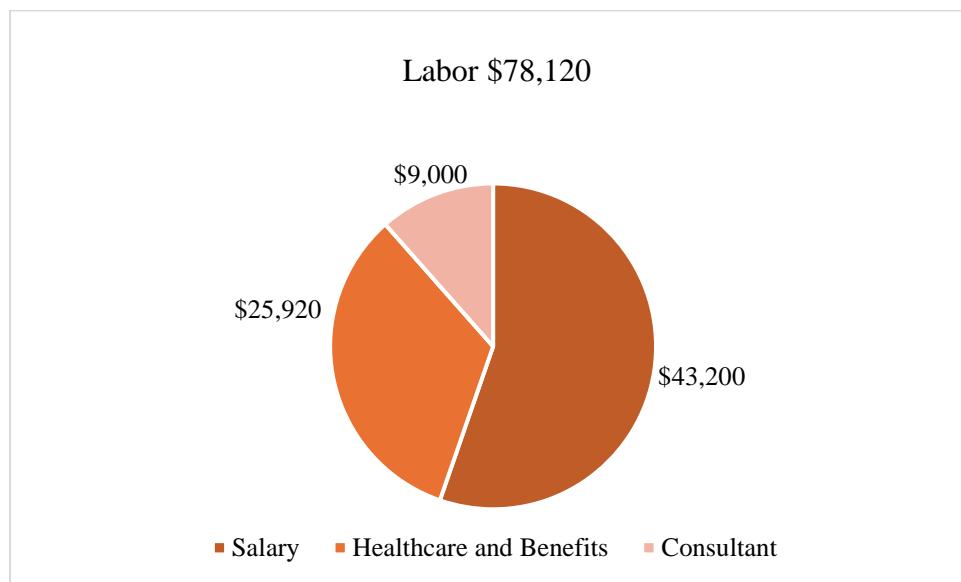


Figure 10.5 Cost of Labor for Engineering Firm

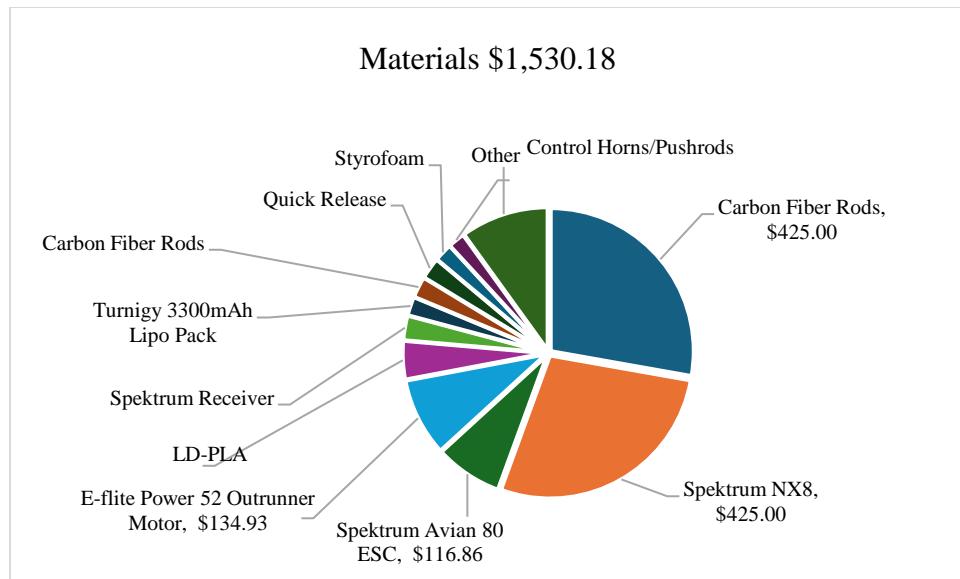


Figure 10.6 Cost of Materials for Engineering Firm

11 Broader Impacts

The team's final design solution involved anticipating and considering broader impacts on a global scale, as well as impacts on individuals and communities. The four main factors considered while designing were public safety, social impact, environmental impact, and economic impact. Each of factor serves in creating a design solution that best creates outcome for the environment, people, and invested parties. These impacts are shown in Table 11.1.

Table 11.1 Broader Impacts and Considerations

Broader Impacts	
Public Safety	<ul style="list-style-type: none"> - Ensured all aircraft components met appropriate safety standards and factors of safety for reliable, safe flight. - Advancements in remote aircraft technology enhance security and strengthen U.S. defense capabilities.
Social Impact	<ul style="list-style-type: none"> - Designed the aircraft for easy manufacturing and replication. - Thorough documentation of the manufacturing process and design reasoning.
Environmental Impact	<ul style="list-style-type: none"> - Focused on sustainability by using reused materials and parts as well as minimizing waste of non-recyclable materials. - Battery-operated design reduces reliance on fossil fuels, mitigating CO₂ emissions.
Economic Impact	<ul style="list-style-type: none"> - Minimal cost due to use of low-cost and reused materials. - Recognized the economic significance for engineering firms: costs, investors, and profitability considerations become critical when scaling the project while adhering to ethical design principles.

12 Summary and Conclusions

Upon completion of fabricating the final aircraft components through structural, aerodynamic, and electronics analysis, it was determined that the team successfully met the primary design considerations described by AIAA for this year's competition. The final aircraft design met the final wingspan requirements by holding a final wingspan of 70 inches as well as having a final prototype weight of 8 lbs. All load-bearing structural components, which include the wing and landing gears, more than exceeded the minimum 1.5 safety factor requirement laid out by the FAA. Results reflected that the selected Clark-Y airfoil was capable of producing a low drag coefficient per the team's desired takeoff velocities, as well as a favorable lift coefficient, lift-to-drag ratios, and cruise velocity. Stability analyses also confirmed the aircraft had favorable static and dynamic stability during all flight missions.

As a result of thrust testing, the current motor and propeller combination produced sufficient thrust but could benefit from an overall lowering of the necessary thrust to reach the design goals to reach a factor of safety of 1.5. Through servo testing, the team found an appropriate servo distribution that provided enough torque to move each control surface and met the total stored energy needs for the motors by selecting a 1200 mAh 2-cell battery. Manufacturing of the final prototype consisted of foam board construction of the wings, various carbon fiber components, as well as laser-cut wing ribs and fuselage. The final prototype suggested the team was able to successfully develop its skills in collaboration, problem-solving, iterative designing, and verification through modeling and testing.

13 Recommendations for Further Study

Although the aircraft was successful in meeting the design requirements for the competition, there are improvements that can be made. Firstly, as previously mentioned, each component was designed with safety factors implemented to account for unpredictable flight conditions and the forces associated with these conditions that might not be accounted for by the analysis that was performed. With additional flight testing, more information can be found about how the aircraft behaves in different flight conditions which would allow safety factors of different components to be reduced while still maintaining confidence that safety concerns are taken into account.

Additionally, with more financial resources provided, different materials and manufacturing techniques could be explored to further reduce the weight of the aircraft. The wing ribs and the fuselage were chosen to be made of cardboard and would likely be changed to be made of foam board to reduce the weight of the carrier plane. Removing weight from these two aspects as well as trimming down the landing gear will allow for that weight to be relocated into the bottles, helping the M2 and M3 score.

Two mission-necessary elements that were not included in the final design were the fuel tanks and the X-1 test vehicle. Because the X-1 was outsourced to the Intro to Aerospace Lab students, the mounting system was not implemented in the final design. Since the DBF project is 2 semesters long, these elements have potential to be worked on with a directed study.

14 Bibliography

- [1] T. Luckett, “Wing Cube Loading,” *RC CAD LLC*, 14-Oct-2020. [Online]. Available: <https://www.rccad2vr.com/aeronautics/wing-cube-loading>. [Accessed: 17-Sep-2024].
- [2] “14 CFR part 107 -- small unmanned aircraft systems,” *Ecfr.gov*. [Online]. Available: <https://www.ecfr.gov/current/title-14/chapter-I/subchapter-F/part-107>. [Accessed: 12-Nov-2024].
- [3] Compass: ASTM International, “ASTM F2910-22 Standard Specification for Design and Construction of a Small Unmanned Aircraft System (sUAS),” ASTM International, West Conshohocken, PA, Dec. 2022.
- [4] “47 CFR Part 15 Subpart C -- Intentional Radiators,” *Code of Federal Regulations*, 04-Jun-2019. [Online]. Available: <https://www.ecfr.gov/current/title-47/chapter-I/subchapter-A/part-15/subpart-C>. [Accessed: 01-Dec-2024].
- [5] Federal Aviation Administration (FAA), “Remote identification (Remote ID) is here. Are you ready?”
- [6] C. T. M. J. Zipay, “NASA,” *The 1.5 & 1.4 Ultimate Factors of Safety for Aircraft & Spacecraft – History, Definition and Applications*, February 2014. [Online]. Available: <https://ntrs.nasa.gov/api/citations/20140011147/downloads/20140011147.pdf>. [Accessed: 03-Dec-2024].
- [7] Compass: American Society for Testing and Materials (ASTM), “ASTM F593-24 Standard Specification for Stainless Steel Bolts, Hex Cap Screws, and Studs,” ASTM International, West Conshohocken, PA, Sep. 2024.
- [8] R. S. Shevell, Fundamentals of flight: United States edition, 2nd ed. Upper Saddle River, NJ: Pearson, 1988.
- [9] P. Barua, T. Sousa, and D. Scholz, “Empennage Statistics and Sizing Methods for Dorsal Fins,” Apr. 2013.
- [10] R. Elliott, “Deflection of Beams,” *Org.uk*, 30-Dec-2010. [Online]. Available: <http://www.clag.org.uk/beam.html>. [Accessed: 06-Dec-2024].
- [11] A. Ugural and S. Fenster, Advanced Mechanics of Materials and Applied Elasticity, 6th ed. Upper Saddle River, NJ: Pearson, 2019.
- [12] H. Ritchie, “What share of global CO₂ emissions come from aviation?,” *Our World in Data*, 08-Apr-2024. [Online]. Available: <https://ourworldindata.org/global-aviation-emissions#:~:text=The%20gains%20in%20efficiency%20have,that%20was%20around%201%20billion>. [Accessed: 15-Dec-2024].
- [13] “DBF Feature,” www. [Online]. Available: <https://www.aiaa.org/dbf>. [Accessed: 10-Sep-2024].

15 Appendices

Appendix A – Complete Drawings

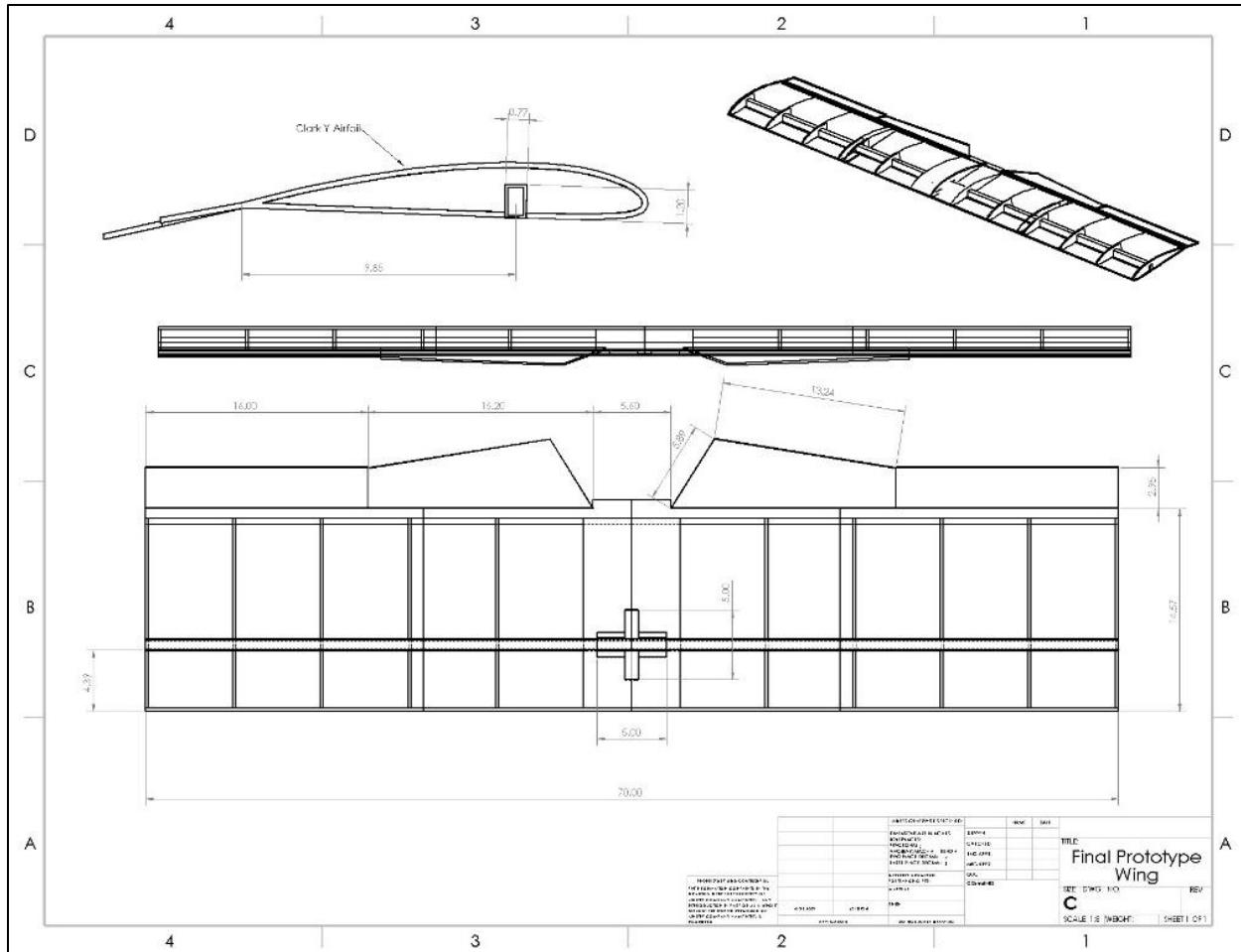


Figure A-1. Full Wing

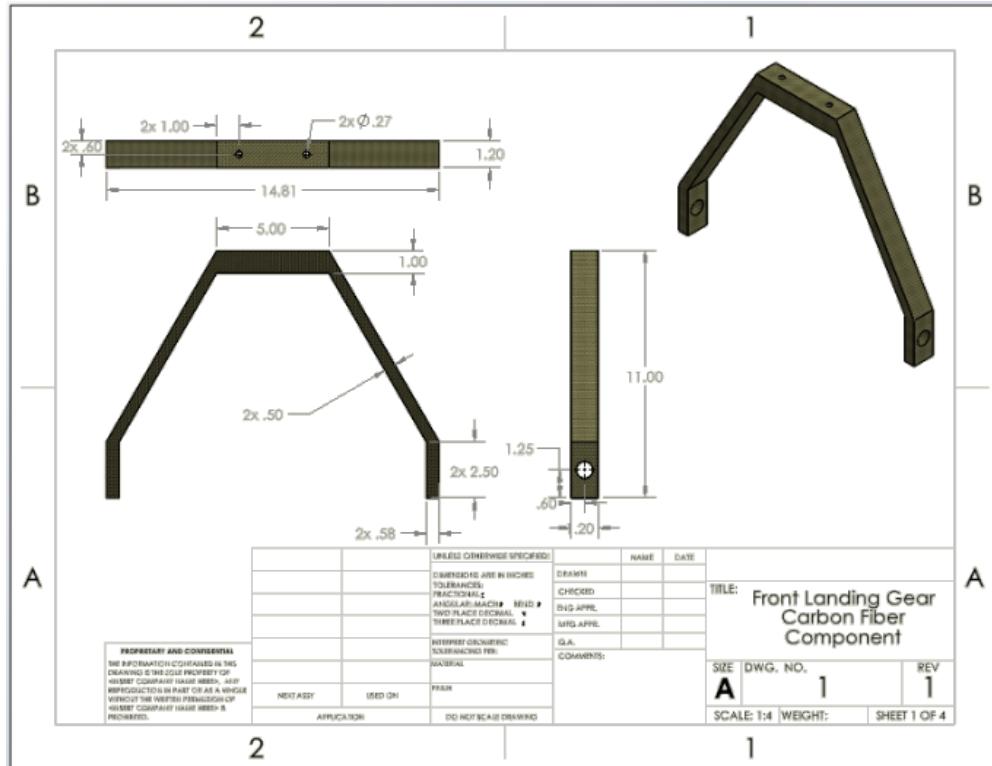


Figure A-2. Front Landing Gear

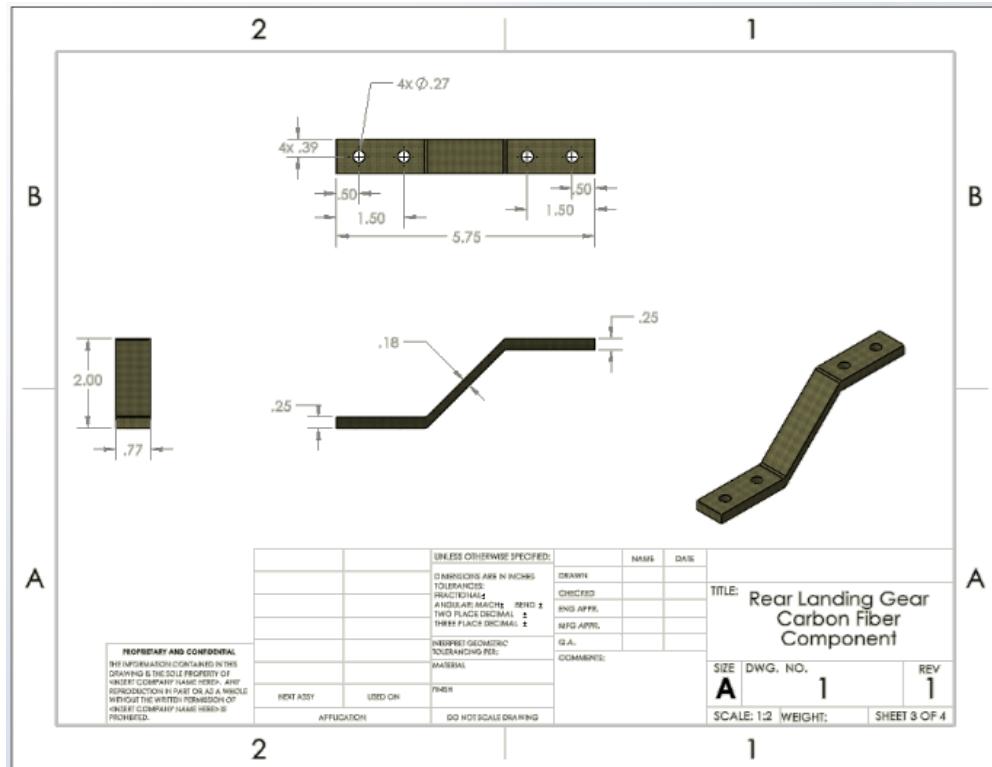


Figure A-3. Rear Landing Gear

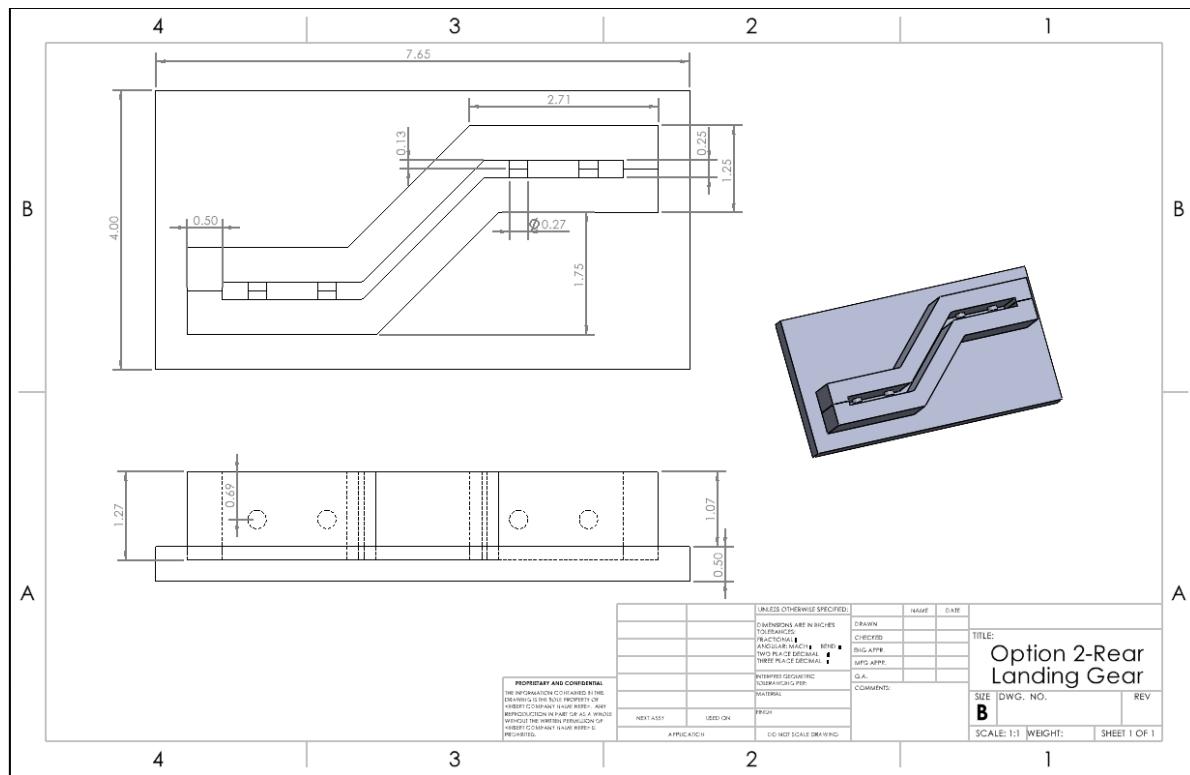


Figure A-4. Option 2 Rear Landing Gear Mold

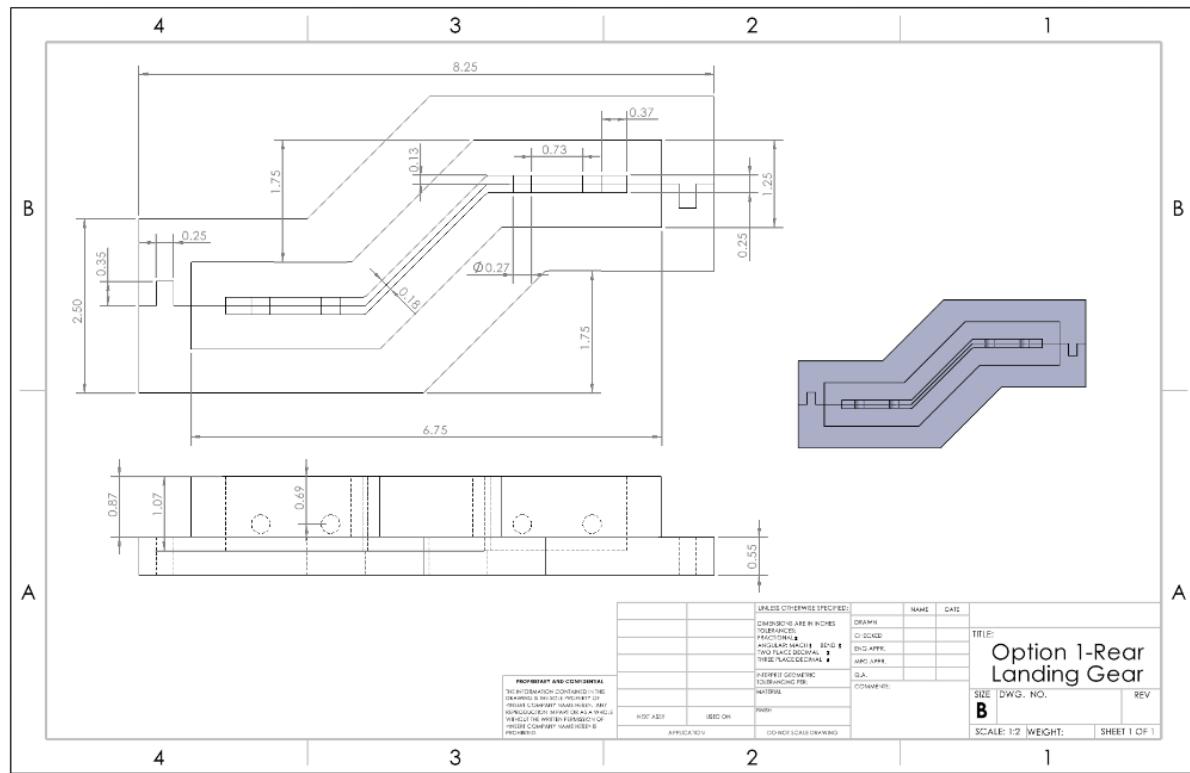


Figure A-5. Option 1 Rear Landing Gear Mold

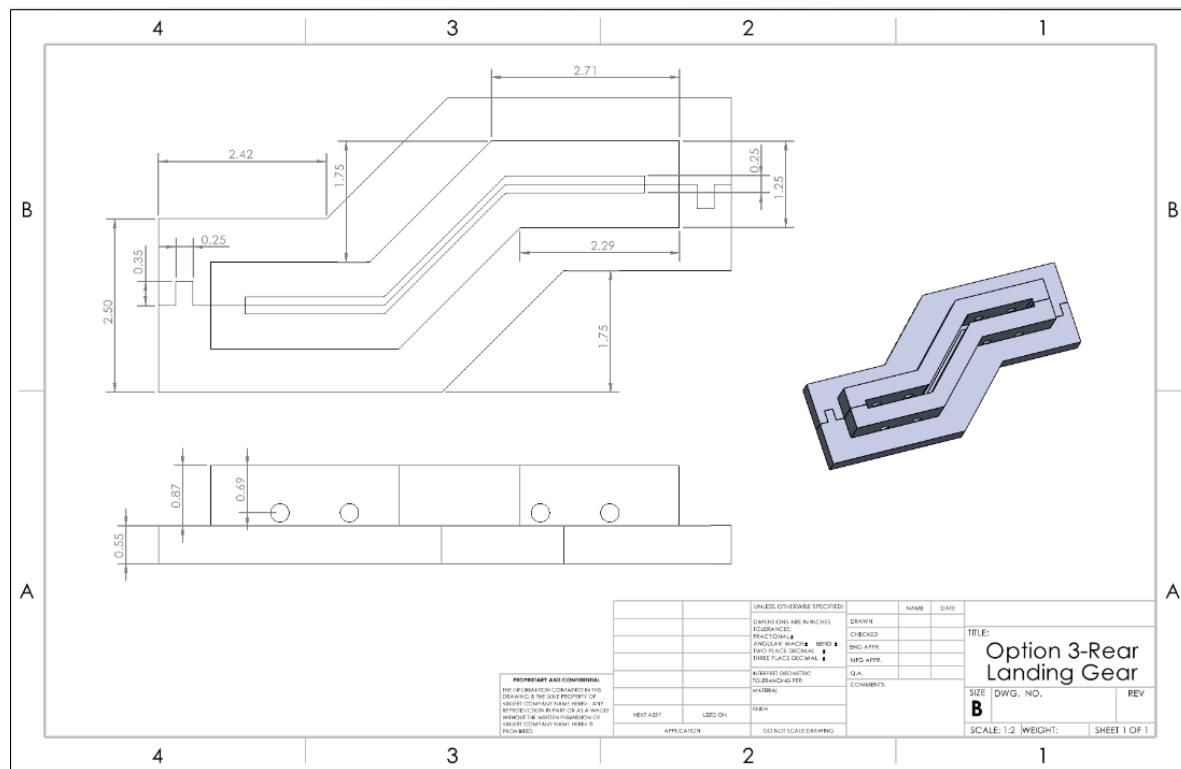


Figure A-6. Option 3 Rear Landing Gear Mold

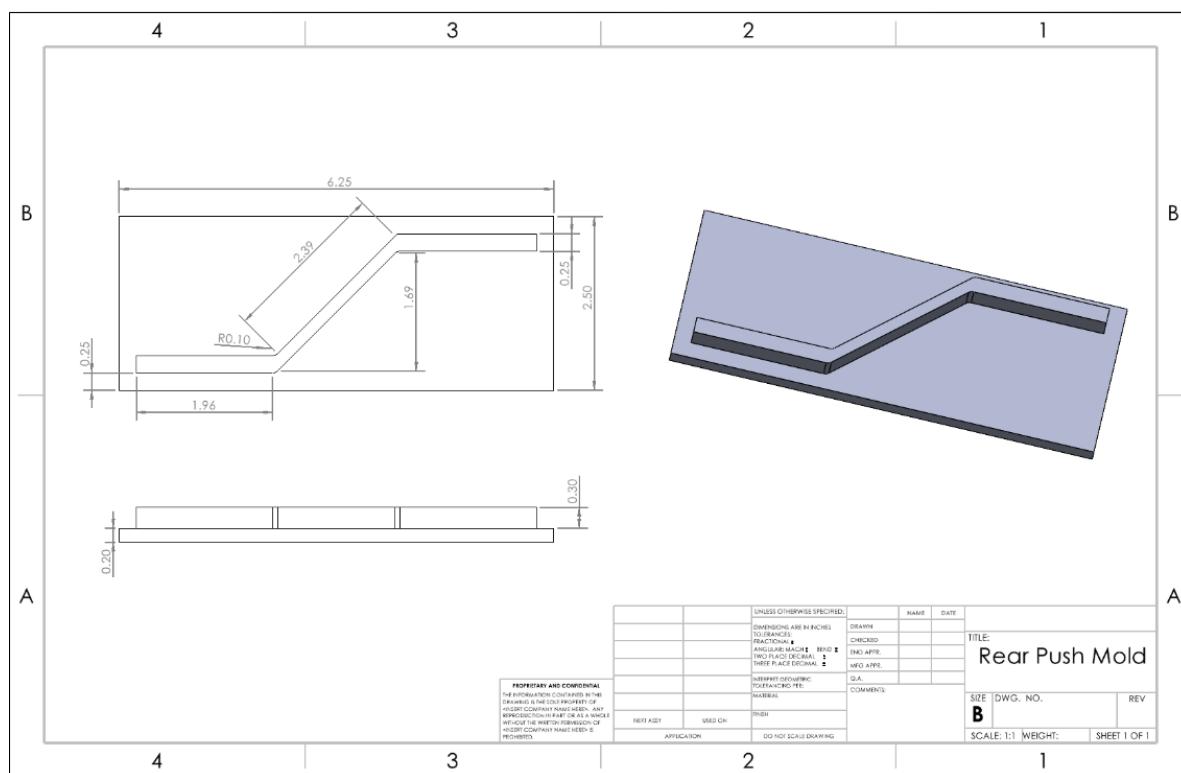


Figure A-7. Rear Push-Down Mold

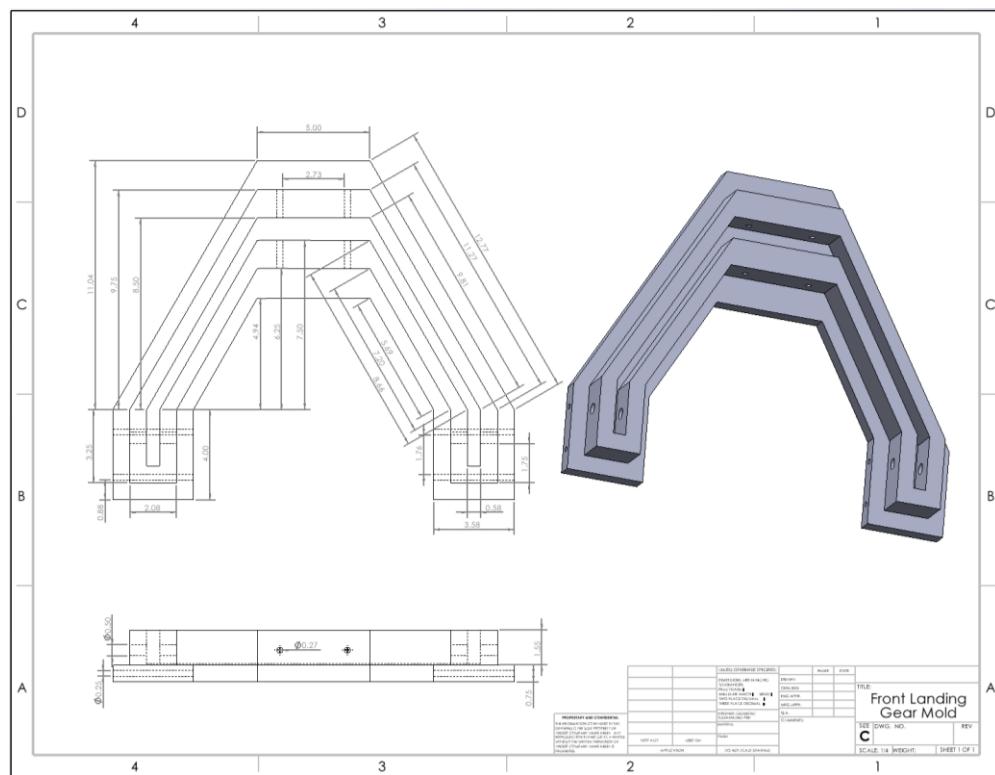


Figure A-8. Front Landing Gear Mold

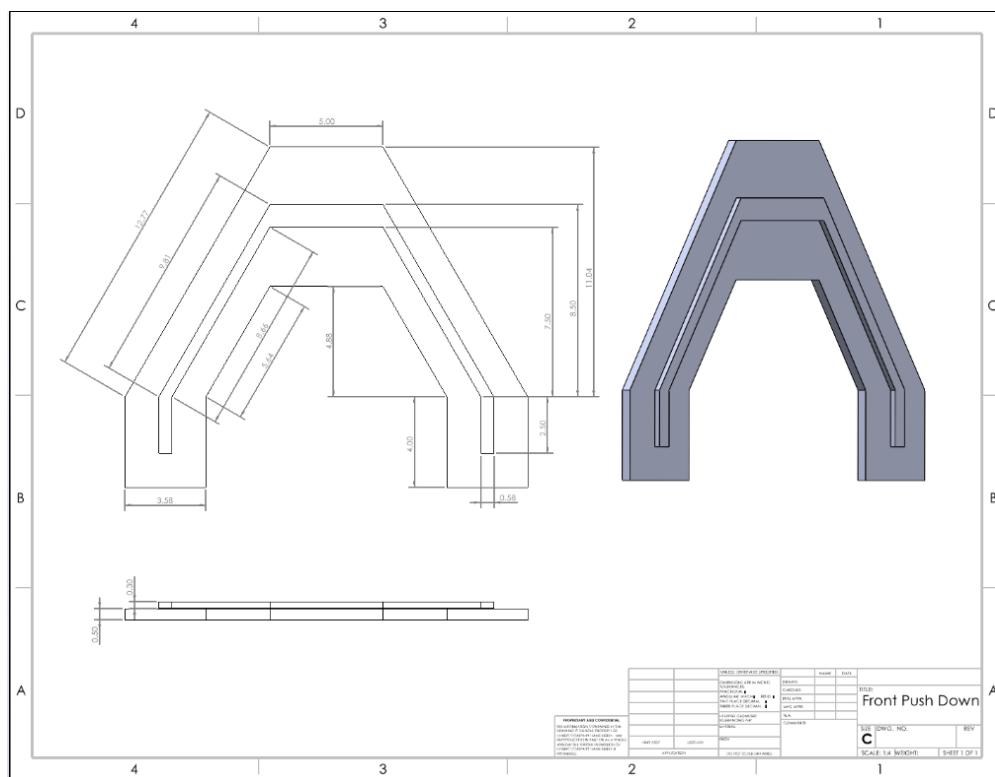


Figure A-9. Front Push Down Mold

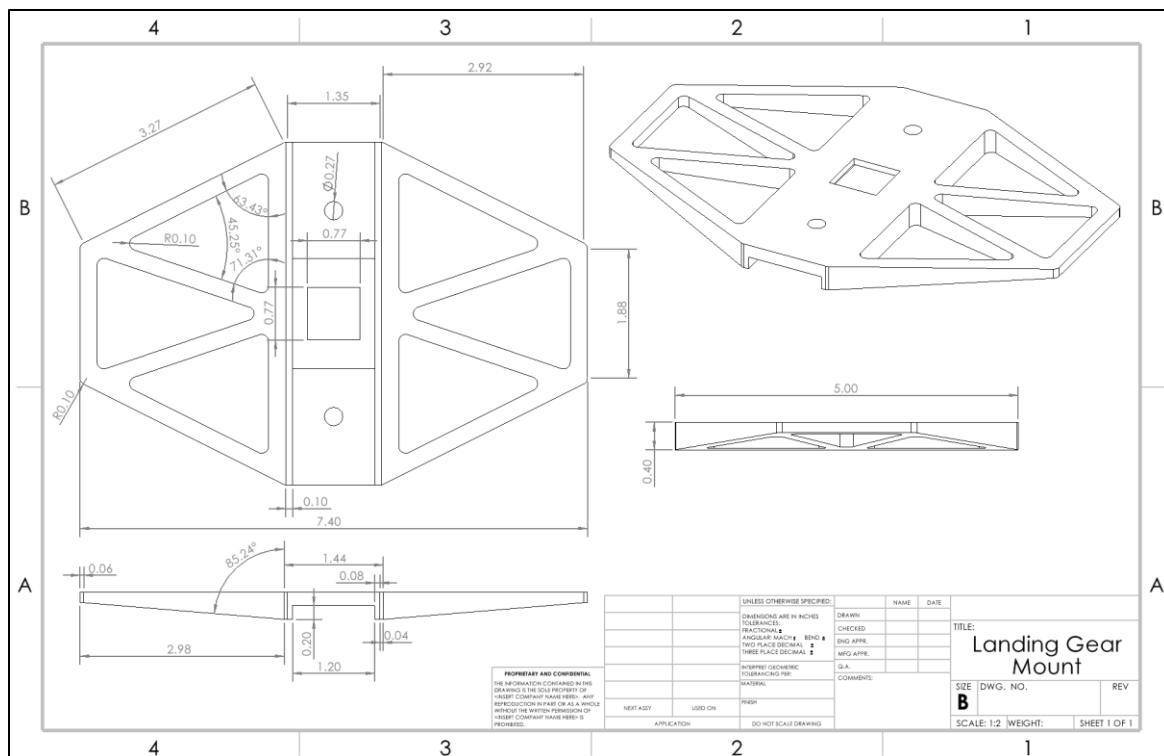


Figure A-10. Landing Gear Mount

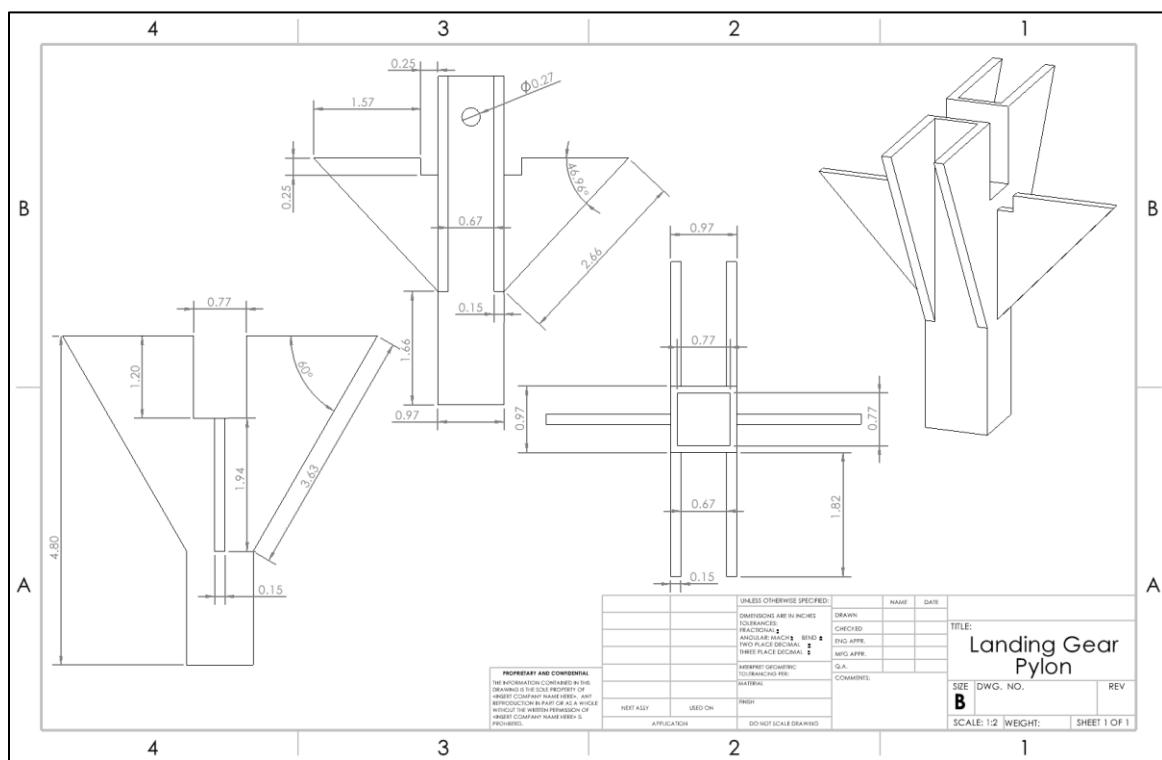


Figure A-11. Landing Gear Pylon

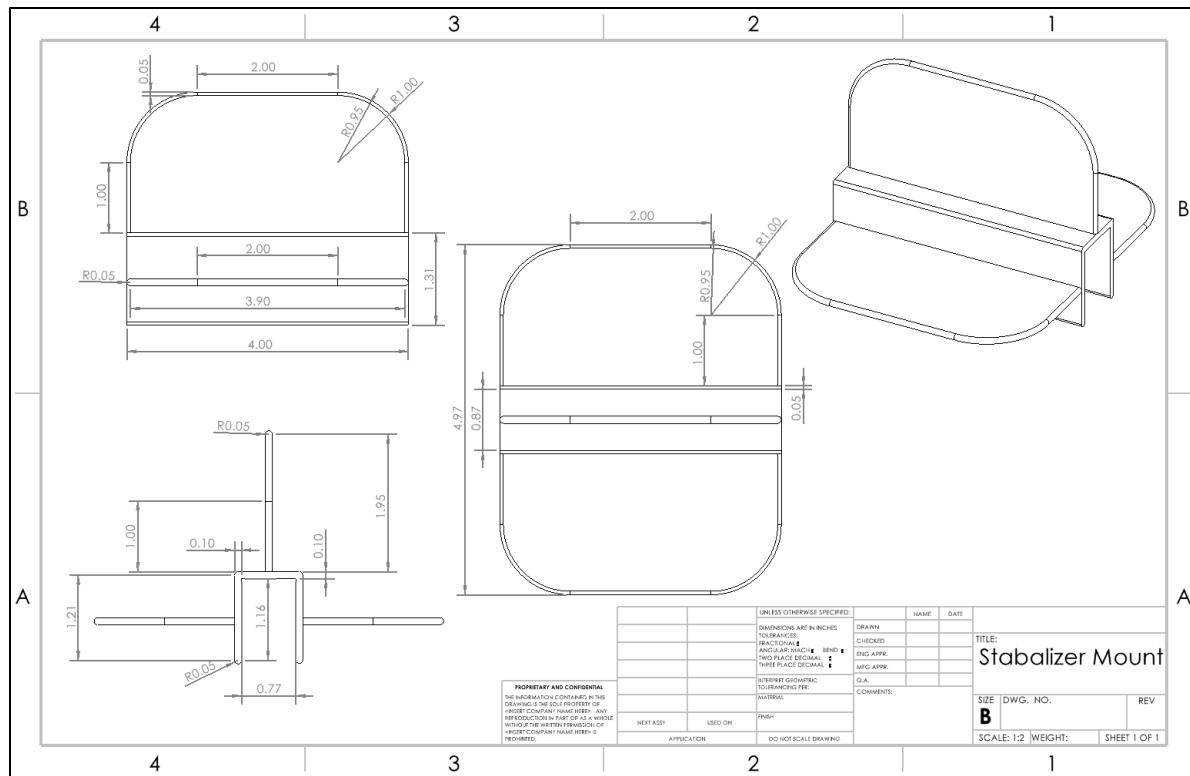


Figure A-12. Stabilizer Mount

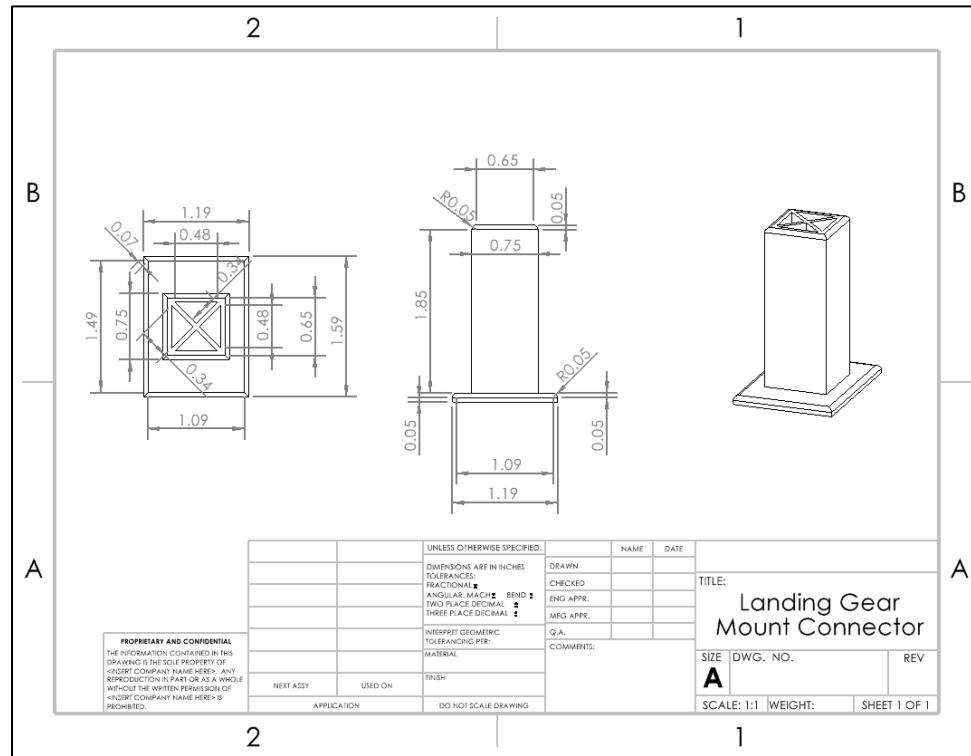


Figure A-13. Landing Gear Mount Connector

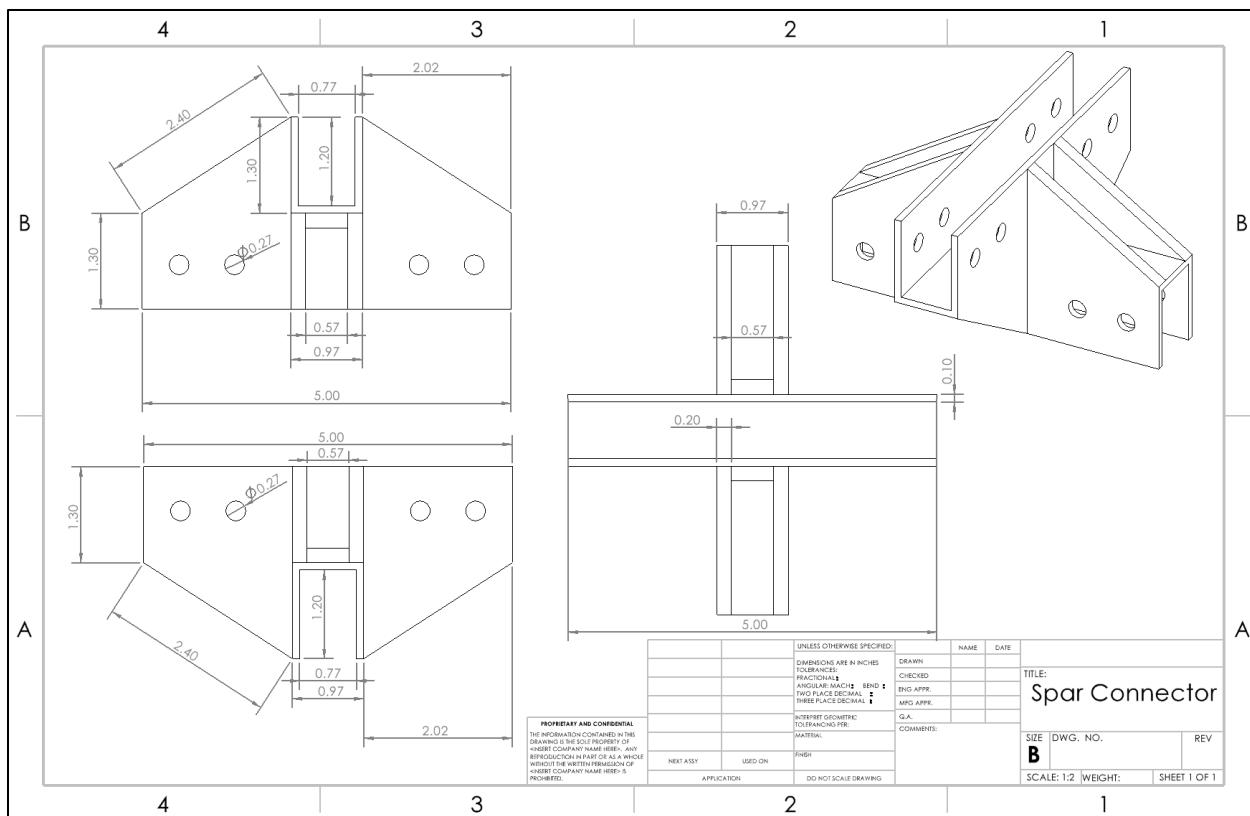


Figure A-14. Spar Connector

Appendix B – Landing Gear Final Design Solution



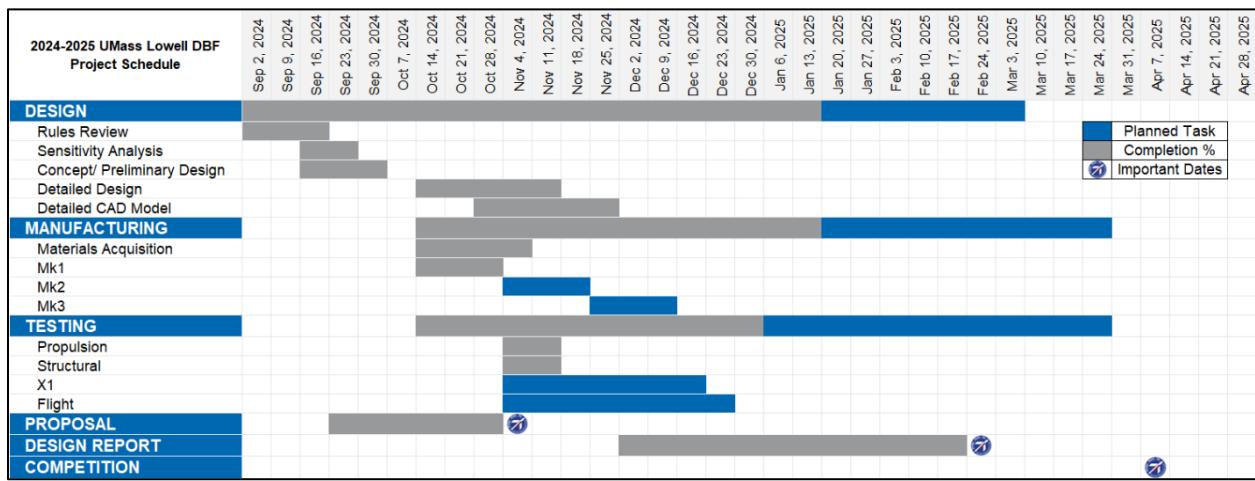
Figure B-1. Front Landing Gear



Figure B-2. Rear Landing Gear

Appendix C – Gantt Chart

Table C-1. Gantt Chart



Appendix D – Programming Code

D-1: Sensitivity Analysis:

```
% Defining the parameters of the best performing team
bestTeam.fuel = 20; % [lbf] Payload weight used by the best performing team
bestTeam.lapsTime = 60; % [s] Mission 2 lap time of the best performing team
bestTeam.laps = 15; % Number of laps flown in Mission 3 by the best performing team
bestTeam.bonus = 2.5; % Number of bonus points earned in Mission 3 by the best performing team
bestTeam.X1Weight = .2; % Weight of X1 vehicle in lbs.
%bestTeam.passengers = 20; % Number of passengers carried by the best performing team in
Mission 3
%bestTeam.batteryCapacity = 25; % [W-h] Battery capacity of the best performing team
bestTeam.GM = 60; % [s] Time taken to complete Ground Mission

% Defining parameters for our team
team = bestTeam;

% Maximum mission scores based on the formulas given in the rulebook
score.Max.M1 = 1;
%score.Max.M2 = 1 + (team.payload / team.lapTime) / (bestTeam.payload / bestTeam.lapTime);
score.Max.M2 = 1 + (team.fuel/team.lapsTime) / (bestTeam.fuel/bestTeam.lapsTime);
%score.Max.M3 = 2 + ((team.laps * team.passengers) / team.batteryCapacity) / ((bestTeam.laps *
bestTeam.passengers) / bestTeam.batteryCapacity);
score.Max.M3 = 2 + (team.laps + (team.bonus/team.X1Weight)) / (bestTeam.laps +
(bestTeam.bonus/bestTeam.X1Weight));
score.Max.GM = bestTeam.GM / team.GM;
score.Max.Total = score.Max.M1 + score.Max.M2 + score.Max.M3 + score.Max.GM;

% Total Score Sensitivity Analysis
% Vary each parameter from -50% to 50% in 10% increments
percentChange = -0.5:0.1:0.5;

% Impact of Mission 2 score on Total Score
score.Total.M2 = score.Max.M1 + (score.Max.M2 * (1 + percentChange)) + score.Max.M3 +
score.Max.GM;
score.Change.M2 = ((score.Total.M2 - score.Max.Total) / score.Max.Total) * 100;

% Impact of Mission 3 score on Total Score
score.Total.M3 = score.Max.M1 + score.Max.M2 + (score.Max.M3 * (1 + percentChange)) +
score.Max.GM;
score.Change.M3 = ((score.Total.M3 - score.Max.Total) / score.Max.Total) * 100;

% Impact of Ground Mission on Total Score
score.Total.GM = score.Max.M1 + score.Max.M2 + score.Max.M3 + (score.Max.GM * (1 +
percentChange));
score.Change.GM = ((score.Total.GM - score.Max.Total) / score.Max.Total) * 100;

% Plotting Total Score Sensitivity Analysis
plots.Y1.Name = {'Mission 2 Score', 'Mission 3 Score', 'Ground Mission Score'};
plots.Y1.Value = [score.Change.M2', 'score.Change.M3', 'score.Change.GM'];

% The sensitivity of the score is directly related to the slope of the
% mission, greater slope = greater sensitivity
figure()
plot((percentChange * 100), plots.Y1.Value)
legend(plots.Y1.Name)
```

```

xlabel('Change in input variable (in %)')
ylabel('Change in mission score (in %)')
title('Total Mission Score Analysis')

% Mission 2 Sensitivity Analysis
% Creating lap time vector of the team
timeInterval = 50;
team.M2.lapsTime = [linspace(bestTeam.lapsTime, 300, timeInterval)]';
team.M2.fuel = (4:4:20) .* ones(timeInterval, 5);
(4:4:20) .* ones(timeInterval, 5);
score.M2 = 1 + ((team.M2.fuel ./ team.M2.lapsTime) / (bestTeam.fuel / bestTeam.lapsTime));

% Plotting Mission 2 Sensitivity Analysis
plots.Y2.Name = {'4 lbs', '8 lbs', '12 lbs', '16 lbs', '20 lbs'};
plots.Y2.Value = score.M2;

figure()
plot(team.M2.lapsTime, plots.Y2.Value);
legend(plots.Y2.Name);
xlabel('Total Time for 3 Laps [s]');
ylabel('Mission 2 Score');
title('Mission 2 Score Analysis');

% Mission 3 Sensitivity Analysis
% Number of Laps and X1 Weight
team.M3.X1Weight = [linspace(bestTeam.X1Weight, .55, 20)]';
team.M3.laps = (4:4:16) .* ones(20, 4);
score.M3.weightToLaps = 2 + (team.M3.laps + (team.bonus ./ team.M3.X1Weight)) ./
(bestTeam.laps + (bestTeam.bonus ./ bestTeam.X1Weight));

% Plotting Mission 3 Sensitivity Analysis of Number of Laps to
% X1 Weight Correlation
plots.Y3.Name = {'4 Laps', '8 Laps', '12 Laps', '16 Laps', '20 Laps'};
plots.Y3.Value = score.M3.weightToLaps;

figure()
plot(team.M3.X1Weight, plots.Y3.Value);
legend(plots.Y3.Name);
xlabel('X1 Weight');
ylabel('Mission 3 Score');
title('X1 Weight to Number of Laps Correlation');

% Number of Laps and Bonus Score Correlation
team.M3.bonus = [0, 1, 2.5]';
team.M3.laps = (4:4:16) .* ones(3, 4);
score.M3.bonusPointsToLaps = 2 + (team.M3.laps + (team.M3.bonus ./ team.X1Weight)) ./
(bestTeam.laps + (bestTeam.bonus ./ bestTeam.X1Weight));

% Plotting Mission 3 Sensitivity Analysis of Number of Laps to Battery Capacity Correlation
plots.Y4.Name = {'4 Laps', '8 Laps', '12 Laps', '16 Laps', '20 Laps'};
plots.Y4.Value = score.M3.bonusPointsToLaps;

figure()
plot(team.M3.bonus, plots.Y4.Value);
legend(plots.Y4.Name);
xlabel('Bonus Points');
ylabel('Mission 3 Score');
title('Bonus Points to Number of Laps Correlation');

% Bonus Points and X1 Correlation
team.laps = 5;

```

```
[X, Y] = meshgrid(team.M3.bonus, team.M3.X1Weight);
score.M3.bonusToX1Weight = 2 + (team.laps + (X ./ Y)) ./ (bestTeam.laps + (bestTeam.bonus ./ bestTeam.X1Weight));
%2 + (team.M3.laps + (team.bonus ./ team.M3.X1Weight)) ./ (bestTeam.laps + (bestTeam.bonus ./ bestTeam.X1Weight));
%Plotting Mission 3 Sensitivity Analysis of Number of Laps to Battery Capacity Correlation
plots.Y5.Value = score.M3.bonusToX1Weight;

figure()
surface(X,Y,score.M3.bonusToX1Weight);

view(3)
xlabel('Bonus Points [W-h]');
ylabel('X1 Weight');
zlabel('Mission 3 Score');
title('Bonus Points to X1 Weight Correlation');
colorbar
```

D-2: Location of Tail

```
clc, clear all, clear vars

% Wing
Wing_Chord_Length = 13.68; % inches
WingQuarter_Chord = (0.25) * (Wing_Chord_Length) % inches

% Vertical Stabilizer
Vertical_Stabilizer_Chord_Length = 5.25; % inches
Vertical_StabilizerQuarter_Chord = (0.25) * (Vertical_Stabilizer_Chord_Length) % inches

% Horizontal Stabilizer
Horizontal_Stabilizer_Chord_Length = 6.75; % inches
Horizontal_StabilizerQuarter_Chord = (0.25) * (Horizontal_Stabilizer_Chord_Length) % inches

% Plane
Distance_BetweenQuarter_Chords = 40; % inches

% To find the distance between the leading edge of the main wing and the
% leading edge of the horizontal stabilizer, take the distance between the
% quarter chords and then add the wing quarter chord and horizontal quarter
% chord dimensions. Do the same but use the vertical stabilizer quarter
% chord dimension to find the distance between the leading edge of the wing
% and leading edge of the vertical stabilizer

% Leading edge of wing to leading edge of horizontal stabilizer distance (for XFLR5)
Wing_to_Horizontal_Stabilizer_Distance = (Distance_BetweenQuarter_Chords) +
(Vertical_StabilizerQuarter_Chord) + (WingQuarter_Chord)

% Approximately 45"

% Leading edge of wing to leading edge of horizontal stabilizer distance (for XFLR5)
Wing_to_Vertical_Stabilizer_Distance = (Distance_BetweenQuarter_Chords) +
(Vertical_StabilizerQuarter_Chord) + (WingQuarter_Chord)

% Approximately 45"
```

D-3: Control Surface and Empennage Sizing

```

clc, clear all, clear vars

%% Wing Dimensions

% Imperial
Wing_Span_in = 70; % in
Chord_Length_in = 13.68; % in
Wing_Area_in_squared = (Wing_Span_in)*(Chord_Length_in); % in^2

% Metric
Wing_Span_meters = (Wing_Span_in)/(39.37); % m
Chord_Length_meters = (Chord_Length_in)/(39.37); % m
Wing_Area_meters_squared = (Wing_Span_meters)*(Chord_Length_meters); % m^2

%% Horizontal Stabilizer Dimensions

% Note: The selected size for the Horizontal Stabilizer span and area was
% determined initially from the preliminary sizing done in excel looking
% at previous years designs and computing an average. The value was then
% increased accordingly based on a linear proportion between this years and
% last year's plane. A more "rigorous" method of computing the horizontal
% stabilizer dimensions can be found later in the code by using Raymers
% tail coefficients

% Imperial

Horizontal_Stabilizer_Span_in = 29.50; % in
Horizontal_Stabilizer_Area_in_squared = 177.00; % in^2

% Metric

Horizontal_Stabilizer_Span_meters = (Horizontal_Stabilizer_Span_in)/(39.37); % m
Horizontal_Stabilizer_Area_meters_squared = (Horizontal_Stabilizer_Area_in_squared)/(1550); % m^2

% Horizontal Stabilizer Chord Length

Horizontal_Stabilizer_Chord_Length_inches =
(Horizontal_Stabilizer_Area_in_squared)/(Horizontal_Stabilizer_Span_in); % in

Horizontal_Stabilizer_Chord_Length_meters =
(Horizontal_Stabilizer_Area_meters_squared)/(Horizontal_Stabilizer_Span_meters); % m

% _____
% From Historical Data/Excel Analysis (I will not be using these numbers)

% Horizontal_Stabilizer_Span_in = 21.82;
% Horizontal_Stabilizer_Area_in_squared = 199.00;
% Horizontal_Stabilizer_Span_meters = 0.554228;
% Horizontal_Stabilizer_Area_meters_squared = 0.128387;

% _____
%% Vertical Stabilizer Dimensions

```

```

% Note: The selected size for the vertical stabilizer span and area was
% determined initially from the preliminary sizing done in excel looking
% at previous years designs and computing an average. The value was then
% increased accordingly based on a linear proportion between this years and
% last year's plane. A more "rigorous" method of computing the vertical
% stabilizer dimensions can be found later in the code by using Raymers
% tail coefficients

% Imperial

Vertical_Stabilizer_Span_in = 13.00;
% inches
Vertical_Stabilizer_Area_in_squared = 68.25;
% inches^2

% Metric

Vertical_Stabilizer_Span_meters = (Vertical_Stabilizer_Span_in)/(39.37);
% meters
Vertical_Stabilizer_Area_meters_squared = (Vertical_Stabilizer_Area_in_squared)/(1550);
% meters^2

% Vertical Stabilizer Chord Length

Vertical_Stabilizer_Chord_Length_inches =
(Vertical_Stabilizer_Area_in_squared)/(Vertical_Stabilizer_Span_in);

Vertical_Stabilizer_Chord_Length_meters =
(Vertical_Stabilizer_Area_meters_squared)/(Vertical_Stabilizer_Span_meters);

% _____
_____
% From Historical Data/Excel Analysis (I will not be using these numbers)

% Imperial

% Vertical_Stabilizer_Span_in = 9.49; % inches
% Vertical_Stabilizer_Area_in_squared = 71.86; % inches^2

% Metric

% Vertical_Stabilizer_Span_meters = 0.241048; % meters
% Vertical_Stabilizer_Area_meters_squared = 0.046361198; % meters^2

% _____
_____
%% Control Surface Sizing

% For flaps and ailerons, we are going to use sizing percent composition
% from previous DBF years as well as previous UML DBF years (notably 2023-2024) given proven
flight success
% We will bias the size of the flaps and ailerons slightly given the significance of high lift
% devices (flaps) in generating sufficient lift for takeoff this year.

% Flaps

% Total Flap Span (Divide this by 2 to get the span of the individual flaps)

```

```

Flaps_Span_in = (46.00/100)*(Wing_Span_in);
Flap_Span_meters = (46.00/100)*(Wing_Span_meters);

% Single Flap Span (LOOK AT ME)
Single_Flap_Span_in = (Flaps_Span_in)/2;
Single_Flap_Span_meters = (Flap_Span_meters)/2;

% Flap Chord Length (LOOK AT ME)
Flap_Chord_Length_in = (15.0/100)*(Chord_Length_in);
Flap_Chord_Length_meters = (15.0/100)*(Chord_Length_meters);

% Single Flap Surface Area
Single_Flap_Surface_Area_in_squared = (Single_Flap_Span_in) * (Flap_Chord_Length_in);
Single_Flap_Surface_Area_meters_squared = (Single_Flap_Span_meters) *
(Flap_Chord_Length_meters);

% Total Flap Surface Area

Total_Flap_Surface_Area = (2) * (Single_Flap_Surface_Area_in_squared);

%_____
% 2024-2025 Percent Make Up

% Flap Span = 46.00 % (of wingspan)
% Flap Chord Length = 15 % (of wing span)
%_____
_____
%_____
% Ailerons

% Total Aileron Span (Divide this by 2 to get the span of the individual ailerons)
Aileron_Span_in = (45/100)*(Wing_Span_in);
Aileron_Span_meters = (45/100)*(Wing_Span_meters);

% Single Aileron Span (LOOK AT ME)
Single_Aileron_Span_in = (Aileron_Span_in)/2;
Single_Aileron_Span_meters = (Aileron_Span_meters)/2;

% Aileron Chord Length (LOOK AT ME)
Aileron_Chord_Length_in = (15.0/100)*(Chord_Length_in);
Aileron_Chord_Length_meters = (15.0/100)*(Chord_Length_meters);

% Single Aileron Surface Area

Single_Aileron_Surface_Area = (Single_Aileron_Span_in) * (Aileron_Chord_Length_in);

% Total Aileron Surface Area

Total_Aileron_Surface_Area = (Single_Aileron_Surface_Area) * (2);

%_____
_____

```

% 2024-2025 Percent Make Up

```
% Aileron Span = 45.00 % (of wingspan)
% Aileron Chord Length = 15.00 % (of wing span)
%
```

%% Empennage Sizing

```
% The goal of this section is to determine the horizontal and vertical tail
% size for the plane. Raymer "tail volume
% coefficients" were utilized to determine appropriate tail size.
%
```

% 3.1 Horizontal Tail Volume Coefficient

```
% Column 1 is the type of plane
% Column 2 is the statistics carried about by the research group
% Column 3 is the suggested tail volume coefficient (this is where you pick
% your value)
```

% Pg 71 Table 3.1 | Suggested Values for tail volume coefficient of horizontal tail (C_H)

% For civil props -> GA - Single Engine -> Avg. = 0.672

C_H = 0.672;

%

% 3.2 Vertical Tail Volume Coefficient

```
% Pg 73 Table 3.2 | Suggested values for tail volume coefficients of
% vertical tail (C_v)
```

% For civil props -> GA - Single Engine -> Avg. = 0.0443

C_v = 0.0443;

%

% Determining the Horizontal and Vertical Stabilizer Surface Area

% Vertical Stabilizer

```
% To do this we need to know what the distance between the quarter chord of
% the wing and both the horizontal and vertical stabilizer is. The equation
% outlined in Raymer and used in the 2023-2024 report below was used to
% determine this distance
```

% From Page 14 and 15| Coefficient of Tail Volume

% $C_v = ((L_v) * (S_v)) / ((S_w) * (b_w))$

% Where

```
% C_v = Coefficient of Tail Volume (Non-Dimensional) | Defined
% L_v = Tail Lever Arm Length (Vertical) | Defined
% S_v = Vertical Stabilizer Area | SOLVE FOR
```

```

% b_w = Wing Span | Defined
% S_w = Wing Area | Defined

% To solve the equation for S_v, we need to know what the tail lever arm
% length is. This value was estimated in conversation based on the 2024-2025 DBF Prototype
% built, the distance
% between the quarter chord of the main wing and the quarter chord of both the horizontal and
% vertical stabilizers were approximately 40". This value will be tested
% and modified if needed.

% Imperial

L_v_in = 40.00; % in (ORIGINALLY 40")
b_w_in = Wing_Span_in; % in^2
S_w_in_squared = Wing_Area_in_squared; % in^2

S_v_inches_squared = ((S_w_in_squared * b_w_in)/(L_v_in))*(C_v); % Vertical Stabilizer Area

% Raymer Vs Preliminary Vertical Stabilizer Surface Area (Percent
% Difference)

Vertical_Stabilizer_Area_Percentage_Over_Preliminary = (1 -
(Vertical_Stabilizer_Area_in_squared/ S_v_inches_squared)) * 100;

%_____
_____

% Now that the surface area is known, we need to be able to find
% appropriate values for the span (height) and chord length of the vertical stabilizer

% The preliminary aero analysis had the vertical stabilizer span (height)
% at 13 inches. Increasing the height by 1.125" multiplied by the same
% chord length from preliminary gets extremely close to the surface area
% specified by Raymer

New_Vertical_Stabilizer_Surface_Area = S_v_inches_squared; % in^2
New_Vertical_Stabilizer_Chord_Length = 5.25; % in
New_Vertical_Stabilizer_Span =
(New_Vertical_Stabilizer_Surface_Area)/(New_Vertical_Stabilizer_Chord_Length);

% Rudder Chord Length and Surface Area

% Based on historical data and previous UML DBF years (notably 2023-2024) analysis, the rudder
% chord length was sized to be
% 33.3% of the vertical stabilizer chord length. This same percent will be utilized.

New_Rudder_Chord_Length_in = (1/3)*(New_Vertical_Stabilizer_Chord_Length);
New_Rudder_Surface_Area_in_squared =
(New_Rudder_Chord_Length_in)*(New_Vertical_Stabilizer_Span);
%_____
_____

% Horizontal Stabilizer

% From Page 14 and 15| Coefficient of Tail Volume

% C_H = ((L_h)*(S_h))/((S_w)*(C_mac))

% Where

```

```
% C_H = Coefficient of Horizontal Tail Volume (Non-Dimensional) | Defined
% L_h = Tail Lever Arm Length (horizontal) | Defined
% S_h = Horizontal Stabilizer Area | SOLVE FOR
% S_w = Wing Area| Defined
% C_mac = Mean Aerodynamic Chord | Defined

% Imperial
L_h_in = 40; % in
S_w_in_squared = Wing_Area_in_squared; % in^2
C_mac = Chord_Length_in; % in

S_h_inches_squared = ((S_w_in_squared * C_mac)/(L_h_in))*(C_H);

% Raymer Vs Preliminary Vertical Stabilizer Surface Area (Percent
% Difference)

Horizontal_Stabilizer_Area_Percentage_Over_Preliminary = (1 -
(Horizontal_Stabilizer_Area_in_squared/ S_h_inches_squared)) * 100;
%
```

% Now that the surface area is known, we need to be able to find
% appropriate values for the span and chord length of the horizontal stabilizer

% The preliminary aero analysis had the horizontal stabilizer span
% at 29.50 inches. Increasing the span by 2.5" (32.00") and chord length
% by 0.75 inches would produce a horizontal stabilizer area of 216 in squared which is
% sufficiently close to the Raymer analysis

```
New_Horizontal_Stabilizer_Surface_Area = 220.00; % in^2
New_Horizontal_Stabilizer_Span = 32.00; % in
New_Horizontal_Stabilizer_Chord_Length = 6.75; % in

% Elevator Chord Length and Surface Area

% Based on historical data and previous UML DBF years (notably 2023-2024) analysis, the rudder  
elevator chord length was sized to be
% 26.7 % of the horizontal stabilizer chord length. This same percent will be utilized based  
on flight success
```

```
New_Elevator_Chord_Length_in = (0.267)*(New_Horizontal_Stabilizer_Chord_Length);
New_Elevator_Surface_Area_in_squared =
(New_Elevator_Chord_Length_in)*(New_Horizontal_Stabilizer_Span);
%
```

%% XFLR5 Analysis Dimensions

% XFLR5 can not model trailing edge control surfaces at a location of 100 % of
% the chord length (because the trailing edge develops zero thickness).
% This inevitably becomes a problem since we want to model what the lift
% behavior of the wing is with the control surfaces at critical moments in
% the aircrafts flight, such as at take off and cruise. One way to
% theoretically get around this is by modeling wing and empennage surfaces
% to have a chord length that is the sum of the primary lifting surface and
% the control surface, effectively making a bigger continuous wing. When running the
% 2-D analysis in XFLR5, a hinge can then be specified at a location where the
% trailing edge of the primary surface should be and then an angle of
% deflection for the control surface can then be specified. Doing this

```
% means we do not run into the issue of trying to tack on a control surface
% at a point on the wing where there is zero-thickness.
```

```
% Wing Definition
```

```
% Wing Total Chord Length (Wing Chord Length + Flap Chord Length)
```

```
Wing_Total_Chord_Length = (Chord_Length_in) + (Flap_Chord_Length_in);
```

```
% T.E. (Trailing Edge) Flap Location as a percentage of total wing chord length
```

```
T_E_Flap_Hinge_Location = ((Chord_Length_in)/(Wing_Total_Chord_Length)) * 100;
```

```
% T.E. Flap Deflection Angle (Determined from visual inspection on prototype)
```

```
T_E_Flap_Angle = -20; % Degrees
```

```
% Span Definition
```

```
% To accurately represent the span of the wing with control surfaces
% correctly, we need to be able to section the wing according to the span of
% each control surface. While the span for both the aileron and flaps are
% known , we need to calculate the difference between the half span and the sum of the aileron
and
% flap span
```

```
% Section 1 (Body -> start of flap)
```

```
Section_1_Span = (Wing_Span_in/2) - (Single_Aileron_Span_in + Single_Flap_Span_in);
```

```
% Section 2 (Start of deflected flap span -> end of flap span)
```

```
Section_2_Span = Single_Flap_Span_in;
```

```
% Section 3 (end of flap span/start of aileron span -> end of aileron
% span/end of the wing
```

```
Section_3_Span = Single_Aileron_Span_in;
```

```
% Length Check
```

```
Section_Sum = (Section_1_Span + Section_2_Span + Section_3_Span);
error = 1*10^(-5);
```

```
if (Section_Sum - Wing_Span_in) <= error
    disp("The sum of the sections is equal to the wing span")
else
    disp("error")
end
```

```
% Section 1 [Start, End] = [0", 3.150"]
% Section 2 [Start, End] = [3.150", 19.250"]
% Section 3 [Start, End] = [19.250", 35.00"]
```

```
% Surface Area Check
```

```
% We need to calculate what the "master" surface area of the wing is (sum
% of the surface area of the main wing, and the main wings control
% surfaces). This is important because we need to ensure that when we use
```

```
% the 15.38" chord length, instead of 13.68", the total wing surface area
% XFLR5 puts out isn't wildly larger than the computed total that will be
% done below

% Wing Surface Area (No Control Surfaces) = 957.60 in^2

% Master Surface Area

Master_Chord_Length = (Chord_Length_in + Flap_Chord_Length_in); % in

Master_Wing_Surface_Area_Analytical = (Wing_Area_in_squared) + (Total_Flap_Surface_Area) +
(Total_Aileron_Surface_Area); % in^2

Master_Wing_Surface_Area_XFLR5 = 1101.24; % in^2

% Surface Area Error

% The difference in surface areas between the analytical calculations and
% XFLR5 modeling is due to the fact that the sum of the control surface
% span is not equal to the total wing span. The control surfaces span
% only 91 % of the total wing span, meaning there is an expected gap
% between the control surfaces and the fuselage much like the prototype.
% In XFLR5, this remaining % is just the regular wing filling in the gap,
% leading to why the surface area is slightly larger. Attempts were made to
% create a gap to better reflect the prototype geometry, but this leads to
% a severe discontinuity in the wing geometry which would lead to
% significant analysis issues. The percent difference is considered
% negligible for this reason

Surface_Area_Percent_Difference = (1 -
(Master_Wing_Surface_Area_Analytical/Master_Wing_Surface_Area_XFLR5)) * 100;

% _____
```

```
% Horizontal Stabilizer Total Chord Length (Horizontal Stabilizer Chord Length + Elevator
Chord Length)

Horizontal_Stabilizer_Total_Chord_Length = (New_Horizontal_Stabilizer_Chord_Length) +
(New_Elevator_Chord_Length_in);

% T.E. (Trailing Edge) Elevator Location as a percentage of total horizontal stabilizer chord
length

T_E_Elevator_Hinge_Location =
((New_Horizontal_Stabilizer_Chord_Length)/(Horizontal_Stabilizer_Total_Chord_Length)) * 100;

% T.E. Elevator Deflection Angle (Determined from visual inspection on prototype)

T_E_Elevator_Angle = -15; % Degrees

% Horizontal Stabilizer (Master Surface Area)

Horizontal_Stabilizer_Master_Chord_Length = Horizontal_Stabilizer_Total_Chord_Length; % in
```

```

Master_Horizontal_Stabilizer_Surface_Area_Analytical =
(Horizontal_Stabilizer_Master_Chord_Length) *(New_Horizontal_Stabilizer_Span); % in^2

Master_Horizontal_Stabilizer_Surface_Area_XFLR5 = 273.66; % in^2

% Surface Area Error

Horizontal_Stabilizer_Surface_Area_Percent_Difference = (1 -
(Master_Horizontal_Stabilizer_Surface_Area_XFLR5/Master_Horizontal_Stabilizer_Surface_Area_An-
alytical)) * 100;

% _____
_____
% Vertical Stabilizer Total Chord Length (Vertical Stabilizer Chord Length + Rudder Chord
Length)

Vertical_Stabilizer_Total_Chord_Length = (New_Vertical_Stabilizer_Chord_Length) +
(New_Rudder_Chord_Length_in)

% T.E. (Trailing Edge) Rudder Location as a percentage of total vertical stabilizer chord
length

T_E_Rudder_Hinge_Location =
((New_Vertical_Stabilizer_Chord_Length)/(Vertical_Stabilizer_Total_Chord_Length)) * 100

% T.E. Elevator Deflection Angle (CFD Analysis will not test effects of rudder deflection)

T_E_Elevator_Angle = 0 % Degrees

% Vertical Stabilizer (Master Surface Area)

Master_Vertical_Stabilizer_Surface_Area_Analytical = (Vertical_Stabilizer_Total_Chord_Length)
* (New_Vertical_Stabilizer_Span);
Master_Vertical_Stabilizer_Surface_Area_XFLR5 = 98.99;

% Surface Area Error

Vertical_Stabilizer_Surface_Area_Percent_Difference = (1 -
(Master_Vertical_Stabilizer_Surface_Area_Analytical/Master_Vertical_Stabilizer_Surface_Area_XF-
LR5)) * 100;

```

D-4: DBF Aero Analysis Plots

```

% Givens/Constants

WingSpan_in = 70; %in
WingSpan_ft = WingSpan_in/12; %ft
RootChordLength_in = 13.68; %in
RootChordLength_ft = RootChordLength_in/12; %ft
TipChordLength_in = 13.68; %in
TipChordLength_ft = TipChordLength_in/12; %ft
WingArea_in = 957.6; %in^2
WingArea_ft = WingArea_in/144; %ft^2
TotalWeight_lb = 15; %lbs
TotalWeight_oz = TotalWeight_lb*16; %ounces

```

```

OswaldEff = 0.7;
Cd_zerolift = 0.04;
AirDensity = 2.3769*10^-3; %lb*s^2*ft^-4
AspectRatio = (WingSpan_in^2)/WingArea_in;
TaperRatio = RootChordLength_in/TipChordLength_in;
MotorEff = 0.9*0.65;

%-----
%-----
%-----


Velocity_fps = (0:2:200);
Velocity_mph = Velocity_fps*0.6818181818;
Velocity_knots = Velocity_fps/1.69; %experiment with changing this to the excel multiplier
instead of 1.69

CoeffLift = TotalWeight_lb./(0.5*AirDensity.* (Velocity_fps.^2).*WingArea_ft);
CoeffLift_rootform = CoeffLift.^ (2/3);
CoeffInducedDrag = (CoeffLift.^2)/(pi()*AspectRatio*OswaldEff);
InducedDrag_lb = 0.5.*CoeffInducedDrag.*AirDensity.* (Velocity_fps.^2)*WingArea_ft; %lbs
ParastiticDrag_lb = 0.5*Cd_zerolift*AirDensity*(Velocity_fps.^2)*WingArea_ft; %lbf
CoeffZeroLiftDrag = 0.04*ones(1,101);
CoeffTotalDrag = CoeffInducedDrag+CoeffZeroLiftDrag;
CoeffLiftToDrag = CoeffLift./CoeffTotalDrag;
TotalDrag_lbf = InducedDrag_lb+ParastiticDrag_lb;

Thrust_lbf = TotalDrag_lbf; %lbf
Power_lbf_ft_s = Thrust_lbf.*Velocity_fps; %lbf*ft/s
Power_Watts = Power_lbf_ft_s*1.355821899; %Watts or J/s
EffectivePower = Power_Watts/MotorEff; %Watts or J/s
EffectivePowerPerVelocity = EffectivePower./Velocity_fps; %W/(ft/s)

%-----
%-----
%-----


% Plots -----

%Drag Force Vs Velocity:
plot(Velocity_fps(10:101), InducedDrag_lb(10:101))
hold on
grid on
grid minor
plot(Velocity_fps(10:101),ParastiticDrag_lb(10:101))
plot(Velocity_fps(10:101),TotalDrag_lbf(10:101))
hold off

```

```

title('Drag vs Velocity')
legend('Induced Drag','Parasitic Drag','Total Drag')
xlabel('Velocity (ft/s)')
ylabel('Drag (lbf)')

%Coefficient of Lift Vs Velocity:
plot(Velocity_fps(10:101), CoeffLift(10:101))
grid on
grid minor
title('Coefficient of Lift vs Velocity')
legend('Coefficient of Lift, C_L')
xlabel('Velocity (ft/s)')
ylabel('Coefficient of Lift, C_L')

%Coefficient of Drag vs Velocity:
plot(Velocity_fps(15:101),CoeffInducedDrag(15:101))
hold on
grid on
grid minor
plot(Velocity_fps(15:101),CoeffTotalDrag(15:101))
plot(Velocity_fps(15:101),CoeffZeroLiftDrag(15:101))
hold off
title('Coefficients of Drag vs Velocity')
legend('Coefficient of Induced Drag','Coefficient of Total Drag','Coefficient of Zero-Lift
Drag')
xlabel('Velocity (ft/s)')
ylabel('Coefficient of Drag, C_d')

%Coefficient of Lift vs Coefficient of Drag:
plot(CoeffTotalDrag(10:101),CoeffLift(10:101))
grid on
grid minor
title('Coefficient of Lift vs Coefficient of Drag')
legend('Coefficient of Lift vs Coefficient of Drag')
xlabel('Coefficient of Total Drag (C_d)')
ylabel('Coefficient of Lift, C_L')

%C_L/C_d vs Velocity:
plot(Velocity_fps(1:101),CoeffLiftToDrag(1:101))
grid on
grid minor
title('C_L/C_d vs Velocity')
legend('C_L/C_d')
xlabel('Velocity (ft/s)')
ylabel('C_L/C_d')

%Effective Power vs Velocity:

```

```

plot(Velocity_fps(1:101),EffectivePower(1:101))
grid on
grid minor
title('Effective Power vs Velocity')
legend('Effective Power (W)')
xlabel('Velocity (ft/s)')
ylabel('Effective Power (W)')

%(C_L)^{2/3} vs Coefficient of Total Drag:
plot(CoeffTotalDrag(1:101),CoeffLift_rootform(1:101))
grid on
grid minor
title('C_L^{2/3} vs Coefficient of Total Drag')
legend('C_L^{2/3} vs C_d')
xlabel('Coefficient of Total Drag')
ylabel('C_L^{2/3}')

% Plots included in the report:
% - CL vs velocity
% - Drag (lbs) vs velocity
% - CL/CD vs velocity

```

D-5-1: Thrust Testing Code, Calibrate

""

Calibration script for Load Cell Data Logger.

This script allows users to either collect calibration data or perform calibration using existing data.

It supports single load cell data collection, fitting linear and polynomial regression models, evaluating model performance, and saving the models for later use.

""

```

from dependencies import * # Centralized imports
from utils import getNumberOfLoadCells, getChannelNumber, loadCalibrationData
from LoadCell import LoadCell

```

""

Purpose:

Fit a linear regression model to the calibration data.

Input Parameters:

- x (numpy.ndarray): Sensor readings (independent variable).
- y (numpy.ndarray): True values (dependent variable).

Returns:

tuple: Contains the fitted model, predictions, mean squared error, and R² score.

Expected Behavior:

Fits a linear regression model to the provided data and evaluates its performance.

""

```
def fitLinearModel(x, y):
    linearModel = LinearRegression()
    linearModel.fit(x, y)
    yPrediction = linearModel.predict(x)
    return linearModel, yPrediction, mean_squared_error(y, yPrediction), r2_score(y, yPrediction)
```

""

Purpose:

Fit a polynomial regression model to the calibration data.

Input Parameters:

- x (numpy.ndarray): Sensor readings (independent variable).
- y (numpy.ndarray): True values (dependent variable).
- degrees (int): Degree of the polynomial to fit.

Returns:

tuple: Contains the fitted model, predictions, mean squared error, and R² score.

Expected Behavior:

Fits a polynomial regression model to the provided data and evaluates its performance.

""

```
def fitPolynomialModel(x, y, degrees):
    polyFeatures = PolynomialFeatures(degree=degrees)
    xPoly = polyFeatures.fit_transform(x)
    polyModel = LinearRegression()
    polyModel.fit(xPoly, y)
    yPrediction = polyModel.predict(xPoly)
    return polyModel, yPrediction, mean_squared_error(y, yPrediction), r2_score(y, yPrediction)
```

""

Purpose:

Save a trained model to a specified filepath.

Input Parameters:

- model (object): The trained model to save.
- filepath (str): Path to save the model.

Expected Behavior:

Saves the model in binary format using pickle.

""

```
def saveModel(model, filepath):
```

```
with open(filepath, 'wb') as file:  
    pickle.dump(model, file)
```

""

Purpose:

Collect voltage ratio changes for a specific load cell.

Input Parameters:

channel (int): The channel to which the load cell is connected.

Expected Behavior:

Continuously prints voltage ratio changes to allow the user to observe and record the values.

Stops on user interruption (Ctrl+C).

""

```
def collectCalibrationData(channel):
```

try:

```
    # Initialize LoadCell object (local connection assumed for simplicity)  
    calibrateCell = LoadCell(forceType='Channel {channel}', channel=channel,  
    fitModel=None, isRemote=False, remoteServer=None)
```

print(f'Collecting calibration data for Load Cell on channel {channel}. Apply known loads
and observe voltage ratio changes.')

print('Press Ctrl+C to stop.')

calibrateCell.open() # Open connection to the load cell

time.sleep(1)

Continuously print voltage ratio changes

while True:

voltageRatio = calibrateCell.device.getVoltageRatio() # Get the current voltage ratio

print(f'Channel {channel} Voltage Ratio: {voltageRatio:.6f} [mV/V]')

except KeyboardInterrupt:

print('\nData collection stopped by user.')

finally:

calibrateCell.close() # Ensure the device is properly closed

""

Purpose:

Main function for the calibration script.

Expected Behavior:

Prompts the user for their desired operation mode (data collection or calibration).

For data collection, prints voltage ratio changes for a specified load cell.

For calibration, processes existing data to fit regression models and save them.

```

"""

def calibrationMain():
    os.system('cls') # Clear terminal for Windows

    # Prompt the user until a valid input is provided
    while True:
        mode = input('Do you want to collect calibration data or calibrate using existing data? (collect/calibrate): ').strip().lower()
        if mode in {'collect', 'calibrate'}:
            break
        print('Invalid input. Please enter "collect" or "calibrate".')

    if mode == 'collect':
        channel = input('Enter the channel number for the load cell (0-3): ').strip()
        # Data collection mode
        while channel not in {'0', '1', '2', '3'}:
            print('Invalid input. Enter a channel number between 0 - 3')
            channel = input('Enter the channel number for the load cell (0-3): ').strip()
        collectCalibrationData(int(channel))

    elif mode == 'calibrate':
        # Calibration mode
        numberOfLoadCells = getNumberOfLoadCells()
        channels = getChannelNumber(numberOfLoadCells)

        # Load calibration data
        calibrationData = loadCalibrationData(CALIBRATION_DATA_FILE, channels)

    for channel in channels:
        x = calibrationData['sensorReading'][channel].reshape(-1, 1)
        y = calibrationData['trueValue'][channel]

        # Fit linear and polynomial models
        linearModel, linearPrediction, linearMSE, linearR2 = fitLinearModel(x, y)
        polyModel, polyPrediction, polyMSE, polyR2 = fitPolynomialModel(x, y,
POLY_DEGREE)

        # Output model performance
        print(f'Channel {channel} - Linear: MSE = {linearMSE:.3f}, R2 = {linearR2:.5f}')
        print(f'Channel {channel} - Polynomial ({POLY_DEGREE}): MSE = {polyMSE:.3f},
R2 = {polyR2:.5f}')



```

```

# Save models
saveModel(linearModel, LINEAR_FIT_MODEL.format(channel))
saveModel(polyModel, POLY_FIT_MODEL.format(channel))

# Plot results
plotData = pd.DataFrame({
    'Sensor Reading': calibrationData['sensorReading'][channel].flatten(),
    'True Value': calibrationData['trueValue'][channel],
    'Linear Prediction': linearPrediction.flatten(),
    'Polynomial Prediction': polyPrediction.flatten(),
})

sns.set_theme(style='whitegrid')
fig, axes = plot.subplots(1, 2)
fig.suptitle(f'Load Cell Connected to Channel {channel}', fontsize=16)

# Linear fit plot
sns.scatterplot(x='Sensor Reading', y='True Value', data=plotData, ax=axes[0],
label='True Values', color='blue')
sns.lineplot(x='Sensor Reading', y='Linear Prediction', data=plotData, ax=axes[0],
label='Linear Fit', color='red')
axes[0].set_title('Linear Fit')
axes[0].set_xlabel('Sensor Reading')
axes[0].set_ylabel('True Value')
axes[0].text(0.05, 0.95, f'MSE: {linearMSE:.3f}\nR2: {linearR2:.5f}',
transform=axes[0].transAxes, fontsize=12, verticalalignment='top', bbox=dict(facecolor='white',
alpha=0.5))

# Polynomial fit plot
sns.scatterplot(x='Sensor Reading', y='True Value', data=plotData, ax=axes[1],
label='True Values', color='blue')
sns.lineplot(x='Sensor Reading', y='Polynomial Prediction', data=plotData, ax=axes[1],
label=f'Polynomial Fit ({POLY_DEGREE} Degrees)', color='green')
axes[1].set_title(f'Polynomial Fit ({POLY_DEGREE} Terms)')
axes[1].set_xlabel('Sensor Reading')
axes[1].set_ylabel('True Value')
axes[1].text(0.05, 0.95, f'MSE: {polyMSE:.3f}\nR2: {polyR2:.5f}',
transform=axes[1].transAxes, fontsize=12, verticalalignment='top', bbox=dict(facecolor='white',
alpha=0.5))

```

```
plot.tight_layout()
plot.show()
```

```
if __name__ == "__main__":
    calibrationMain()
```

D-5-2: Thrust Testing Code, Config

```
""
```

Configuration file for Load Cell Data Logger.

This file contains constants and settings used throughout the application, such as timeout durations, plotting parameters, and file paths.

```
""
```

```
import os
```

```
# -----
# Device Configuration
# -----
```

```
TIMEOUT = 5000 # Maximum time in milliseconds before connection to a device times out
DATA_COLLECTION_INTERVAL = 100 # [ms] Collect Data within this interval
```

```
# -----
# Data Plotting
# -----
```

```
MAX_READINGS = 1000 # Maximum number of sensor readings to store for plotting
PAUSE_TIME = 0.101 # [s]
```

```
# -----
# File Paths
# -----
```

```
# Filepath to the Calibration data
CALIBRATION_DATA_FILE = os.path.join('Calibration Data', 'LoadCellCalibrationData.xlsx')
```

```
# Filepath template for the linear fit model (channel-specific)
LINEAR_FIT_MODEL = os.path.join('Calibration Data', 'linearFitModel_channel{}.pkl')
```

```
# Filepath template for the linear fit model (channel-specific)
POLY_FIT_MODEL = os.path.join('Calibration Data', 'polyFitModel_channel{}.pkl')
```

```
# Filepath template for the Excel sheet used for storing load cell data
```

```

EXCEL_FILE = os.path.join('Logged Data', 'loadCellData_{ }.xlsx')

# -----
# Calibration Configuration
# -----


POLY_DEGREE = 4 # Degree of polynomial for polynomial regression

CALIBRATION_DATA_COLUMNS = {
    'true_value': 'Input Weight [N]', # Column name for true input weight in calibration data
    'sensor_reading': 'Sensor Reading [mV/V]' # Column name for sensor readings
}

```

D-5-3: Thrust Testing Code, Dependencies

```

# Standard Library
import os
from datetime import datetime
import pickle
import subprocess
import time

# Third-party Libraries
import numpy as np
import pandas as pd
from openpyxl import Workbook, load_workbook
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import PolynomialFeatures
from Phidget22.Phidget import *
from Phidget22.Net import *
from Phidget22.Devices.VoltageRatioInput import *

# Application-specific Imports
from config import *

```

D-5-4: Thrust Testing Code, Loadcell

""

Class to interact with Phidget load cell devices.

Provides methods to configure the device, read voltage ratios, convert readings to force, and handle taring for zeroing the sensor. Includes support for remote connections.
""

```
from config import *
```

```
from Phidget22.Phidget import *
from Phidget22.Net import *
from Phidget22.Devices.VoltageRatioInput import *
import numpy as np
import pickle
import subprocess
```

""

LoadCell class for interacting with Phidget load cell devices.

This class provides methods for configuring the device, handling voltage ratio changes, predicting forces using a pre-trained model, and zeroing the sensor (taring).
Supports both local and remote connections.

""

class LoadCell:

""

Purpose:

Initialize the Load Cell class and configure the device.

Input Parameters:

forceType (str): Type of force the load cell is measuring (e.g., Lift, Drag, Thrust).

channel (int): The channel number the load cell is connected to (0-3).

fitModel (str): Filepath to the trained model used to convert voltage ratios to force.

isRemote (bool): Whether to connect to the load cell over a remote server.

remoteServer (tuple): Remote server details (serverName, address, port, password), or None for local connection.

Expected Behavior:

Creates an instance of the LoadCell class, associates it with the specified channel, and loads the pre-trained model. Configures remote connection if applicable. Sets the data collection interval to

the configured value (`DATA_COLLECTION_INTERVAL`).

Raises:

RuntimeError: If the connection to the remote server fails.

""

```
def __init__(self, forceType, channel, fitModel, isRemote, remoteServer):
```

Initialize key attributes like forceType, channel, and tare offset

self.forceType = forceType # Store the type of force being measured (Lift, Drag, Thrust)

self.channel = channel # Store the channel number to which the load cell is connected

self.sensorReading = [] # Initialize an empty list to store sensor readings over time

self.plotWindow = [] # Initialize an empty list to store sensor readings over time (This will get modified and plotted)

self.tareOffset = 0 # Initialize with 0 tare

try:

 self.model = self.loadFitModel(fitModel) # Load the pre-trained model from the provided filepath

```

except RuntimeError as error:
    print(f"Warning: {error}. Calibration required for {self.forceType} load cell on channel
{self.channel}.")
    self.model = None # Placeholder until calibration is performed

    self.device = VoltageRatioInput() # Create a VoltageRatioInput object to interact with the
load cell hardware

    # Enable remote server discovery if connecting remotely
    if isRemote:
        Net.enableServerDiscovery(PhidgetServerType.PHIDGETSERVER_DEVICEREMOTE
)
        self.device.setIsRemote(True) # Remote connection
        # Add the specific remote server if provided
        if remoteServer:
            serverName, address, port, password = remoteServer
            try:
                Net.addServer(serverName, address, port, password, True)
                print(f"Connected to remote server { serverName } at { address }: { port }")
            except PhidgetException as error:
                raise RuntimeError(f'Failed to connect to remote server: { error }')

    self.device.setChannel(self.channel) # Set the channel on the device to the provided
channel number

"""

```

Purpose:

Load the pre-trained model for converting voltage ratios to force.

If the model file is missing, prompts the user to run the calibration script.

Input Parameters:

filepath (str): The filepath to the pre-trained model.

Returns:

object: The loaded model for predicting forces.

Expected Behavior:

Attempts to load the model from the specified filepath.

If the file is missing or corrupted, prompts the user to run calibration and restarts the model loading.

Raises a RuntimeError if the user declines calibration.

""

```

def loadFitModel(self, filepath):
    modelLoaded = False
    while not modelLoaded:
        try:
            # Attempt to open the model file and load it using pickle
            with open(filepath, 'rb') as model:
                modelLoaded = True

```

```

        return pickle.load(model)
    except FileNotFoundError as error:
        print(f"Error loading model file '{filepath}': {error}.")
        response = input("No calibration model found. Would you like to run the calibration script now? (y/n): ").strip().lower()
        while response not in {'y', 'n'}:
            print('Invalid input. Please enter "y" or "n"')
            response = input("No calibration model found. Would you like to run the calibration script now? (y/n): ").strip().lower()
        if response == "y":
            # Run the calibration script
            try:
                subprocess.run(["python", "Force Balance Data Logger\\calibrate.py"],
check=True)
            except subprocess.CalledProcessError as calError:
                raise RuntimeError("Calibration script execution failed. Please retry.") from calError
            print("Calibration completed. Retrying model loading...")
        else:
            raise RuntimeError("Calibration required but skipped by the user.")
    except (Exception, pickle.UnpicklingError) as error:
        # Print a generic error message for any other exception during model loading
        raise RuntimeError(f'An unexpected error occurred while loading the model from "{filepath}": {error}')
    ...

```

Purpose:

Open the device connected to the specified channel and wait for attachment.

Input Parameters:

None.

Expected Behavior:

Opens the connection to the load cell and waits for attachment.

Raises:

RuntimeError: If the connection to the load cell fails within the timeout period.

...

def open(self):

try:

```

        # Attempt to open the device and wait for attachment, with a timeout
        self.device.openWaitForAttachment(TIMEOUT)
        self.device.setDataInterval(DATA_COLLECTION_INTERVAL) # Set Data Collection Rate (in Milliseconds) for the loadcell
        self.device.setOnVoltageRatioChangeHandler(self.onVoltageRatioChange) # Assign a handler function for voltage ratio change events
    except PhidgetException as error:
        # Raise a RuntimeError if the connection to the load cell fails

```

```
    raise RuntimeError(f'Failed to connect to { self.forceType } load cell on channel { self.channel }\nConnection Timed Out') from error
```

'''

Purpose:

Close the connection to the load cell device.

Input Parameters:

None.

Expected Behavior:

Closes the connection to the load cell and releases resources.

'''

```
def close(self):
```

```
    self.device.close() # Close the connection to the load cell device
```

'''

Purpose:

Event handler that triggers whenever the voltage ratio changes. This function calculates the predicted

force from the voltage ratio, adjusts it for the tare offset, appends it to the cumulative sensor readings

(`sensorReading`), and updates the temporary readings (`temp`) for plotting.

Input Parameters:

device (Phidget): The Phidget device that triggered the event.

voltageRatio (float): The new voltage ratio reading from the load cell.

Expected Behavior:

Converts the voltage ratio to a force using the pre-trained model and appends it to sensor readings.

Removes the oldest reading if the maximum number of readings is exceeded.

Prints the voltage ratio and predicted force to the console.

'''

```
def onVoltageRatioChange(self, device, voltageRatio):
```

```
    # Convert the voltage ratio to a predicted force using the model
```

```
    forcePrediction = self.predictForce(voltageRatio) - self.tareOffset
```

```
    # Append the predicted force to the sensorReading list
```

```
    self.sensorReading.append(forcePrediction)
```

```
    self.plotWindow.append(forcePrediction)
```

```
    # If the temp list exceeds MAX_READINGS, remove the oldest entry
```

```
    if len(self.plotWindow) > MAX_READINGS:
```

```
        self.plotWindow.pop(0) # Keep only the latest 1000 readings to plot
```

```
    # Print the current voltage ratio and predicted force to the console
```

```
    print(f'Voltage Ratio for {self.forceType} Load Cell: {voltageRatio}, Predicted Force: {forcePrediction}')
```

'''

Purpose:

Convert the voltage ratio to a force using the pre-trained model.

Input Parameters:

voltageRatio (float): The voltage ratio reading from the load cell.

Returns:

float: The predicted force corresponding to the voltage ratio.

Expected Behavior:

Uses the loaded model to convert the voltage ratio to a force.

Raises:

RuntimeError: If the model is not loaded or an error occurs during prediction.

""

```
def predictForce(self, voltageRatio):
```

```
    reshapedVoltageRatio = np.array([[voltageRatio]]) # Reshape the voltage ratio into a 2D
array for the model
```

```
    return self.model.predict(reshapedVoltageRatio) # Placeholder: model would predict force
here (self.model.predict(reshapedVoltageRatio))
```

""

Purpose:

Set the tare offset to the current voltage ratio to account for baseline drift or zeroing the sensor.

Input Parameters:

None.

Expected Behavior:

Reads the current voltage ratio from the load cell and sets it as the tare offset.

Prints the new tare offset to the console for confirmation.

Raises:

PhidgetException: If reading the voltage ratio fails.

""

```
def tare(self):
```

try:

```
    # Read the current voltage ratio from the device
```

```
    currentVoltageRatio = self.device.getVoltageRatio()
```

```
    # Set the tare offset to the current voltage ratio
```

```
    self.tareOffset = self.predictForce(currentVoltageRatio);
```

```
    # Print a confirmation message
```

```
    print(f'Tared {self.forceType} Load Cell. Offset set to: {self.tareOffset}')
```

except PhidgetException as e:

```
    # Print an error message if reading the voltage ratio fails
```

```
    print(f'Failed to read voltage ratio for taring: {e}')
```

D-5-5: Thrust Testing Code, Main

""

Main script for running the Load Cell Data Logger.

This script initializes load cells, sets up real-time data visualization,

and handles logging data to Excel files. It supports both USB and remote connections to Phidget devices. Includes options for taring and saving real-time data from load cells.

""

```
from dependencies import * # Centralized imports
from utils import *
from calibrate import calibrationMain
from LoadCell import LoadCell
```

""

Purpose:

Main function for the Load Cell Data Logger application.

Expected Behavior:

Prompts the user to configure load cells, sets up real-time data visualization, and logs data to Excel files. Supports USB and remote connections.

Workflow:

1. Prompts the user for connection type (USB or remote).
2. Configures load cells, including channel numbers and force types.
3. Initializes data visualization and logging.
4. Continuously updates plots with real-time load cell readings.
5. Allows taring (zeroing) of load cells with "Ctrl+T".
6. Saves all collected data to an Excel file only upon program termination.
7. Handles shutdown on user interruption (Ctrl+C).

""

def main():

```
os.system('cls') # Clear terminal for Windows
```

```
time_step = 0 # Initialize time step for Excel saving
```

```
stop_flag = [False] # Mutable flag to indicate when to stop
```

```
timestamp = datetime.now().strftime('%Y_%m_%d-%H-%M') # Time the Program was run to
timestamp data collection
```

```
# Ask if the user wants a remote connection
```

```
useRemote = input("Do you want to connect to a remote Phidget server? (y/n):
```

```
").strip().lower() == "y"
```

```
remoteServer = None
```

```
if useRemote:
```

```
    # Prompt for remote server details
```

```
    serverName = input("Enter the server name: ").strip()
```

```
    address = input("Enter the server IP address: ").strip()
```

```
    port = int(input("Enter the server port (default 5661): ").strip() or 5661)
```

```
    password = input("Enter the server password (leave empty if none): ").strip()
```

```
    remoteServer = (serverName, address, port, password)
```

```
    # Local Computer
```

```
    #remoteServer = ('AADITH', '10.248.51.199', 5661, '')
```

```

numberOfLoadCells = getNumberOfLoadCells()
channels = getChannelNumber(numberOfLoadCells)
loadCells = {}

# GET CHANNEL NUMBERS OF THE LOAD CELLS
for channel in channels:
    forceType = getForceType(channel)
    try:
        loadCells[forceType] = LoadCell(forceType=forceType, channel=channel,
fitModel=LINEAR_FIT_MODEL.format(channel), isRemote=useRemote,
remoteServer=remoteServer)
    except RuntimeError as error:
        print(f'{ error }')
    return

workbook = initializeExcel(EXCEL_FILE.format(timestamp), loadCells) # Initialize or load
the Excel file

try:
    # OPEN THE CONNECTION TO ALL THE LOAD CELLS
    for forceType, loadCell in loadCells.items():
        try:
            loadCell.open() # Open the connection to the current load cell
        except RuntimeError as error:
            print(f'{ error }')
        return

    # Enable interactive mode for Matplotlib to allow real-time plot updates
    plot.ion()

    # Create a single plot with three lines (one for each load cell)
    figure, axes, lines = createPlot(loadCells)

    print('Running... Press Ctrl+C to Stop') # Inform the user that the program is running

    # Continuously update the plots with real-time data
    while not stop_flag[0]:
        # Register the keypress handler
        registerKeypressHandler(figure, loadCells, stop_flag)

        updatePlot(figure, axes, lines, loadCells) # Update the plot with the sensor reading
        time_step += DATA_COLLECTION_INTERVAL # Increment time step by the number
of readings cleared
        finally:
            plot.ioff() # Turn off Matplotlib's interactive mode

```

```

plot.close('all') # Close all open plot windows

# Close the connection to all load cells
for forceType, loadCell in loadCells.items():
    loadCell.close()

saveData(workbook, loadCells)
workbook.save(EXCEL_FILE.format(timestamp)) # Save changes to the Excel file
print('Load Cell Connections Closed') # Inform the user that the connections have been
closed

if __name__ == "__main__":
    main() # Run the main function

```

D-5-6: Thrust Testing Code, Postprocessing

""

Post-processing script for the Load Cell Data Logger.

This script prompts the user to select a saved Excel file containing load cell data, processes the data from all sheets, and plots force vs. time for each load cell.

""

from dependencies import * # Centralized imports

""

Purpose:

Main function for the post-processing script.

Prompts the user to select a saved Excel file, processes its data, and generates plots.

Input Parameters:

None.

Expected Behavior:

1. Prompts the user to enter the name of the Excel file to process.
2. Reads data from the specified file, processes each sheet, and plots force vs. time.
3. Handles errors gracefully, such as missing files or invalid data.

""

def postProcess():

Prompt the user for the Excel file name

while True:

file_name = input("Enter the name of the Excel file to process (e.g., 'loadCellData_YYYY_MM_DD-HH-MM.xlsx'): ").strip()

file_path = os.path.join('Logged Data', file_name)

if os.path.exists(file_path):

print(f"File '{file_name}' found. Proceeding...")

break

else:

```

print(f"File '{file_name}' not found. Please try again.")

# Load the workbook and list all sheets
workbook = pd.ExcelFile(file_path)
print("\nAvailable Sheets in the File:")
print(workbook.sheet_names) # Display available sheets
print("\n")

# Process each sheet (assuming each sheet corresponds to a load cell)
for sheet_name in workbook.sheet_names:
    print(f"Processing data for sheet: {sheet_name}...")

    # Load the data from the sheet into a DataFrame
    data = pd.read_excel(file_path, sheet_name=sheet_name)

    # Validate columns
    if 'Time [s]' not in data.columns or 'Force [N]' not in data.columns:
        print(f"Sheet '{sheet_name}' does not contain the required columns. Skipping...")
        continue

    # Extract time and force data
    time = data['Time [s]']
    force = data['Force [N]']

    # Plot the data
    plot.figure(figsize=(10, 6))
    plot.plot(time, force, label=sheet_name)
    plot.title(f"{sheet_name} Load Cell Data")
    plot.xlabel("Time [s]")
    plot.ylabel("Force [N]")
    plot.grid(True)
    plot.legend()

# Display all plots
plot.show()
"""

```

Purpose:

Entry point for the script. Executes the `postProcess` function.

Expected Behavior:

Runs the post-processing workflow to generate plots for saved load cell data.

```

"""
if __name__ == "__main__":
    postProcess()

```

D-5-7: Thrust Testing Code, Utils

Utility functions for the Load Cell Data Logger.

Includes functions for plotting, user input validation, and Excel file management.

""

```
from dependencies import * # Centralized imports
```

""

Purpose:

Prompt the user to input the number of load cells they want to configure.

Input Parameters:

None.

Returns:

int: The number of load cells to configure.

Expected Behavior:

Prompts the user to enter a valid integer greater than 0. Repeats the prompt if the input is invalid.

""

```
def getNumberOfLoadCells():
```

 while True:

 try:

 # Ask the user for the number of load cells to configure

 numLoadCells = int(input('Enter the number of load cells to configure: '))

 return numLoadCells # Return the valid number of load cells

 except ValueError:

 print(f'Invalid Input. Please try again.')

""

Purpose:

Prompt the user to input the channel numbers for the load cells.

Input Parameters:

 numberOfLoadCells (int): The number of load cells to configure.

Returns:

list: A list of unique channel numbers (0-3) entered by the user.

Expected Behavior:

Ensures no duplicate channels are assigned and channels are within the valid range (0-3).

Prompts the user again if input is invalid or duplicate.

""

```
def getChannelNumber(numberOfLoadCells):
```

 assignedChannels = []

 for number in range(numberOfLoadCells):

 while True:

 try:

 # Ask the user for the channel number of the load cells

```

channel = int(input('Enter the channel that load cell { number + 1 } is connected to: '))
if channel < 0 or channel > 3:
    # Channel number must be between 0 and 3 for valid configuration
    print('Incorrect Input. Please try again.')
elif channel not in assignedChannels:
    assignedChannels.append(channel)
    break
else:
    # Prevent assigning the same channel to multiple load cells
    print('Channel { channel } is already assigned to another load cell')
except ValueError:
    print('Invalid Input. Channel Number must be between 0 - 3. Please try again.')
return assignedChannels

```

""

Purpose:

Prompt the user to input the type of force the load cell is measuring.

Input Parameters:

channelNumber (int): The channel number of the load cell.

Returns:

str: The type of force being measured (e.g., Lift, Drag, Thrust).

Expected Behavior:

Prompts the user for a valid force type. Repeats the prompt if the input is invalid.

""

def getForceType(channelNumber):

while True:

try:

Prompt the user for the force type

forceType = input('Enter the type of force that the load cell in channel {channelNumber} measures (e.g., Lift, Drag, Thrust): ')

return forceType # Return the valid channel number

except ValueError:

Print an error message and prompt the user to try again

print('Invalid Input. Please try again')

""

Purpose:

Prompt the user for the name of the Excel sheet containing calibration data for a load cell.

Input Parameters:

channel (int): The channel number of the load cell.

Returns:

str: The name of the sheet entered by the user.

Expected Behavior:

Prompts the user to input the name of the sheet for each load cell.

""

def getSheetName(channel):

```

sheetName = input(f'Enter the name of the sheet that contains the calibration data for the load
cell connected to channel {channel}: ')
return sheetName
"""

```

Purpose:

Load calibration data from the specified file.

Input Parameters:

filepath (str): Path to the Excel file containing calibration data.

Returns:

dict: A dictionary containing true values and sensor readings for each load cell.

Expected Behavior:

Reads calibration data for all specified load cells and organizes it into true values
(expected forces) and sensor readings (voltage ratios).

"""

```
def loadCalibrationData(filepath, channels):
```

```
    rawData = {}
```

```
    calibrationData = {
        'trueValue': {},
        'sensorReading': {}
    }
```

for channel in channels:

```
    sheetName = getSheetName(channel) # Get the sheet name for each load cell
```

```
    # Read data from the specified sheet in the Excel file
```

```
    rawData[channel] = pd.read_excel(filepath, sheet_name=sheetName, skiprows=1,
index_col=0)
```

```
    # Extract true values and sensor readings
```

```
    calibrationData['trueValue'][channel] = rawData[channel]['Input Weight [N]'].values
    calibrationData['sensorReading'][channel] = rawData[channel]['Sensor Reading
[mV/V]'].values
```

```
return calibrationData # Return organized calibration data
```

"""

Purpose:

Ensures all load cells have a dedicated sheet with appropriate headers.

If the file does not exist, creates a new workbook.

If it exists, appends new sheets for missing load cells.

Input Parameters:

file_name (str): The name of the Excel file to create or load.

loadCells (dict): A dictionary of load cell objects, where keys are force types
(e.g., Lift, Drag) and values are LoadCell instances.

Returns:

Workbook: The Excel workbook object, with sheets for each load cell.

Expected Behavior:

Creates a new workbook with headers for each load cell if the file does not exist.

If the file exists, ensures all required sheets are present for appending data.

""

```
def initializeExcel(file_name, loadCells):
    if not os.path.exists(file_name):
        workbook = Workbook()
        if len(loadCells) != 0:
            default_sheet = workbook["Sheet"]
            workbook.remove(default_sheet)
        for forceType in loadCells:
            sheet = workbook.create_sheet(title=forceType)
            sheet.append(["Time [ms]", "Time [s]", "Force [N]"]) # Add headers
        workbook.active = workbook[sheetnames[0]]
        workbook.save(file_name)
    else:
        workbook = load_workbook(file_name)
        for forceType in loadCells:
            if forceType not in workbook.sheetnames:
                sheet = workbook.create_sheet(title=forceType)
                sheet.append(["Time [ms]", "Force [N]"])
        if not workbook.active:
            workbook.active = workbook[sheetnames[0]]
    return workbook
```

""

Purpose:

Save the sensor readings from all load cells to an Excel file.

Input Parameters:

workbook (Workbook): The Excel workbook object where the data will be saved.

loadCells (dict): A dictionary of load cell objects, where keys are force types

(e.g., Lift, Drag) and values are LoadCell instances.

time_step (int): The current time step to associate with the data.

Expected Behavior:

Appends all collected sensor readings for each load cell to its respective sheet in the Excel workbook.

The time for each reading is calculated based on its index and the configured data collection interval (`DATA_COLLECTION_INTERVAL`).

""

```
def saveData(workbook, loadCells):
    for forceType, loadCell in loadCells.items():
        # Check if the sheet for this load cell exists, create it if not
        if forceType not in workbook.sheetnames:
            sheet = workbook.create_sheet(title=forceType)
```

```

sheet.append(["Time [ms]", "Time [s]", "Force [N]"]) # Add headers
else:
    sheet = workbook[forceType]

# Append the sensor readings with their time steps
for index, reading in enumerate(loadCell.sensorReading):
    sheet.append([index * DATA_COLLECTION_INTERVAL, (index * 
DATA_COLLECTION_INTERVAL) / 1000, reading.item()])

```

""

Purpose:

Create and configure a plot for visualizing load cell readings.

Input Parameters:

loadCells (dict): A dictionary of load cell objects, where keys are force types
(e.g., Lift, Drag) and values are LoadCell instances.

Returns:

tuple: A tuple containing:

- figure (matplotlib.figure.Figure): The figure object for the plot.
- axes (matplotlib.axes.Axes): The axes object where the data is plotted.
- lines (dict): A dictionary of Line2D objects for each load cell, used for updating the plot.

Expected Behavior:

Sets up a plot with labeled axes and legends for real-time updates of load cell data.

""

def createPlot(loadCells):

```

figure, axes = plt.subplots() # Create a figure and a set of subplots (axes)
axes.set_title('Load Cell Readings [N] vs. Time [s]')
axes.set(ylabel='Load Cell Reading [N]', xlabel='Time [ms]')
lines = {}
# Create an empty Line2D object for each load cell and store them in a dictionary
for forceType in loadCells:
    lines[forceType] = axes.plot([], [], label=f'{forceType} Load Cell')[0]

```

axes.legend()

return figure, axes, lines # Return the figure, axes, and line for later use

""

Purpose:

Update the plot with the latest sensor readings for the specified load cell.

Input Parameters:

figure (matplotlib.figure.Figure): The figure object that contains the plot.
axes (matplotlib.axes.Axes): The axes object where the data is plotted.
lines (dict): A dictionary of Line2D objects for each load cell.
loadCells (dict): A dictionary of LoadCell objects to update data from.

Expected Behavior:

Updates the x-data and y-data of each Line2D object with the latest sensor readings.

Handles exceptions for plot rendering errors.

```

"""
def updatePlot(figure, axes, lines, loadCells):
    for forceType, loadCell in loadCells.items():
        # Ensure the x and y arrays are of the same length
        y_data = loadCell.plotWindow
        x_data = np.arange(len(y_data)) # Create x data of the same length as y_data
        #print(f'y_data: { y_data }, x_data: { x_data }')
        # Check if the lengths of x_data and y_data are the same before updating the plot
        if len(x_data) == len(y_data):
            lines[forceType].set_xdata(x_data) # Update the x-axis data
            lines[forceType].set_ydata(y_data) # Update the y-axis data
        else:
            print(f"Length mismatch: x_data has length {len(x_data)}, but y_data has length
{len(y_data)}")
            axes.relim() # Recompute the limits of the data to rescale the axes
            axes.autoscale_view() # Automatically adjust the view to fit the new data

    try:
        figure.canvas.draw() # Redraw the figure to reflect updated data
    except Exception as error:
        # Handle plotting errors, especially in headless environments
        print(f"Plot rendering error: {error}")
        plot.pause(PAUSE_TIME) # Brief pause to allow the UI to update
"""

```

Purpose:

Register a keypress event handler to handle custom keyboard inputs (e.g., Ctrl+T for taring, Ctrl+C for stopping).

Input Parameters:

figure (matplotlib.figure.Figure): The figure object where the event handler is attached.
loadCells (dict): A dictionary of LoadCell objects to perform taring.

stop_flag (list): A mutable flag (e.g., list with one boolean) used to signal stopping the program.

Expected Behavior:

Detects the 'Ctrl+T' key combination and triggers the taring function for all load cells.
Detects the 'Ctrl+C' key combination and sets the stop flag to True.

""

```

def registerKeypressHandler(figure, loadCells, stop_flag):
    def on_key(event):
        # Check for 'Ctrl+T' to tare load cells
        if event.key == 'ctrl+t':
            print("Taring load cells...")
            for loadCell in loadCells.values():
                loadCell.tare()

        # Check for 'Ctrl+C' to stop the program

```

```

if event.key == 'ctrl+c':
    print("Stopping the Program...")
    stop_flag[0] = True

# Connect the keypress handler to the figure
figure.canvas.mpl_connect('key_press_event', on_key)

```

D-5-8: Thrust Testing Code, Calibration data

Channel 0

Input Mass [kg]	Input Weight [N]	Sensor Reading [mV/V]
0.1	0.981	0.000079500
0.2	1.962	0.000074400
0.3	2.943	0.000069000
0.4	3.924	0.000064250
0.5	4.905	0.000059150
0.6	5.886	0.000054000
0.7	6.867	0.000048900
0.8	7.848	0.000043800
0.9	8.829	0.000038700
1	9.81	0.000033600
1.115	10.93815	0.000027700
1.494	14.65614	0.000008480
2.074	20.34594	-0.000021000
2.158	21.16998	-0.000026600
2.66	26.0946	-0.000054700
4.53592	44.4973752	-0.000146500
9.07184	88.9947504	-0.000376900

Channel 2

Input Mass [kg]	Input Weight [N]	Sensor Reading [mV/V]
0.1	0.981	0.000095100
0.2	1.962	0.000100200
0.3	2.943	0.000105000
0.4	3.924	0.000110400
0.5	4.905	0.000115600
0.6	5.886	0.000120700
0.7	6.867	0.000125800
0.8	7.848	0.000130900
0.9	8.829	0.000136100
1	9.81	0.000141200

1.115	10.93815	0.000147000
1.494	14.65614	0.000165000
2.047	20.08107	0.000194700
2.158	21.16998	0.000200400
2.66	26.0946	0.000226200
4.53592	44.4973752	0.000322500
9.07184	88.9947504	0.000554900

Appendix E – Full Spreadsheets for Calculations, Data, Project Tracking

Table E-1. Wing Cube Loading Analysis

Previous Teams				
Area		Weight		
ft^2	in^2	lbs.	oz.	Wing Cube Loading
11.30	1627.06	18.50	295.95	7.79
6.90	993.60	15.97	255.47	14.09
0.66	94.46	0.56	8.98	16.90
5.00	720.00	11.24	179.79	16.08
5.52	794.88	13.70	219.20	16.90
5.56	800.64	17.80	284.80	21.72
7.00	1008.00	10.77	172.27	9.30
8.50	1224.00	13.30	212.85	8.59
5.70	820.80	11.50	184.00	13.52
6.05	871.20	17.97	287.47	19.32
6.53	940.32	13.98	223.68	13.40
8.09	1164.96	15.15	242.35	10.53
5.78	774.38	10.89	174.27	13.97
Area		Weight		
ft^2	in^2	lbs.	oz.	Wing Cube Loading
Average	6.35	910.33	13.18	210.85
Approximate Area + Weight				
Area		Weight		
ft^2	in^2	lbs.	oz.	Wing Cube Loading
6.65	957.60	15.00	240.00	14.00

Table E-2. Empennage and Primary Control Sizing

Component	Dimension	Value
Vertical Stabilizer (NACA 0009)	Span (Height of Vertical Tail)	14.125 in
	Chord Length	5.25 in
	Surface Area	74.25 in ²
Rudder	Span (Height of Rudder)	14.125 in
	Chord Length	1.75 in
	Surface Area	24.7175 in ²
Horizontal Stabilizer (NACA 0009)	Span	32.00 in
	Chord Length	6.75 in
	Surface Area	220.00 in ²
Elevator	Span (Width of Horizontal Tail)	32.00 in
	Chord Length	1.75 in
	Surface Area	56.00 in ²
Flap	Single Flap Span	16.00 in
	Flap Chord Length	2.00 in
	Single Flap Surface Area	32.00 in ²
	Total Flap Surface Area	64.00 in ²
Aileron	Single Aileron Span	15.75 in
	Aileron Chord Length	2.00 in
	Single Aileron Surface Area	31.5 in ²
	Total Aileron Surface Area	63.00 in ²

Table E-3. Bill of Materials

Component	Part	Quantity	Units	Price Per Unit	Estimated Shipping + Tax	Price Per Part
Materials		29				
	Foam Board	20	Pcs	\$1.25	\$11.49	\$36.49
	1/4-20 Nylon Screws	1	Pack	\$17.76	\$10.33	\$28.09
	1/4-20 Nylon Nuts	1	Pack	\$13.24	\$8.87	\$22.11
	1/4" Wooden Dowels	1	Pack	\$4.99	\$7.30	\$12.29
	Control Horns/Pushrods	2	Pack	\$10.38	\$8.29	\$29.05
	Styrofoam	1	Pack	\$22.99	\$8.43	\$31.42
	LD-PLA	2	Spool	\$27.99	\$10.49	\$66.47
Electronics		2				
	9g Servos	1	Pack	\$18.99	\$8.18	\$27.17
	Quick Release	1	Pcs	\$28.99	\$8.80	\$37.19
Components		3				
	Tail Gear	1	Pcs	\$11.99	\$10.25	\$22.24
	18x12 E Propeller	1	Pcs	\$12.03	\$6.00	\$18.03
	16x9 Propeller	1	Pcs	\$20.99	\$1.31	\$22.30
					Total Cost	\$353.45
Donated or Reused Items						
	Hockey Sticks	4	Pcs	\$100	\$6.25	\$425
	Spektrum NX8	1	Pcs	\$400	\$25	\$425
	Spektrum Receiver	1	Pcs	\$40	\$2.50	\$42.5
	ESC	1	Pcs	\$109.99	\$6.87	\$116.86
	Motor	1	Pcs	\$126.99	\$7.94	\$134.93
	Battery	1	Pcs	\$25.28	\$7.16	\$32.44
					Total Cost	\$1,176.73

Table E-4. Makerspace Usage

Equipment	Cost Per Hour	Hours Used	Cost
Laser Cutter	\$35	1	\$35
3D Printer	\$35	70	\$2450
Total Cost		\$2485	

Appendix F – ME Departmental Supply and Resource Request Forms



Department of Mechanical and Industrial Engineering
One University Avenue
Lowell, Massachusetts 01854

Capstone Materials & Supplies Order Form

To be completed by Requester

Student Name(s): Gregory Beaudoin, Hamza Khan, Matthew Matta, Bradley Moore, Kyle Pierson, Patrick Rohleder, Will Russo, Ryan Webster

Course Section: 806

Capstone Instructor: David Willis

Capstone Project ID: ME-F24-12

Date: 25-October-2024

Description of Supplies: The supplies listed on the separately attached Bill of Materials are additional materials and supplies needed beyond the already supplemented materials from prior Design-Build-Fly Teams. These include various forms of wood board, foam board, 3D Printing Supplies, Various Hardware, and various electronics as needed.

Attach a complete bill of materials (including item description, part number, unit price, quantity, extended price, and supplier). An estimate of shipping costs must also be included. All items should be from UMass Lowell approved vendors (e.g., McMaster and Grainger). Total purchase cannot exceed \$100/enrolled student.

To be completed by Department Staff (Barrett O'Brien)

I certify that the total Bill of Materials of this project (including shipping estimate) is below the Department allotment, or otherwise approved by the Department Chair and that all items are from an approved UMass Lowell vendor.

Department Staff Signature: _____ **Date:** _____

To be completed by Capstone Instructor

I certify that the capstone project, including this form and associated attachments, has satisfactorily completed the design review, and is approved to use the described supplies and materials.

Capstone Instructor Signature: _____ **Date:** _____

Capstone Resource Form

To be completed by Requester

Student Name(s): Gregory Beaudoin, Hamza Khan, Matthew Matta, Bradley Moore, Kyle Pierson, Patrick Rohleider, Will Russo, Ryan Webster

Course Section: 806

Capstone Instructor: David Willis

Capstone Project ID: ME-F24-**12**

Date: 25-October-2024

Description of Resource(s)/Service(s): The team did not deem it necessary as we are not in need of any additional department resources or services.

Attach all drawings for machining/fabrication, electrical schematics for electronic fabrication, testing standards and test matrix. Include details about what you will be doing under supervision and what you need to have performed by department staff. Include number of parts or samples.

To be completed by Department Staff

Resource Hours Required:

Approximate Dates/Times:

Department Staff Signature: _____ **Date:** _____

To be completed by Capstone Instructor

I certify that the capstone project, including this form and associated attachments, has satisfactorily completed the design review, and is approved to use/arrange the described resources.

Capstone Instructor Signature: _____ **Date:** _____

Capstone student(s) should submit these forms to the department staff to schedule the resources or services only after signed by both the Department Staff and Capstone Instructor. Students should retain a copy of this form to include in Capstone Final Report.



Appendix G – Honors Add-On

Improving AIAA-DBF Performance by Analyzing, Designing, and Prototyping Carbon Fiber Parts

By

Matthew Matta
Kyle Pierson
Will Russo



SUBMITTED AS PART OF THE HONORS ADD-ON OF THE CAPSTONE PROJECT
HONORS REQUIREMENT FOR THE DEGREE OF B.S. IN MECHANICAL ENGINEERING

DEPARTMENT OF MECHANICAL ENGINEERING
FRANCIS COLLEGE OF ENGINEERING
UNIVERSITY OF MASSACHUSETTS LOWELL

12-DECEMBER-2024

Signatures of Authors:
Matthew Matta *Kyle Pierson* *William Russo*

Capstone Advisor: David Willis, Ph.D.
Professor, Associate Chair for Undergraduate Studies
Department of Mechanical and Industrial Engineering

Capstone Instructor: Patrick Drane, Ph.D.
Assistant Director of Baseball Research Center / Technical Program Manager
Department of Mechanical and Industrial Engineering

Honors Representative: Rae Mansfield, Ph.D.
Associate Director of Honors Scholarship and Curriculum
Honors College



1) Abstract:

The American Institute for Aeronautics and Astronautics, AIAA, hosts an annual event called the Design, Build, Fly competition. Every year, teams across the world are given a new mission set of one ground mission and three flight missions for a remote-controlled aircraft. The DBF competition for the 2024-2025 year is in Tucson, Arizona. The mission set for this year was based on the X-1 Super Sonic Flight Program which involves a larger aircraft flying a supersonic aircraft up to its flight window before releasing it with some initial velocity. In addition, the plane must carry bottles under the wings to simulate gas tanks.

This report focused on the design and analysis of carbon fiber landing gears of said aircraft. Several design iterations of these landing gears were created through SolidWorks CAD simultaneously with a series of ANSYS simulations. Through this process, it was ensured that the carbon fiber landing gear would meet safety factor regulations. Upon the completion of the landing gear designs, the team shifted its focus to manufacturing methods for epoxy/carbon fiber parts through push-down molds. After performing preliminary research and discussing various design options, the team manufactured and printed molds, to which the molding process underwent. The landing gear parts were molded successfully, but the team determined there would be room for future improvements.

2) Acknowledgments:

The team would like to express appreciation to Professor David Willis for helping guide the team through design and prototyping of the team's landing gear and its molds, as well as analyses performed on iterations on the designs of the landing gears. The team would also like to thank all members of the AIAA Design/Build/Fly Capstone team for their support and contributions to the efforts discussed in this report.

3) Table of Contents:

<i>1. Abstract</i>	2
<i>2. Acknowledgements</i>	2
<i>3. Table of Contents</i>	3
<i> 3.1 List of Figures</i>	3
<i> 3.2 List of Tables</i>	4
<i>4. Introduction</i>	5
<i> 4.1 Project Motivation</i>	5
<i> 4.2 Background Information</i>	5
<i> 4.3 Literature Review</i>	6
<i>5. Design Methodology</i>	7
<i> 5.1 Discussion of Procedures and Methods</i>	7
<i> 5.2 Design Considerations and Constraints</i>	8
<i> 5.3 Design Alternatives and Decision-Making Process</i>	9
<i> 5.3.1 Front and Rear Landing Gear</i>	9
<i> 5.3.2 Landing Gear Molds</i>	10
<i> 5.4 Design Iteration and Refinement</i>	11
<i> 5.5 Analysis Methods</i>	13
<i> 5.6 Evaluation of Design through Testing Procedures</i>	14
<i>6. Design Solution and Final Specification</i>	16
<i> 6.1 CAD Drawings</i>	16
<i> 6.2 Codes and Standards</i>	20
<i> 6.3 Critical Calculations and Analyses</i>	20
<i> 6.4 Final Design Solution</i>	22
<i>7. Results</i>	23
<i>8. Broader Impacts</i>	24
<i>9. Summary and Conclusion</i>	25
<i>10. Recommendations for Further Study</i>	26
<i>11. Bibliography</i>	28

3.1 List of Figures

<i>Figure 5.1. Rear Landing Gear Mold Option 1</i>	11
<i>Figure 5.2. Rear Landing Gear Mold Option 2</i>	11
<i>Figure 5.3. Rear Landing Gear Mold Option 3</i>	11
<i>Figure 5.4. Rear Landing Gear Push Down Mold</i>	11
<i>Figure 5.5. Design Iteration Examples for the Front Landing Gear</i>	12
<i>Figure 5.6. Design Iteration Examples for the Rear Landing Gear</i>	12
<i>Figure 5.7. Beam Bending Testing Apparatus</i>	14
<i>Figure 6.1. Front Landing Gear CAD Drawing</i>	16
<i>Figure 6.2. Rear Landing Gear CAD Drawing</i>	16
<i>Figure 6.3. Option 1 Rear Landing Gear Mold</i>	17
<i>Figure 6.4. Option 2 Rear Landing Gear Mold</i>	17
<i>Figure 6.5. Option 3 Rear Landing Gear Mold</i>	18
<i>Figure 6.6. Rear Push Down Mold</i>	18
<i>Figure 6.7. Front Push Down Mold</i>	19

<i>Figure 6.8. Front Landing Gear Mold.....</i>	19
<i>Figure 6.9. Replication of Beam Bending Test Conditions in ANSYS.....</i>	21
<i>Figure 6.10. Rear Landing Gear Mold Final Result</i>	23
<i>Figure 6.11. Front Landing Gear Mold Final Result.....</i>	23
<i>Figure 7.1. Front Landing Gear ANSYS Simulation Result</i>	23
<i>Figure 7.2. Rear Landing Gear ANSYS Simulation Result.....</i>	24

3.2 List of Tables

<i>Table 5.1: Landing Gear Considerations and Constraints</i>	8
<i>Table 5.2: Decision Matrix for Landing Gear Configuration</i>	9
<i>Table 5.3. Comparison of ANSYS Simulation and Beam Bending Testing.....</i>	15
<i>Table 6.1. Codes and Standards</i>	20

4) Introduction:

4.1 Project Motivation

One of the main objectives of the competition is to carry the most weight possible. Due to this, the team wanted to build strong landing gear to carry the final decided weight of the plane and bottles which provided a good project to hand off to the honors section of the Capstone team. In addition, the team was interested in learning more about composite materials manufacturing which led to carbon fiber as the material choice.

4.2 Background Information

The AIAA Design/Build/Fly (DBF) competition is a yearly competition where universities from around the world design, build, and fly a radio-controlled (RC) aircraft, each year holding a unique set of mission requirements. For the 2024-2025 iteration of this competition, the goal was to simulate an X-1 Supersonic Flight Program [1]. This represents the controlled test of an experimental aircraft, which is typically performed by utilizing a larger aircraft with a higher flight ceiling. Among the several components of every aircraft is its landing gear. When designing the landing gear for an RC aircraft, it is imperative that it can withstand the anticipated loads upon landing.

Historically, capstone teams who work on this project have purchased off-the-shelf carbon fiber landing gears. In selecting carbon fiber as the landing gear material, which has a high strength-to-weight ratio, it was ensured that the landing gear did not significantly deflect upon landing. While previous teams have successfully integrated landing gear into an aircraft design, whether or not the landing gear has held up through the entire competition has been a consistent concern [2]. As a result, the team decided it would be best to design and manufacture its own landing gear for this year's competition.

Due to this decision, the team used this as an opportunity to learn about the nature of epoxy/carbon fiber composite materials, as well as manufacturing methods to mold the landing



gear. In addition to a reasonably high strength-to-weight ratio, these epoxy/carbon fiber composites are incredibly lightweight, which is beneficial for the overall weight of the plane, and is also incredibly stiff. In the context of ensuring the landing gear can withstand the loads associated with landing the aircraft, this was beneficial because it allowed the team to over-design the landing gear with minimal concern of weighing down the plane. Although this material is ideal for most RC aircraft applications, it is rather expensive. With that, the team aimed to find a balance between making sure the landing gear was robust enough to handle all anticipated external loads during the competition while not wasting unnecessary material in the manufacturing process. When taking these considerations into account, the team performed several iterations of Computer Aided Design (CAD) Models and ANSYS Finite Element Analysis (FEA) Simulations.

4.3 Literature Review:

The landing gear and its molds were designed and analyzed utilizing knowledge obtained from the following courses taken at the University of Massachusetts Lowell:

1. MECH.2010 – Computer Aided Design: SolidWorks was utilized to create a CAD model of all landing gears and the molds.
2. MECH.2020 – Manufacturing Laboratory: Manufacturing techniques as they pertain to molding fundamentals were learned in this course that aided the landing gear mold design.
3. MECH.2960 – Material Science for Engineers: Key takeaways from this course relate to the understanding of the isotropic and anisotropic nature of specific materials, and allowed the team to understand the material properties and behaviors of carbon fiber.
4. MECH.3110 – Applied Strength of Materials: The loading conditions on the landing gear during landing were analyzed using ANSYS, a software used commonly during this course.
5. MECH.4510 – Dynamics Systems Analysis: Modeled different force patterns commonly seen in dynamic systems to predict system behavior.

5 Design Methodology:

5.1 Discussion of Procedures and Methods

The team identified that the aircraft required an epoxy/carbon fiber front and rear landing gear that was lightweight and stiff which did not lead to the damage of other aircraft components upon landing. Approaching the design and analysis of these landing gear components required a general understanding of landing gear designs. To start, the team analyzed the reports of high-performing teams from previous competitions to gain an understanding of how front and rear landing gear is designed and integrated into the broader design of the aircraft. In doing so, this provided opportunities to compare different designs and incorporate improvements. To decide between alternative designs, the team utilized decision matrices to determine the final configuration of the landing gear designs. The team decided that a taildragger landing gear configuration would be best suited for the project requirements based on the considerations that were weighed in the decision-making process. The requirements developed for the front and rear landing gear components were driven primarily by safety factors, in which the landing gear must not fail up to the load limit of the carbon fiber. These safety factors were developed based on team discussion, guidance from the team's advisor, and applicable engineering codes and standards.

Determining these requirements led to beginning the preliminary designs of the landing gear using SolidWorks CAD software. Upon each iteration of the front and rear landing gear designs, ANSYS Finite Element Analysis was utilized to determine whether the landing gear held up against the aforementioned safety factors. At this stage, it was decided would model the carbon fiber as an isotropic material even though most variations of carbon fiber display anisotropic material properties. The team made this assumption primarily to simplify the analysis process, which included inputting certain material properties into an "Isotropic" material window.

After the designs of the front and rear landing gear were finalized, the team pivoted to the design of the molds for the landing gear, which led to the manufacturing of the landing gear. This stage of the design process allowed the team to research different manufacturing methods utilized in

epoxy/carbon fiber molds. Upon completion of this research, the team then designed 3 configurations of potential molds for the rear landing gear. This was done due to the rear landing gear being much smaller leading it to be quicker and easier to manufacture. Once the 3 configurations were tested, the best option was picked to be recreated for the mold of the front landing gear. The front landing gear mold was created and printed and then used to create the front landing gear.

5.2 Design Considerations and Constraints

Along with the team's design considerations and constraints shown in Table 5.1, which were set by AIAA, the team also acknowledged public health, safety and welfare while also evaluating economic factors [3]. The team understood that the usage of epoxy/carbon fiber composites in manufacturing could be hazardous and therefore dangerous to human health and safety. It was imperative to understand that Personal Protective Equipment (PPE) be used when this material was handled, which included avoiding inhaling the potential toxic fumes as safely as possible [4]. As previously mentioned, safety factors were heavily considered in the design of the landing gear to protect aircraft operators and other spectators of the aircraft's flight in the event of potential failures. The team also discussed minimizing the size, particularly the thickness, of the landing gear as much as possible to ensure that the team limits the use of the epoxy/carbon fiber composite in the sense that the design is cost-effective.

Table 5.1. Landing Gear Considerations and Constraints

Consideration	Constraints
Clearance	Front and rear landing gear must be tall enough to prop the fuselage up at least 4 inches off the ground to fit the height of the X-1 glider.
Geometry	Front and rear landing gear must be at least 12 inches apart to be able to fit the length of the glider.
Strength	Both landing gear components must hold up to a minimum safety factor of at least 1.5.

5.3 Design Alternatives and Decision-Making Process

5.3.1 Front and Rear Landing Gear

The choice to use a taildragger landing gear configuration was based on a decision matrix, which weighed the cost, ease of manufacturing, weight, and durability as important design factors. The breakdown of the different factors that were examined in the selection are shown in Table 5.2 below.

Table 5.2. Decision Matrix for Landing Gear Configuration

	Criteria 1	Criteria 2	Criteria 3	Criteria 4	
CRITERIA	Cost	Ease of Manufacturing	Weight	Durability	
WEIGHT	20%	30%	30%	20%	100%
OPTIONS	Criteria 1 SCORES	Criteria 2 SCORES	Criteria 3 SCORES	Criteria 4 SCORES	
Tricycle	8.33	7.33	7.67	7.00	7.57
Tail Dragger	7.33	7.67	7.00	8.67	7.60
Quadricycle	5.00	5.67	4.33	5.00	5.00

Ultimately, the main deciding factor in choosing a tail dragger configuration is the increased durability. An important note about the competition is that the landing space for the plane is anticipated to be very rough and uneven. Generally, taildraggers are considered better equipped for this type of situation which is reflected in the durability score for each configuration [5].

Once the landing gear configuration was determined, preliminary CAD models were created to simulate different loading conditions in ANSYS. An important consideration made during the design process was leaving enough space underneath the fuselage to fit the X-1 glider. The approximate size of the glider was predicted to be about 12" long and 4" tall, so the dimensions of the landing gear were set accordingly to account for this.

Determining the setup for the ANSYS simulations required predicting the impact force that the landing gear would experience. Given the uncertain flying conditions for the competition, the team decided to take a very conservative approach to the setup. As previously mentioned, carbon

fiber being a lightweight material was useful because it allowed the team to overdesign the landing gear to an extreme degree without concern of adding substantial weight to the aircraft. The team ultimately decided that the front landing gear should be able to withstand 10 times the weight of the plane and the rear landing gear should be able to withstand 4 times the weight of the plane. Based on historical data collected from previous reports, the target weight of the aircraft was set at 15 pounds, so 150 lbf and 60 lbf were applied to the front and rear landing gear respectively [1].

5.3.2 Landing Gear Molds

The goal of the landing gear molds were to allow for a quick and easy ability to manufacture the designed front and rear landing gear. To do this, multiple designs were considered from how to make the molds to however, due to the resources available, 3D printing the molds was decided as it was free through the Makerspace while being relatively fast to create. The rear mold was designed first due to it being substantially smaller which meant that it could be used to test different molds designs before creating the front landing gear mold. 3 designs were taken into account, with the first design relying on 2 halves being pushed together to form a cavity that would be filled with epoxy and carbon fiber, as displayed in Figure 5.1. They would be locked together by cutting an inlet into each half with each half also having an extrusion that creates a snug fit to lock them into position [6]. To create the screw holes, an extrusion was made on the mold to simulate the screw holes so that the carbon fiber would settle around them. The 2nd design, shown in Figure 5.2, relied on a 3rd part in the form of a base to lock in the 2 halves instead of using the inlet and extrusion method. The extrusion to create the screw holes was also used for the 2nd design.

The third design, shown in Figure 5.3. was the same as the first with the only difference being how the screw holes were made. Instead of creating an extrusion, holes were instead made so that rods or screws could be pushed through the mold and cavity to create the screw holes. All 3 used the same push down mold. The 3rd design was chosen as the best solution due to the ease of taking off the mold whence the carbon fiber has settled. This was due to the rods being able to be removed without the risk of them snapping or breaking, as was the case with the 3D printed

holes due to them being not as strong. In addition, due to the 3D printed hole halves not perfectly fitting against each other, there was more post processing in the form of sanding that would have to be done. If it was required, by using screws in the holes, threads could also be created in the carbon fiber which offers additional options for the final design of how the landing gear would be connected to the plane. Once the rear landing gear design was selected, the same design methodology was applied to the mold design of the front landing gear.

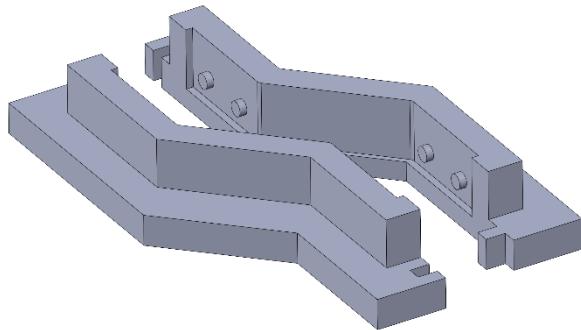


Figure 5.1. Rear Landing Gear Mold Option 1

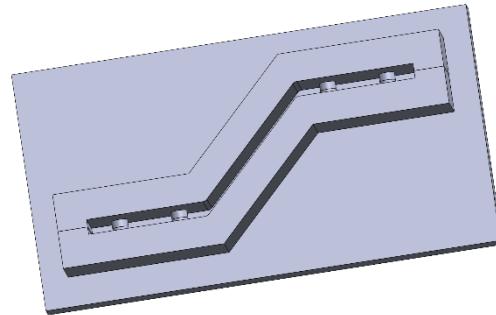


Figure 5.2. Rear Landing Gear Mold Option 2

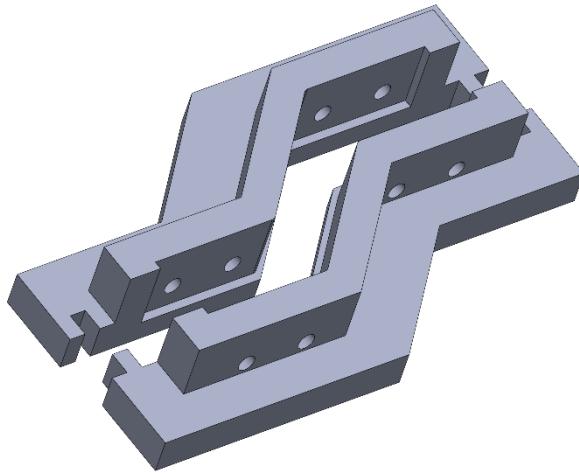


Figure 5.3. Rear Landing Gear Mold Option 3

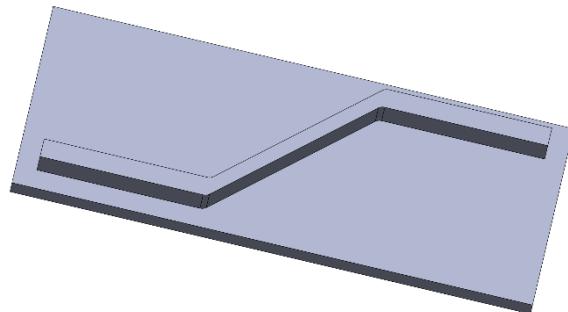


Figure 5.4. Rear Landing Gear Push-Down Mold

5.4 Design Iteration and Refinement

Iterations to the landing gear were made based on the results of the ANSYS simulations and feedback from the other sub-teams. As previously mentioned, the landing gear was intentionally designed to withstand more weight than is expected during the actual mission landings due to how critical the landing gear is to the durability of the aircraft. When doing the simulations,

target safety factors of 10 and 3 for the front and rear landing gear were chosen respectively based on the weights applied to each component. This was more than was required per the relevant standards, but the team wanted to ensure the landing gear would be able to hold up under the anticipated loading conditions. When running the simulations for each component, the dimensions of the landing gear were altered iteratively until the safety factors approached these target values. Figures 5.5 and 5.6 compare the original designs of the front and rear landing gear to the respective final designs.

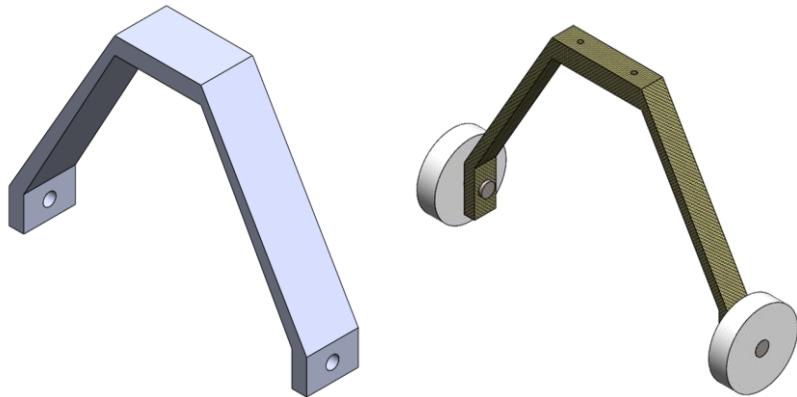


Figure 5.5. Design Iteration Examples for the Front Landing Gear

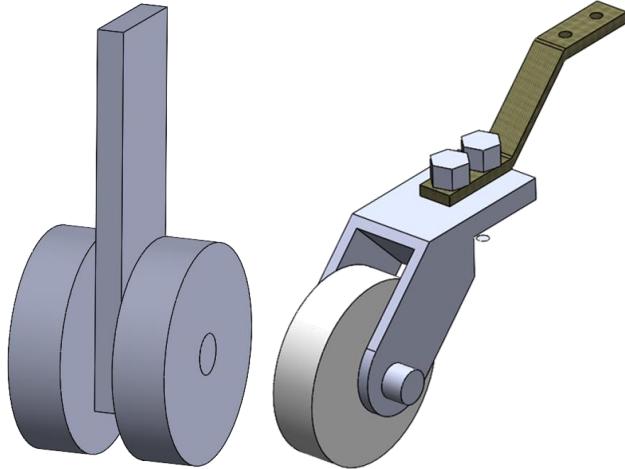


Figure 5.6. Design Iteration Examples for the Rear Landing Gear

5.5 Analysis Methods

The majority of the work done in the analysis involved analyzing external loading conditions on the components of the landing gear. This required comprehensive knowledge of Applied Strength of Materials and Material Science because the constraints and expected failure modes were important considerations for the system setup. Design for Manufacturing (DFM) considerations also drove the analysis as it was critical that the components were designed in a way that they could actually be integrated into the body of the aircraft. Finally, the ability to analyze the pros and cons of different design alternatives to make choices that best suit the team's project, and timeline was a driver of the analysis.

One assumption in the design process was modeling the carbon fiber as an isotropic material. Typically, carbon fiber components have the fibers aligned in the same orientation, leading to anisotropic material properties. However, the landing gear for this application was made from forged carbon fiber, which led to the fibers having varying orientations across the profile of each component. For this reason, the team decided to use isotropic properties for the carbon fiber components under the assumption that the material properties along each axis of the landing gear components would be similar due to the different orientations of the fibers [7]. This was considered a reasonable assumption by the team because even though this would slightly compromise the overall strength of the landing gear, it would increase the consistency of behavior across the different axes making it easier to predict in the simulations.

Another assumption that was made was the impact forces that would be applied to each component of the landing gear. Originally, the impact force was calculated by using Equation 1 below [8].

$$mV_y = F_y \Delta t \quad (1)$$

$$F_y = \frac{m_y V}{\Delta t}$$

Originally, the angle of attack was approximated at 3.5° based on the preliminary dimensions and locations of the landing gear components. After consulting with the aerodynamics sub-team, MECH.4230-806 Fall 2024 Capstone Design

the ideal angle of attack changed multiple times as the overall design of the aircraft changed. For this reason, the team eventually deviated from this method of force calculation and settled on a conservative estimate of 150 lbf for the front landing gear and 60 lbf on the rear landing gear under the assumption that the front landing gear would absorb the majority of the impact force. As a result of these preliminary steps, the team decided to utilize ANSYS Finite Element Analysis software as well as beam bending validation testing to determine the effects on the landing gear per the aforementioned conditions.

5.6 Evaluation of Design through Testing Procedures

Once the CAD models were created, ANSYS simulations were conducted on the load-bearing components of the aircraft. These simulations were used to predict how the landing gear structure would hold up during landing. The only testing procedures used during this process involved a beam bending test with a 2.50" x 20.25" x 0.50" aluminum bar. This test setup, displayed in Figure 5.7, was not used directly to validate the landing gear simulations, but rather to reinforce the team's ability to accurately model external forces acting on aircraft components.

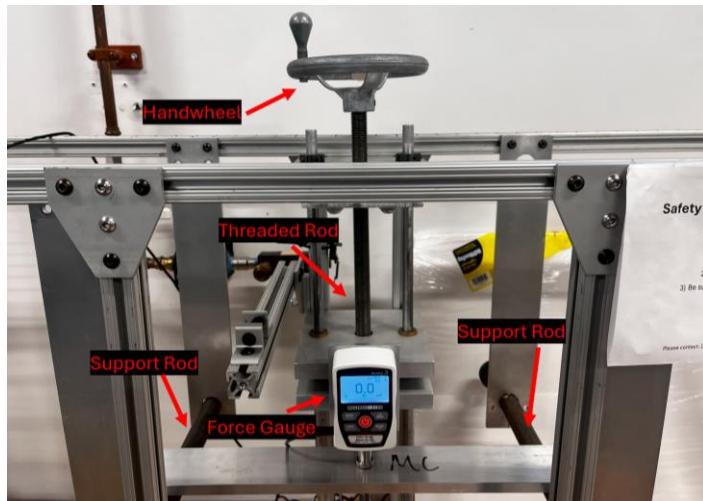


Figure 5.7. Beam Bending Testing Apparatus

The expected calculated deflection of the beam was calculated based on the geometric and material properties of the beam for further validation. The calculated deflection, which is representative of the beam deflection with two simple supports, is shown in Equation 2 [9].

$$\delta = \frac{F * L^3}{48 * E * I} \quad (2)$$

The accuracy of the simulation with aluminum provided confidence that the same could be achieved for carbon fiber. Plus, it is more difficult to measure the deflection of carbon fiber with the provided setup, shown in Figure 5.7. In this setup, both ends of the beam were constrained constraining the ends and a 100 lbf downward force was applied in the middle. An ANSYS simulation was set up with the same dimensions and material properties as the actual beam that was tested for the deflection comparison.

Table 5.3. Comparison of ANSYS Simulation and Beam Bending Testing

	FEA	Hand Calculations
Expected Result (in.)	0.0563	0.0671
Measured Result (in.)	0.0625	0.0625
% error	10.97%	6.800%

This comparison yielded a 10.97% discrepancy with the FEA, and 6.800% as a result of the formula in Equation 2, which was deemed acceptable as there were likely imperfections in the actual beam as well as lack of measurement with a precise tool. While Aluminum 6061 material properties were utilized in the FEA and hand calculations, these true properties were difficult to account for in the real life model, as that model may have different properties based on differences in manufacturing or wear [10]. A ruler was also used to measure the deflection as opposed to a more precise tool such as a dial indicator. Considering that the deflection is on a small scale, this error was considered negligible. Based on the results of this test setup validation, it was deemed that the ANSYS simulations performed using the landing gear was accurate.

Simulations were also conducted on the front and rear landing gear by fixing the top of the landing gear and placing a 150 lbf upward force on the wheels, which is further discussed in Section 7. Per standard 14CFR 25.303, all components of an aircraft that are flight tested should be able to withstand 1.5 or more times the loads applied to it [3]. For the landing gear specifically, it was decided that the front gear should be built to a safety factor of at least 10 and the rear landing gear should be built to a safety factor of at least 3. This was done for a few

different reasons. As previously mentioned, the landing gear, specifically the front landing gear will absorb the majority of the force during landing, so any damage during the missions would cause significant issues for the rest of the aircraft. Additionally, the landing gear was modeled as an isotropic material even though carbon fiber is generally anisotropic due to the orientation of the fibers [7]. The method of manufacturing the landing gear resulted in fibers that had varying orientations, so the decision to model the landing gear as isotropic was made under the assumption that it would be more conservative and a more accurate simulation on average.

6 Design Solution and Final Specifications:

6.1 CAD Drawings

The CAD drawing for the final front landing gear, final rear landing gear, and their corresponding molds can be seen in Figures 6.1-6.8. The front and rear landing gear can be seen in 6.1 and 6.2. The 3 rear mold configurations Figure 6.3-6.5 with 6.5 being the configuration chosen to be used when making the front landing gear mold. The push down mold that was used for all rear landing gear configurations is Figure 6.1 with the front push down mold being Figure 6.7. Finally, the front landing gear mold is Figure 6.8.

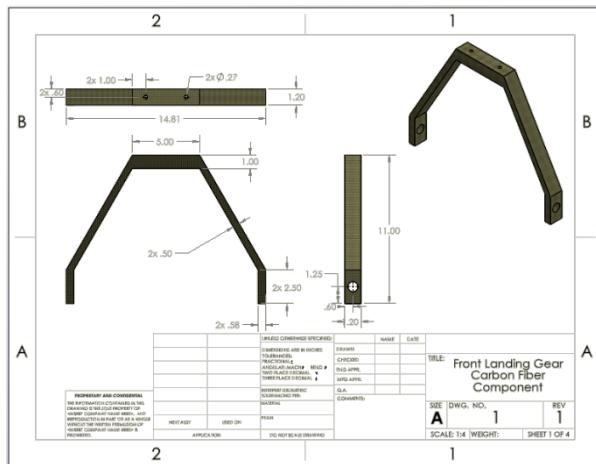


Figure 6.1. Front Landing Gear CAD Drawing

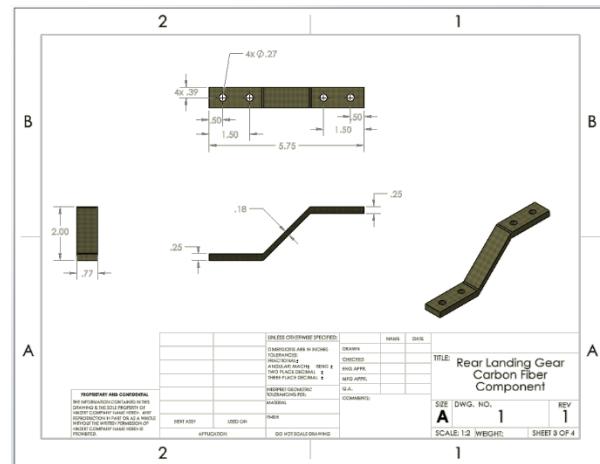


Figure 6.2. Rear Landing Gear CAD Drawing

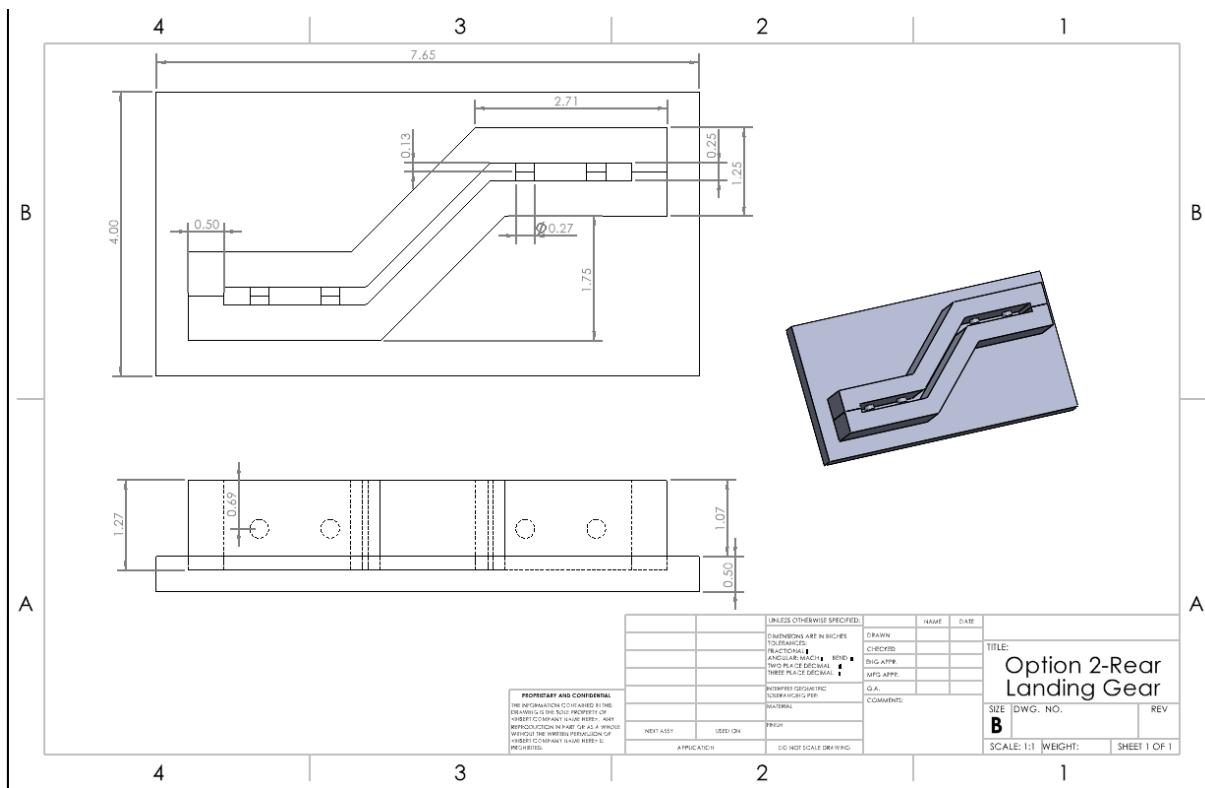


Figure 6.3. Option 1 Rear Landing Gear Mold

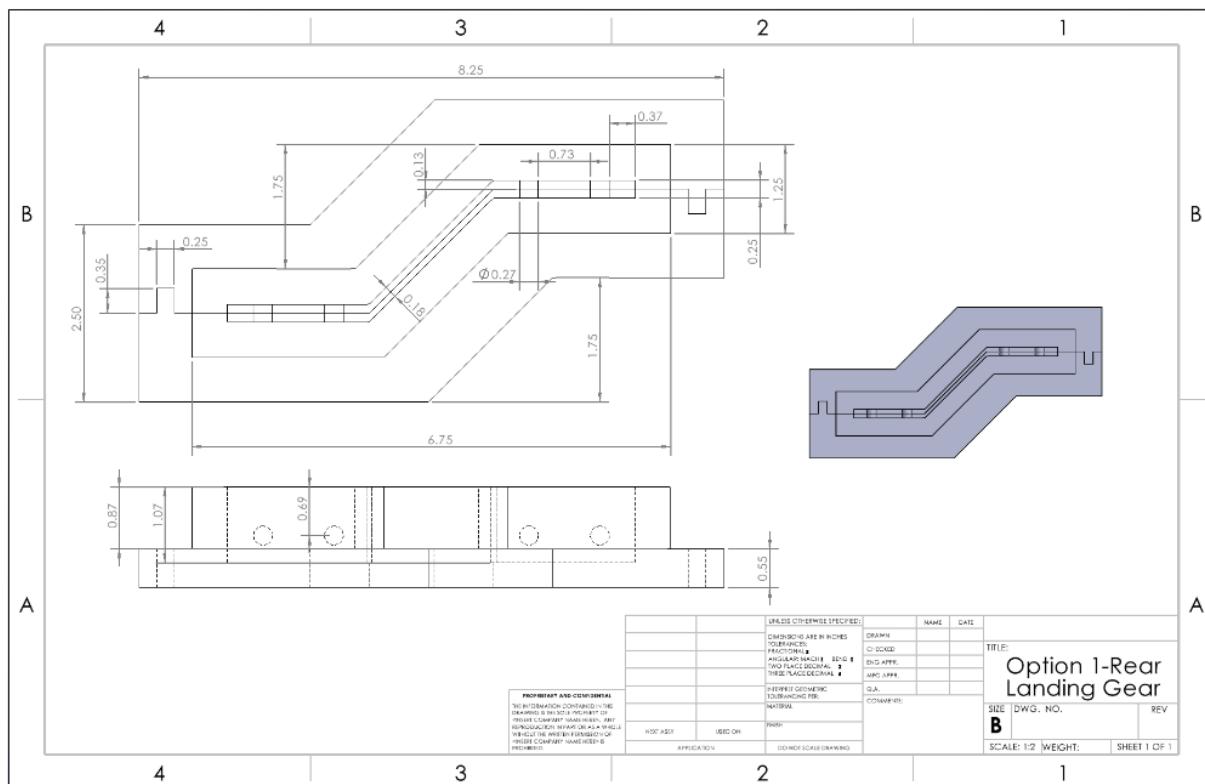


Figure 6.4. Option 2 Rear Landing Gear Mold

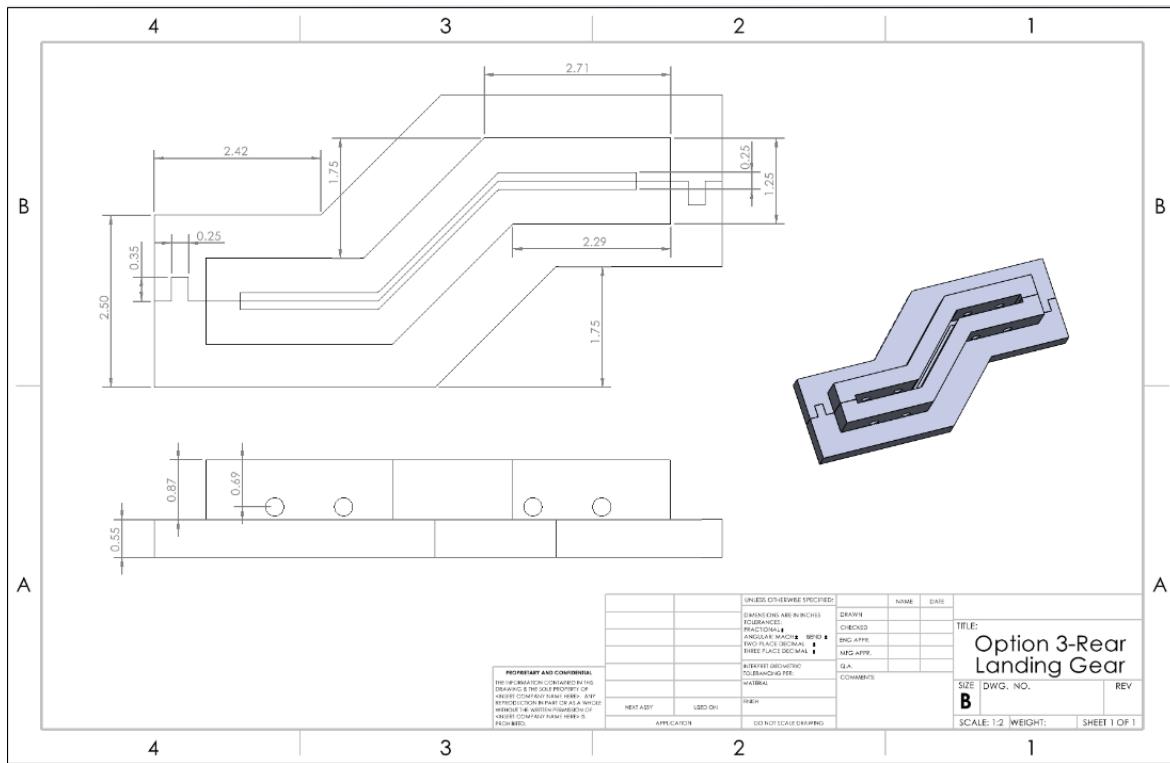


Figure 6.5. Option 3 Rear Landing Gear Mold

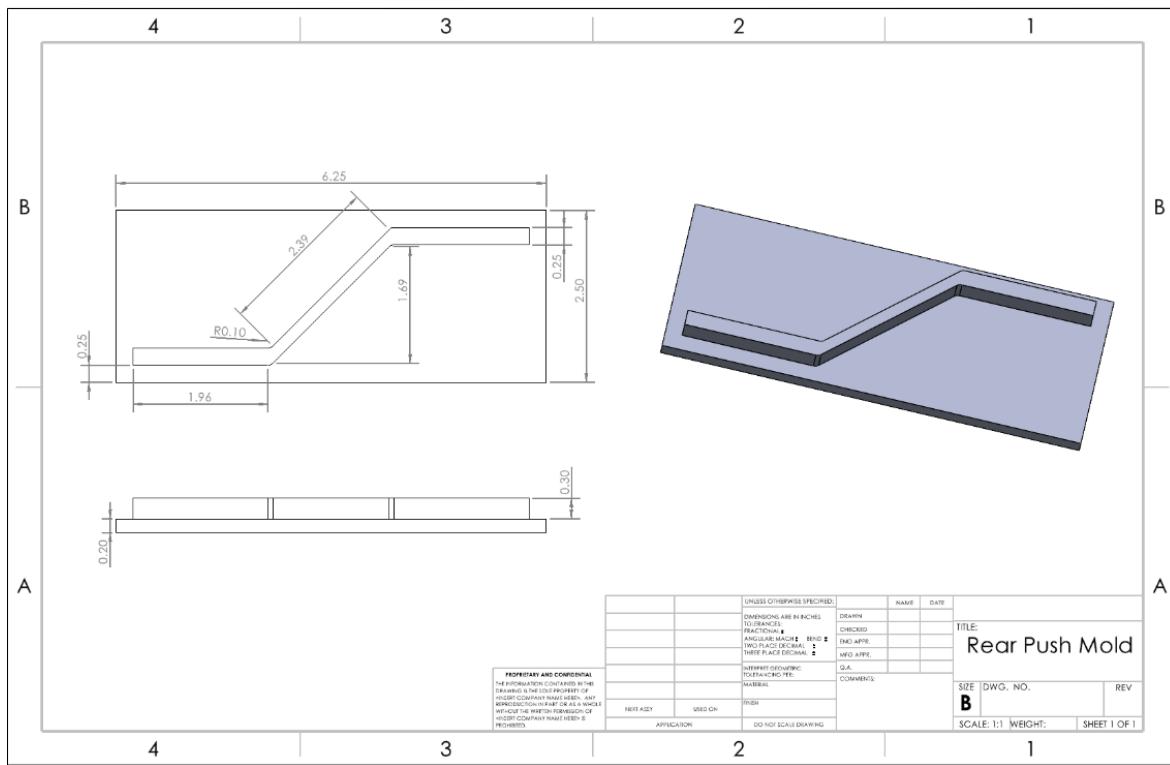


Figure 6.6. Rear Push Down Mold

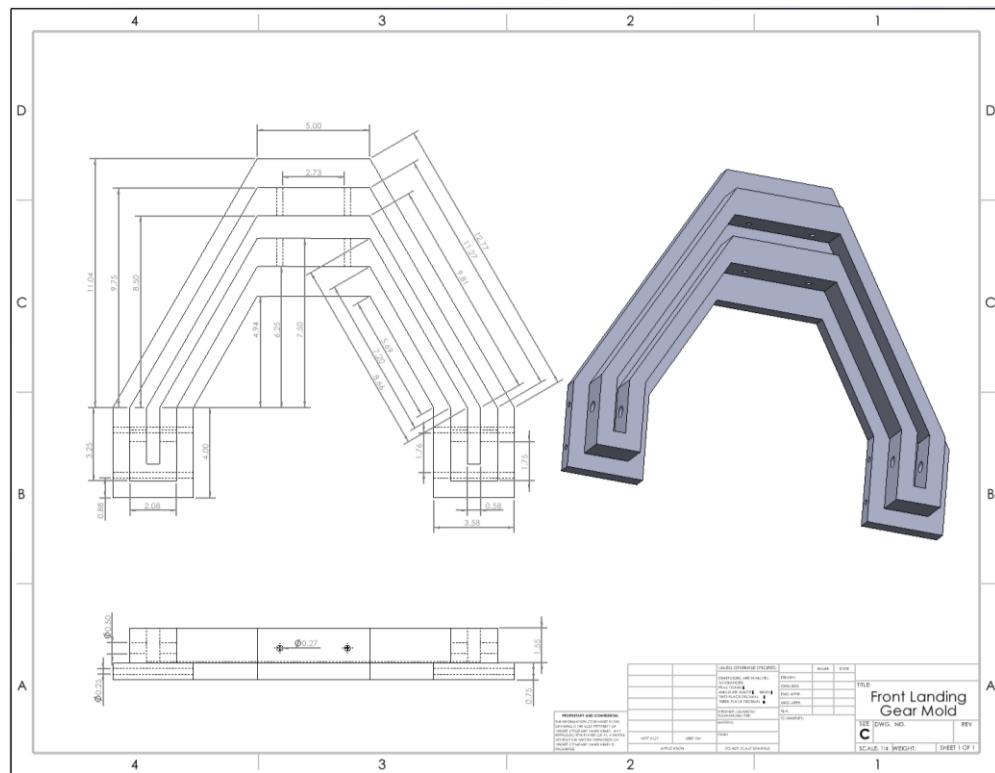


Figure 6.7. Front Push Down Mold

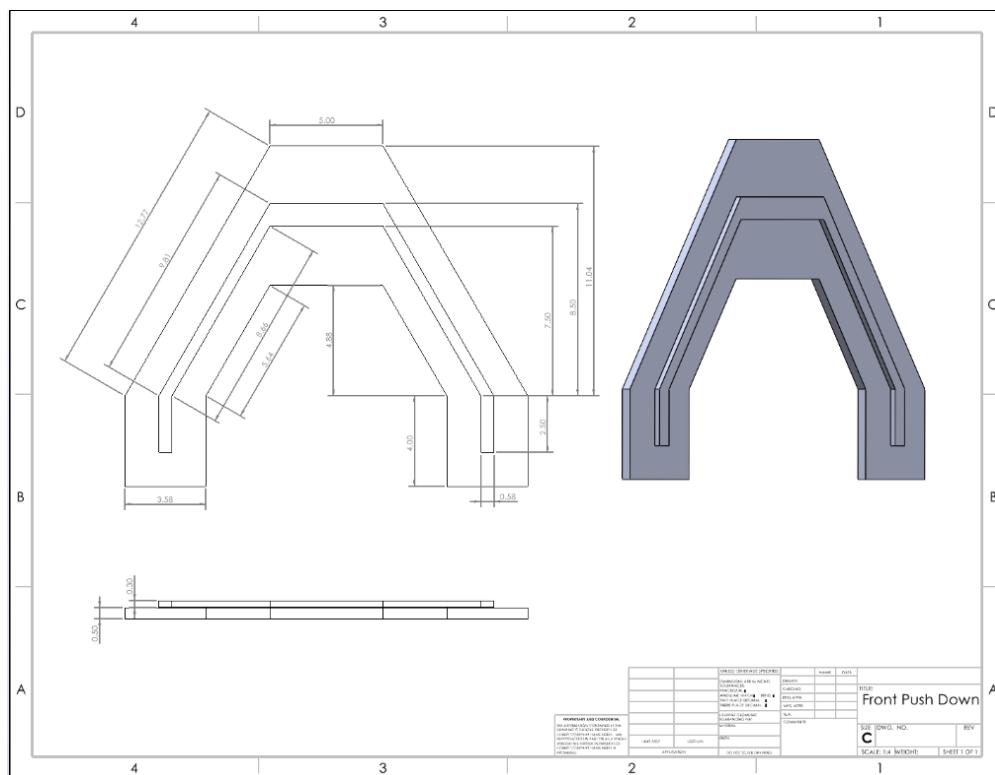


Figure 6.8. Front Landing Gear Mold

6.2 Codes and Standards

There are many codes and standards set by the FAA and other agencies that are required to be followed to ensure safe flight. The codes that were most important for this project were centered around the licenses required to fly the type of aircraft that was being built as well as following the necessary safety factor regulations that are necessary for all aircraft. The overarching regulation for flying an RC aircraft over 0.5 pounds is Part 107 as that is the license required to fly this aircraft. For manufacturing the Federal Airworthiness Regulation requires a safety factor of at least 1.5 for all parts of an aircraft. There is not anything specific for aircraft for screw regulations but there are general ASTM standards that were followed which are general requirements for the screws and bolts being used.

Table 6.1. Codes and Standards

Code/Standard	What Does the Code or Standard Say
Part 107 [11]	<ul style="list-style-type: none"> • License for flying aircraft over 250g • Aircraft must be registered with FAA • Aircraft must have a transponder if not flown in a FRIA
Remote ID Information [12]	If not flying in FRIA site, aircraft must broadcast: <ul style="list-style-type: none"> • Drone ID (Serial Number) • Drone location and altitude • Drone Velocity • Takeoff Location and elevation • Time Mark
14 CFR § 25.303 Factor of Safety [3]	Federal Airworthiness Regulation Part 25.303 says that a safety factor of 1.5 must be applied to the prescribed limit loads that are considered external loads on the structure
ASTM F593 [13]	This specification covers the requirements for stainless steel bolts, hex cap screws, and studs 0.25 to 1.50 in., inclusive, in nominal diameter in a number of alloys in common use and intended for service applications requiring general corrosion resistance

6.3 Critical Calculations and Analyses

As discussed in Section 5.5, the team performed calculations to validate the beam bending test setup. With deformation being the simplest quantity to measure based on the equipment used, deformation formulas were utilized to calculate the expected deformation based on the conditions in the true test. Displayed in Figure 6.9 are the conditions of the beam bending test as set up in ANSYS.

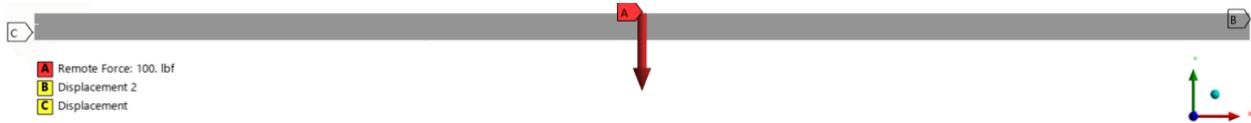


Figure 6.9. Replication of Beam Bending Test Conditions in ANSYS

As shown in Figure 6.9, the two displacements are representative of the conditions in the true beam bending test, where the beam was free to move in the x-direction, but fixed in the y and z directions. Equation 2 is the deflection of a beam with two simple supports yields the expected deflection, which is dependent on material properties and geometry of the beam, shown below [9]. The two “simple supports” are the two displacements shown in Figure 6.9.

$$\delta = \frac{F * L^3}{48 * E * I} = \frac{F * L^3}{48 * E * \left(\frac{1}{12} b h^3\right)} = \frac{F * L^3}{4 * E * b h^3}$$

$$\delta = \frac{(100 \text{ lbf}) * (20.25 \text{ in.})^3}{4 * \left(9906077 \frac{\text{lbf}}{\text{in.}^2}\right) * (2.5 \text{ in.})(0.5 \text{ in.})^3} \quad \delta = \mathbf{0.0671 \text{ in.}}$$

Another critical calculation was the impact force being applied to the landing gear components. Initially, this was modeled as an impulse force based on an angle of attack of 3.5° as shown in Equation 1 below [8]. It is important to note that although this equation was used to gain an initial understanding of the magnitude of the impact forces, the angle of attack changed multiple times during the aerodynamic analyses so the team decided that using 10 G's (150 lbf) and 6 G's (60 lbf) on the front and rear landing gear respectively would be a conservative estimate that did not depend on the angle of attack.

$$\begin{aligned} mV_y &= F_y \Delta t \\ F_y &= \frac{m_y V}{\Delta t} \\ F_y &= \frac{(6.80 \text{ kg}) \left(\left(30.48 \frac{\text{m}}{\text{s}}\right) \sin(3.5^\circ) \right)}{0.025 \text{ s}} \\ F_y &= \mathbf{506.12 \text{ N} = 113.88 \text{ lbf}} \end{aligned}$$

Safety factor calculations were also used to gauge if the dimensions were appropriate for each landing gear component. To do this, the equivalent stresses output by the ANSYS simulations were compared to the ultimate stress of carbon fiber composite materials. The calculated safety factors of the front and rear landing gear components are shown in Equation 3 below [14].

$$SF = \frac{\sigma_{ultimate}}{\sigma_{allowable}} \quad (3)$$

$$SF_{front} = \frac{\sigma_{ultimate}}{\sigma_{allowable}}$$

$$SF_{front} = \frac{139961 \text{ psi}}{13329 \text{ psi}}$$

$$\mathbf{SF_{front} = 10.5}$$

$$SF_{rear} = \frac{\sigma_{ultimate}}{\sigma_{allowable}}$$

$$SF_{rear} = \frac{139961 \text{ psi}}{41342 \text{ psi}}$$

$$\mathbf{SF_{rear} = 3.38}$$

6.4 Final Design Solution

The biggest consideration when making the landing gear was the safety factor as the primary reason for making the landing gear was to make sure the plane would not crash when landing. In addition, the height and distance between the front and rear landing gear were considered so that the glider would be able to fit beneath the aircraft. By building the landing gear, the team was able to apply loads to different designs until the simulations showed an acceptable amount of deformation which showed that the landing gear would be able to survive a worst-case scenario when landing. Once the designs were chosen molds were created and the landing gears were made.



Figure 6.10. Rear Landing Gear Mold Final Result



Figure 6.11. Front Landing Gear Mold Final Result

7 Results:

For each component of the landing gear, target safety factors were determined that guided the analysis. Per Standard 14 CFR § 25.303 Factor of Safety, load-bearing components of aircraft must have a safety factor of 1.5 [3]. The team decided that due to the importance of the landing gear to the overall durability of the aircraft, target safety factors of 10 and 3 were chosen for the front and rear landing gear respectively based on the expected distribution of impact force. For the ANSYS simulations, testing was done iteratively to see how close the safety factors would be to the target values based on the current geometry. From there, the dimensions were altered slightly between each test until the safety factor was close to the target values. Images of the simulation results are shown in Figures 7.1 and 7.2.

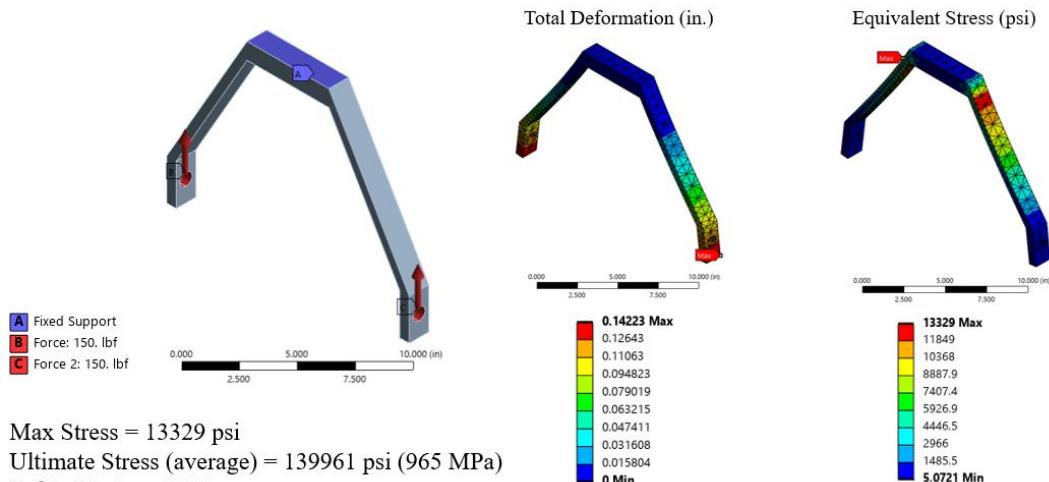


Figure 7.1. Front Landing Gear ANSYS Simulation Result

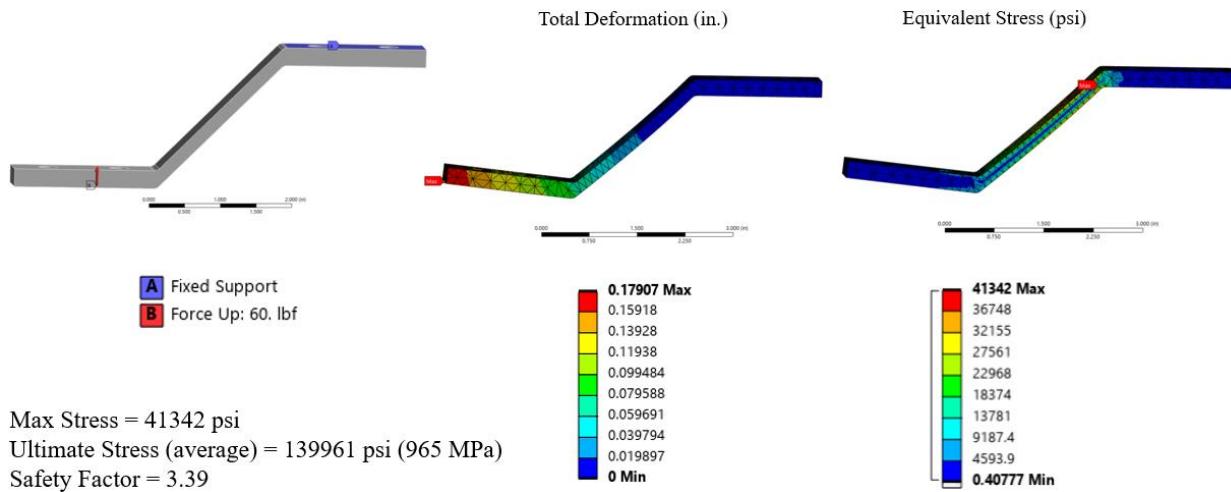


Figure 7.2. Rear Landing Gear ANSYS Simulation Result

Looking at the deformation, the goal of the landing gear was to have some deformation to allow the impact force to be absorbed by the landing gear more effectively. The team decided that the amount of deformation experienced by each component was acceptable based on the total size of the components and the requirements for accommodating the X-1 glider underneath the fuselage. The stress values were also considered acceptable based on the ultimate stress of carbon fiber and the proximity of the safety factors to the pre-determined target values [7].

8 Broader Impacts:

During the design, analysis, and manufacturing, the team acknowledged the potential implications as it pertains to global, economic, environmental and social contexts. While the team considered global contexts, the scope of the development of epoxy/carbon fiber composite landing gears for this project would not involve any large-scale collaboration, production, or distribution, so global impacts were not a primary concern.

The environmental implications of the carbon fiber landing gear were considerably significant to examine. All carbon fiber composite parts, while recyclable, take hundreds of years to decompose, which leaves a significant carbon footprint. For future teams that choose to refine this design, it is crucial to understand that manufacturing with carbon fiber composites is highly

dangerous, as the curing process produces harmful emissions [4]. Despite these shortcomings, the lightweight nature of this material indicates that energy consumption is significantly less, and with that, requires much less energy to recycle. It is critical that the final design solution of the landing gears are properly reused and recycled as necessary if future teams are to consider creating their own design iterations.

Economic implications of the landing gear pertain improvements from the Spring 2024 AIAA Design/Build/Fly (DBF) team. According to the previous team, "...further study of landing gear be done for future design iterations. This could prevent the time and money spent on fixing the landing gear and design after each failed test flight attempt" [2]. Although the team designed a thicker landing gear than was originally discussed, the team was able to develop front and rear landing gears that were incredibly durable, and therefore, reduces the financial burdens that future teams undergo as frequent repairs and tests are not expected. By strengthening the landing gear design comparatively to previous teams, it allows future teams to not only refine the design outlined in this report if they choose, but they may also invest time, money, and other resources into other aspects of the aircraft.

Participating in the structures sub-team as part of the Fall 2024 AIAA Design/Build/Fly (DBF) allowed the team to work toward custom landing gears, which no prior DBF team had achieved or sought after. The team was able to develop new capabilities and technical knowledge as it pertains to designing carbon fiber composite molds for custom rear and front landing gears. The final design solution allows future teams to build upon and refine the designs outlined in this report, which contributes to making new discoveries. This result has set a new standard for future teams and promotes positive learning and effective team collaboration.

9 Summary and Conclusion:

After fabrication of the final design solution for the landing gears and its molds, the team was able to successfully meet the primary design considerations and constraints. The team's goal was to create a durable, lightweight, and cost-effective set of epoxy/carbon fiber composite landing gears that met FAA safety factor regulations. Both final design solutions of the landing gears met

the constraints set out by AIAA, which involved clearance for the X-1 glider below the primary aircraft as well as its length. An iterative design process using SolidWorks CAD to design the landing gear and its molds were used [1]. After performing research on common landing gear designs, iteration was performed for both landing gears by cycling through a process of various CAD designs and various ANSYS simulations. As a result of these simulations, the front and rear landing gears achieved safety factors of 10 and 3, respectively, which exceeded the FAA regulation of 1.5 for all aircraft components [3]. Considering that carbon fiber is a lightweight material, it allowed the team to overdesign the landing gear to an extreme degree without concern of adding substantial weight to the aircraft, however, the thickness could have been lowered and still would meet FAA regulations.

The resulting push-down landing gear molds reflected the feasibility of being able to manufacture the landing gear accurately and successfully. Iterations of various landing gear molds yielded two final push-down molds that utilized a set of rods that were pushed through the mold and cavity to create screw holes. The resulting manufacturing of the rear landing gear provided insight into the creation of the front landing gear due to the simpler mold design. In doing so, design for manufacturing (DFM) principles were met such that the landing gear was successfully created with minimal error. The resulting molds successfully yielded a set of landing gears that met all project considerations and constraints.

10 Recommendations for Further Study:

For future teams who wish to pursue this work further, several recommendations have been for the front and rear landing gear designs to be improved. For instance, several codes and standards, such as FAA 14 CFR § 25.303 suggest that a safety factor of 1.5 be used for the landing gear as well as other landing gear components [3]. With safety factors including formulas of axial stress, which is dependent on cross-sectional area, and therefore, thickness, future teams may consider lowering the thickness of future landing gears. In reducing the thickness of the landing gear, safety factors are also being decreased. While the team had safety factors of the front and rear landing gears of 10 and 3, respectively, it would be practical to see the effects of the landing gears' reduction in safety factor. Furthermore, reducing the thickness of the part would allow

future teams to be more efficient with carbon fiber composite use. In doing so, there would be a reduction in weight, however, weight was not a major concern in the final design solution.

During the design process, more emphasis must be placed on the recyclability and reusability of epoxy/carbon fiber composite parts. While the team was able to successfully design a durable landing gear, future teams should keep in regards the use of all carbon fiber parts after use, considering the potential environmental implications.

While the team focused on push down molds to manufacture the final front and rear landing gears, results using other manufacturing methods may also be explored. For example, in the early design stages, vacuum bag molding was considered as a manufacturing option. Using this manufacturing method involves preparing a mold that is sealed within a vacuum bag and is compressed by connecting a pump that removes the air from the bag and is molded into the desired shape. After that, as with any mold, the composite must cure, and vacuum pressure and temperature must be monitored frequently to ensure a proper mold [15]. While this process is more involved than the final design solution, it may be of interest to future teams to understand the benefits and drawbacks of other molding methods. In doing so, more methods would be compared to better understand a more efficient way to prepare landing gear molds.

Further verification and validation of the testing setups is also recommended. While the beam bending test setup instilled confidence in the team's setup of the landing gear ANSYS simulations, having further tests to perform would also be beneficial. For example, creating a fixture to recreate the constraints for the front landing gear that are explained in Section 7 would allow the team to visually understand the deflection of the landing gear. Considering that deflection is a simpler way to validate setups based on the intricacy of the equipment, it may be of interest to implement digital image correlation to inspect the change in deflection from before and after the application of the load. Furthermore, this would reduce the potential error seen from utilizing a ruler as a measurement tool. Conducting flight tests may also enhance our understanding of how the landing gear reacts to true landing conditions. Performing flight tests would allow the team to observe where the thickness may be decrease to match the characteristics of more traditional landing gear, while still upholding a safety factor greater than 1.5 as to meet FAA regulations.

11 Bibliography

- [1] “DBF Feature,” www. [Online]. Available: <https://www.aiaa.org/dbf>. [Accessed: 10-Sep-2024].
- [2] Aadith Arasu, Aidan Skidmore, Isaac Venezia, Joseph Martinelli, Juan Pablo Barrero, Nefi Diaz Santos, Pedro Guerra, “AIAA DBF,” The University of Massachusetts Lowell, Lowell, MA, 2024.
- [3] C. T. M. J. Zipay, “NASA,” *The 1.5 & 1.4 Ultimate Factors of Safety for Aircraft & Spacecraft – History, Definition and Applications*, February 2014. [Online]. Available: <https://ntrs.nasa.gov/api/citations/20140011147/downloads/20140011147.pdf>. [Accessed: 03-Dec-2024].
- [4] C. C. Uk, “From strength to sustainability: Unravelling carbon fibre’s environmental impact,” *Composites Construction UK*, 15-Aug-2023. [Online]. Available: <https://www.fibrwrap-ccuk.com/blog/from-strength-to-sustainability-unravelling-carbon-fibres-environmental-impact/>. [Accessed: 11-Dec-2024].
- [5] Pilot Institute, “Landing gear configurations (types, features, uses),” *Pilot Institute*, 16-Jun-2023. [Online]. Available: <https://pilotinstitute.com/landing-gear-configurations/>. [Accessed: 8-Dec-2024].
- [6] Easy Composites Ltd., *Make Forged Carbon Fibre Parts Using Compression Moulding*. United Kingdom: YouTube, 2021.
- [8] “Momentum change and impulse,” *Physicsclassroom.com*. [Online]. Available: <https://www.physicsclassroom.com/class/momentum/Lesson-1/Momentum-and-Impulse-Connection>. [Accessed: 15-Dec-2024].
- [7] “Overview of Materials for Epoxy/Carbon Fiber Composite,” *Matweb*. [Online]. Available: <https://www.matweb.com/search/datasheet.aspx?matguid=39e40851fc164b6c9bda29d798bf3726>. [Accessed: 22-Nov-2024].
- [9] R. Elliott, “Deflection of Beams,” *Org.uk*, 30-Dec-2010. [Online]. Available: <http://www.clag.org.uk/beam.html>. [Accessed: 06-Dec-2024].

- [10] “Aluminum 6061-T6; 6061-T651,” *Matweb*. [Online]. Available: <https://www.matweb.com/search/datasheet.aspx?MatGUID=b8d536e0b9b54bd7b69e4124d8f1d20a&ckck=1>. [Accessed: 22-Nov-2024].
- [11] “14 CFR part 107 -- small unmanned aircraft systems,” *Ecfr.gov*. [Online]. Available: <https://www.ecfr.gov/current/title-14/chapter-I/subchapter-F/part-107>. [Accessed: 12-Nov-2024].
- [12] Federal Aviation Administration (FAA), “Remote identification (Remote ID) is here. Are you ready?”
- [13] Compass: American Society for Testing and Materials (ASTM), “ASTM F593-24 Standard Specification for Stainless Steel Bolts, Hex Cap Screws, and Studs,” ASTM International, West Conshohocken, PA, Sep. 2024.
- [14] A. Ugural and S. Fenster, Advanced Mechanics of Materials and Applied Elasticity, 6th ed. Upper Saddle River, NJ: Pearson, 2019.
- [15] “What is Vacuum Bagging Process?,” NitPro Composites. [Online]. Available: <https://www.nitprocomposites.com/blog/what-is-vacuum-bagging-process>. [Accessed: 12-Dec-2024].