

# Department of Aerospace Engineering IIT Madras



**AS5100:** Mini Project

**Group 5:** UAV Fabrication Report

- **AE21B002 :** Abhigyan Roy
- **AE23M004 :** Vinu Mathew
- **AE23M006 :** Aditya Sai Deepak Rachagiri
- **AE23M008 :** Anish Konar
- **AE23M014 :** Gautham Anil

January 25, 2025

# Contents

<b>1 Problem Definition</b>	<b>6</b>
1.1 Introduction . . . . .	6
1.1.1 Mission Statement . . . . .	6
1.1.2 Mission Motivation and Description . . . . .	6
1.1.3 Mission Profile . . . . .	6
1.2 UAV Aerodynamic Analysis Details . . . . .	8
<b>2 Wing</b>	<b>10</b>
2.1 Schrenk's Method for Lift Distribution . . . . .	10
2.2 Spar Loading . . . . .	11
2.2.1 Shear Force Diagram . . . . .	12
2.2.2 Bending Moment Diagram . . . . .	12
<b>3 Spar Design</b>	<b>14</b>
3.1 Material Selection . . . . .	14
3.2 Spar Calculations . . . . .	14
3.2.1 Allowable Stress Calculation . . . . .	14
3.2.2 Case (i): Considering Skin Also Takes Bending Loads . . . . .	15
3.2.3 Case (ii): Spars Taking All the Bending Loads . . . . .	15
3.3 Conclusion . . . . .	17
<b>4 Wing Shear Design</b>	<b>18</b>
4.1 Shear Flow Calculation . . . . .	18
4.1.1 Structural Idealisation . . . . .	18
4.1.2 1-Cell Analysis of Wing Cross Section Without Spar and Present Control Surface . . . . .	19
4.1.3 2-Cell Analysis of Wing Cross Section With Attachment Spar and No Present Control Surface . . . . .	23
4.2 Conclusion . . . . .	25
<b>5 Torque Estimation for Servo Selection</b>	<b>26</b>
5.1 Torque Estimation . . . . .	26
5.1.1 Torque Estimation for Ailerons . . . . .	26
5.1.2 Torque Estimation for Rudder . . . . .	26
5.1.3 Torque Estimation for Elevator . . . . .	27
5.2 Servo Torque . . . . .	27
<b>6 Fuselage Design</b>	<b>28</b>
6.1 Fuselage Load Distribution and Structural Analysis . . . . .	28
6.1.1 Component Placement . . . . .	28
6.2 Shear Force Analysis . . . . .	28
6.3 Bending Moment Analysis . . . . .	29
<b>7 Longerons and Fuselage Bulkhead Design</b>	<b>30</b>
7.1 Longerons . . . . .	30
7.1.1 Estimating Number of Longerons required . . . . .	30
7.1.2 Longeron Sizing . . . . .	33
7.2 Bulkhead . . . . .	33

<b>8 Landing Gear Design</b>	<b>38</b>
8.1 Landing gear Configuration . . . . .	38
8.1.1 Landing Gear Retraction . . . . .	39
8.2 Landing Gear Geometry . . . . .	39
8.2.1 Position of Main LG and Nose LG . . . . .	39
8.2.2 Landing Gear Track . . . . .	39
8.2.3 Landing Gear Height . . . . .	39
8.2.4 Landing Gear Overturn Angle . . . . .	41
8.3 Load on each Landing Gear . . . . .	41
8.4 Selection of Tyres . . . . .	42
8.5 Nose Wheel Steering System . . . . .	42
8.6 Main Landing Gear Assembly . . . . .	42
<b>9 Bill of Materials</b>	<b>44</b>
<b>10 Fabrication</b>	<b>45</b>
10.1 Brief of Assembly . . . . .	45
10.2 Main Wing . . . . .	45
10.2.1 Rib . . . . .	45
10.2.2 Stringer . . . . .	46
10.2.3 Spar . . . . .	46
10.2.4 Wing Skin . . . . .	47
10.2.5 Challenges Faced during Wing Manufacturing . . . . .	48
10.3 Fuselage . . . . .	48
10.4 Tail and Empennage . . . . .	50
10.5 Landing Gear . . . . .	52
10.6 Summary of Challenges Faced . . . . .	53
10.6.1 Challenges Faced . . . . .	53
10.6.2 Lessons Learnt . . . . .	53
<b>11 Appendix</b>	<b>54</b>
11.1 Lift Distribution of Wing using Schrenks' Method & SFD and BMD Plots Calculation . .	54
11.2 Spar Calculations . . . . .	56
11.3 Shear Flow Calculations . . . . .	58
11.4 Bulkhead Calculations . . . . .	63
11.5 SFD and BMD of Fuselage . . . . .	66
11.6 BMD of Landing Gear . . . . .	68

# List of Figures

1.1.1 Mission Profile . . . . .	7
2.1.1 Chord calculated using Schrenk's Method . . . . .	11
2.1.2 Lift Distribution calculated using Schrenk's Method . . . . .	11
2.2.1 Wing Shear Force Diagram . . . . .	12
2.2.2 Wing Bending Moment Diagram . . . . .	13
3.2.1 Required area moment of inertia variation . . . . .	16
3.2.2 Flange Width Calculation . . . . .	17
3.3.1 Shortened Spar . . . . .	17
4.1.1 Boom Idealisation of a Panel . . . . .	18
4.1.2 1-Cell Boom Structure of Wing Cross Section Excluding Present Control Surface <b>away from the root</b> . . . . .	19
4.1.3 2-Cell Boom Structure of Wing Cross Section with Fuselage Attachment Spar and No Control Surface <b>near the root</b> . . . . .	19
4.1.4 Shear Flow Distribution along chord . . . . .	20
4.1.5 % of $F_{cr,initial}$ of the shear flow . . . . .	21
4.1.6 Shear Flow Distribution with Stringer Placement . . . . .	22
4.1.7 % of $F_{cr}$ of the shear flow <b>with Stringers</b> . . . . .	22
4.1.8 Shear flow Distribution of 2-cell Wing with Spar without any control surfaces . . . . .	23
4.1.9 Shear flow Distribution with Previous Stringer Placement . . . . .	24
4.1.10% of $F_{cr}$ for Shear Flows around wing with spar and stringers . . . . .	24
6.2.1 Shear Force Diagram (SFD) along the fuselage . . . . .	29
6.3.1 Bending Moment Diagram (BMD) along the fuselage . . . . .	29
7.1.1 Fuselage Cross Section with Boom Idealization . . . . .	31
7.1.2 Fuselage Cross Sectional Shear Flow Distribution . . . . .	31
7.1.3 Fuselage Cross Sectional Shear Flow Distribution wrt Angle . . . . .	32
7.1.4 $F_{cr}$ vs No. of Longerons . . . . .	32
7.1.5 6 Longerons placed along Cross Section of Fuselage at $60^\circ$ angles <b>starting from the mid section where the shear flow is maximum</b> . . . . .	33
7.2.1 Symmetrical Half of Bulkhead Section . . . . .	34
7.2.2 Bending Moment Diagram of Bulkhead . . . . .	36
8.0.1 Landing Gear Parameters [2] . . . . .	38
8.1.1 Tricycle Landing Gear . . . . .	39
8.2.1 Take off Ground Clearance . . . . .	40
8.2.2 Clearance Angle Estimation . . . . .	40
8.2.3 Estimation of Overturn Angle . . . . .	41
8.3.1 Loads acting on the LG . . . . .	41
8.6.1 Free Body Diagram of Landing Gear . . . . .	42
8.6.2 Bending Moment Diagram of Landing Gear . . . . .	43
10.2.1 Main Wing Rib . . . . .	46
10.2.2 Wing internal assembly with 1st iteration of Attachment Spar (circled in green) . . . . .	46

10.2.3	Wing Skin Section with Aileron . . . . .	47
10.2.4	Wing Skin Section with Flap Cutout . . . . .	47
10.3.1	1 <sup>st</sup> iteration of Fuselage . . . . .	48
10.3.2	Longerons sticking out due to less tolerance . . . . .	49
10.3.3	Spar Slot for fuselage 1st iteration . . . . .	49
10.3.4	CAD Model of 2 <sup>nd</sup> iteration of Fuselage and Landing Gear Adapter . . . . .	50
10.3.5	2 <sup>nd</sup> iteration of Fuselage with Longerons, Landing Gear Adapter, Attachment Spar and Tail Boom . . . . .	50
10.4.1	Tail Adapter CAD Model . . . . .	51
10.4.2	3D printed Horizontal Tail Rib . . . . .	51
10.4.3	Empennage consisting Horizontal Tail and Tail Boom connected with Tail Adapter . . . . .	51
10.5.1	Main Landing Gear . . . . .	52
10.5.2	Front Landing Gear Wheel and Steerable Ball Bearing . . . . .	52

# List of Tables

1.2.1 General UAV Operational Specifications . . . . .	8
1.2.2 UAV Dimensional Specifications . . . . .	9
3.1.1 Material Properties . . . . .	14
4.1.1 Buckling load of Skin Panel with Stringer . . . . .	22
6.1.1 Component Loads and Positions in the Fuselage . . . . .	28
9.0.1 Structural Components . . . . .	44
9.0.2 Propulsion Components . . . . .	44
9.0.3 Avionics Components . . . . .	44
9.0.4 Tools . . . . .	44

## Chapter 1

# Problem Definition

### 1.1 Introduction

#### 1.1.1 Mission Statement

Monitoring of Flora and Fauna and the environment of forests in the Chennai AOR and Mapping of Forest Cover, including security-based Surveillance of the forest within the said Area of Responsibility.

#### 1.1.2 Mission Motivation and Description

Forests play a crucial role in combating climate change by absorbing CO<sub>2</sub> and releasing oxygen through photosynthesis. They also regulate climates by influencing moisture levels and reflecting sunlight. Deforestation disrupts this balance, reducing Earth's ability to mitigate greenhouse gas emissions. Preserving and restoring forests is vital for the protection of biodiversity and to mitigate the impacts of climate change.

To efficiently manage the forest cover within Chennai city, constant monitoring is required due to the dynamic and ever-changing nature of forests. Traditional methods of monitoring can be time-intensive, labor-intensive, and expensive, requiring data acquisition across large spatial regions. As a cost-effective alternative, medium-range, medium-endurance UAVs can be employed for short-term, scheduled surveys that provide real-time data on the environmental state of forests.

Based on the mission motivation, the group has defined the mission description as follows:

##### 1. Design Objective:

- To design a UAV with sufficient range and endurance to fly within the Chennai City Limits for monitoring forest cover.
- The UAV should initially cruise at an altitude of 80 meters for an aerial survey, followed by a lower altitude flight at 40 meters for detailed observation of the target forest area.

##### 2. Sensor Capabilities:

- Equip the UAV with sensors to monitor environmental parameters such as temperature, relative humidity, and harmful gases in the air.
- The UAV should be capable of detecting intruders under the forest canopy using infrared sensors.

#### 1.1.3 Mission Profile

##### Phase I - Take Off and Climb:

The UAV should have a very low startup time and hence be readily deployed on short notice to achieve the desired objective. Also, it would need to have a low take-off distance and a high Rate of Climb is desirable to be able to reach cruising altitude as soon as possible.

##### Phase II - Cursory Data Acquisition:

Based on the area of coverage of the Chennai City limits and considering the height of the tallest tree

within the city limits to be about 40m, Two cruising altitudes of 80 meters and 40 meters have been identified for the UAV operation. In the Cursory data acquisition phase with an altitude of 80m, the area of coverage comes upto about  $6943\text{ m}^2$ . This would be sufficient for a cursory scan of the target area. Here, UAV is expected to be able to loiter for about 20–30 minutes which would enable us to carry out a high range aerial survey to collect higher spatial data and identify areas of interest.

### **Phase III - Detail Data Acquisition:**

Post completion of the Cursory Data Acquisition phase, the UAV would be employed to loiter at this lower cruise altitude of about 40m for about 45–60 minutes to monitor the environment of the forest at tree top levels. The area of coverage here would be about  $3471\text{ m}^2$ . This cruise altitude will also provide detailed information of the target AOR at a higher resolution. This would enable the user to extract detailed source information for a better assessment of the data acquired.

### **Phase IV - Descent and Landing:**

The UAV must then carry out a descent at nominal descent angles, maintaining correct approach speeds, to carry out a safe landing at the desired landing strip.

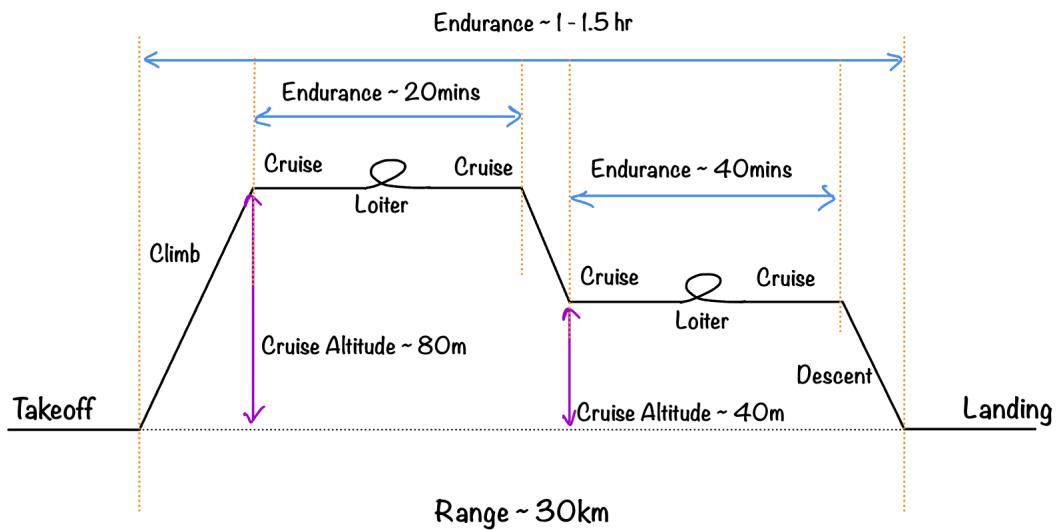


Figure 1.1.1: Mission Profile

## 1.2 UAV Aerodynamic Analysis Details

Parameter	Value
Cruise Speed	18 m/s
Max Speed	22.5 m/s
Stall Speed	10 m/s
Rotation Speed	11 m/s
Take off Speed	12 m/s
Climb Speed	11.51 m/s
Max Climb Rate	2 m/s
Max Climb Angle	12 Deg
Absolute Ceiling	100m
Cruise Altitude	80m/40m
Propeller Efficiency	0.5
L/D	19.48
Battery Capacity	18000 mAh
$C_{L_{max}}$	1.56
$C_{L_{TakeOff}}$	1.21
$C_{L_{Stall}}$	1.42
$C_{L_{Climb}}$	1.31
$C_{L_{Cruise}}$	0.46
Oswalds Efficiency factor	0.8113
$C_{D_0}$	0.03
Wing Loading	90 N/ $m^2$

Table 1.2.1: General UAV Operational Specifications

Parameters	Value
MTOW	8.80 kg
Max Payload Weight	2.00 kg
Design Payload Weight	1.50 kg
Powerplant Weight	2.30 kg
CG location (from nose)	0.50 m
Wing Area	$0.96 \text{ m}^2$
Wing Span	2.80 m
Wing Taper Ratio	1
Wing Root Chord	0.34 m
Wing Tip Chord	0.34 m
Wing Aspect Ratio	8.34
Wing Twist Angle	0 Deg
Wing Sweep Angle	0 Deg
Wing Dihedral Angle	2 Deg
Wing Setting Angle	1 Deg
Wing Aerofoil	GOE 553
Aileron Area	$0.05 \text{ m}^2$
Aileron Chord	0.09 m
Aileron Span	0.56 m
Fuselage Length	1.50 m
Fuselage Diameter	0.25 m
Fuselage Width	0.20m
Horizontal Tail Area	$0.42 \text{ m}^2$
Horizontal Tail Span	1.50 m
Horizontal Tail Taper Ratio	1
Horizontal Tail Root Chord	0.28 m
Horizontal Tail Tip Chord	0.28 m
Horizontal Tail Aspect Ratio	5.0
Horizontal Tail Twist Angle	0 Deg
Horizontal Tail Sweep Angle	0 Deg
Horizontal Tail Dihedral Angle	0 Deg
Horizontal Tail Setting Angle	2 Deg
Horizontal Tail Aerofoil	NACA 0014
Elevator Area	$0.10 \text{ m}^2$
Elevator Chord	0.07 m
Elevator Span	1.50 m
Vertical Tail Area	$0.14 \text{ m}^2$
Vertical Tail Span	0.40 m
Vertical Tail Taper Ratio	1
Vertical Tail Root Chord	0.35 m
Vertical Tail Tip Chord	0.35 m
Vertical Tail Aspect Ratio	1.14
Vertical Tail Twist Angle	0 Deg
Vertical Tail Sweep Angle	0 Deg
Vertical Tail Dihedral Angle	0 Deg
Vertical Tail Setting Angle	0 Deg
Aerofoil	NACA 0014
Rudder Area	$0.04 \text{ m}^2$
Rudder Chord	0.10 m
Rudder Span	0.40 m

Table 1.2.2: UAV Dimensional Specifications

## Chapter 2

# Wing

### 2.1 Schrenk's Method for Lift Distribution

The lift distribution over an elliptical wing is proportional to its chord distribution, but this does not hold for non-elliptical wings. To estimate lift distribution for such wings, Schrenk's Method offers a simple and efficient approximation, which is particularly useful for preliminary structural design.

According to Schrenk's Method, the lift distribution is proportional to the average chord distribution of both the elliptical equivalent and the actual non-elliptical wing. The steps involved in the calculation are as listed below:

1. **Equivalent Elliptical Wing Chord Distribution:** Compute the chord distribution for an equivalent elliptical wing with the same span length ( $b$ ) and area ( $S$ ) as the actual wing using:

$$c_{\text{elliptical}}(y) = \frac{4S}{\pi b} \sqrt{1 - \left(\frac{2y}{b}\right)^2} \quad (2.1.1)$$

2. **Average Chord Distribution:** Determine the average chord distribution,  $c_{\text{schrenk}}(y)$ , by averaging the elliptical and actual wing chord distributions:

$$c_{\text{schrenk}}(y) = \frac{c(y) + c_{\text{elliptical}}(y)}{2} = c(y) \left( \frac{1}{2} + \frac{2S}{\pi b c(y)} \sqrt{1 - \left(\frac{2y}{b}\right)^2} \right) \quad (2.1.2)$$

3. **Lift Distribution:** The lift per unit span distribution is then proportional to  $c_{\text{schrenk}}(y)$ .

4. **Proportionality Factor:** Equate the required total lift to the calculated lift to find the proportionality factor  $k$ :

$$k = \frac{W/2}{\int_0^{b/2} c_{\text{schrenk}}(y) dy} \quad (2.1.3)$$

Figure 2.1.1 illustrates the chord calculated by Schrenk's Method showing the average between the chosen rectangular chord distribution and the equivalent elliptic chord distribution. Figure 2.1.2 illustrates the corresponding lift distribution computed using Schrenk's Method and compares it to the elliptic lift distribution. The maximum lift in the distribution obtained from this method is 34.8 N.

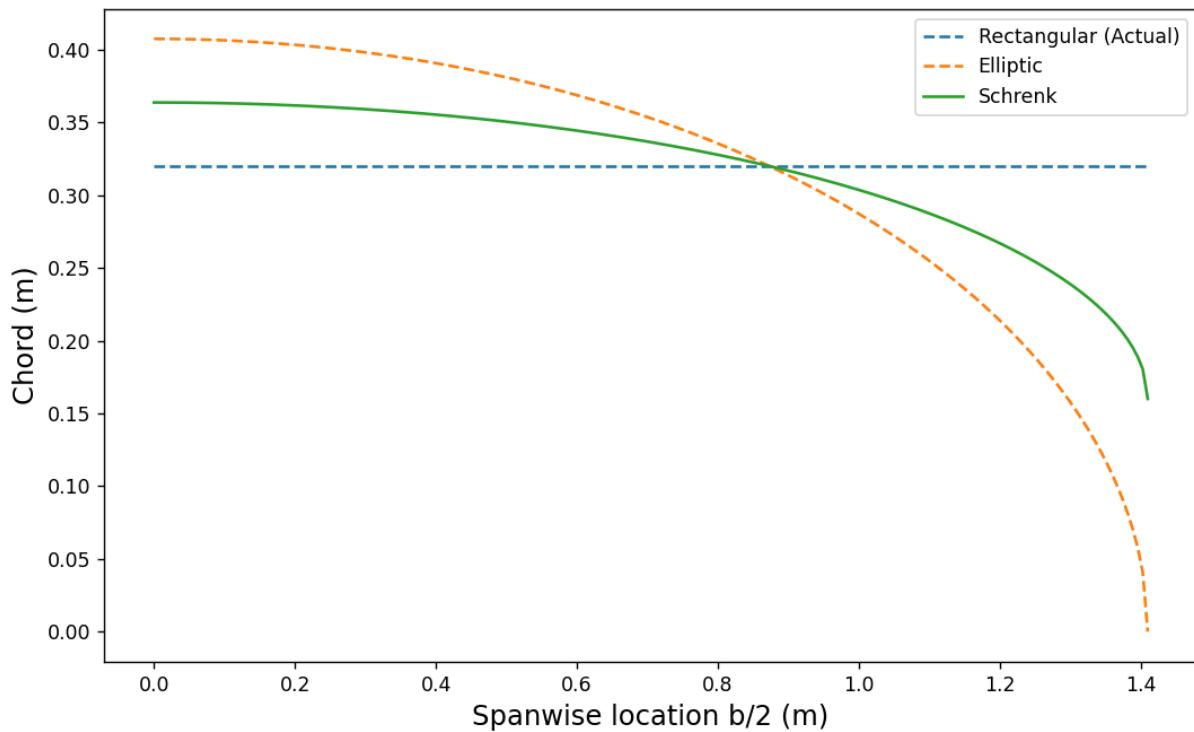


Figure 2.1.1: Chord calculated using Schrenk's Method

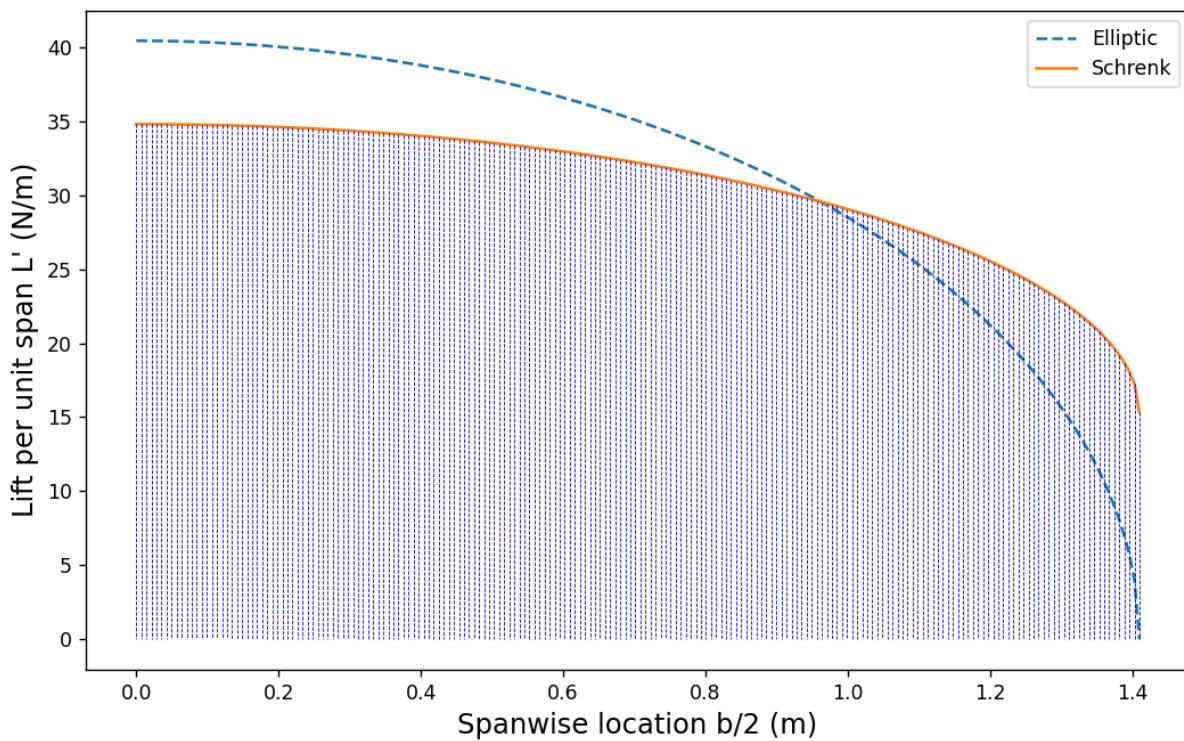


Figure 2.1.2: Lift Distribution calculated using Schrenk's Method

## 2.2 Spar Loading

The Shear Force Diagram (SFD) and Bending Moment Diagram (BMD) for the wing, based on lift forces estimated using Schrenk's method, are depicted in Figures 2.2.1 and 2.2.1.

### 2.2.1 Shear Force Diagram

The shear force at each section or station along the wing's semi-span is determined by summing the elemental lift forces from that  $y$  station to the wingtip ( $y = b/2$ ), as shown in Figure 2.2.1. Since, it is a free end, there is no shear force at the wing tip.

The shear force  $V(y)$  at a distance  $y$  from the root is given by:

$$V(y) = - \int_0^y L_{eq}(y') dy'$$

The resulting shear force diagram for the wing is presented in Figure 2.2.1. It is observed that the maximum shear force, 43 N, occurs at the wing root.

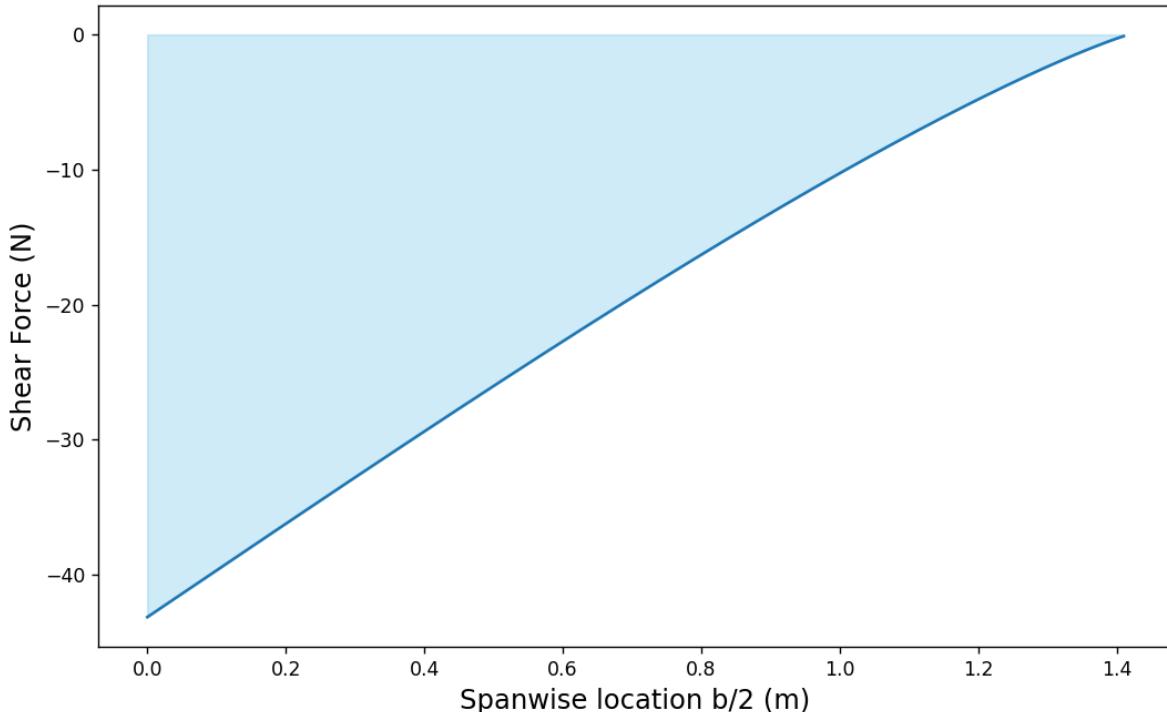


Figure 2.2.1: Wing Shear Force Diagram

### 2.2.2 Bending Moment Diagram

The bending moment at a section is calculated by summing the moments of forces from that arbitrary point  $y$  to the wingtip ( $y = b/2$ ). Alternatively, it can be determined by calculating the area under the shear force diagram from  $y = 0$  to  $b/2$ . Since, it is a free end, there is no bending moment at the wing tip.

The bending moment  $M(y)$  at a distance  $y$  from the root is:

$$M(y) = - \int_0^y V(y') dy'$$

The resulting bending moment diagram for the wing is presented in Figure 2.2.2. It is observed that the maximum bending moment, 28 Nm, occurs at the wing root.

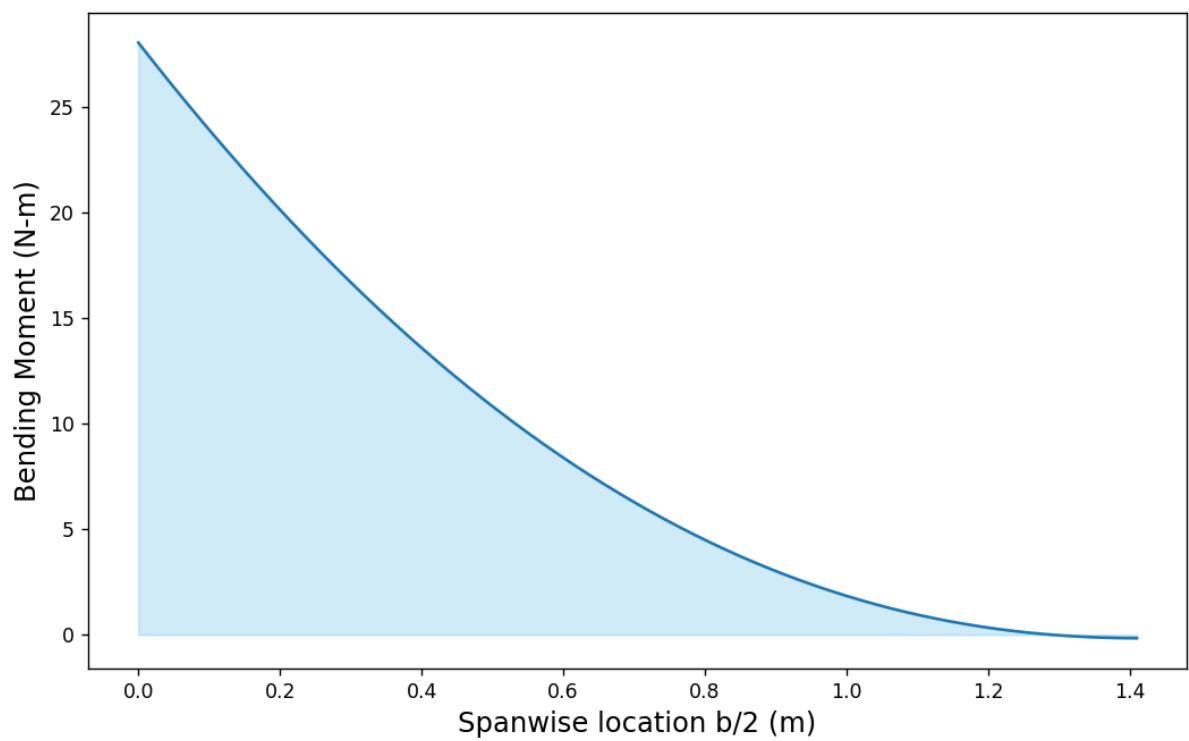


Figure 2.2.2: Wing Bending Moment Diagram

## Chapter 3

# Spar Design

Wings generate most of an aircraft's lift and bear some of the heaviest loads within the aircraft's structure. Fundamentally, a wing consists of a framework made from spars and ribs, all covered with metal. Additionally, stringers are incorporated in a semi-monocoque construction for added support.

Spars are the principal structural components of the wing, typically running from the wing root to the wingtip. They are responsible for supporting all the loads carried by the wing and are engineered to possess significant bending strength.

### 3.1 Material Selection

The selected material for this design is Aluminum alloy (Al 6061-T6). This material is commonly used in the aerospace industry due to its ease of fabrication and adequate strength. The properties of the material are detailed below in Table 3.1.1.

Property	Value (Unit)
Density ( $\rho$ )	2700 kg/m <sup>3</sup>
Yield strength ( $\sigma_Y$ )	270 MPa
Young's modulus ( $E$ )	69 GPa
Poisson's ratio ( $\nu$ )	0.33

Table 3.1.1: Material Properties

### 3.2 Spar Calculations

The wing spars of an aircraft in flight are subject to bending stresses. The following calculations [11.2] are done to get the dimensions and location of the required spar.

#### 3.2.1 Allowable Stress Calculation

The allowable stress is calculated by dividing the yield stress by several factors. These factors include:

- Stress Concentration Factor ( $k$ )
- Fatigue Factor ( $f$ )
- Factor of Safety ( $m$ )
- Maximum Load Factor ( $n$ )

Using assumptions,  $k = 1.5$ ,  $m = 1.5$ ,  $n = 3$ ,  $f = 1.5$ , based on standard industry values, the allowable stress is given by:

$$\sigma_{\text{allowable}} = \frac{\sigma_Y}{n \times f \times k \times m} = \frac{\sigma_Y}{10.125} = 26.67 \text{ MPa} \quad (3.2.1)$$

### 3.2.2 Case (i): Considering Skin Also Takes Bending Loads

The area moment of inertia about the centroid of the airfoil cross-section is given by:

$$I_{\text{centroid}} = \int_A y^2 dA$$

where:

- $y$  is the vertical distance from the centroidal axis,
- $dA$  is the differential area element of the cross-section.

The centroid of the airfoil cross-section was calculated, and the moment of inertia calculation was performed with the centroid as the origin.

The moment of inertia of the skin panel with a thickness of 0.5 mm is calculated to be  $I_{xx} = 9.38 \times 10^{-8} \text{ m}^4$  or  $I_{xx} = 93800 \text{ mm}^4$ .

The moment of inertia offered by the skin is higher than the actual inertia requirement. This implies that the design can be made without the requirement for spars. However, we have chosen to keep spars to serve the following purposes.

- To assist in strengthening the wing to counter the loads acting due to the tail
- To provide strengthening to cater for Wing-Fuselage attachment.
- To provide strengthening to cater for Ailerons and Flaps.

### 3.2.3 Case (ii): Spars Taking All the Bending Loads

The cross-section of the spar is determined by calculating the moment of inertia required if the spar takes all the bending loads. The allowable stress equation is given by:

$$\sigma_{\text{allowable}} = \frac{M \cdot y}{I} \quad (3.2.2)$$

where  $M$  is the bending moment,  $I$  is the moment of inertia, and  $y = \frac{t_{\max}}{2}$  (for symmetric sections). From the bending moment diagram, a moment curve is reported from which  $M_{\max} = 28 \text{ Nm}$ . The height is determined by the thickness of the airfoil ( $t_{\max}$  here), which is roughly 45 mm.

Now the total required moment of Inertia is given by the equation below. A required MOI distribution is thus plotted along the span.

$$I_{\text{total}} = \frac{M_{\max} \cdot y}{\sigma_{\text{allowable}}} = I_{\text{spar}} + I_{\text{skin}} \quad (3.2.3)$$

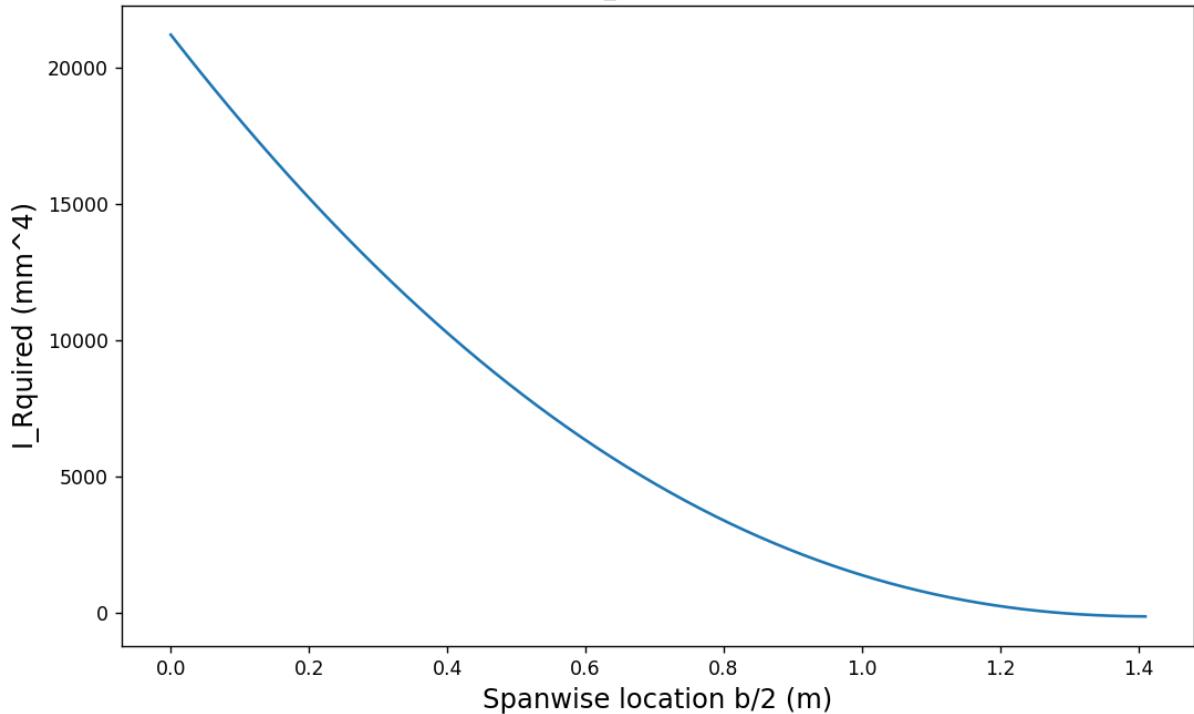


Figure 3.2.1: Required area moment of inertia variation

For a C section spar, the thickness is fixed according to standard skin thickness available in the market. Using the moment of inertia of spar formula, where  $t$  is the sheet thickness taken as 0.5 mm,  $h$  ( $= t_{\max}$ ) is the web height from the max chord thickness of 45 mm and  $w$  is the flange width (assuming  $I_{\text{skin}} = 0$ ) :

$$I_{\text{spar}} = \frac{th^3}{12} + w \times t \times \left(\frac{h}{2}\right)^2 \times 2 = \frac{M_{\max} \cdot y}{\sigma_{\text{allowable}}} - I_{\text{skin}} \quad (3.2.4)$$

The spar is designed to attach the wing to the fuselage. Thus the flange width variation required along the wing is calculated from the bending moment variation, and the root flange width of  $w = 40$  mm is selected. For attachment, it is estimated that the flange would need to connect to the first two root ribs. With 20 ribs along the wing span of 2.8 m, the rib spacing is 0.14 m. Thus each half wing spar is selected to extend to 0.28 m.

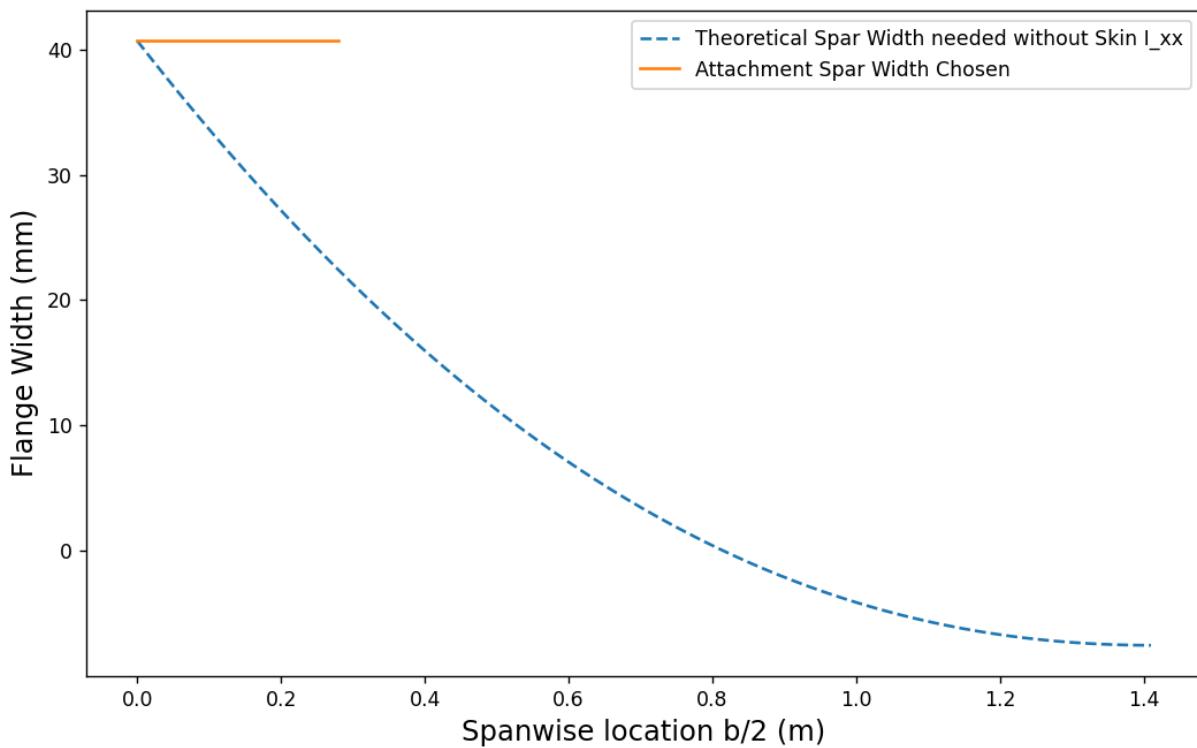


Figure 3.2.2: Flange Width Calculation

### 3.3 Conclusion

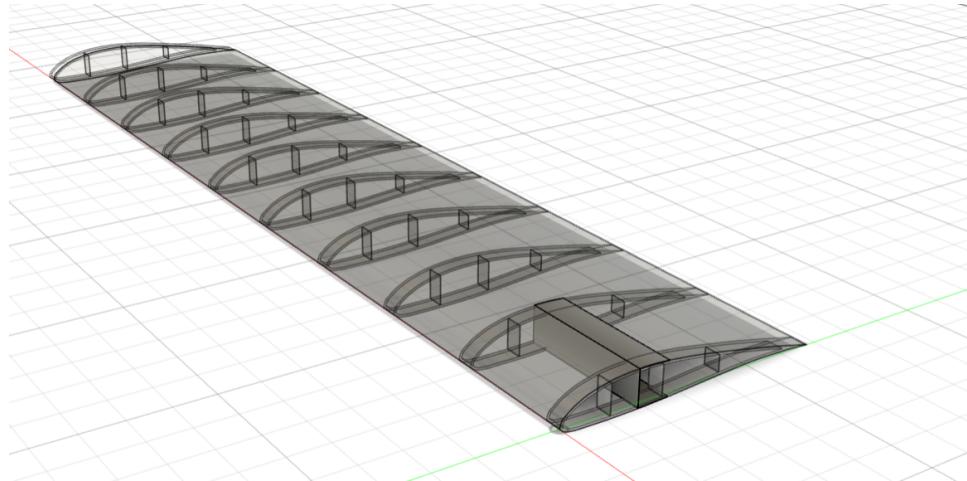


Figure 3.3.1: Shortened Spar

Although the skin alone can provide the required moment of inertia, a spar is included in the design at maximum thickness (29.6 % of chord length) for structural purposes. *For optimum weight reduction and manufacturability, we shorten the spar length to just between the rib at the root and the next rib and keep a constant flange width.*

## Chapter 4

# Wing Shear Design

Shear flow is a fictitious concept used in thin-walled structures to describe the distribution of shear forces over a given cross-section of a beam or structural element. Taking the **wing skin thickness** as 0.5 mm of Aluminium 6061, the objective of this chapter is to design a practical wing which includes, a **selection on the number of ribs** along the span of the wing, and a **selection of the number of stringers** required and their positioning along the cross section of the wing to prevent the buckling of the thin wing skin due to shear stresses acting on it. This is done by calculating the shear flows in the cross section of the wing along the chord at the root of the wing, where the shear forces due to the lift distribution is the highest, and comparing it with the buckling stress computed by an empirical formulation.

### 4.1 Shear Flow Calculation

#### 4.1.1 Structural Idealisation

Owing to its curved unsimplified geometry, an airfoil section with thin walls is best described as a group of panels. Using a 2-boom approximation (also known as boom idealisation [1]), these panels can be roughly represented as a series of direct stress carrying booms and shear stress carrying skin panel sections, as illustrated in figure 4.1.1.

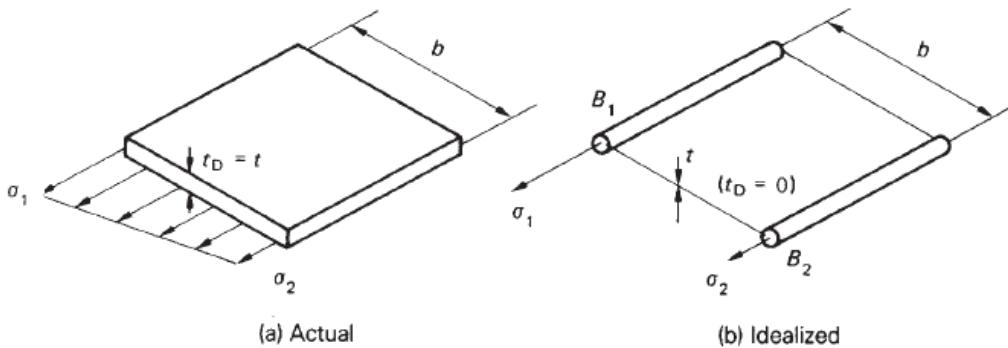


Figure 4.1.1: Boom Idealisation of a Panel

The area of the booms,  $B_1$  and  $B_2$ , are given by equations 4.1.1 and 4.1.1.

$$B_1 = \frac{t_D b}{6} \left( 2 + \frac{\sigma_2}{\sigma_1} \right) \quad (4.1.1)$$

$$B_2 = \frac{t_D b}{6} \left( 2 + \frac{\sigma_1}{\sigma_2} \right) \quad (4.1.2)$$

The section is required to resist bending moments in a vertical plane. The direct stresses ( $\sigma$ ) acting on a boom is directly proportional to its distance from the neutral axis ( $\sigma_k \propto y_k$ ).

The initial part of the wing connecting the wing to the fuselage near the root contains an attachment spar and no control surfaces, and the approximated boom structure of the skin panel is shown in fig 4.1.3. The approximated boom structure of the skin panel for the rest of the wing which is devoid of a spar and contains control surfaces; namely the flap and the aileron is shown in fig 4.1.2.

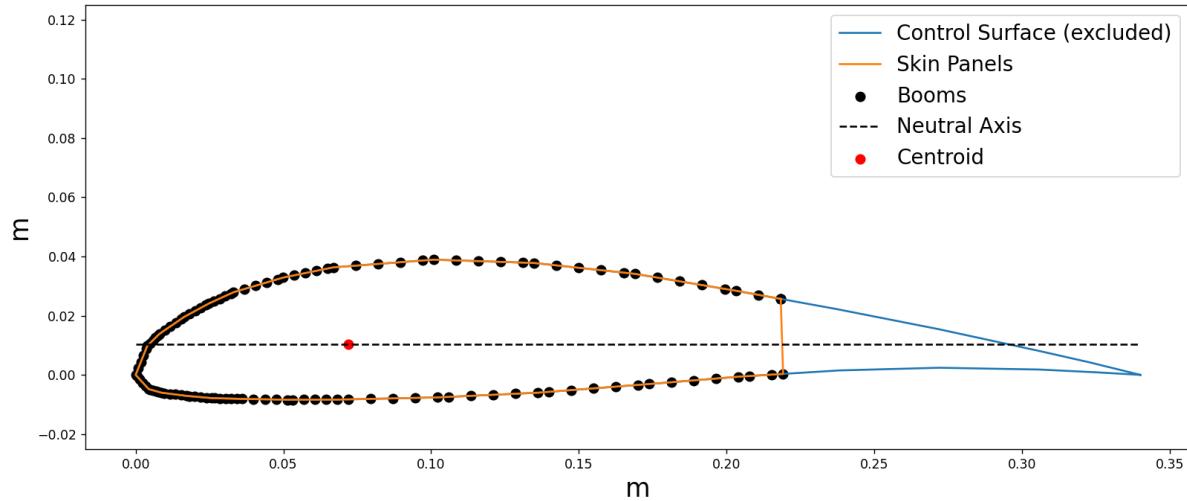


Figure 4.1.2: 1-Cell Boom Structure of Wing Cross Section Excluding Present Control Surface **away from the root**

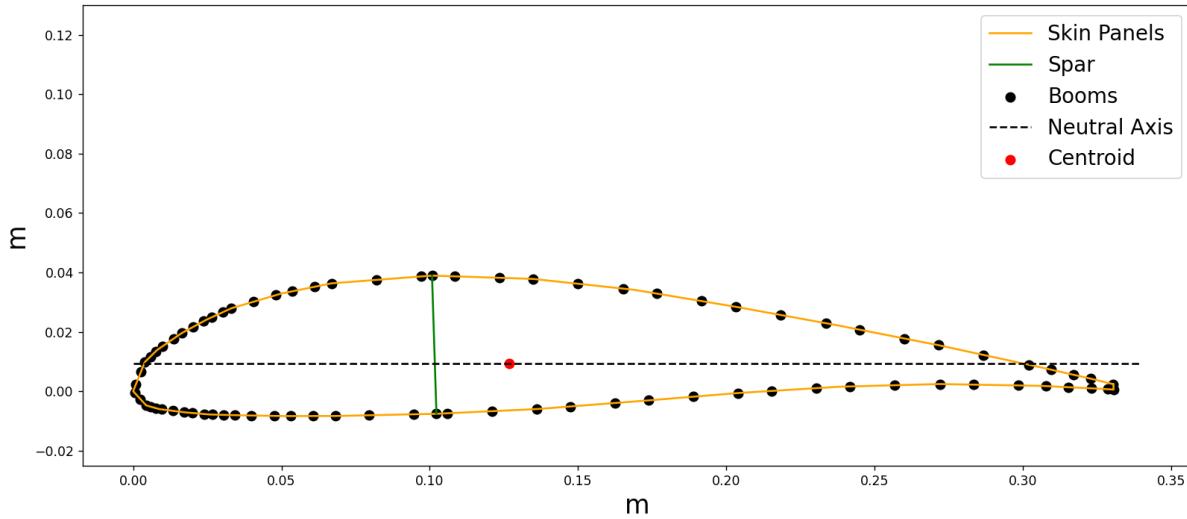


Figure 4.1.3: 2-Cell Boom Structure of Wing Cross Section with Fuselage Attachment Spar and No Control Surface **near the root**

#### 4.1.2 1-Cell Analysis of Wing Cross Section Without Spar and Present Control Surface Shear Flow Calculation

For a closed 1-cell section, a fictitious cut is initially made to calculate the basic shear flow. To close the structure and get the total shear flow, a constant shear flow  $q_{s,0}$  is calculated by satisfying the moment equilibrium or by equating the rate of twist acting on the wing cross-section to 0, and is added to all the cells. In the following no-spar analysis, the rate of twist equation is taken to close  $q_{s,0}$ . Thus, the shear flow for a cell is given by:

$$q_s = - \left( \frac{S_x I_{xx} - S_y I_{xy}}{I_{xx} I_{yy} - I_{xy}^2} \right) \sum_r B_r x_r - \left( \frac{S_y I_{yy} - S_x I_{xy}}{I_{xx} I_{yy} - I_{xy}^2} \right) \sum_r B_r y_r + q_{s,0} \quad (4.1.3)$$

where:

- $S_x$  and  $S_y$  are the shear forces in the  $x$  and  $y$  directions, respectively.
- $I_{xx}$  and  $I_{yy}$  are the second moments of area about the  $x$  and  $y$  axes, respectively.
- $I_{xy}$  is the product of inertia.
- $B_r$  is the area of the  $r^{th}$  boom encountered in that cell.
- $x_r$  and  $y_r$  are the coordinates of the  $r^{th}$  boom in that cell.
- $q_{s,0}$  is the constant shear flow added to satisfy moment equilibrium for that cell.

Which can be simplified by assuming  $I_{xy} \approx 0$ , and considering the design specification only specifies an acting  $S_y$  and taking  $S_x = 0$ ,

$$q_s = -\frac{S_y}{I_{xx}} \sum_r B_r y_r + q_{s,0} \quad (4.1.4)$$

Where the basic shear flow  $q_b$  is  $-\frac{S_y}{I_{xx}} \sum_r B_r y_r$ .

The shear flow  $q_{s,0}$  is closed by equating the total rate of twist  $\frac{d\theta}{dz}$  acting on the cross section to 0 assuming the load acts through the shear center.

The rate of twist for the 1-cell section with the load acting though the shear center is given by:

$$\frac{d\theta}{dz} = \frac{1}{2A_k G} \int \frac{q_s ds}{t} = 0 \quad (4.1.5)$$

where:

- $A_k$  is the area of the  $k$ -th cell,
- $G$  is the shear modulus,
- $q_s$  is the shear flow,
- $ds$  is the length element along the section,
- $t$  is the thickness of the cell panel.

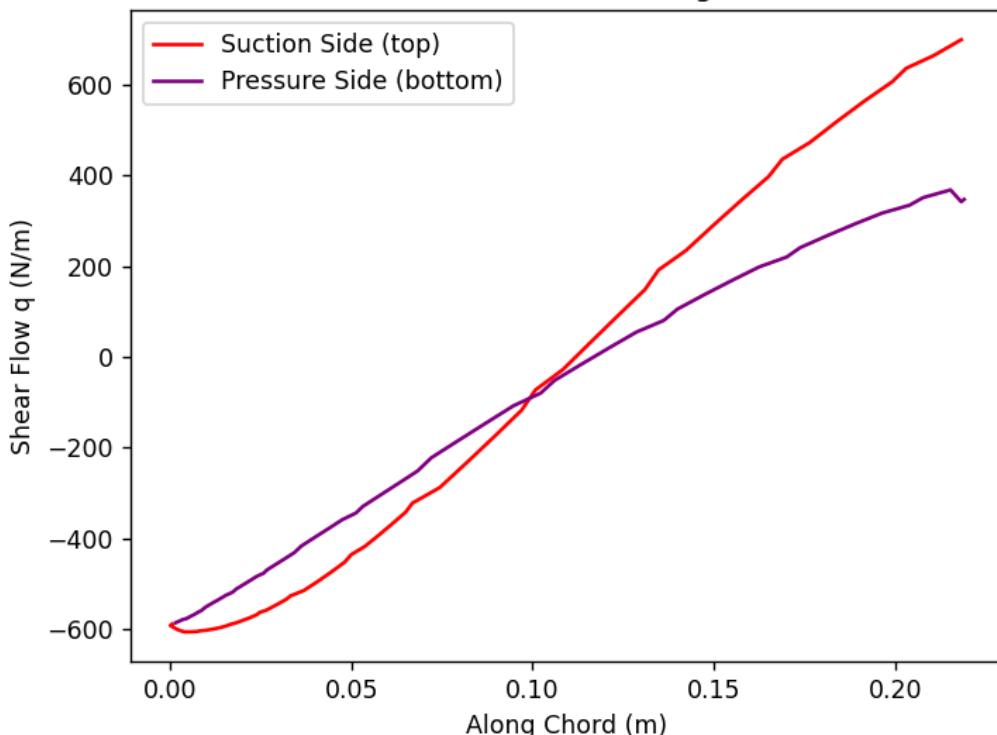


Figure 4.1.4: Shear Flow Distribution along chord

## Buckling Analysis

Calculating the critical buckling load is necessary to ensure that our skin panel will not buckle under the applied shear stresses. The critical buckling load  $F_{cr}$  is found using the empirical formulae given below and compared to the shear flow acting along the cross section of the idealized wing.

Critical load per unit area (for buckling) is given by:

$$F_{cr} = K_{ss} \frac{\pi^2 E}{12(1-\nu^2)} \left(\frac{t}{b}\right)^2 \left[ R_a + \left(\frac{R_a - R_b}{2}\right) \left(\frac{b}{a}\right)^3 \right] \quad (4.1.6)$$

where

- $K_{ss} = 5.34 + \frac{4}{\left(\frac{a}{b}\right)^2}$ , where  $\mathbf{a}$  is the distance between 2 consecutive ribs and  $\mathbf{b}$  is the spacing between 2 stringers, taking 10 ribs along each half wing (20 ribs for wing span 2.8 m)
- $R_a$  and  $R_b$  are the rotation end fixity conditions. We assume  $R_a = R_b = 1$  for simply supported.
- $\nu$  is the Poisson ratio, which is equal to 0.33.
- $E$ , the elastic modulus is 70 GPa
- The thickness ( $t$ ) of the aluminum sheet used for the skin panel is 0.5 mm

The objective is to design a wing which does not buckle. For this a safety factor is calculated taking the max +ve load factor  $n = 3$  and the Factor of Safety  $FOS = 1.5$  and an additional factor of  $d = 2$  to account for direct stresses. Thus the total factor taken into consideration is

$$F = n \times FOS = 3 \times 1.5 \times 2 = 9 \quad (4.1.7)$$

Thus, we design the shear flows to be less than  $\approx 11\%$  of  $F_{cr}$  i.e.

$$q_{total} \leq \frac{F_{cr}}{9} \quad \text{OR} \quad q_{total} \leq 11\% \text{ of } F_{cr} \quad (4.1.8)$$

Taking the length of the whole cross section as 1 panel and placing no stringers, an initial  $F_{cr,initial}$  is calculated and the % of  $F_{cr,initial}$  of the shear flows is reported by calculating  $q_{total}/F_{cr} \times 100$ , which serves as an indication of the location where stringers need to be placed.

$$F_{cr,initial} = 1836 \text{ N/m} \quad (4.1.9)$$

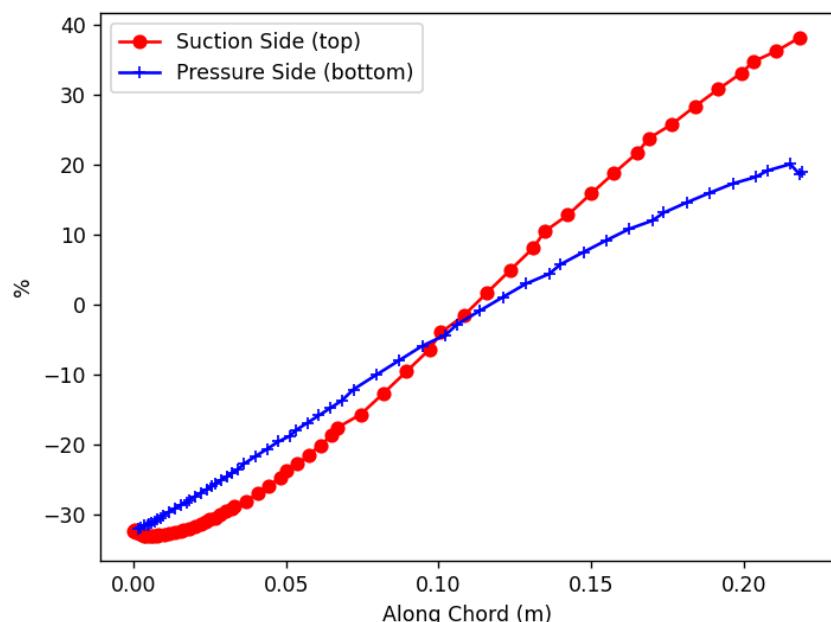


Figure 4.1.5: % of  $F_{cr,initial}$  of the shear flow

Observing from the above diagram, the shear flow clearly exceed the set limit of 20% of  $F_{cr,initial}$ . Thus the skin is divided into panels to increase the critical buckling load by decreasing the panel length  $b$ , placing 4 stringers: 2 on suction side and 2 on the pressure side, with 2 stringers close to the Leading Edge and 2 stringers close to the Trailing Edge. The Stringer Placement is reported in the figure below, and the new critical buckling loads  $F_{cr}$  are tabulated, and the new % of  $F_{cr}$  of the shear flows have been plotted against the chord.

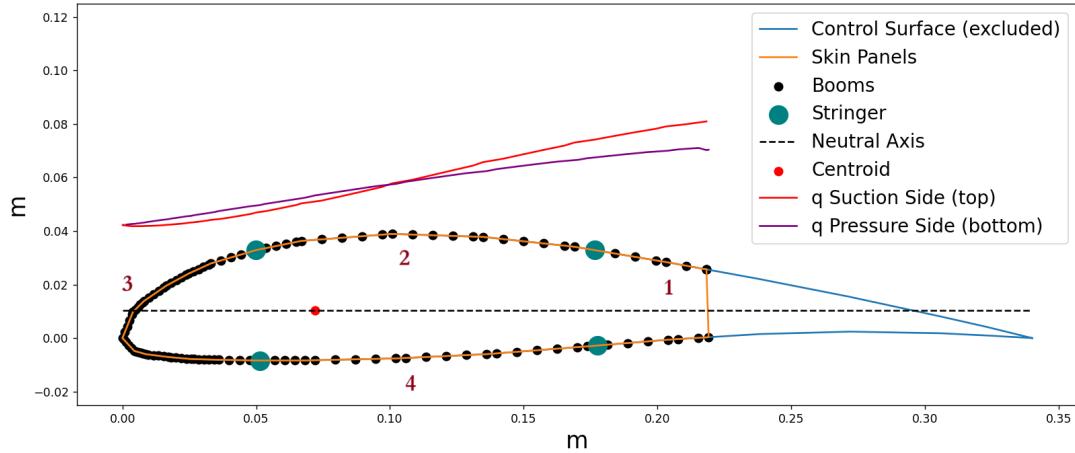


Figure 4.1.6: Shear Flow Distribution with Stringer Placement

Panel	Critical Load (N/m)
1	7760
2	4332
3	4853
4	4332

Table 4.1.1: Buckling load of Skin Panel with Stringer

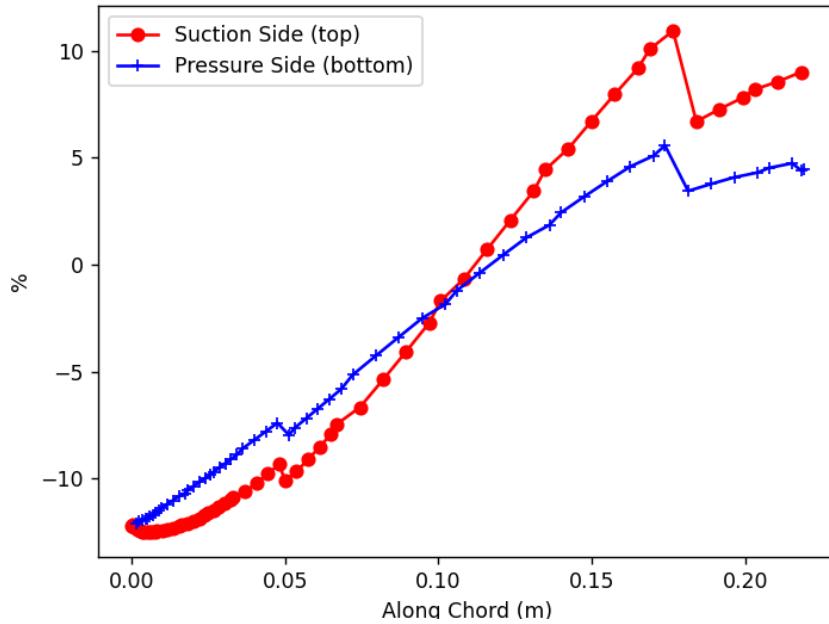


Figure 4.1.7: % of  $F_{cr}$  of the shear flow with Stringers

Thus with the current configuration of 4 stringers placed according to 4.1.7, the above plot reports that the shear flows along the wing cross section are now less than 11% of  $F_{cr}$ , and would prevent the buckling

of the wing skin.

A follow up 2-cell analysis is done in the next section to check if the same configuration and positioning of stringers would be effective for a wing cross section right at the root, which would include an attachment spar and have no control surfaces attached to it.

#### 4.1.3 2-Cell Analysis of Wing Cross Section With Attachment Spar and No Present Control Surface

For a **2-cell analysis** which includes a spar dividing the wing cross-section into 2 cells, 1 cut is made in each cell and the basic shear  $q_b$  is calculated similarly as before. But to close the 2 constant shear flows in the 2 cells  $q_{s,0,1}$  and  $q_{s,0,2}$ , 2 equations are solved. The 1<sup>st</sup> equation is a rate of twist compatibility equation, equating the rate of twist contribution to the spar by each cell. The 2<sup>nd</sup> equation is a moment balance equation, which is equated to 0 taking the shear load  $S_y$  acting along the spar, thus taking the moment balance around the point of intersection of the line of action of  $S_y$  and the Neutral Axis, and equating it the moment to 0. Thus the equations are:

$$\frac{d\theta}{dz}|_1 = \frac{d\theta}{dz}|_2 \quad (4.1.10)$$

For each cell, the total moment balance equation due to shear flow in each cell is given by:

$$\sum_{R=1}^N M_{q,R} = \sum_{R=1}^N \oint_R q_b p_0 ds + \sum_{R=1}^N 2A_R q_{s,0,R} = 0 \quad (4.1.11)$$

where:

- $M$  is the moment due to shear flow,
- $A$  is the enclosed area of the closed section,
- $q_{0,i}$  is the constant shear flow in the closed section
- $q_s$  is the shear flow distribution along the open section,
- $s$  is the distance from a reference point,
- $ds$  is the length element along the section.

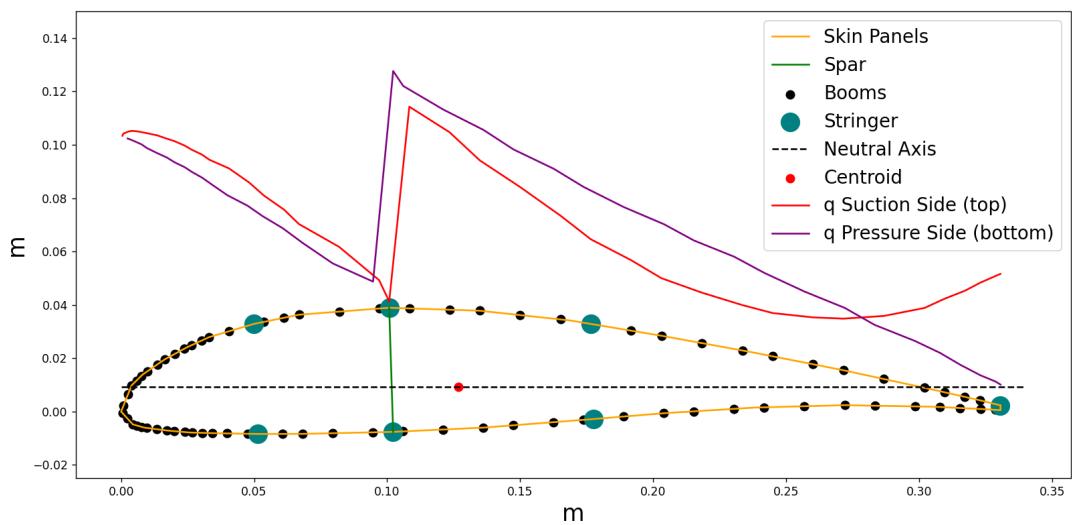


Figure 4.1.8: Shear flow Distribution of 2-cell Wing with Spar without any control surfaces

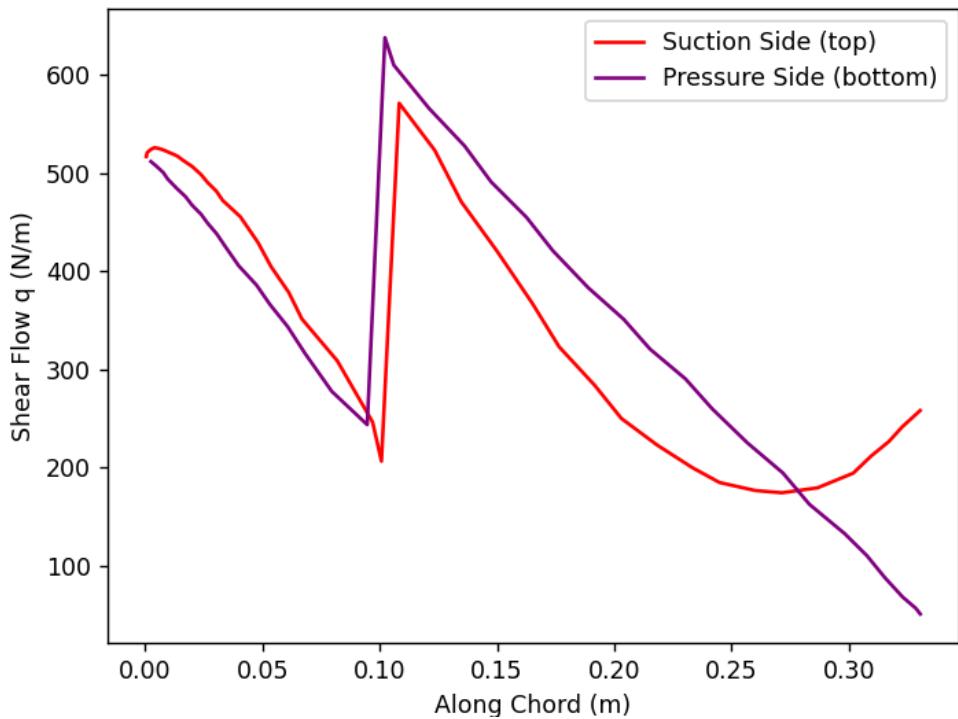


Figure 4.1.9: Shear flow Distribution with Previous Stringer Placement

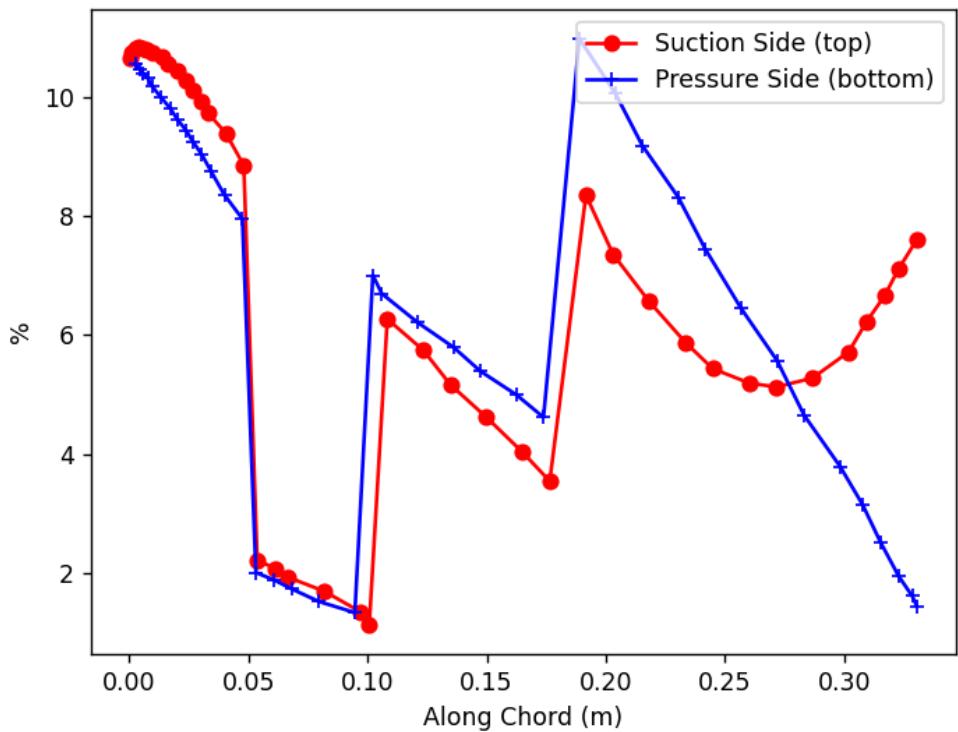


Figure 4.1.10: % of  $F_{cr}$  for Shear Flows around wing with spar and stringers

First, the stringer positions from the 1-cell analysis are replicated here to check if it meets the current design requirement of  $q_{total} \leq 11\%$  of  $F_{cr}$ . It is observed from the graph below that the requirement is met and  $\% F_{cr}$  is limited to less than 11%. Thus the stringer placements are finalized.

## **4.2 Conclusion**

From the above, we have concluded that the wing design requires 7 stringers (3 on Upper surface and 4 on lower surface) positioned as shown in fig 4.1.10. Also, to reduce weight and to ensure that the wingtips are easily replaceable (for cases of wingtip damages due to excessive bank during landing) the last 0.1m of wing section on either wing will be fabricated using PLA.

## Chapter 5

# Torque Estimation for Servo Selection

For an optimal design, the selection of the servo motor responsible for the movement of the control surfaces must ensure that it can effectively accommodate the entire range of control surface deflection, while minimizing any weight penalty.

### 5.1 Torque Estimation

To estimate the torque generated by the control surface, it is essential to consider both the angle of deflection of the control surface and the angle of deflection of the servo motor. The following equation gives us the torque generated:

$$\text{Torque (oz.-in)} = 8.5 \times 10^{-6} \times \left( \frac{C^2 \times V^2 \times L \times \sin(S_1) \times \tan(S_1)}{\tan(S_2)} \right)$$

where:

- $C$  = Control surface chord in cm
- $L$  = Control surface length in cm
- $V$  = Speed in mph
- $S_1$  = Max control surface deflection in degrees
- $S_2$  = Max servo deflection in degrees

#### 5.1.1 Torque Estimation for Ailerons

As per our Aileron design specifications, we have:

$$\begin{aligned} C &= 8.5 \text{ cm} \\ L &= 56.4 \text{ cm} \\ V &= 20 \text{ m/s} = 44.74 \text{ mph} \\ S_1 &= 45^\circ \\ S_2 &= 60^\circ \end{aligned}$$

Hence, we get:

$$\text{Torque} = 28.30 \text{ oz-in} = 2.04 \text{ kg-cm}$$

#### 5.1.2 Torque Estimation for Rudder

As per our Rudder design specifications, we have:

$$\begin{aligned} C &= 10.5 \text{ cm} \\ L &= 40 \text{ cm} \\ V &= 20 \text{ m/s} = 44.74 \text{ mph} \\ S_1 &= 45^\circ \\ S_2 &= 60^\circ \end{aligned}$$

Hence, we get:

$$\text{Torque} = 30.63 \text{ oz-in} = 2.20 \text{ kg-cm}$$

### 5.1.3 Torque Estimation for Elevator

As per our Elevator design specifications, we have:

$$C = 7 \text{ cm}$$

$$L = 75 \text{ cm}$$

$$V = 20 \text{ m/s} = 44.74 \text{ mph}$$

$$S_1 = 45^\circ$$

$$S_2 = 60^\circ$$

Hence, we get:

$$\text{Torque} = 25.52 \text{ oz-in} = 1.84 \text{ kg-cm}$$

## 5.2 Servo Torque

Considering our design requirements, we have chosen the ES 3352 Metal Gear Digital 12G Servo. This servo has a stall torque limit of 2.4 kg-cm at 4.8 Volts and 2.8 kg-cm at 6.0 Volts. The servo motor weighs about 12.4g and is compact in size, making it suitable for our UAV.

## Chapter 6

# Fuselage Design

### 6.1 Fuselage Load Distribution and Structural Analysis

The fuselage serves as the main structure of the aircraft, housing critical components and bearing various loads during flight. In this design, the fuselage is constructed using an aluminum sheet with a thickness of 0.5 mm and is reinforced by bulkheads and longerons made of PLA and Aluminum respectively. The fuselage is streamlined to minimize drag, featuring a circular central section with a length of 1.2 meters and diameter of 0.23m. The empennage includes a tail boom made from PVC and houses both the horizontal and vertical tails whose internal structure is made of PLA and skin is made of Aluminum for weight reduction and enhanced aerodynamic stability.

#### 6.1.1 Component Placement

The internal components are carefully placed along the fuselage to ensure balanced load distribution and structural integrity. Table 6.1.1 provides details about the components housed in the fuselage, their corresponding weights, and their positions along the beam (fuselage). The positions are measured from the nose of the aircraft.

Component	Weight (N)	Position from the Nose (m)
Motor	8.98	0.053
Propellor	5.46	0.053
Battery	22.27	0.186
Wiris	4.21	0.276
Gimbal	4.56	0.276
Environmental Sensor	2.23	0.476
Avionics	1.96	0.481
Empty Weight	29.43	0.326
Nose Landing Gear	0.77	0.15
Main Landing Gear	1.55	0.35

Table 6.1.1: Component Loads and Positions in the Fuselage

### 6.2 Shear Force Analysis

The structural integrity of the fuselage was analyzed using a shear force diagram (SFD) and a bending moment diagram (BMD). The shear force at any point along the fuselage is the result of the cumulative effect of all loads acting up to that point. The bending moment, which is a measure of the bending effect at each section, is obtained by integrating the shear force.

Figure 6.2.1 shows the shear force distribution along the length of the fuselage. Positive values indicate downward forces, while negative values represent upward forces.

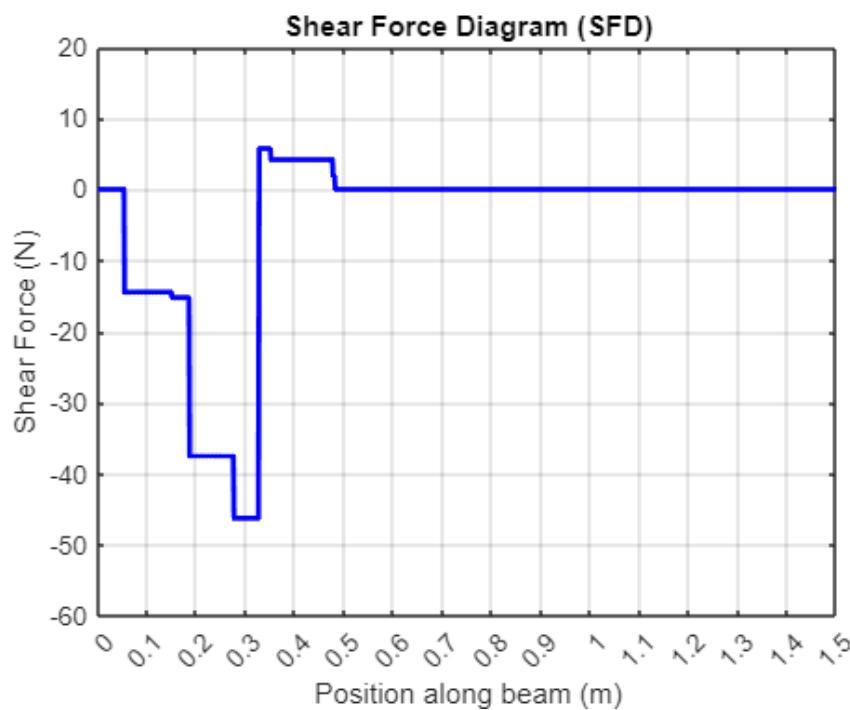


Figure 6.2.1: Shear Force Diagram (SFD) along the fuselage

**Maximum Shear Force: 46.25 N**

### 6.3 Bending Moment Analysis

The bending moment diagram, shown in Figure 6.3.1, illustrates how the bending moment varies along the fuselage. The maximum bending moment occurs near the center of the fuselage due to the combined effect of the structural weight and aerodynamic loads.

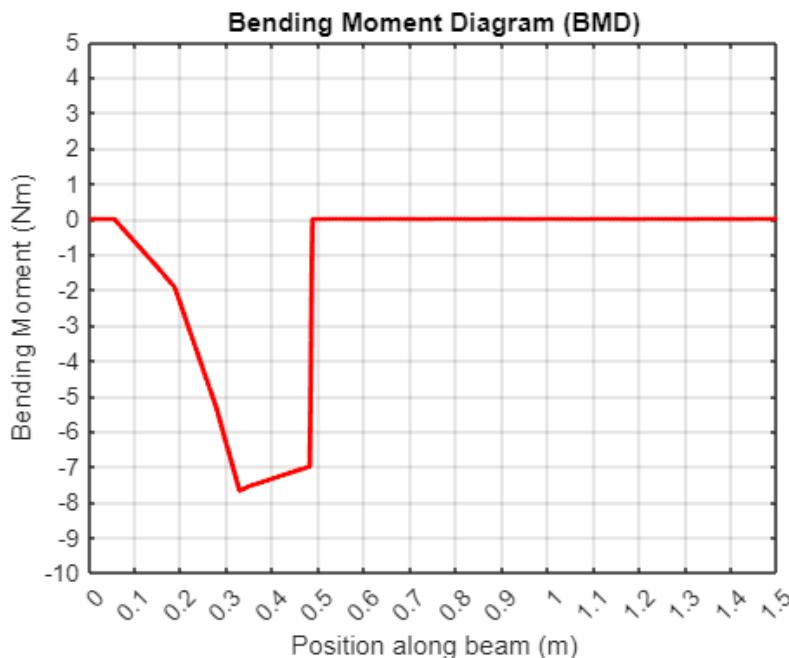


Figure 6.3.1: Bending Moment Diagram (BMD) along the fuselage

**Maximum Bending Moment: 7.68 Nm**

## Chapter 7

# Longerons and Fuselage Bulkhead Design

### 7.1 Longerons

#### 7.1.1 Estimating Number of Longerons required

According to our aerodynamic design, a circular fuselage was chosen. Taking a cross-section of the fuselage at the location of the maximum shear force acting along the fuselage, the shear flow acting along the circumference of the cross section is calculated and hence a design decision is made on the number of longerons required to prevent the buckling of the fuselage skin.

The cross section of the fuselage with a diameter of  $0.230\text{ m}$  or radius of  $0.115\text{ m}$  and skin thickness  $t_D = 0.5\text{ mm}$  was made and idealized by placing 38 equi-distant booms along the circumference as shown in fig 7.1.1. The boom areas were calculated by the following equation :

$$B_1 = \frac{t_D b}{6} \left( 2 + \frac{\sigma_2}{\sigma_1} \right) = 5.38\text{ mm}^2$$

The maximum shear force acting along the fuselage was found to be  $46\text{ N}$  in the previous section. Setting  $S_y = 46\text{ N}$ , the shear flow along the cross section is calculated.

$$q_{total} = q_b + q_{s,0} = \frac{-S_y}{I_{xx}} \sum_{r=1}^n B_r y_r + q_{s,0} \quad (7.1.1)$$

An initial cut is made in the right most section between the 1st and 2nd booms, and the basic shear flow  $q_b$  is calculated across each boom. The closed section shear flow  $q_{s,0}$  is estimated using the moment balance equation, and is then added to  $q_b$  to get the total shear flow across each boom  $q_{total}$ .

$$0 = \sum_{r=1}^n \oint_R q_b p_0 ds + \sum_{r=1}^n 2A_R q_{s,0,R} \quad (7.1.2)$$

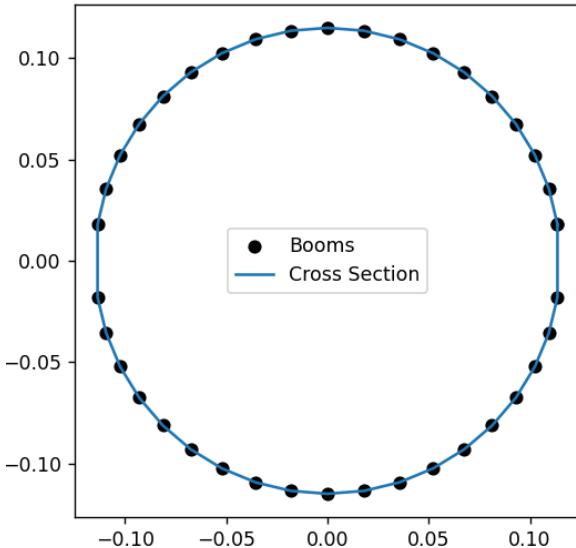


Figure 7.1.1: Fuselage Cross Section with Boom Idealization

The distribution of the total shear flow  $q_{total}$  is reported as a distribution around the fuselage cross section in fig 7.1.2 as well as it's variation with the angle from the centre in fig 7.1.3. It is observed that the shear flow has a maximum value of  $q = 120 \text{ N/m}$  and is maximum at the right and left side of the cross section while it goes to 0 at the top and bottom. This is also reflected in the plot of shear flow wrt angle, where  $0^\circ$  &  $180^\circ$  corresponds to the right and left side where the shear flow is maximum at  $120 \text{ N/m}$ , while  $90^\circ$  &  $270^\circ$  corresponds to the top and bottom side where the shear flow goes to  $0 \text{ N/m}$ .

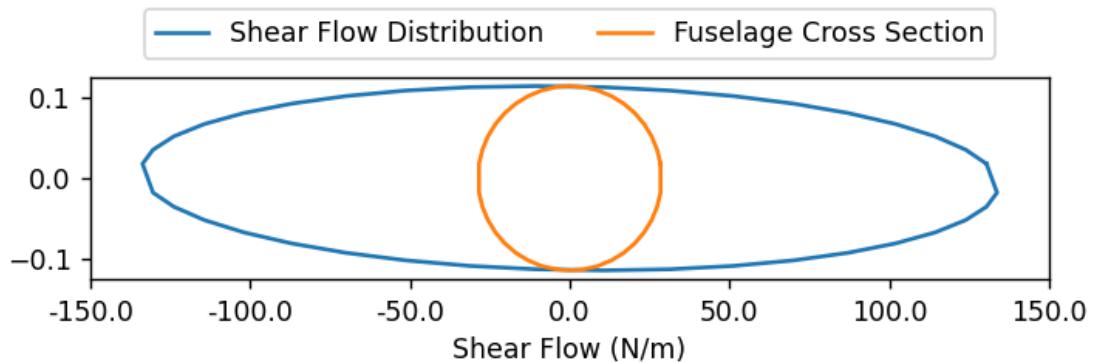


Figure 7.1.2: Fuselage Cross Sectional Shear Flow Distribution

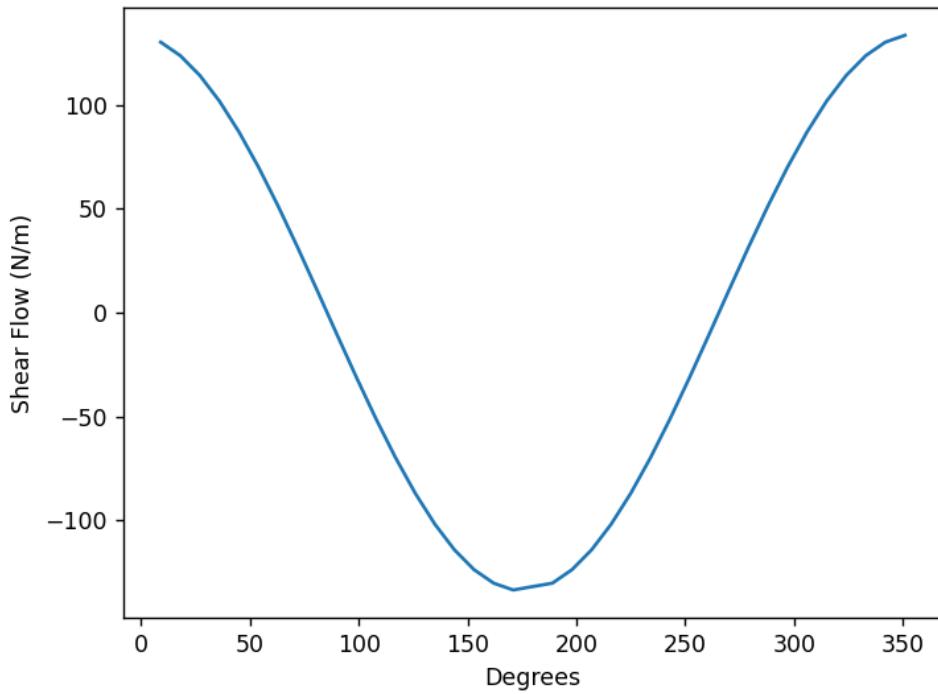


Figure 7.1.3: Fuselage Cross Sectional Shear Flow Distribution wrt Angle

Thus obtaining the shear flow acting along the circumference of the fuselage cross section, the number of longerons required to prevent the buckling of the fuselage is estimated from calculating the critical Force  $F_{cr}$  reported in equation 7.1.3, where  $a$  = length between each bulkhead =  $0.65/4 = 0.16\text{ m}$  and  $b$  = length between each longeron,  $R_a = R_b = 1$  for simply supported joints. Using this equation, a plot of  $F_{cr}$  in  $\text{N}/\text{m}$  was created w.r.t number of longerons reported in fig 7.1.4.

$$F_{cr} = K_{ss} \frac{\pi^2 E}{12(1-\nu^2)} \left(\frac{t}{b}\right)^2 \left[ R_a + \left(\frac{R_a - R_b}{2}\right) \left(\frac{b}{a}\right)^3 \right] \quad (7.1.3)$$

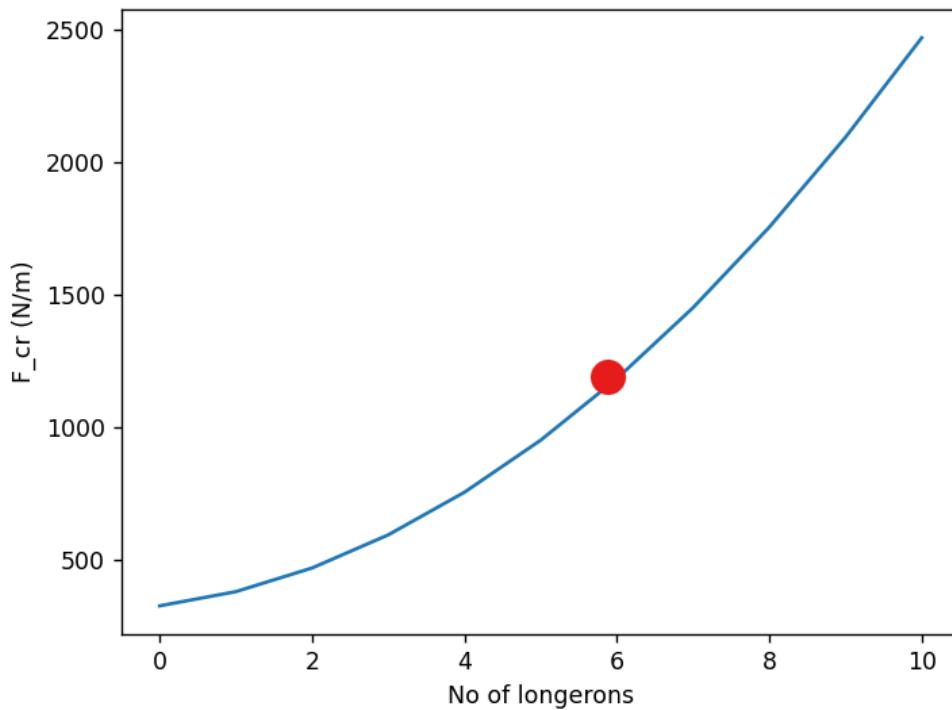


Figure 7.1.4:  $F_{cr}$  vs No. of Longerons

Taking a factor of safety of  $FOS \approx 10$  from the initial spar calculation chapter by considering the stress concentration factor, fatigue factor, factor of safety and maximum load factor, the required  $F_{cr} = 1200N/m$ , which is obtained by selecting 6 equi-distant longerons along the circumference. Thus, this is placed accordingly and illustrated in fig 7.1.5.

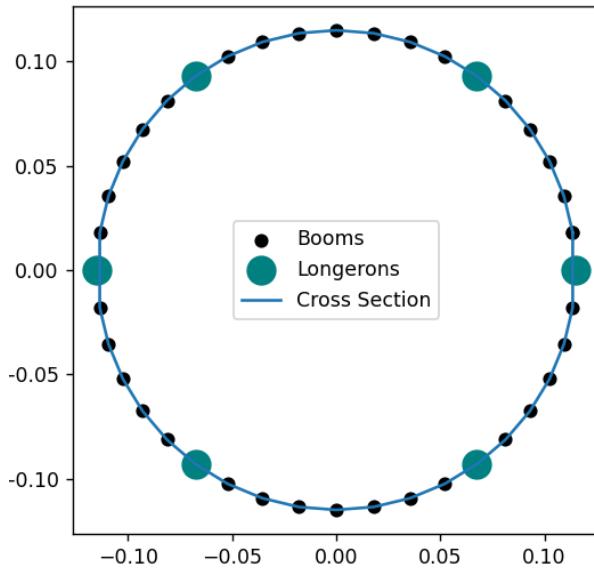


Figure 7.1.5: 6 Longerons placed along Cross Section of Fuselage at  $60^\circ$  angles **starting from the mid section where the shear flow is maximum**

### 7.1.2 Longeron Sizing

In this subsection, the area of the longeron is estimated from the maximum bending moment exerted along the fuselage. From the Bending moment diagram along the fuselage, the maximum values is  $M_{max} = 7.7 N-m$ . Taking a total factor of safety as  $FOS = 10$ , which includes a fatigue factor of 1.5, a stress concentration factor of 3 and a load factor of 3,  $\sigma_{allowable}$  is found to be  $193/10 = 19.3 MPa$ , where  $\sigma_{Al,yield} = 193 MPa$ . Taking  $y$  as the radius of the fuselage i.e.  $y = 0.115 m$  and plugging it in the following equation, we get the required  $I_{xx} = 93000 mm^4$  for the section.

$$I_{total} = \frac{M_{max} \cdot y}{\sigma_{allowable}} = 93000 mm^4 \quad (7.1.4)$$

Now to design the longeron, first the  $I$  contribution by the skin is calculated by the following equation

$$I_{skin} = \frac{\pi * (R^4 - (R - t)^4}{4} = 143000 mm^4 \quad (7.1.5)$$

Since the  $I_{skin}$  already exceeds the desired  $I_{total}$ , no special design requirement needs to be taken for the longeron, and for manufacturability a C-section Aluminium longeron of 0.5 mm thickness with a web of 10mm and flanges of 8mm is chosen.

## 7.2 Bulkhead

A circular bulkhead was designed with a C Cross Section with an outer diameter of 230mm which will be made of PLA and bulkhead thickness of 15mm.

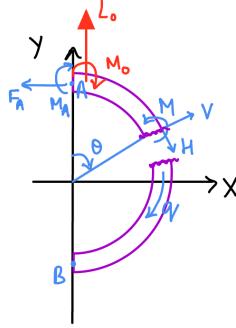


Figure 7.2.1: Symmetrical Half of Bulkhead Section

We write the equilibrium equations for force (H,V) and moment(M) taking a random cut in one half of the bulkhead.  $F_{qy}$  and  $F_{qx}$  are the vertical and horizontal forces and  $M_q$  is the moment due to shear flow.

$$\begin{aligned} F_{qx} &= q \cdot r \cdot \sin(\theta) \\ F_{qy} &= q \cdot r \cdot (1 - \cos(\theta)) \\ M_q &= 2 \cdot \bar{A} \cdot q \end{aligned}$$

$L_0$  and  $M_0$  are the moment load and the moment transfer from the wing.  $F_A$  and  $M_A$  are the reaction forces at the middle / end of the cross-section of the bulkhead.

We start by rearranging the equilibrium equations for horizontal and vertical forces.

### Horizontal Force Equilibrium

The horizontal force equilibrium is given by:

$$H = \frac{F_A - F_{qx} - V \cdot \sin(\theta)}{\cos(\theta)} \quad (7.2.1)$$

### Vertical Force Equilibrium

The vertical force equilibrium is:

$$V = \frac{F_{qy} + H \cdot \sin(\theta) - L_0}{\cos(\theta)} \quad (7.2.2)$$

Now, solve for  $H$  in terms of  $V$  from the horizontal force equation. Substitute this into the vertical force equation:

$$V = \frac{F_{qy} - \left( \frac{F_A - F_{qx} - V \cdot \sin(\theta)}{\cos(\theta)} \right) \cdot \sin(\theta) - L_0}{\cos(\theta)}$$

Simplify this equation to isolate  $V$ :

$$V \cdot \cos(\theta) = F_{qy} + \left( \frac{F_A - F_{qx} - V \cdot \sin(\theta)}{\cos(\theta)} \right) \cdot \sin(\theta) - L_0$$

$$V \cdot (\cos^2(\theta) + \sin^2(\theta)) = F_{qy} \cdot \cos(\theta) + F_A \cdot \sin(\theta) - F_{qx} \cdot \sin(\theta) - L_0 \cdot \cos(\theta)$$

Since  $\cos^2(\theta) + \sin^2(\theta) = 1$ , the equation simplifies to:

$$V = F_{qy} \cdot \cos(\theta) + F_A \cdot \sin(\theta) - F_{qx} \cdot \sin(\theta) - L_0 \cdot \cos(\theta) \quad (7.2.3)$$

Finally, substitute  $V$  back into the equation for  $H$ :

$$H = F_A \cos(\theta) - F_{qx} \cos(\theta) + F_{qy} \sin(\theta) + L_0 \sin(\theta) \quad (7.2.4)$$

## Moment Equilibrium

The moment  $M$  is given by:

$$M = M_A - F_A \cdot r \cdot (1 - \cos(\theta)) + M_q + M_0 \quad (7.2.5)$$

Since we have 2 extra unknowns in the form of  $F_A$  and  $M_A$  we use Castiglano's theorem to solve for it.

## Castiglano's Theorem

The total strain energy  $U_{\text{total}}$  is given by:

$$U_{\text{total}} = \frac{1}{2} \int \frac{M^2}{E \cdot I} dx + \frac{1}{2} \int \frac{H^2}{E \cdot A} dx + \frac{1}{2} \int \frac{V^2}{G \cdot A} dx \quad (7.2.6)$$

where  $A$  and  $I$  are the longitudinal cross-section areas and second moment of areas and  $E = 3.5$  GPa and  $G = 4$  GPa are the elasticity modulus and shear modulus of PLA.

## Failure Analysis

To determine the reaction forces  $M_A$  and  $F_A$ , we apply the energy method. This involves assuming a virtual displacement and virtual rotation caused by the reaction forces and moments, which are directly linked to the partial derivatives of the total strain energy with respect to these forces and moments. The displacements and rotations are considered to be zero and hence, the partial derivatives are also zeros. The coordinates are transformed from rectangular to polar coordinates with  $x = r \cdot \sin(\theta)$ , implies  $dx = x = r \cdot \cos(\theta) \cdot d\theta$

The derivative of the total strain energy with respect to  $F_A$  is:

$$\frac{\partial U_{\text{total}}}{\partial F_A} = \int \frac{M \cdot (r \cdot (1 - \cos(\theta)))}{E \cdot I} r \cos \theta d\theta + \int \frac{H \cdot \cos(\theta)}{E \cdot A} r \cos \theta d\theta + \int \frac{V \cdot \sin(\theta)}{G \cdot A} r \cos \theta d\theta \quad (7.2.7)$$

The derivative of the total strain energy with respect to  $M_A$  is:

$$\frac{\partial U_{\text{total}}}{\partial M_A} = \int \frac{M}{E \cdot I} r \cos \theta d\theta \quad (7.2.8)$$

Solving using the above equations [11.4] we get,

$$F_A = 3.75 \text{ N}$$

$$M_A = 25.93 \text{ Nm}$$

Using this we obtain the variation of Bending Moment around the bulkhead as shown in figure 7.2.2.

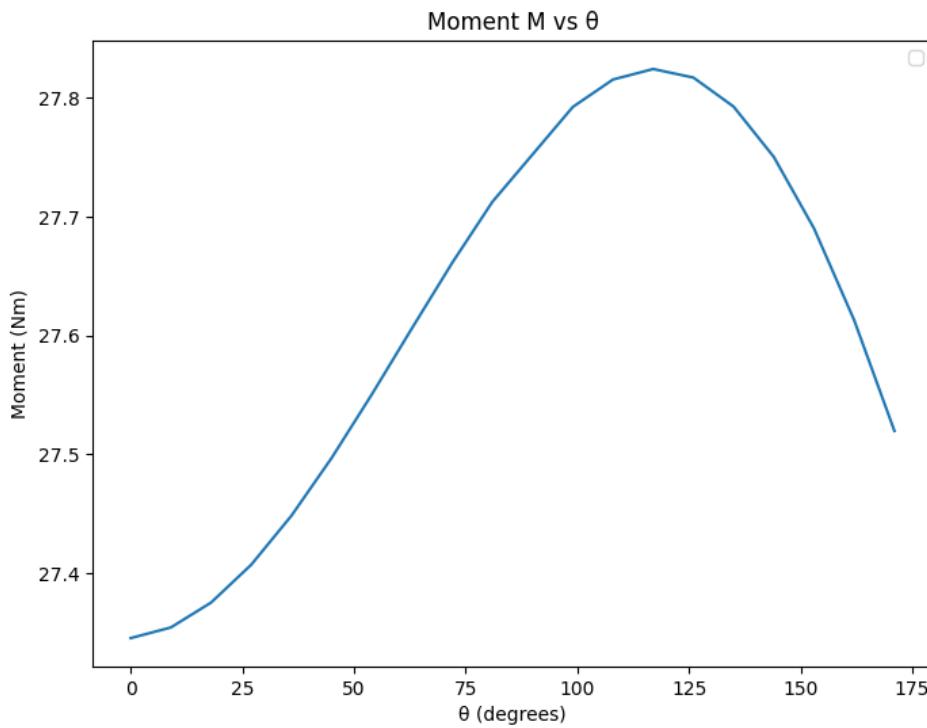


Figure 7.2.2: Bending Moment Diagram of Bulkhead

The maximum bending moment from this plot,  $M_{max} = 27.8 \text{ Nm}$

Using this, we can now calculate the required moment of the bulkhead cross section,  $I_{req}$ . The yield stress is based on a factor of safety (FOS), which accounts for uncertainties in material strength and loading conditions.

Given, for PLA:

- Maximum bending moment,  $M_{max} = 27.8 \text{ in N}\cdot\text{m}$
- Factor of Safety, FOS = 10
- Yield stress of the material,  $\sigma_{yield} = 50 \text{ MPa}$
- Maximum distance from the neutral axis,  $y_{max} = \frac{t}{2}$ , where  $t$  is the thickness of the material

The yield stress considering the factor of safety is given by:

$$\sigma_{yield} = \frac{\sigma}{FOS} = 5 \text{ MPa}$$

Using the formula for the required moment of inertia:

$$I_{req} = \frac{M_{max} \cdot y_{max}}{\sigma_{yield}}$$

Substituting the values for  $M_{max}$ ,  $y_{max}$ , and  $\sigma_{yield}$ , we calculate  $I_{req}$  to ensure the structure remains within the allowable stress limits.

$$I_{req} = \frac{M_{max} \cdot \frac{t}{2}}{\sigma_{yield}}$$

The required moment of inertia of bulkhead cross-section is  $41000 \text{ mm}^4$ .

Using this required I, we do the bulkhead sizing. Since it is a C section, the area moment of inertia is given as

$$I_{req} = 2 \left( \frac{w}{2} \right)^2 t_s + \frac{t_s w^3}{12}$$

where:

- $t_s$  is the thickness of the flange (10 mm).
- $w$  is the unknown width we need to calculate.
- $t$  is the thickness of bulkhead considered (15mm).

The width of web of the C section of bulkhead comes out to be around 21mm.

## Chapter 8

# Landing Gear Design

For an appropriate design of a landing gear system for our UAV, we shall describe the following parameters.

- Configuration
- Fixed or retractable
- Landing Gear Geometry
- Load on each strut
- Tire sizing
- Nose Wheel Steering

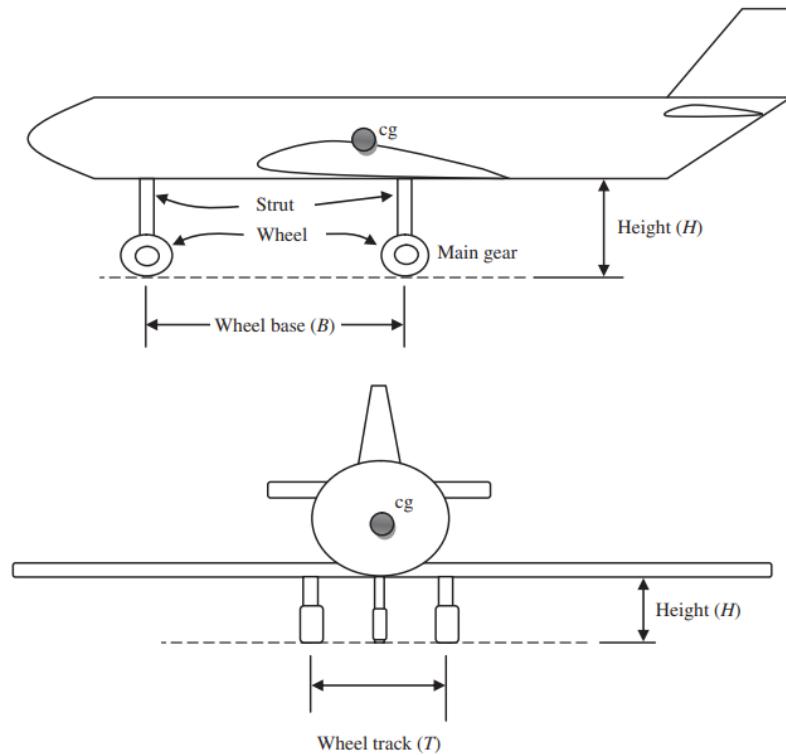


Figure 8.0.1: Landing Gear Parameters [2]

### 8.1 Landing gear Configuration

Based on our CG and design specifications, the **Tricycle LG** is the best fit. The Tricycle LG is the most widely used landing gear configuration. Figure 8.1.1 shows the Tricycle Landing gear in a typical aircraft. Two main gears are at the same distance from the CG in the longitudinal axis and are symmetrical about it; thus both carry the same load. The forward gear is far from the CG (compared with the main gear); hence, it carries a much smaller load.

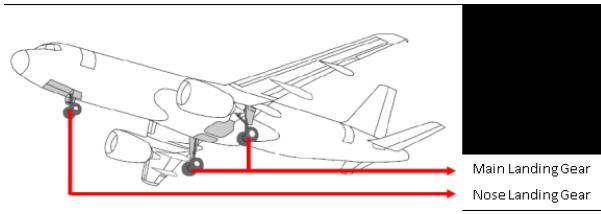


Figure 8.1.1: Tricycle Landing Gear

As a major design configuration for our UAV is low weight and low cost and to minimize complexity it is prudent to design a Fixed Landing gear setup so as to reduce the cost and weight. A general assessment of the same is presented below.

### 8.1.1 Landing Gear Retraction

Additionally, since the maximum speeds to be attained are relatively low, the need for retraction of the Landing Gear is not envisaged.

## 8.2 Landing Gear Geometry

### 8.2.1 Position of Main LG and Nose LG

From our design specifications and fuselage design, we have a fuselage length of 1.5 m. Based on the distribution of the loads within the fuselage, we have our CG location to be about 0.32 m from the nose. Taking into account a CG margin of 10% of the total length of the fuselage, we get forward CG point to be about 0.29 m and the aft CG point to be about 0.35 m.

Based on the above specifications, we have chosen our nose gear to be ahead of the forward CG point and the main gear to be behind the aft CG point as shown above. Additionally, considering the location of the bulkheads, we have chosen the distance of the Nose LG from the nose to be about 0.15 m and the main LG from the nose has been chosen to be about 0.35 m.

This gives us a **wheel base of 0.2 m.**

### 8.2.2 Landing Gear Track

The next parameter to be defined in the geometry is the track of the wheels. The landing gear track is usually set as a percentage of the wingspan or width of the fuselage (according to 9.5.3 of Sadrey [2]). This ensures adequate stability during taxiing, takeoff, and landing, without making the gear too wide.

Since our design has a *circular fuselage with a diameter of 0.23 m and we have a wing span of about 2.8 m*, we have chosen a **wheel track of 0.75 m.**

### 8.2.3 Landing Gear Height

With the position of the main LG and the nose LG properly defined, we can now determine the height requirements of the landing gear. The two main conditions for the determination of the LG height are as listed below,

1. Aircraft General Ground Clearance Requirement
2. Take-Off Rotation Ground Clearance Requirement

We shall now look at each of these in detail.

#### Aircraft General Ground Clearance Requirement

One of the primary functions of the landing gear is to protect the aircraft structure from the ground. This job is performed by providing a clearance from the ground. The clearance is measured from the lowest point of the aircraft to the ground. In our case, the prop tip will be the lowest point closest to the ground.

As a general rule (As per 9.5.1.2 of Sadrey[2]), for a propeller aircraft we take a clearance of 0.2 m from the propeller tip as the height of the Landing gears. As per our design, we have a fuselage diameter of 0.23 m with a propeller diameter of 0.406 m. Applying the 0.2 m ground clearance to these design specifications, we get the **landing gear height from the fuselage to be 0.288m** (including tyres).

### Take-Off Rotation Ground Clearance Requirement

An aircraft is usually rotating about the main gear in order to increase the lift to prepare for take-off (see Figure 8.2.1). This is also true for landing operation, in which the aircraft rotates to gain a high angle of attack. The height of the landing gear must be set so that the tail or rear fuselage does not strike the ground during the take-off rotation or landing with a high angle of attack.

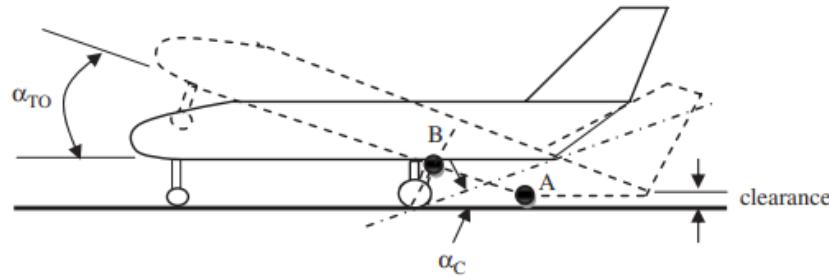


Figure 8.2.1: Take off Ground Clearance

The take-off rotation ground clearance requirement to prevent a fuselage hit is as follows:

$$\alpha_C \geq \alpha_{TO} \quad (8.2.1)$$

where the clearance angle is, [5]

$$\alpha_C = \tan^{-1} \left( \frac{H_f}{AB} \right)$$

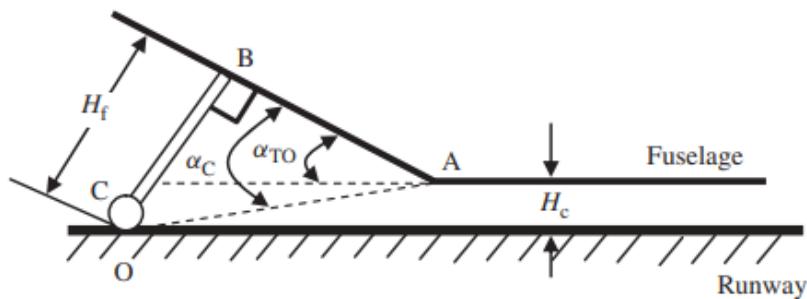


Figure 8.2.2: Clearance Angle Estimation

From our design specifications,  $H_f$  is **0.288 m** and  $AB$  as **0.3 m** as depicted in Fig 8.2.2. Hence the **clearance angle would be  $43.83^\circ$** . In our case the **take off angle is about  $12^\circ$**  and hence our design ensures that the tail would not touch during take off or landing.

#### 8.2.4 Landing Gear Overturn Angle

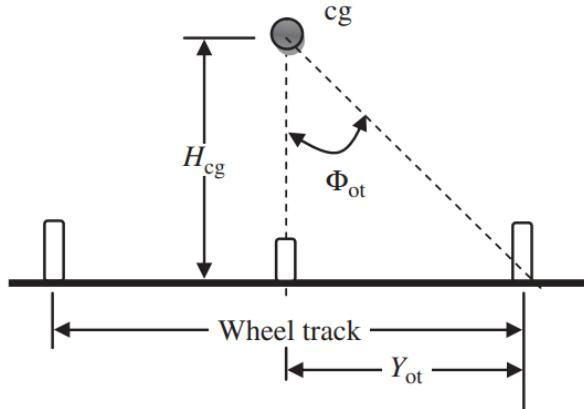


Figure 8.2.3: Estimation of Overturn Angle

The wheel track of the main wheel should be arranged so that the aircraft cannot roll over too easily due to wind or during a ground turn. The overturn angle is the angle which is critical to the aircraft overturn. To determine the overturn angle, look at the aircraft front view, the angle between the vertical line passing through the aircraft CG and the line between the aircraft CG and that of the main wheels is the overturn angle  $\phi_{ot}$  (shown in Fig 8.2.3). In the figure, the parameter  $H_{cg}$  is the height of the aircraft CG from the ground. As a rule of thumb, the wheel track must be such that the overturn angle ( $\phi_{ot}$ ) is inside the following recommended limit,

$$\phi_{ot} \geq 25^\circ$$

Based on our dimensions, we find that the overturn angle ( $\phi_{ot}$ ) is  $52.47^\circ$ . Which meets the stability condition for overturn criteria.

#### 8.3 Load on each Landing Gear

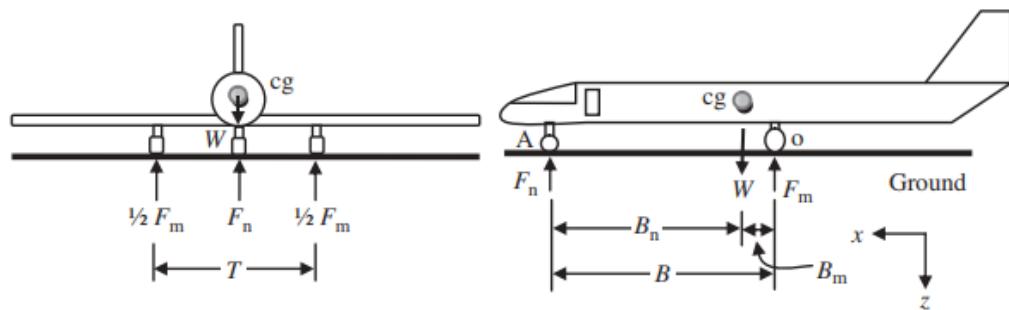


Figure 8.3.1: Loads acting on the LG

Figure 8.3.1 shows a stationary aircraft with a tricycle landing gear on the ground. The aircraft weight ( $W$ ) is carried by three wheels (i.e., two main and one nose gear). The loads on nose and main gears are denoted by  $F_n$  and  $F_m$  respectively. Due to the ground mobility (i.e., steering) requirement, typically the nose gear must not carry less than about 5% of the total load and also must not carry more than about 20% of the total load (e.g., aircraft weight). Thus, the main gear carries about 80–95% of the aircraft load.

Calculation of the static loads on each gear is performed by employing equilibrium equations. Since the aircraft is in static equilibrium, the summation of all forces in the  $z$  direction must be zero:

$$\Sigma F_z = 0 \implies F_n + F_m = W \quad (8.3.1)$$

Furthermore, the summation of all moments about O is zero.

$$\Sigma M_o = 0 \implies F_n b - WB_m = 0 \quad (8.3.2)$$

Applying the above equations with our design specifications.

We find the Load on the Nose LG is 10.34N and that being applied on the main LG is 75.9 N (**Each main LG will have a load of 37.95N**).

This implies that the nose LG carries a load of about 12% and the main LG carries about 88% of the total load. These specifications also ensure good ground mobility of the mini UAV.

## 8.4 Selection of Tyres

Based on our load calculations, design parameters, ground clearance requirements, extensive market survey and based on tyres being utilized by UAVs of similar weight and dimensions we have selected a *3.5 inch PU tyre*. The main features of the same are as follows:-

- Core made of bore retrofitted copper pipe, force balanced and wear-resistant
- Golden Aluminum Hub
- Size: 3.5inch
- Weight: 68g

## 8.5 Nose Wheel Steering System

Considering the maneuvering requirement during taxi, minor course adjustments to be catered for during take off roll and landing run the team has chosen to use a Nose Wheel Steering system. The nose wheel will have a freedom of rotation of 45 Deg and enable the aircraft to efficiently maneuver on ground. The nose wheel steering will be connected to the rudder and the flight controller to enable its control on ground.

## 8.6 Main Landing Gear Assembly

To design the Main LG, we must first quantify the moments and the loads acting on the landing gear assembly. To quantify the moments, we first draw the Free Body Diagram of the Landing Gear as shown in 8.6.1. Here, **Section 1 has length of 0.24m** and **Section 2 has a length of 0.13m**.

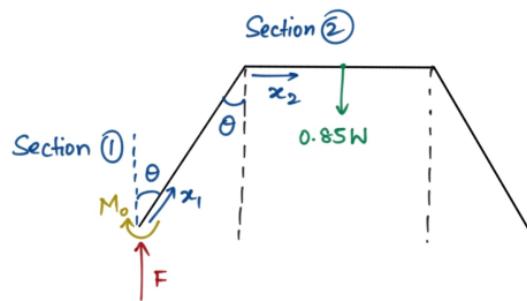


Figure 8.6.1: Free Body Diagram of Landing Gear

Based on the above, we calculate the moment acting at various sections of the Landing Gear [11.6] and accordingly plot the Bending Moment Diagram as shown below 8.6.2.

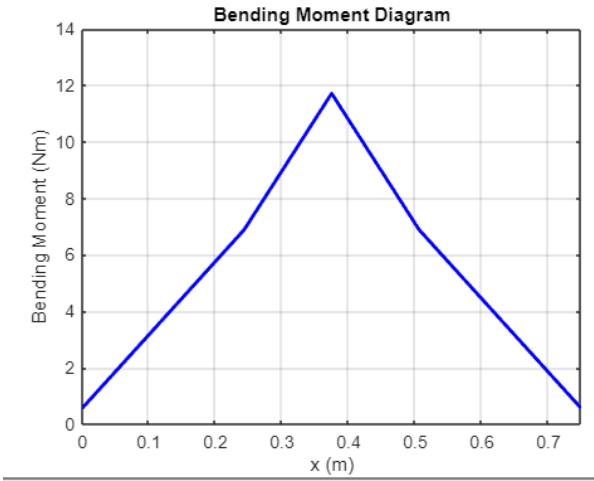


Figure 8.6.2: Bending Moment Diagram of Landing Gear

From, the BMD we find that Maximum Bending moment to be **11.71 Nm**.

We know that,

$$I_{\text{required}} = \frac{M_{\text{max}}y}{\sigma_{\text{allowable}}} \quad (8.6.1)$$

Hence, to optimize the size and weight of the Landing gear strut we chose a C Section cross section. From the optimization matrix [11.6] and considering manufacturing constraints we find a **C Section with web of 25mm and flange of 35mm with a thickness of 1mm** to be the optimised solution.

Additionally, we must also ensure that the Landing Gear Assembly does not fail under buckling load. For which we calculate the axial stress acting on Landing Gear Assembly and compare the same with the critical load for the section using the equations below.

The axial force in the landing gear can be calculated as:

$$F_{\text{axial}} = F \cos(\theta) \quad (8.6.2)$$

Therefore, the maximum axial stress in the landing gear is:

$$\sigma_{\text{axial}} = \frac{F_{\text{axial}}}{A} \quad (8.6.3)$$

For the rectangular section, the critical load for buckling is:

$$\sigma_{\text{cr}} = \frac{\pi^2 EI}{AL_{\text{eff}}^2} \quad (8.6.4)$$

For our specifications, we have

- $\theta = 45^\circ$
- $E = 69 \text{ GPa}$
- $L_{\text{eff}} = \frac{1}{\sqrt{2}}$
- $F = 36.68 \text{ N}$
- $A = 95 \times 10^{-6} \text{ m}^2$

We find that the **Max Axial Stress** ( $\sigma_{\text{axial}}$ ) is about **273 KPa** and applying a **factor of safety of 10.5**, we get the **axial stress to be about 27.3 MPa**. Comparing this Axial stress with the critical load for the section of about 134.14 MPa, we can conclude that the section would not fail under buckling load.

However, during fabrication, it was observed that the C section strut was unable to satisfactorily withstand torsional loads and failed during twisting. Hence, the team decided to redesign the landing gear strut with a hollow rectangular strut. The dimensions of the hollow rectangular strut was chosen based on the nearest available size which would be able to withstand the bending, buckling and torsional loads. A hollow rectangular strut of 1 mm thick aluminum alloy with a cross section of 20mm x 35mm was thus chosen.

## Chapter 9

# Bill of Materials

Sr No.	Component	Vendor	Specification	Quantity (kg or pc)
1	Aluminum sheets	LNG Enterprises	Al 6061 Sheets 0.5 mm Thickness	10
2	Landing Gear Assy	Amazon	Aluminum Main Landing Gear Wheel Kit RC Airplane	1
3	Engine mounting brackets	Amazon	RC Airplane Motor Mount, Aluminum Alloy Motor Support	1
4	Nuts bolts	Sri Krishna Hardware	Slotted M3 nut bolt pair with washers	50
5	Rivets	Amazon	0.2mm X 20mm Steel Aluminum Blind Rivets	200
6	Tyres	rchyderabad.com	3.5 Inches 89mm PU Wheels	3
7	Servo	rchyderabad.com	Metal Gear Digital 12G Servo	7
8	Control rods	Amazon	Steel Pushrods, Nylon Lock on Control Horns	10
9	Link mechanisms	Amazon	Adjustable Pushrod Connector Linkage Stopper	1
10	PLA for 3D printer	Robu.in	eSun PLA+ 1.75mm Filament 1kg	2
11	Tail boom	Amazon	6061 Aluminum Tube, 48" Length	1
12	Hinges for door	Amazon	PVC Pinned Hinges for RC Plane	10
13	Foam filling	Amazon	EPE Foam Sheet 12x12 Inches, 15mm Thickness	3

Table 9.0.1: Structural Components

Sr No.	Component	Vendor	Specification	Quantity (kg or pc)
1	Motor	Robu.in	T Motors AT7215 30CC KV220	1
2	Propeller	Raceway Impex	TF 16 * 8 Propeller	1
3	ESC	Robokits India	T-Motor AT 115A 14S ESC	1

Table 9.0.2: Propulsion Components

Sr No.	Component	Vendor	Specification	Quantity (kg or pc)
1	Flight Controller	Robu.in	Pixhawk Combo Kit	1
2	Battery	Imagine Innovation	22.8v Tattu Battery 6s 22.2v 18000mah Lipo	1
3	Transmitter & Receiver	Robu.in	FlySky FS-T6 6CH Transmitter with FS-R6B Receiver	1
4	Connecting Wires	Amazon	Silicone Wire	1

Table 9.0.3: Avionics Components

Sr No.	Component
1	Spanner multi-size
2	Screwdriver Set
3	Riveting tool
4	Drilling machine
5	Twister
6	DMM

Table 9.0.4: Tools

## Chapter 10

# Fabrication

### 10.1 Brief of Assembly

The assembly process for joining the individual parts to the skin was using rivets of 3mm dia. The fuselage consists of the Nose Cone, Platform structure, Longerons, Bulkheads (with provision for main wing and tail attachment). The nose cone is the housing of the motor and the platform is made to attach the individual avionics equipment.

The Main wing and the fuselage was joined by adding an extension of the spar into a slot provided inside the fuselage. The joint between the Empennage and the fuselage was made using a tail boom of 4 inch dia. The Empennage assembly with the Vertical tail and Horizontal tail were planned to be joined to each other using a tail boom attachment. The complete fuselage along with the main wing and the Empennage assembly were supported by two main Landing gears and a single steerable nose wheel assembly.

The assembly was started with the individual wings. The wing ribs were 3D printed and attached to each other using the four stringers on each side. The spars, made using machining techniques were attached to the first 3 ribs and an extension was provisioned for attachment to the fuselage. While the team was awaiting the 3D print of the bulkheads, we continued making the longerons for the fuselage. Once the 3D printed bulkheads were received, they were attached to the longerons to complete the fuselage skeleton. 2 different iterations of the bulkheads were 3D printed and trialed on. Further, the skin was attached to the wings to complete the main wing assembly on either side. Simultaneously, the team was working on similar lines to fabricate the horizontal tail. The horizontal tail was then attached to the tail boom attachment and further connected to the fuselage using the tail boom. The team was unable to complete the design and print of the vertical tail due to paucity of time.

### 10.2 Main Wing

The wing design consists of 18 ribs, with 9 ribs on each half of the wing, and 6 stringers for additional support. The ribs were made from PLA, while the stringers were constructed using 0.5mm thick Al 6061. The wing skin was also made from 0.5mm Al 6061.

The wing features ailerons and separate flaps, both 3D printed in PLA to maintain a lightweight design. The fuselage attachment spar, also made of 0.5mm Al 6061, connects to the second rib on each half of the wing, ensuring attachment between the wing and fuselage. The wing structure was designed to meet performance requirements while minimizing weight and material usage.

#### 10.2.1 Rib

The ribs of the main wing of the UAV were fabricated using 3D printing technology with PLA (Polylactic Acid) material. To ensure a balance of strength and weight, the infill density was set to 30%, and a rectilinear infill pattern was used for optimal support and structural stability. This approach allowed for precise, cost-effective manufacturing while maintaining the necessary rigidity for the wing's performance. The use of PLA also ensures durability and ease of production, making it an ideal choice for the rib structure.

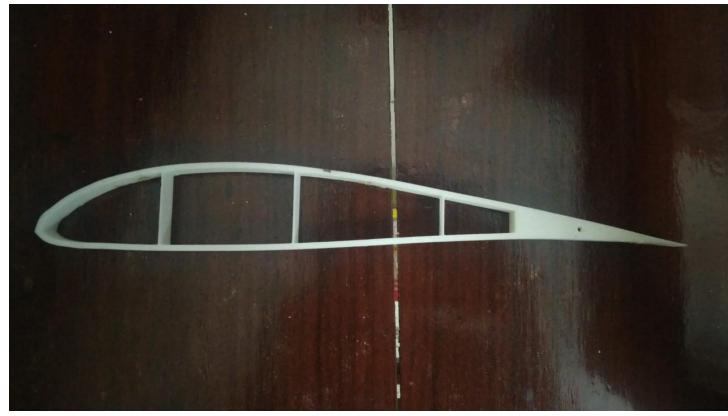


Figure 10.2.1: Main Wing Rib

### 10.2.2 Stringer

The stringers were designed as L-sections with dimensions of 10mm x 10mm and a thickness of 0.5 mm, positioned at 6 optimized locations around the wing's cross section. These locations were carefully chosen based on shear loading analysis, ensuring efficient distribution of stresses across the wing structure. The stringers extend continuously along the length of each half-wing, measuring 0.14 meters in total.

The positions of the stringers were marked along the ribs, based on the optimized graphic design. To fabricate the stringers, precise cuts were made on a 0.14 meter long sheet of 0.5 mm Al 6061. A press brake machine was then used to shape the stringers into the required L-section.

### 10.2.3 Spar

The attachment spar was designed to connect the 2nd rib of each wing half and extend continuously through the wing structure. Based on design calculations that considered bending moments (outlined in Chapter 3), the final specifications determined that 0.5mm thick Al 6061 would be the appropriate material for the spar. The spar was designed with a C-section, featuring a web height of 40mm and flange lengths of 45mm, providing the necessary support while keeping the weight minimal.



Figure 10.2.2: Wing internal assembly with 1st iteration of Attachment Spar (circled in green)

Two iterations of the spar-rib coupling were tested during the design process. The first iteration split the attachment rib in half, as the maximum thickness of the rib matched the web height of the spar. In this version, the rear half of the rib was positioned beneath the spar, while the front half connected to the spar's leading edge.

The second iteration reduced the spar's web height to 36 mm to better fit through the rib at the point of maximum thickness, and is 100 cm from tip to tip. This adjustment allowed the spar to pass through the rib while keeping the first two ribs intact. However, this change led to a slight compromise in the spar's ability to resist bending loads, as the web height was reduced to accommodate the rib's geometry.

#### 10.2.4 Wing Skin

According to shear loading calculations done in Chapter X, the wing skin thickness required was taken as 0.5 mm of Al 6061. The length along the rib according to Fusion360 was 668 mm, thus 4 wing skin sections of 670 mm length and  $0.14/2 = 0.07$  m in width was marked and cut using the shear cutter. After which, the position of the ribs and the cut-outs for the flaps and ailerons were marked out.



Figure 10.2.3: Wing Skin Section with Aileron

Each section was attached from the bottom surface of the wing, as 2 men pulled and stretched the skin as another made the appropriate holes through the skin and underlying ribs using a 0.4 mm drilling machine which was subsequently attached by blind riveting.



Figure 10.2.4: Wing Skin Section with Flap Cutout

### **10.2.5 Challenges Faced during Wing Manufacturing**

The L-Section Stringers were planned to be made by cutting the 0.5mm sheet into long strips and then bend them by 90 degrees to make the L Section. However, it was realized that the shear cutting machine available in the workshop was unable to cut the 0.5mm sheet and hence the team was forced to manual cut long strips of the stringers by using shear cutters. This process was tedious and produced inaccuracies which then further caused us to redo the process again. Also, during the process of bending the long strips, it was observed that the bending machine was only able to bend the strip upto about 120 deg. Hence, the stringers had to be manually bent to 90 Degrees using wooden mallets. This hammering process further reduced the strength of the stringers and on occasions the stringer had to remade from scratch.

During the process of riveting of the wing ribs to the stringers, the team faced challenges due to a few faulty rivets. This challenge was further compounded as the process of de-riveting, the original hole diameter increased. This caused the team to use bigger rivet size and on occasions also add a small sheet of metal to aid the riveting process. However, on one occasion, one of the ribs was damaged beyond repair and further assembly of the main wing was halted as the particular rib had to be reprinted. The team was also unable to further proceed with the assembly of the stringer to the wing ribs as the stringers were made as a single long element extending from the wing rib to the wing tip.

During the attachment of the wing skin, the 0.5mm thickness aluminum sheets proved difficult to work with and shape along the airfoil manually. This introduced errors at the trailing edge and the wing-skin sections had not been fully fastened parallel to the ribs.

## **10.3 Fuselage**

During the design process, the team had opted for a circular fuselage. The main fuselage contained the nose, bulkheads, longerons, landing gear adapter, and platform structure. Since the bulkheads were circular and the nose was chosen to be an elliptical cone, the bulkheads and nose were 3D printed using PLA. The longerons were made using 0.5mm Al 6061 sheets while two different iterations of the platform structure was created using a slide-through PLA 3D print and a 1.0mm Al 6061 sheet.

### **1<sup>st</sup> Iteration**

A total of four bulkheads were selected during the design and calculation phase. The 1<sup>st</sup> iteration of the bulkhead was designed utilizing 1 bulkhead attached to the nose-cone, 2 bulkheads to support the fuselage structure through which the attachment spar would go through, and the final bulkhead at the end of the fuselage to support the empennage. Longeron slots were also given with a 0.2 mm tolerance, as well as a C-section hole in the fuselage structure to slide the attachment spar through.



Figure 10.3.1: 1<sup>st</sup> iteration of Fuselage

**Challenges Faced:**

Due to dimensional limitations of the 3D printer, the bulkheads had to split into multiple parts. Also, as the dimensions were large there was a considerable delay in getting the bulkheads 3D printed.



Figure 10.3.2: Longerons sticking out due to less tolerance

To cater for tolerances, 0.2mm gap was given at the attachment points. However, once the assembly was started, it was realized that the 0.2mm gap was insufficient. Also, as reprinting the bulkheads would cause further delay in assembly, the team manually enlarged the gap using hacksaw blades.

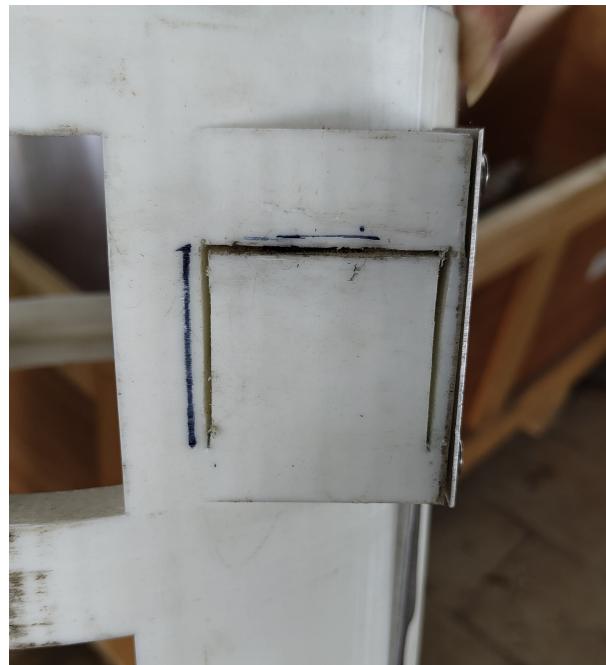


Figure 10.3.3: Spar Slot for fuselage 1st iteration

Similar issue was also faced during the assembly of the spar to the fuselage. However, the challenge was further amplified due to the limited accessibility in this area. Hence the team opted to use waterjet machining to enlarge the gap in the fuselage but the attempt was unsuccessful as the water jet facility was unable to fix the structure in the rig. The team then opted to use a heated hacksaw blade to enlarge the gap.

## 2<sup>nd</sup> Iteration

The 2<sup>nd</sup> iteration of the bulkheads increased the tolerance to 1 mm which gave the longerons a slide-in friction fit. Additional changes were made; the bulkhead spacing was kept constant as was fixed by the initial design analysis, and the tail boom housing was shortened to remove excessive weight. The bulkheads were modularized to benefit 3D printing, as they were printed separately. The spar slot was changed from a C section slot to a Box slot to facilitate the easier sliding in of the attachment spar. The tail boom hole was also reduced from 0.5 m diameter to a 0.4 m diameter, significantly cutting down the weight addition from it.



Figure 10.3.4: CAD Model of 2<sup>nd</sup> iteration of Fuselage and Landing Gear Adapter

The nose cone was fastened using 6 screws to the 1<sup>st</sup> bulkhead, and the longerons were riveted to the 10 mm extensions from the bulkhead with riveting holes in it. Additionally, the landing gear adapter was modelled as a 3D printed C section and attached to the last bulkhead with three 7 cm screws going through the bulkhead.



Figure 10.3.5: 2<sup>nd</sup> iteration of Fuselage with Longerons, Landing Gear Adapter, Attachment Spar and Tail Boom

## 10.4 Tail and Empennage

The empennage of the UAV consists of a tail boom made from a PVC pipe, extending 0.65 meters from the final bulkhead and secured to the last three bulkheads. Two PVC pipes were used in the design: one with a 5 cm diameter and 1 cm thickness for the first fuselage iteration, and a second with a 4 cm diameter and 0.5 cm thickness for the subsequent iteration.

A 3-way tail adapter was modeled and 3D printed to hold the horizontal and vertical tails in place. This adapter featured sockets for each tail.

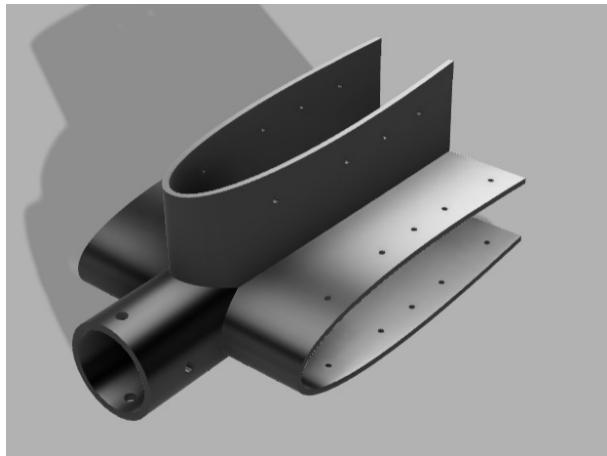


Figure 10.4.1: Tail Adapter CAD Model

The tail ribs were designed using the NACA 0014 airfoil and the specified chord length. Four ribs were chosen for each half of the tail, which were connected by a continuous spar running along the length of the tail. A 0.3mm thick Al 6061 sheet was wrapped around to form the tail skin, which was then riveted into place.



Figure 10.4.2: 3D printed Horizontal Tail Rib

The horizontal and vertical tail sections were attached to the tail adapter using rivets, while the tail adapter itself was fastened to the PVC tail boom with two 6 mm diameter screws.

**Challenges Faced:** All the Vertical Tail Ribs necessary for fabrication were not printed on time as well as the elevator and rudder panels and thus the could not be assembled before the deadline.



Figure 10.4.3: Empennage consisting Horizontal Tail and Tail Boom connected with Tail Adapter

## 10.5 Landing Gear

The original design of the main landing gear was to be made of a C Section with web of 25mm and flange of 35mm from a 1mm AL 6061 sheet. However, post fabrication of the main Landing gear, it was realized that the design was unable to withstand torsional loads satisfactorily. Hence, to strengthen the design, the cross section was changed to a hollow rectangular box section of 20mm x 35mm with a thickness of 1mm .



Figure 10.5.1: Main Landing Gear

Similarly, the nose landing gear design was designed in two parts namely nose wheel support strut and the nose wheel assembly with a spring shock absorber. The nose wheel support strut connects the nose bulkhead to the nose wheel assembly. The presence of the inbuilt spring shock absorber helped to damp the impact on the avionic components in the fuselage.

However, the attachment of the said part was unable to withstand the weight on the forward section of the fuselage. Additionally, a roller bearing was added to the nose landing gear for provisioning of the nose wheel steering. However, the attachment of the roller bearing to the nose wheel strut (made of PLA) was challenging. The team used soldering iron to mold the PLA around the roller bearing. However, the said assembly process failed during fatigue loading.

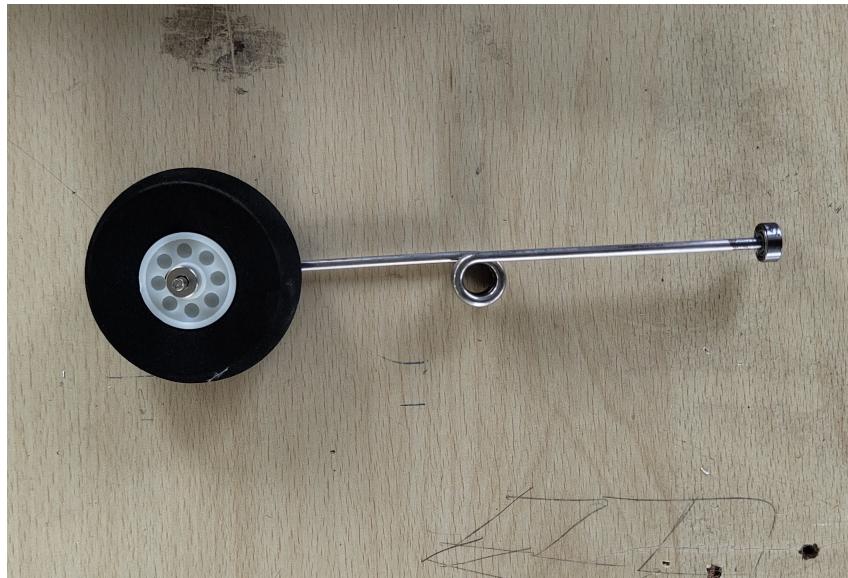


Figure 10.5.2: Front Landing Gear Wheel and Steerable Ball Bearing

## 10.6 Summary of Challenges Faced

### 10.6.1 Challenges Faced

During the fabrication process, the following challenges were encountered:

1. The stringers (L-Section) were planned to be fabricated by cutting a 0.5mm sheet into long strips and bending them to 90 degrees. However, the available shear cutting machine could not cut the sheet, necessitating manual cutting. This was tedious and led to inaccuracies, causing repeated attempts. Additionally, bending the strips to 90 degrees required manual effort using wooden mallets, which weakened the stringers.
2. Faulty rivets during the riveting of wing ribs to stringers caused de-riveting issues, increasing the hole diameter. This necessitated larger rivets or additional support sheets, delaying the process. One rib was damaged beyond repair and required reprinting, halting the assembly.
3. Bulkheads, due to their circular cross-section, were 3D printed. Limited 3D printer capacity caused delays, as the bulkheads had to be printed in parts.
4. Tolerances in the attachment points for longerons to bulkheads were underestimated, requiring manual enlargement using hacksaw blades.
5. The spar-to-fuselage joint required enlarging the gap, but waterjet machining failed due to fixture issues. The team resorted to using a heated hacksaw blade.
6. Weight estimation did not account for internal support structures, leading to a significant weight increase above the predicted 8.8 kg. Additionally, the original 50mm aluminum tail boom was unavailable, and a heavier PVC pipe was used, later reduced to 40mm diameter.
7. To address earlier challenges, the fuselage skeleton was redesigned with lighter bulkheads and increased tolerances. However, printing delays persisted.
8. The Main Landing Gear's C-section design using a 1mm sheet was inadequate for torsional loads. It was redesigned as a rectangular box section for strength.
9. The Nose Landing Gear, designed with a spring action for impact absorption, faced structural issues. The PLA-made strut failed under fatigue loading, and soldering iron molding for roller bearing attachment was unsuccessful.
10. During the final assembly, the landing gear was attached before the fuselage skin due to time constraints. Since the fuselage skin was not wrapped before, the structure was unable to support itself and there was a bending noticed in the longerons.
11. Limited availability and poor quality of tools (e.g., drilling machines, drill bits, riveting guns) caused delays.

### 10.6.2 Lessons Learnt

1. Practical knowledge of machining processes and riveting is crucial. Workshops on these topics should be conducted before fabrication.
2. Upgrading workshop machines (e.g., for cutting/bending aluminum sheets) would enhance accuracy and efficiency.
3. Knowledge of 3D printing design considerations (e.g., tolerances) is essential. Training sessions would reduce waste and speed up processes.
4. Availability of adequate tools and consumables is critical. Tools should either be distributed to groups or shared efficiently.
5. Flexible workshop hours and dedicated spaces for teams would improve productivity and coordination.

## Chapter 11

# Appendix

### 11.1 Lift Distribution of Wing using Schrenks' Method & SFD and BMD Plots Calculation

---

```
import matplotlib.pyplot as plt
import numpy as np
import math

b = 2.82
chord = 0.32
W = 8.8*9.81
S = b*chord

y = np.linspace(0,b/2,200)

# Tau calculation
V_inf = 18
rho_inf = 1.225
C_L = 0.5
tau_0 = (2*V_inf*S*C_L)/(b*math.pi)
tau = tau_0*np.sqrt(1-np.power(2*y/b,2))
Ly = rho_inf*V_inf*tau

# Elliptic chord distr
c_elliptic = 4*S/(math.pi*b)*np.sqrt(1-np.power(2*y/b,2))

# Average chord (i need a shrink)
c_schrenk = (chord + c_elliptic)/2

plt.figure(figsize = (10,6))
plt.plot(y, np.ones(len(y))*chord, linestyle='--')
plt.plot(y, c_elliptic, linestyle='--')
plt.plot(y, c_schrenk)
# plt.title("Chord Distribution")
plt.legend(["Rectangular (Actual)", "Elliptic", "Schrenk"])
plt.xlabel("Spanwise location b/2 (m)", fontsize = 14)
plt.ylabel("Chord (m)", fontsize = 14)
plt.show()

integral = np.trapz(c_schrenk, y)

k = (W/2)/(integral)
# print(k)
L_eq = k*c_schrenk

plt.figure(figsize = (10,6))
plt.plot(y, Ly, linestyle='--')
```

```

plt.plot(y, L_eq)

for xi, yi in zip(y, L_eq):
    plt.plot([xi, xi], [yi, 0], color='blue', linestyle='--', linewidth=0.5)

plt.legend(["Elliptic", "Schrenk"])
plt.xlabel("Spanwise location b/2 (m)", fontsize = 14)
plt.ylabel("Lift per unit span L' (N/m)", fontsize = 14)
plt.title("Load Distribution")
plt.show()

# L_eq : Lift values
# y : Span values
R = 0
M = 0
dy = (b/2)/len(y)
# Finding Reaction force at root

# Euler integration for dy/dt = y
for i in range(len(y)):
    R = R-L_eq[i]*dy

print(R, "N")

SF = np.zeros(len(y))
# Looping over every span element
for i in range(len(y)):
    SF_element = 0
    for j in range(0,i):
        SF_element = SF_element + L_eq[j]*dy
    SF[i] = R + SF_element

plt.figure(figsize = (10,6))
plt.plot(y, SF)
plt.xlabel("Spanwise location b/2 (m)", fontsize = 14)
plt.ylabel("Shear Force (N)", fontsize = 14)
plt.fill_between(y, SF, color='skyblue', alpha=0.4)
plt.show()

M = 0
for i in range(len(y)):
    M = M + L_eq[i]*dy*y[i]

BM = np.zeros(len(y))
for i in range(len(y)):
    BM_element = 0
    for j in range(0,i):
        BM_element = BM_element + SF[j]*dy
    BM[i] = M + BM_element

print(M, "Nm")

plt.figure(figsize = (10,6))
plt.plot(y, BM)
plt.xlabel("Spanwise location b/2 (m)", fontsize = 14)
plt.ylabel("Bending Moment (N-m)", fontsize = 14)
plt.fill_between(y, BM, color='skyblue', alpha=0.4)
plt.show()

```

## 11.2 Spar Calculations

---

```
# I_skin
dat_file = r"C:\Users\anirb\OneDrive\Documents\goe553.dat"
# Centroid Calculation
data = np.loadtxt(dat_file)
x_airfoil = data[:, 0]
y_airfoil = data[:, 1]
# MOI
fig, ax = plt.subplots()
ax.plot(x_airfoil,y_airfoil)
ax.set_aspect('equal')
plt.show()

import numpy as np
import os

def read_airfoil_dat(filename):
    """Reads the airfoil coordinates from a .dat file, skipping the first line."""
    with open(filename, 'r') as file:
        # Skip the first line
        next(file)
        # Read the remaining lines and extract coordinates
        coordinates = np.loadtxt(file)
    return coordinates

def scale_airfoil(coordinates, chord_length):
    """
    Scales both x and y coordinates of the airfoil to match the desired chord length.
    Assumes the airfoil coordinates are in unit scale.
    """
    scale_factor = chord_length # Scale both x and y by the chord length
    scaled_coordinates = coordinates * scale_factor
    return scaled_coordinates

def calculate_centroid(coordinates):
    """
    Calculates the centroid (x_bar, y_bar) of the airfoil section.
    Assumes the airfoil coordinates define a closed polygon.
    """
    x = coordinates[:, 0]
    y = coordinates[:, 1]

    #from fusion360 model
    Cx = 166
    Cy = 11.2

    return Cx, Cy

def shift_coordinates_to_centroid(coordinates, centroid):
    """
    Shifts the airfoil coordinates so that the centroid is at the origin.
    """
    x_shifted = coordinates[:, 0] - centroid[0]
    y_shifted = coordinates[:, 1] - centroid[1]
    return np.column_stack((x_shifted, y_shifted))

def calculate_area_moment_of_inertia(coordinates, thickness):
```

```

Calculates the area moment of inertia (Ixx) of the airfoil section
using the given coordinates and fixed skin thickness, with respect to the centroid.

Assumes the airfoil coordinates define a closed polygon.
"""
x = coordinates[:, 0]
y = coordinates[:, 1]

# Initialize moment of inertia components
Ixx = 0
n = len(x)

for i in range(n - 1):
    # dx is the difference in x between consecutive points
    dx = x[i + 1] - x[i]
    # Midpoint of y for the strip between consecutive points
    y_mid = (y[i] + y[i + 1]) / 2
    # Width of the strip is the distance between consecutive x points
    width = np.abs(dx)
    # Contribution to moment of inertia from this strip
    dIxx = (y_mid ** 2) * thickness * width
    Ixx += dIxx

return Ixx*1e-12

# Example usage
filename = r'D:\IITM\4th Year\Sem 7\AS5100\GOE553.dat' # Use the raw string notation (r) for
# Windows paths
chord_length = 340 # Chord length in mm
thickness = 0.3 # Skin thickness in m

# Read the airfoil coordinates
coordinates = read_airfoil_dat(filename)

# Scale the airfoil to the desired chord length
scaled_coordinates = scale_airfoil(coordinates, chord_length)

# Calculate the centroid
centroid = calculate_centroid(scaled_coordinates)

# Shift the coordinates to the centroid
shifted_coordinates = shift_coordinates_to_centroid(scaled_coordinates, centroid)

# Calculate the area moment of inertia for the shifted airfoil with the fixed thickness
Ixx = calculate_area_moment_of_inertia(shifted_coordinates, thickness)

print(f"Centroid: ({centroid[0]:.2f}, {centroid[1]:.2f}) mm")
print(f"Area Moment of Inertia (Ixx): {Ixx} m^4")

I_skin = Ixx

I_skin = 0 # mm^4

sigma_yield = 290e6 #Pa, Al 2024
sigma_allowable = sigma_yield/10
t = 0.3e-3 #Thickness, m
airfoil_thickness_percent = 13.7

sw_web = airfoil_thickness_percent/100 * chord # m

I_spar = BM*sw_web/(2*sigma_allowable) * 10**12 - I_skin

plt.figure(figsize = (10,6))

```

```

plt.plot(y, I_spar)
plt.title("I_Required")
plt.xlabel("Spanwise location b/2 (m)", fontsize = 14)
plt.ylabel("I_Rquired (mm^4)", fontsize = 14)
plt.show()

sw_web = sw_web*10**3
t = t*10**3
# width_flange = I_spar*2/(t*10**3*(sw_web*10**3)**2)
width_flange = (I_spar - (t*sw_web**3)/12)/(2*t*(sw_web/2)**2)

plt.figure(figsize = (10,6))
plt.plot(y, width_flange, linestyle='--')
plt.plot([0,1.4], [66.2,20])
plt.title("width flange")
plt.xlabel("Spanwise location b/2 (m)", fontsize = 14)
plt.ylabel("Flange Width (mm)", fontsize = 14)
plt.legend(["Theoretical", "Manufacturable Estimate"])
plt.show()

```

---

### 11.3 Shear Flow Calculations

```

import numpy as np
import matplotlib.pyplot as plt
import math
from scipy.interpolate import interp1d
from sympy import symbols, Eq, solve

chord = 0.34 # m
max_thickness = 13.7/100*chord
max_pos = 29.6/100*chord
thickness = 0.3e-3 # m
web_area = 0.066*thickness
stringer_area = 0.0001
print("max position = ", max_pos)

# Loading Airfoil Points
dat_file = r"C:\Users\anirb\OneDrive\Documents\goe553_extend.dat"
data = np.loadtxt(dat_file) # TE index 0, goes to LE and comes back, 33 points
data = data*chord
x_airfoil = data[:, 0]
y_airfoil = data[:, 1]
# points = np.column_stack((x_airfoil, y_airfoil))

# Shave off TE for Structural Idealization
x_airfoil_SI = x_airfoil[45:-45]
y_airfoil_SI = y_airfoil[45:-45]

x_airfoil_SI = np.append(x_airfoil_SI, x_airfoil_SI[0])
y_airfoil_SI = np.append(y_airfoil_SI, y_airfoil_SI[0])
points = np.column_stack((x_airfoil_SI, y_airfoil_SI))
print("airfoil points = ", len(x_airfoil))
print("SI points = ", len(x_airfoil_SI))

# Mean Camber Line
y_camber = (y_airfoil[:int(len(x_airfoil)/2)] + y_airfoil[int(len(x_airfoil)/2)-1:-1][::-1])/2
y_camber = y_camber[30:int(len(x_airfoil)/2)]
# y_camber = (y_airfoil_SI[:13] + y_airfoil_SI[14:-1][::-1])/2

# Spar Location

```

```

diff_top = np.abs(x_airfoil_SI[:130] - max_pos)
sorted_indices = np.argsort(diff_top)
spar_indices = sorted_indices[:1]

diff_bottom = np.abs(x_airfoil_SI[130:-1] - max_pos)
sorted_indices = np.argsort(diff_bottom)
spar_indices = np.append(spar_indices, sorted_indices[1]+130)

x_spar = x_airfoil_SI[spar_indices]
y_spar = y_airfoil_SI[spar_indices]
spar_length = round(math.sqrt((x_spar[1] - x_spar[0]) ** 2 + (y_spar[1] - y_spar[0]) ** 2),3)
print("Spar Length = ", spar_length)
# print(x_spar)
# print(y_spar)

# Calculate Area of the 2 sections
A1 = np.sum(np.multiply((x_airfoil_SI[0:spar_indices[0]-1] - x_airfoil_SI[1:spar_indices[0]]),
    np.abs(y_airfoil_SI[0:spar_indices[0]-1]))) +
    np.sum(np.multiply((x_airfoil_SI[spar_indices[1]+1:-1] -
        x_airfoil_SI[spar_indices[1]:-2]), np.abs(y_airfoil_SI[spar_indices[1]:-2])))
A2 = np.sum(np.multiply((x_airfoil_SI[spar_indices[0]:int(len(x_airfoil_SI)-1/2)-1] -
    x_airfoil_SI[spar_indices[0]+1:int(len(x_airfoil_SI)-1/2)]),
    y_airfoil_SI[spar_indices[0]:int(len(x_airfoil_SI)-1/2)-1])) +
    np.sum(np.multiply((x_airfoil_SI[int(len(x_airfoil_SI)-1/2):spar_indices[1]-1] -
        x_airfoil_SI[int(len(x_airfoil_SI)-1/2)+1:spar_indices[1]]),
    y_airfoil_SI[int(len(x_airfoil_SI)-1/2):spar_indices[1]-1]))
print("A1 = ", A1, "m^2")
print("A2 = ", A2, "m^2")

centroid_x = np.mean(x_airfoil_SI)
centroid_y = np.mean(y_airfoil_SI)
print(centroid_x, centroid_y)

fig, ax = plt.subplots()
ax.plot(x_airfoil,y_airfoil)
# ax.plot(x_airfoil[30:160],y_airfoil[30:160])
# ax.plot(x_airfoil_SI[:50],y_airfoil_SI[:50])
ax.plot(x_airfoil_SI,y_airfoil_SI)
# ax.scatter(x_airfoil_SI,y_airfoil_SI)
# ax.scatter(x_airfoil_SI,y_airfoil_SI, s = 20)
ax.plot(x_spar, y_spar)
ax.scatter(centroid_x, centroid_y, s = 30)
# ax.plot(x_airfoil[30:int(len(x_airfoil)/2)], y_camber)
# ax.plot([0,0.34], [0,0.34*math.sin(2*math.pi/180)])
# ax.scatter(x_airfoil_SI[-2], y_airfoil_SI[-2])
ax.set_aspect('equal')
plt.title("Airfoil and Spar")
plt.show()

#Idealize
def segment_length(start_index, end_index) :
    segment = points[start_index:end_index + 1]
    distances = np.sqrt(np.sum(np.diff(segment, axis=0)**2, axis=1))
    length = np.sum(distances)
    return round(length, 3)

def length_from_camber(index):
    if index not in spar_indices:
        length = round(y_airfoil_SI[index]-y_camber[index%int(len(x_airfoil_SI)/2)-1]],3)
    elif index in spar_indices:
        if spar_indices[0] == index:

```

```

        length = round(spar_length/2,3)
    else:
        length = round(-spar_length/2,3)
    return round(length, 3)

def length_from_NA(index):
    aoa = 2 # deg
    A = math.sin(aoa*math.pi/180)
    B = -1
    C = 0.008
    x0 = x_airfoil_SI[index]
    y0 = y_airfoil_SI[index]
    numerator = A * x0 + B * y0 + C
    denominator = np.sqrt(A**2 + B**2)
    distance = numerator / denominator
    return -round(distance, 3)

# boom_indices = np.array([0, spar_indices[0], spar_indices[1], len(x_airfoil_SI)-2,
#     len(x_airfoil_SI)-1]) # TE Index is First Index, Last index is repeated
# boom_indices = np.array([0,spar_indices[0], spar_indices[0]+30,
#     spar_indices[1]-30,spar_indices[1] ,len(x_airfoil_SI)-2, len(x_airfoil_SI)-1])
boom_indices = np.array([0, 18, spar_indices[0], spar_indices[0]+30,
    spar_indices[1]-30,spar_indices[1] , len(x_airfoil_SI)-20, len(x_airfoil_SI)-2,
    len(x_airfoil_SI)-1])
boom_points = points[boom_indices] [-1] # For plotting
spar_boom_index = []
for index in range(len(boom_indices)-1):
    if boom_indices[index] in spar_indices:
        spar_boom_index.append(index)

segment_1 = segment_length(boom_indices[0], boom_indices[1])
segment_2 = segment_length(boom_indices[1], boom_indices[2])
segment_3 = segment_length(boom_indices[2], boom_indices[3])
segment_4 = segment_length(boom_indices[3], boom_indices[4])
# print(segment_1)
# print(segment_2)
# print(segment_3)
# print(segment_4)

# print(length_from_camber(boom_indices[0]))
# print(length_from_camber(boom_indices[1]))
# print(length_from_camber(boom_indices[2]))
# print(length_from_camber(boom_indices[3]))

def distance_from_origin(x,y):
    return np.sqrt(x**2+y**2)

def idealize(boom_indices, spar_indices):
    # spar_boom_area = 66e-3*0.3e-3
    # stringer_boom_area = 1
    boom_areas = np.zeros(len(boom_indices)-1)
    Ixx = 0
    s = np.zeros(len(boom_indices))
    # if boom_indice not spar_indice -> Stringer, 2 panel
    # if boom_indice = spar_indice -> Spar, 3 panel
    for index in range(0,len(boom_indices)-1):
        if boom_indices[index] in spar_indices:
            b1 = segment_length(boom_indices[index], boom_indices[index+1])
            b2 = segment_length(boom_indices[index-1], boom_indices[index])
            b3 = spar_length

            y_boom = length_from_NA(boom_indices[index])
            y1 = length_from_NA(boom_indices[index+1])

```

```

y2 = length_from_NA(boom_indices[index-1])

other_spar_index = spar_indices[1] if spar_indices[0] == boom_indices[index] else
    spar_indices[0]
y3 = length_from_NA(other_spar_index)
# print(index, b1, b2, b3, y_boom, y1, y2)
boom_area = thickness*b1/6*(2+y1/y_boom) + thickness*b2/6*(2+y2/y_boom) +
    thickness*b3/6*(2-y3/y_boom) + web_area
boom_areas[index] = boom_area
Ixx = Ixx + boom_area*y_boom**2
s[index] = b1

elif boom_indices[index] not in spar_indices:
    b1 = segment_length(boom_indices[index], boom_indices[index+1])
    b2 = segment_length(boom_indices[index-1], boom_indices[index])

    y_boom = length_from_NA(boom_indices[index])
    y1 = length_from_NA(boom_indices[index+1])
    y2 = length_from_NA(boom_indices[index-1])
    if index-1 == -1:
        # Only for TE
        b2 = segment_length(boom_indices[-2], boom_indices[-1])
        y2 = length_from_NA(boom_indices[-2])
    # print(index, b1, b2, y_boom, y1, y2)
    boom_area = thickness*b1/6*(2+y1/y_boom) + thickness*b2/6*(2+y2/y_boom) #+
        stringer_area
    boom_areas[index] = thickness*b1/6*(2+y1/y_boom) + thickness*b2/6*(2+y2/y_boom)
    Ixx = Ixx + boom_area*y_boom**2
    s[index] = b1

s[-1] = spar_length

return boom_areas, Ixx, s

boom_areas, Ixx, s = idealize(boom_indices, spar_indices)
# print(f"Segment Lengths = ", s)
print(boom_areas*10**4, "cm^2")
# print(boom_points)
print(f"Ixx = {Ixx*10**12} mm^4")

fig, ax = plt.subplots()
ax.plot(x_airfoil,y_airfoil)
ax.plot(x_airfoil_SI,y_airfoil_SI)
ax.plot(x_spar, y_spar)
# ax.plot(x_airfoil[30:int(len(x_airfoil)/2)], y_camber)
ax.scatter(boom_points[:,0], boom_points[:,1], s = 50, color = "black")
ax.plot([0,0.34], [0.008,0.34*math.sin(2*math.pi/180)+0.008], linestyle = "--", color =
    "black")
# ax.plot([0,0.34], [0.008,0.008])
ax.scatter(centroid_x, centroid_y, s = 50, color = "red")
plt.legend(["Control Surface (excluded)", "Skin Panels", "Spar", "Booms", "Neutral Axis",
    "Centroid"])
# ax.scatter(x_airfoil_SI[-2], y_airfoil_SI[-2])
ax.set_aspect('equal')
plt.ylim((-0.025,0.1))
plt.title("Idealized Airfoil")
plt.show()

# Shear Flow Calculation
# At root
G = 26*10**9 #Shear Modulus of Al, Pa
Sy = -43 # N

```

```

qb_skin = np.zeros(len(boom_indices))
sect_2_marker = []
qb_skin[0] = 0 # Cut on Section 1
qb_skin[spar_boom_index[0]] = 0 # Cut on Section 2
qb_prev = 0
print(f"Ixx = {Ixx} m^4, Sy = {Sy} N")
for index in range(len(boom_indices)-1):
    if index != 0:
        qb_prev = qb_skin[index-1]

    Br = boom_areas[index]
    yr = length_from_NA(boom_indices[index])
    print(f"Br = {Br}, yr = {yr}")
    # print(qb_prev, Br, yr)
    if index == spar_boom_index[0]:
        print("qb_prev = ", qb_prev, index)
        # print(f"Br = {Br}, yr = {yr}")
        qb_skin[-1] = qb_prev - Sy/Ixx*(Br*yr)
        sect_2_marker.append(index)
    if index == spar_boom_index[1]:
        qb_prev = qb_skin[-1]
        sect_2_marker.append(index)

    qb_skin[index] = qb_prev - Sy/Ixx*(Br*yr)

    # Cut qb BC
    qb_skin[0] = 0 # Cut on Section 1
    qb_skin[spar_boom_index[0]] = 0 # Cut on Section 2

    print("Segment Lengths = ", s, "m")
    print("Basic Shear = ", np.round(qb_skin,1), "N/m")
    print("Basic Shear Force = ", np.round(np.multiply(qb_skin,s)), "N")
    # print(spar_boom_index)

    s_1 = np.concatenate((s[:spar_boom_index[0]], s[spar_boom_index[1]:]))
    s_2 = np.concatenate((s[spar_boom_index[0]:spar_boom_index[1]], [s[-1]]))
    total_s_1 = np.sum(s_1)
    total_s_2 = np.sum(s_2)

    qb_s = np.multiply(qb_skin,s)
    qb_s_1 = np.concatenate((qb_s[:spar_boom_index[0]], qb_s[spar_boom_index[1]:]))
    qb_s_2 = np.concatenate((qb_s[spar_boom_index[0]:spar_boom_index[1]], [qb_s[-1]]))

    total_qb_s_1 = np.sum(qb_s_1)
    total_qb_s_2 = np.sum(qb_s_2)
    # print(qb_s_1)
    # print(qb_s_2)

    # print(total_s_1)
    # print(total_s_2)
    # print(total_qb_s_1)
    # print(total_qb_s_2)

    q0_1, q0_2 = symbols('q0_1 q0_2')
    lhs_dtdz = 1/(2*A1*G*thickness)*(total_qb_s_1 + total_s_1*q0_1 - spar_length*q0_2)
    rhs_dtdz = 1/(2*A2*G*thickness)*(total_qb_s_2 + total_s_2*q0_2 - spar_length*q0_1)

    p0 = np.sqrt(x_airfoil_SI**2+y_airfoil_SI**2)
    p0_panel_points_split = np.split(p0, boom_indices[1:-2])
    p0_panel_averages = [np.mean(arr) for arr in p0_panel_points_split]
    p0_panel_averages.append(x_airfoil_SI[-2])
    p0_panel_averages.append(x_airfoil_SI[spar_indices[0]])
    # print(len(p0_panel_points_split))

```

```

# print(p0_panel_averages)

moment_contrib = q0_1 * A1 + q0_2 * A2 - np.sum(qb_skin * p0_panel_averages * s)

moment_eq = Eq(moment_contrib, 0)
dtdz_eq = Eq(lhs_dtdz, rhs_dtdz)
solution = solve((moment_eq, dtdz_eq), (q0_1, q0_2))
print("Solution:", solution)
# p0_panel_averages = [np.mean(arr) for arr in p0_panel_points_split]

q_total = np.zeros(len(qb_skin))

for i in range(len(q_total)-1):
    if i < spar_boom_index[0] or i >= spar_boom_index[1]:
        q_total[i] = qb_skin[i] + solution[q0_1]
    elif i >= spar_boom_index[0] and i < spar_boom_index[1]:
        q_total[i] = qb_skin[i] + solution[q0_2]
q_total[-1] = qb_skin[-1] + solution[q0_1] - solution[q0_2]

print(f"q_total = {np.round(q_total,1)} N/m")
print(f"q_force = {np.round(np.multiply(q_total, s),1)} N")

# print(f"q_force/area = {np.multiply(q_total, 1/thickness)} Pa")

# centroid_x = np.mean(x_airfoil_SI)
# centroid_y = np.mean(y_airfoil_SI)
# print(centroid_x, centroid_y)

# Fcr calculation

K_ss = lambda a, b: 5.34 + 4 / ((a / b) ** 2) # K_ss equation
R_a = 1 # Given Ra
R_b = 1 # Given Rb
nu = 0.33 # Poisson's ratio
E = 70e9 # Elastic modulus in Pascals (converted from MPa)
t = 0.3e-3 # Thickness in meters (converted from mm)
# span = 1.4

# Define dimensions (modifiable)
a = 1.4/15 # Length in meters
# b = 0.4 # Width in meters

# Critical buckling load equation
F_cr = []
for b in s:
    F_cr.append((K_ss(a, b) * (np.pi ** 2 * E) / (12 * (1 - nu ** 2)) * (t / b) ** 2 * (R_a +
        ((R_b - R_a) / 2) * (b / a) ** 3))*b*thickness)
print(f"F_cr = {np.round(F_cr,1)} N")

```

## 11.4 Bulkhead Calculations

---

```

import numpy as np
from scipy.optimize import fsolve
import matplotlib.pyplot as plt
from sympy import symbols, Eq, solve

# Constants
E = 3.5e9 # Young's modulus
G = 4e9 # Shear modulus
L0 = 43 # lift by wing
M0 = -27 # moment by wing

```

```

# Given values
d_outer = 0.230 # Outer diameter in meters
t = 15e-3 # Thickness of bulkhead in meters (comes from longeron, subject to change)

# Calculate inner diameter
d_inner = d_outer - t

r = (d_inner+d_outer)/2 # radius

# Calculate moment of inertia I
I = (np.pi / 64) * (d_outer**4 - d_inner**4)

print(f"Moment of inertia I of bulkhead: {I:.6e} m^4")

A = 0.25*np.pi*(d_outer**2 - d_inner**2) # Cross-sectional area

# Given q values as numpy array
q = np.array([10.5217594151503, 31.3061724477080, 51.3196990310784, 70.0696369811366,
87.0942138922379, 101.974193460033, 114.343226239021, 123.896791717115,
130.399609213066, 133.691548220268, 130.399609213066, 123.896791717115,
114.343226239021, 101.974193460033, 87.0942138922379, 70.0696369811366,
51.3196990310784, 31.3061724477080, 10.5217594151502])

# Create theta array with the same number of points, going from 0 to pi
#theta = np.linspace(0, np.pi, len(q))

thetas = np.array( [ 0. , 9.00003038, 18.00009851, 26.99998459, 35.99986409,
45., 54.00013591, 63.00001541, 71.99990149, 80.99996962,
99.00003038, 108.00009851, 116.99998459, 125.99986409, 135.,
144.00013591, 153.00001541, 161.99990149, 170.99996962])
theta = np.deg2rad(thetas)

# Known variables
F_qy = q * r * (1 - np.cos(theta))
F_qx = q * r * np.sin(theta)
M_q = 2 * q * A * theta * (1/(2*np.pi))

# Define equilibrium equations as functions
def equilibrium(vars):
    F_A, M_A = vars

    # Initialize arrays for H, V, M
    H = np.zeros_like(theta)
    V = np.zeros_like(theta)
    M = np.zeros_like(theta)

    # Solve equilibrium equations pointwise
    #H = (F_A - F_qx - V * np.sin(theta)) / np.cos(theta)
    #V = (F_qy - H * np.sin(theta) - L) / np.cos(theta)

    # Simplified expressions for H and V
    # Calculate V using the final expression
    V = (F_qy - L0) * np.cos(theta) + (F_A - F_qx) * np.sin(theta)

    # Calculate H using the expression for H in terms of V
    H = (F_A - F_qx)*np.cos(theta) + (F_qy + L0)* np.sin(theta)

    # Moment equation
    M = M_A + F_A * r * (1 - np.cos(theta)) + M_q + MO

    # Calculate total strain energy

```

```

U_total = (1 / 2) * np.trapz((M**2) / (E * I), theta) \
+ (1 / 2) * np.trapz((H**2) / (E * A), theta) \
+ (1 / 2) * np.trapz((V**2) / (G * A), theta)

# Derivatives of U_total with respect to F_A and M_A (Castigliano's Theorem)
dU_dFA = np.trapz((r*np.cos(theta))*(M * (r * (1 - np.cos(theta)))) / (E * I), theta) \
+ np.trapz((r*np.cos(theta))*(H * np.cos(theta)) / (E * A), theta) \
+ np.trapz((r*np.cos(theta))*(V * np.sin(theta)) / (G * A), theta)

dU_dMA = np.trapz((r*np.cos(theta))*M / (E * I), theta)

return [dU_dFA, dU_dMA]

# Initial guess for F_A and M_A
initial_guess = [1e5, 1e5]

# Solve the system of equations
F_A_sol, M_A_sol = fsolve(equilibrium, initial_guess)

# Output the solutions
print("\nBulkhead at wing loc")
print(f"Solution for F_A: {F_A_sol} N")
print(f"Solution for M_A: {M_A_sol} N/m")

# Compute moment based on solved values
M = M_A_sol - F_A_sol * r * (1 - np.cos(theta)) + M_q

# Plot the moment M as a function of theta
plt.figure(figsize=(8, 6))
plt.plot(np.rad2deg(theta), M)
plt.xlabel(' (degrees)')
plt.ylabel('Moment (Nm)')
plt.title('Moment M vs ')
plt.legend()
plt.show()

#compute shear forces based on solved values
V = (F_qy - L0) * np.cos(theta) + (F_A_sol - F_qx) * np.sin(theta)

H = (F_A_sol - F_qx)*np.cos(theta) + (F_qy + L0)* np.sin(theta)

plt.figure(figsize=(8, 6))
plt.plot(np.rad2deg(theta), H, label="H (Horizontal Force)")
plt.plot(np.rad2deg(theta), V, label="V (Vertical Force)")
plt.xlabel('Theta (degrees)')
plt.ylabel('Force (N)')
plt.title('Horizontal and Vertical Forces vs Theta')
plt.legend()
plt.show()

# Compute total strain energy
U_total = (1 / 2) * np.trapz((M**2) / (E * I), theta) \
+ (1 / 2) * np.trapz((H**2) / (E * A), theta) \
+ (1 / 2) * np.trapz((V**2) / (G * A), theta)

print(f"Total strain energy U_total: {U_total}")

# Given values
FOS = 10 # Factor of Safety
sigma = 50e6 #yield stress
M_max = max(abs(M)) # Maximum bending moment in N/m

```

```

print(f"Maximum bending moment: {M_max} Nm")

# Calculate yield stress () based on FOS
sigma_yield = sigma / FOS

# Calculate maximum distance from the neutral axis (y_max)
y_max = t/2

# Calculate moment of inertia (I)
I_req = abs((M_max * y_max) / sigma_yield)

print(f"Required moment of inertia I: {I_req:.6e} m^4")

ts = 10e-3
#web_length = np.sqrt(2*I_req/(t*ts))
#print(f"Required web length: {web_length} m")

# Redefine the equation to solve for w
def equation(w):
    return 2 * (w/2)**2 * t * ts + (ts * w**3) / 12 - I_req

# Use fsolve to find the root
w_initial_guess = 1.0 # Provide an initial guess for w
w_solution = fsolve(equation, w_initial_guess)

print(f"Required web length: {w_solution} m")

```

---

## 11.5 SFD and BMD of Fuselage

```

clear all;
close all;

% Define beam length
L = 1.5; % Beam length (m)

% Define point loads and their positions
loads = [8.98, 5.46, 22.27, 4.21, 4.56, 2.23, 1.96, 0.77, 1.55, 29.43]; % Loads in N
positions = [0.053, 0.053, 0.186, 0.276, 0.276, 0.476, 0.481, 0.150, 0.350, 0.326]; %
    Positions of loads in m
FL = 81.42; % Lift force (N)
xL = 0.326; % Position of lift force (m)

% Discretize the beam into small sections
num_points = 1000;
x = linspace(0, L, num_points); % X positions along the beam

% Initialize arrays for shear force (V) and bending moment (M)
V = zeros(1, num_points);
M = zeros(1, num_points); % Bending moment array

% Compute shear force at each point
for i = 1:num_points
    xi = x(i); % Current position along the beam

    % Calculate shear force by adding all the loads applied before the current point
    V(i) = 0; % Start with zero shear force
    for j = 1:length(loads)
        if positions(j) <= xi
            V(i) = V(i) - loads(j); % Subtract the loads
        end
    end

```

```

    end

    % Add the lift force at its location
    if xi >= xL
        V(i) = V(i) + FL;
    end
end

% Now, calculate the bending moment by integrating the shear force
for i = 2:num_points
    dx = x(i) - x(i-1); % Small section length
    M(i) = M(i-1) + V(i-1) * dx; % Integrate shear force to get moment

    % If we reach the last loading point x=0.481 (y=-6.81), linearly interpolate to zero at
    % x=0.486
    if x(i) >= 0.481 && x(i) <= 0.486
        M481 = -6.81; % Bending moment value at x = 0.481
        M(i) = M481 + (0 - M481) * (x(i) - 0.481) / (0.486 - 0.481); % Linear interpolation
    end

    % After x = 0.486m, set the BMD to remain constant (i.e., zero)
    if x(i) > 0.486
        M(i) = 0; % Set the BMD to zero after x = 0.486m
    end
end

% Find maximum shear force and bending moment values
[max_shear_force, idx_shear] = max(abs(V)); % Maximum absolute shear force and its index
[max_bending_moment, idx_moment] = max(abs(M)); % Maximum absolute bending moment and its index

% Display the maximum shear force and bending moment values
fprintf('Maximum Shear Force: %.2f N at x = %.2f m\n', V(idx_shear), x(idx_shear));
fprintf('Maximum Bending Moment: %.2f Nm at x = %.2f m\n', M(idx_moment), x(idx_moment));

% Plot Shear Force Diagram (SFD)
figure;
plot(x, V, 'b', 'LineWidth', 2);
title('Shear Force Diagram (SFD)');
xlabel('Position along beam (m)');
ylabel('Shear Force (N)');
grid on;
xlim([0, 1.5]); % Set X-axis limits
ylim([-60, 20]); % Set Y-axis limits
% Increase grid lines
xticks(linspace(0, 1.5, 16)); % 15 divisions along the x-axis
yticks(-60:10:20); % Adjust as needed for y-axis

% Plot Bending Moment Diagram (BMD)
figure;
plot(x, M, 'r', 'LineWidth', 2);
title('Bending Moment Diagram (BMD)');
xlabel('Position along beam (m)');
ylabel('Bending Moment (Nm)');
grid on;
xlim([0, 1.5]); % Set X-axis limits
ylim([-10, 5]); % Set Y-axis limits
% Increase grid lines
xticks(linspace(0, 1.5, 16)); % 15 divisions along the x-axis
yticks(-10:1:5); % Adjust as needed for y-axis

```

## 11.6 BMD of Landing Gear

---

```
clear all;
close all;

% Parameters
F = 0.85 * 8.8 * 9.81 / 2; % Force calculation
M0 = F * (31e-3) / 2;      % Moment at x = 0
theta = 45;                % Angle in degrees
theta_rad = deg2rad(theta); % Convert to radians for calculations

% Section 1: x from 0 to 0.24355
x1 = linspace(0, 0.24355, 100);
M1 = M0 + x1 * F * sin(theta_rad); % Moment calculation for Section 1

% Calculate the moment at x = 0.24355 (end of Section 1)
M_at_24355 = M1(end); % Moment carried over to Section 2

% Section 2: x from 0.24355 to 0.375
x2 = linspace(0.24355, 0.375, 100);
M2 = M_at_24355 + F * (x2 - 0.24355); % Moment calculation for Section 2

% Calculate the moment at x = 0.375 (end of Section 2)
M_at_375 = M2(end); % Moment passed to Section 3

% Section 3: x from 0.375 to 0.50645
x3 = linspace(0.375, 0.50645, 100);
M3 = M_at_375 - F * (x3 - 0.375); % Moment calculation for Section 3

% Section 4: x from 0.50645 to 0.75
x4 = linspace(0.50645, 0.75, 100);
M4 = M0 + (0.75 - x4) * F * sin(theta_rad); % Moment calculation for Section 4

% Combine sections for plotting
x_total = [x1, x2, x3, x4]; % Combine x values
M_total = [M1, M2, M3, M4]; % Combine moment values

% Plotting BMD
figure;
plot(x_total, M_total, 'b', 'LineWidth', 2);
xlabel('x (m)');
ylabel('Bending Moment (Nm)');
title('Bending Moment Diagram');
xlim([0, 0.75]); % Set limits for the entire beam length
ylim([0, 14]);
grid on;

% Calculate and display M(x) at x = 0.375 m
disp(['The value of M(x) at x = 0.375 m is: ', num2str(M_at_375), ' Nm']);
```

---

# Bibliography

- [1] TH Megson - Aircraft Structures
- [2] Aircraft design: A Systems Engineering Approach - Mohammad H. Sadraey
- [3] Workswell Wiris Enterprise - Integrated Thermal and Optical Sensor
- [4] Prana PM2.5 - Air Quality Monitor
- [5] Aircraft Design: A Conceptual Approach - Daniel P Raymer
- [6] T-MOTOR AT7215
- [7] TF16\*8 Propellers