Preliminary Stability & Dynamics Study

Team 7:
Cooper LeComp, John A. Papas Dennerline,
Ian Greene, Christos Levy, Daniel Qi, Hans Suganda

26th September 2022

Contents

0.1 Preliminary Theoretical Basis

Based on inviscid analysis of an airfoil, there is a point in the airfoil called the aerodynamic center wherein the moments experienced about that point is insensitive to angle of attack. The concept of aerodynamic center can be extended to wings on an aircraft. The moment about the aerodynamic center of an aircraft wing is often non-zero, but it is unchanging regardless of the aircraft's orentiation.

0.1.1 Single Wing Thought Experiment

Let us consider a simple system where we have a single finite wing. Let us have some reference point P located about the zero-α line, but downwind of the wing.

The moment experienced about point P would be the moments about the aerodynamic center plus the influence of the lift force by the finite wing with the moment arm length evaluated at the aerodynamic center of the finite wing.

Imagine that this point P is located far away from the aerodynamic center of the finite wing. If the angle of attack is changed by say $5\,deg$, then this would result in a very large change of moments about point P because the moment arm of the changing lift force is large. As point P is moved closer and closer to the aerodynamic center of the finite wing, the change of $5\,deg$ in angle of attack affects the moments about point P less and less because the moment arm is increasingly small.

Eventually if point P is on the aerodynamic center of the finite wing, then the moment about P does not change with the angle of attack the wing is at.

0.1.2 Double Wing Thought Experiment

Let us now consider a slightly more complex system where we have 2 finite wings. Wing 1 is going to be upstream of wing 2 and wing 2 is going to be downstream of wing 1. Let us now have point P be located in between wing 1 and wing 2.

Now the moments about point P depends on the summation of the moments at the aerodynamic center of both wing 1 and 2 (unchanging with angle of attack) plus the influence of changing lift due to wing 1 and wing 2.

Suppose the angle of attack for the system was increased by $5\,deg$. Both wings 1 and 2 will provide additional lift and drag, but let us ignore the drag for now. The increase in lift of wing 1 would provide a moment that tends to pitch the system ”up”, in the clockwise direction. The increase in lift of wing 2 would provide a moment that tends to pitch the system ”down”, in the counter-clockwise direction.

If point P is moved backwards closer to wing 2, then the moments influence of wing 2 would be smaller when the angle of attack has been increased and the moments influence of wing 1 would be greater when the angle of attack is increased. This is because the moment arm length is smaller to wing 2 and greater for wing 1. Conversely, if we move point P front, closer to wing 1, then the moment influence of wing 1 is reduced in the event of a change of angle of attack due to smaller moment arm, and the influence of wing 2 is increased in the event of a change of angle of attack due to a larger moment arm.

Eventually, if the lift force increases linearly with angle of attack (valid assumption) there must be a unique point in between the 2 wings wherein the moments about that point is unchancing with angle of attack, because the influence of clockwise moments due to wing 1 is exactly cancelled by the influence of counter-clockwise moments due to wing 2. This is what is often known as the neutral point of the system.

0.1.3 Longitudinal Static Stability

It is often convenient to consider moments about the center of mass.

If the center of mass is behind the neutral point, the ”influence” of wing 2 would be reduced due to a shortening of the moment arm, and the ”influence” of wing 1 would be increased due to a lengthening of the moment arm. This means that if the 2-wing system (aircraft) pitches up slightly, wing 1 (forward wing) is going to have a winning influence compared to wing 2 (back wing) and the aircraft will have a tendency to keep pitching up. This means that an aircraft whose center of mass is behind the neutral point is statically unstable; a small pitch in any direction will give a larger tendency to deviate from level steady flight.

If the center of mass is in front of the neutral point, the ”influence” of wing 1 would now be reduced due to the shortning of the moment arm, and the ”influence” of wing 1 would be increased due to lengthening of the moment arm. This is obvoius because now the center of mass (the point we are evaluating moments at) is closer to wing 1 and further way from wing 2. This means now if the 2-wing aircraft system pitches up slightly, wing 2 (backward wing) is going to have a winning influence compared to wing 1 (front wing) and the aircraft would pitch back down. This means that for an aircraft to be statically stable, the center of mass has to be in front or upwind of the neutral point.

0.1.4 Trim Conditions

Arguably, the most important trim condition for an aircraft would be the longitudinal trim. We can model the lift produced by the finite wings to be a linear function of angle of attack. We can also model the drag produced by the finite wings as a quadratic function of lift. This means we can also model the drag produced by the finite wings as a quadratic function of angle of attack.

The lift to drag ratio is an incredibly important parameter in aerodynamic efficiency. Since the lift is equivalent to weight for level and steady flight, then the drag can be wholly determined from the lift to drag ratio for the entire aircraft. Remembering that lift is linear with angle of attack and drag is quadratic to angle of attack, a general expression of lift to drag ratio as a function of angle of attack,

$$\frac{L}{D} = \frac{k_1 + k_2\alpha}{k_3 + k_4\alpha + k_5\alpha^2}$$

We can find the optimum lift to drag ratio by differentiation,

$$\frac{d}{d\alpha}\left[\frac{L}{D}\right] = \frac{d}{d\alpha}\left[\frac{k_1 + k_2\alpha}{k_3 + k_4\alpha + k_5\alpha^2}\right]$$

$$\frac{d}{d\alpha}\left[\frac{L}{D}\right] = \frac{k_2(k_3 + k_4\alpha + k_5\alpha^2) - (k_1 + k_2\alpha)(k_4 + 2k_5\alpha)}{(k_3 + k_4\alpha + k_5\alpha^2)^2}$$

The expression above can be simplified algebraically, but if we set the derivative above to 0, we basically set the numerator of the expression above to 0. Since the numerator is an quadratic function of α, we have 2 maxima/minima points for the L/D. One is going to correspond to positive lift and the other to negative lift, so we can be assured that a single optimum angle of attack α would give us the best positive L/D. We will twist our entire wing to make sure that the aircraft flies at its optimum angle of attack at its ideal trim conditions.

0.2 Algorithm Development

0.2.1 Planning

1. From the Aerodynamic Preliminary Design, we should have an airfoil cross-section that suits our needs. We can design our wing from this airfoil-cross section

2. After we design our main wing, we need to determine its optimum angle of attack. We can run `avl` at a range of angles of attack and figure out the maximum Lift to Drag ratio achievable.

3. After we have our optimum angle of attack, we need to determine the moments of coefficient at our optimum angle of attack.

4. We then need an estimation of our center of mass. We will have to choose a static margin based on historical data. This static margin will control how "manuverable" our aircraft is going to be.

5. This permits a range of possibilities of tail boom and tail size. We will attempt to minimize: cost = weight of tail boom + weight of tail + downwash produced by tail at cruise.

6. We can make changes to the size of the rudder and adjust the dihedral angle to get rid of roll subsidence and dutch roll.

Longitudinal Modes:

1. Phugoid: Not important

2. Short Period Oscillations: Not that important

Lateral-Direction Modes:

1. Roll subsidence: This is affected by dihedral. If we have problems, just increase the dihedral.

2. Dutch Roll: Yaw Damper is supposed to get rid of this, increase size of tail if necessary

3. Spiral Divergence: Deadly and IDK how to offset

0.2.2 Trim_Wing

This is a `Bash` script which calls on all of the other functionalities to perform an aerodynamic "trim" of the wing.

```bash
#!/bin/bash

#Generate the Wing Configuration File
./Gen_Wing_AVL.py Wing_Parameters.tex Naca_6412.dat > Wing_Trim/Simple_Taper.avl

#Generate avl Instructions
./Gen_Instr Wing_Trim/ 5 0.3 24

#Run avl
cd Wing_Trim/
avl Simple_Taper.avl < Stab_Instr_AVL.txt

#Move the data out
mv Outhouse.txt ..

#Get out of there
cd ..

#Cleanup the data for Python
./Clean_Data Outhouse.txt

#Figure out final trim angle
./Comp_Trim_Angle.py Trim_Main_Wing.tex
```

0.2.3 Gen_Wing_AVL.py

The solver of choice is avl due to its scriptability. We need to produce a text file that contains a textual description of the ”aircraft” we want to model. Because we are ”trimming” the main wing of the aircraft, we this Python script only defines the bare minimum for a wing. Note the wing type which is a tapered wing desribed in this script is essentially chosen from the Aerodynamics Preliminary Design and just because of ease of manufacturing. Note that this takes an input file from the previous aerodynamic study.

```
#!/bin/python

#Library import statements
import numpy as np
import sys

#This is the name of the specified input file
Input_File = str(sys.argv[1])
#This is what choice of airfoil to use
Airfoil_Selection = str(sys.argv[2])

#Reading in the data
data = np.genfromtxt(Input_File)

#Writing for incompressible FLows
print("Simple_Tapered_Wing")
print("#Mach\n", 0.0)
print("#IYsym␣␣␣IZsym␣␣␣Zsym\n", 0, 0, 0)
print("#Sref␣␣␣␣Cref␣␣␣␣Bref")

#WRiting Reference area, chord reference and span reference
print(data[0], end="␣") #Writing of Reference Area
print(data[2], end="␣") #Writing of Reference Chord
print(data[4]) #Writing of Reference Wing Span

#TODO:AUTOMATE THIS PART
#Writing Center of Mass (Unimportant)
print("#Xref␣␣␣␣Yref␣␣␣␣Zref\n", 0.232*data[2],0,0)

#TODO:AUTOMATE THIS PART
#Writing Drag obtained from XFoil
print("#CDp\n", 0.05597)

#Surface Header
print("#################################################################")
print("SURFACE")
print("WING")
print("#Nchordwise␣␣␣␣␣Cspace␣␣Nspanwise␣␣␣Sspace")
print(30, 1.0, 30, 1.0)
print("YDUPLICATE\n", 0.0)
print("ANGLE\n", 0.0)
print("#␣x,y,z␣bias␣for␣whole␣surface")
print("TRANSLATE\n", 0.0, 0.0, 0.0)

#Definition of wing at the roots
print("#################################################################")
print("SECTION")
print("#␣␣␣Xle␣␣␣␣␣Yle␣␣␣␣␣Zle␣␣␣␣␣chord␣␣␣␣␣␣␣angle␣␣␣␣␣␣␣Nspan␣␣␣␣␣␣␣Sspace")
print(0, 0, 0, data[2], 0, 30, 1.0)
print("AFIL\n", Airfoil_Selection)

#Definition of wing at the tips
print("#################################################################")
print("SECTION")
print("#␣␣␣Xle␣␣␣␣␣Yle␣␣␣␣␣Zle␣␣␣␣␣chord␣␣␣␣␣␣␣angle␣␣␣␣␣␣␣Nspan␣␣␣␣␣␣␣Sspace")
print((data[2]-data[3])/2, data[4]/2, 0, data[3], 0, 30, 1.0)
print("AFIL\n", Airfoil_Selection)
```

To use this `Python` script, invoke the command,

```
python Gen_Wing_AVL.py Wing_Parameters.tex Naca_6412.dat
```

The contents of the file `Wing_Parameters.tex` is shown below,

```
#Wing Area (m^2)|Taper Ratio|Wing Root Chord (m)|Tip Root Chord (m)|Wing Span (m)|Aspect Ratio
0.903224 1.0 0.3556 0.3556 2.54 7.142857142857142
```

A sample output of this is shown below,

```
Simple_Tapered_Wing
#Mach
 0.0
#IYsym   IZsym   Zsym
 0 0 0
#Sref    Cref    Bref
0.903224 0.3556 2.54
#Xref    Yref    Zref
 0.08249920000000001 0 0
#CDp
 0.05597
#################################################################
SURFACE
WING
#Nchordwise     Cspace  Nspanwise   Sspace
30 1.0 30 1.0
YDUPLICATE
 0.0
ANGLE
 0.0
# x,y,z bias for whole surface
TRANSLATE
 0.0 0.0 0.0
#################################################################
SECTION
#   Xle     Yle     Zle     chord       angle       Nspan       Sspace
0 0 0 0.3556 0 30 1.0
AFIL
 Naca_6412.dat
#################################################################
SECTION
#   Xle     Yle     Zle     chord       angle       Nspan       Sspace
0.0 1.27 0 0.3556 0 30 1.0
AFIL
 Naca_6412.dat
```

0.2.4 Gen_Instr

This `Bash` script generates the instruction for `avl` to make and run all of the test cases. It's outputs are rather complicated especially if you want a large number of increments,

```
#!/bin/bash

#Arg1: Name of address to write Instructions to
#Arg2: Starting Angle of Attack
#Arg3: Increments of AOA's
#Arg4: Number of AOA increments

#Basic Names of the instruction file
Base_Instr_Name="Stab_Instr_AVL.txt"
```

```
#Name of AVL Output File
Out_File="Outhouse.txt"

#Appending the address of where instruction files should be written to
Full_Instr_Name=$1$Base_Instr_Name

#Enter Operational mode
echo "OPER" > $Full_Instr_Name

#Iterate over the different increments
for ((index = 1; index <= $4; index=index+1)) ;do
    #Set the Angle of Attack
    AOA=$(echo $2"+"$index"*"$3 | bc)
    echo "A␣A␣" $AOA >> $Full_Instr_Name
    #Execute the Case
    echo "X" >> $Full_Instr_Name
    #Write into File
    echo "W" >> $Full_Instr_Name
    if [ $index -eq 1 ]; then
        #For the first run case, enter the output filename
        echo $Out_File >> $Full_Instr_Name
    else
        #For subsequent cases, just leave it blank, avl should know
        echo "␣" >> $Full_Instr_Name
    fi
    #Create New Case
    echo + >> $Full_Instr_Name
done

#Exit Operation Mode
echo "␣" >> $Full_Instr_Name

#Quit AVL
echo "QUIT" >> $Full_Instr_Name
```

A small part of the output is shown below,

```
OPER
A A  5.3
X
W
Outhouse.txt
+
A A  5.6
X
W

+
...
```

0.2.5 Clean_Data

So the text file named `Outhouse.txt` (appropriately named) produced by `avl` is in a messy format. The `Bash` script here basically extracts only the important useful things and compiles them into a file which can be conveniently read in `Python`.

```
#!/bin/bash

#Arg1: Name of Data File

#Clean Up the alpha file
Alpha_File="ALPHA_DELETE_THIS.txt"
```

```
grep 'Alpha' $1 > $Alpha_File
sed -i 's/Alpha␣=//' $Alpha_File
sed -i 's/p/\n/' $Alpha_File
sed -i '/^b/d' $Alpha_File

#Clean up the total Lift coefficeint File
CL_tot_File="CL_DELETE_THIS.txt"
grep 'CLtot' $1 > $CL_tot_File
sed -i 's/CLtot␣=//' $CL_tot_File

#Clean up the Total Drag Coefficient File
CD_tot_File="CD_DELETE_THIS.txt"
grep 'CDtot' $1 > $CD_tot_File
sed -i 's/CDtot␣=//' $CD_tot_File

#Re-arrange into columns
paste $Alpha_File $CL_tot_File $CD_tot_File | column -s $'\t' -t > Trim_Main_Wing.tex

#Clean-up intermediate files
rm $Alpha_File $CL_tot_File $CD_tot_File

#Add the file headers
sed -i '1i\#Angle␣of␣Attack␣(deg)␣|␣Lift␣Coefficient␣|␣Drag␣Coefficient' Trim_Main_Wing.tex
```

0.2.6 Comp_Trim_Angle.py

This Python file takes in the already cleaned data from Outhouse.txt and figures out the maximum lift to drag ratio as well as the correspoonding angle of attack.

```
#!/bin/python

#Library Import Statements
import numpy as np
import sys

#Name of Data File
Dat_File = str(sys.argv[1])

#Read in Data File
data = np.genfromtxt(Dat_File)

#Comutation of Lift to Drag Ratio
LDR = data[:,1]/data[:,2]

#Location of Maximum Lift to Drag Ratio
LDR_M_Loc = np.argmax(LDR)

#Maximum Lift to Drag Ratio
LDR_Max = LDR[LDR_M_Loc]

#Angle of Attack which Max L/D Occurs
alpha_LDR_Max = data[LDR_M_Loc, 0]

print("#Alpha␣L/D␣Max␣(deg)␣|␣L/D␣max␣Value")
print(alpha_LDR_Max, LDR_Max)
```

0.2.7 Plot_Wing_Performance.py

This Python script is only used for data visualization and to plot a predicted lift curve slope compared to the avl results. This is used to generate a ”better” view of the whole story.

```
#!/bin/python

#Library Import Statements
import numpy as np
import matplotlib.pyplot as plt
import sys

#Name of Data File
Dat_File = str(sys.argv[1])

#Read in Data File
data = np.genfromtxt(Dat_File)
A_r = 7.142857142857142
zero_loc = np.argmin(data[:,0]**2)
#This is the alpha values for the predicted line
alpha_pred = [data[zero_loc,0], data[-1,0]]
#This is the approximate derivative of the predicted lift curve
cl_grad_pred = 2*np.pi*(A_r/(A_r+2))
#This is predicted lift coeff at lower value of alpha, make it match
cl_lower = data[zero_loc, 1]
#This is higher predicted lift coeff at higher value of alpha
cl_upper = cl_lower + cl_grad_pred*np.deg2rad(alpha_pred[1]-alpha_pred[0])
#Forming the cl predicted array
cl_pred = [cl_lower, cl_upper]

#Plot Lift Coefficient Vs Alpha
figure_CL, axis_CL = plt.subplots()
axis_CL.plot(data[:,0], data[:,1], ’kx-’,label=’AVL␣Wing’)
axis_CL.plot(alpha_pred, cl_pred, ’r--’,label=’Predicted’)
axis_CL.set_xlabel(’Angle␣of␣Atack’)
axis_CL.set_ylabel(’Lift␣Coefficient’)
axis_CL.set_title(’Lift␣Curve␣of␣a␣Wing’)
axis_CL.legend()
axis_CL.grid()

#Plot Drag Curve
figure_CD, axis_CD = plt.subplots()
axis_CD.plot(data[:,0], data[:,2], ’rx-’,label=’Wing’)
axis_CD.set_xlabel(’Angle␣of␣Atack’)
axis_CD.set_ylabel(’Drag␣Coefficient’)
axis_CD.set_title(’Drag␣Curve␣of␣a␣Wing’)
axis_CD.legend()
axis_CD.grid()

#Plot Drag Polar
figure_CDCL, axis_CDCL = plt.subplots()
axis_CDCL.plot(data[:,1], data[:,2], ’gx-’,label=’Wing’)
axis_CDCL.set_xlabel(’Lift␣Coefficient’)
axis_CDCL.set_ylabel(’Drag␣Coefficient’)
axis_CDCL.set_title(’Drag␣Polar␣of␣a␣Wing’)
axis_CDCL.legend()
axis_CDCL.grid()

#Plot L/D Ratio
figure_LDR, axis_LDR = plt.subplots()
axis_LDR.plot(data[:,0], data[:,1]/data[:,2], ’cx-’,label=’Wing’)
axis_LDR.set_xlabel(’Angle␣of␣Attack␣(deg)’)
axis_LDR.set_ylabel(’L/D␣Ratio’)
axis_LDR.set_title(’L/D␣Ratio␣for␣a␣wing’)
axis_LDR.legend()
axis_LDR.grid()

#Show Plots
```

```
plt.show()
```

0.3 Results

The results for the lift to drag ratio seems unrealistic and so is the angle of attack. The main issue is that `avl` is a vortex lattice method, which is an inviscid solver. This means that drag due to viscous effects are severely underestimated. Since drag is the denominator in the lift to drag ratio, this inaccurate predictions of drag leads to wildly inaccurate results for the main wing's optimum angle of attack.

We have also attempted to "inject" viscous drag taken from `xfoil` directly for the airfoil operating at 50000 Reynold's number and the results are still unrealistic. On the other hand, the lift curve slope matches well with the approximate expression, indiciating that the approximation for the lift curve slope is usable for modelling purposes.

0.4 Theoretical Tail Sizing

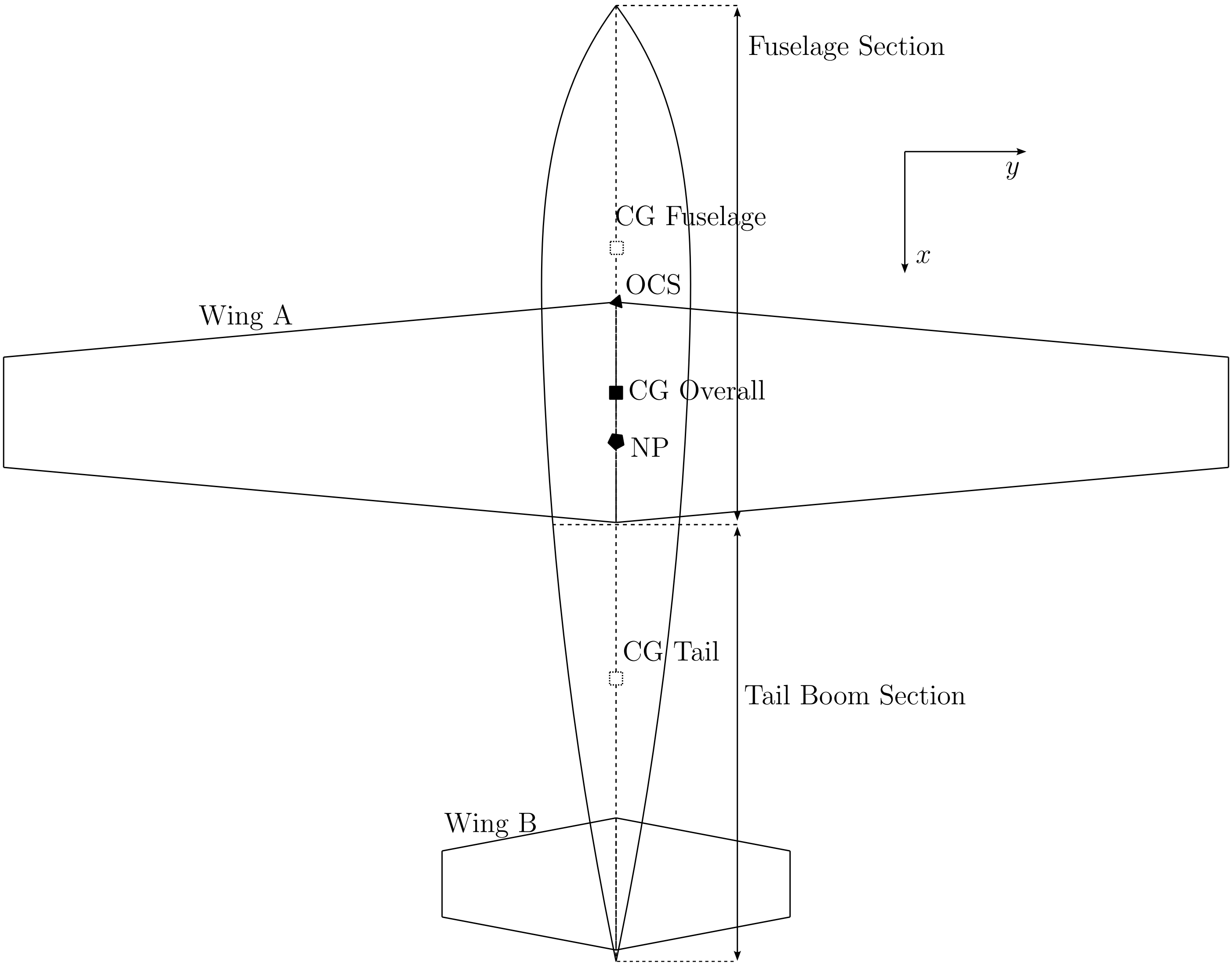

Figure 1: Schematic of Aircraft

We are going to label the main wing that is upstream of the stabilizer as wing A. We are going to label the stabilizer as wing B. Wing B is going to be downstream of wing A. We are going to adopt

AVL's coordinate system wherein x represents distance downstream of aircraft, y goes out of the right wing of the aircraft, and z points up from the body of the aircraft. We are also going to nest the coordinate system's origin at the leading edge of the main wing.

The aircraft is going to be divided into 2 sections, the fuselage section and the tail section. The fuselage section are is indicated in the diagram and includes wing A as well. The tail section is also going to include wing B.

Here is a list of abbreviations used in the diagram and an explanation of what the abbreviations represent.

1. `OCS`: Origin of Coordinate System
2. `CG Overall`: Overall Center of Gravity accounting both fuselage section and tail section
3. `CG Fuselage`: Center of Gravity of the forward fuselage section
4. `CG Tail`: Center of Gravity of the aft tail section
5. `NP`: Neutral Point of the Aircraft

0.4.1 Center of Gravity

The definition for center of gravity for a system of discrete masses is shown below,

$$x_{cg,o} = \frac{\sum_{i=1}^{n} [m_i x_i]}{\sum_{i=1}^{n} [m_i]} \tag{1}$$

We can use the equation above to find out the overall center of mass for the aircraft if we have the mass and center of mass location of both the fuselage section and the tail section. Considering for the case wherein $n = 2$,

$$x_{cg,o} = \frac{m_f x_{cg,f} + m_t x_{cg,t}}{m_f + m_t}$$

wherein m_f and m_t represents the total mass of the fuselage and the tail section respectively. x_{cgf} represents the center of gravity for the fuselage section and x_{cgt} represents the center of gravity for the tail section. We can use equation **??** to determine the overall center of gravity if we want to consider decompose the tail section into the effect of the tail boom, wing B, and other masses including rudder.

$$x_{cg,o} = \frac{m_f x_{cg,f} + m_{wb} x_{cg,wb} + m_{tb} x_{cg,tb} + m_{tm} x_{cg,tm}}{m_f + m_{wb} + m_{tb} + m_{tm}} \tag{2}$$

wherein wb is supposed to represent wing B, tb is supposed to represent tail boom, tm represents tail miscellaneous mass.

0.4.2 Static Margin

Because of the coordinate system that was declared, there should be a slight modification to the definition of static margin,

$$s_m = \frac{x_{np} - x_{cg,o}}{c_{hr,a}}$$

wherein $c_{hr,a}$ represents the chord length of wing A. The equation above can be re-manipulated for the location of the neutral point,

$$s_m c_{hr,a} = x_{np} - x_{cg,o}$$

$$x_{np} = s_m c_{hr,a} + x_{cg,o} \tag{3}$$

0.4.3 Lift Curve Slopes

The neutral point is going to be defined where the Moments acting on the aircraft about that point is unchanging with angle of attack. This means that for an increase in angle of attack, the increase in moments of the wings upstream of the neutral point must be matched by the decrease in moments of the wings downstream.

$$0 = \sum_{i=1}^{n_w} \left[L_{\alpha,i}(x_{np} - x_{ac,i}) \right]$$

wherein $x_{ac,i}$ represents the i^{th} wing on the system and n_w represents the number of wings in the aircraft. $L_{\alpha,i}$ represents the derivative of lift force with respect to angle of attack α for the i^{th} wing. Just for convenience, we can multiply both sides by -1 and obtain,

$$0 = \sum_{i=1}^{n_w} \left[L_{\alpha,i}(x_{ac,i} - x_{np}) \right]$$

This is the same expression and concept for center of gravity, so we should not be surprised if we can express x_{np} similarly to how center of gravity is expressed. Let us substitute for the conventional system where we are only dealing with 2 wings,

$$0 = \sum_{i=1}^{2} \left[L_{\alpha,i}(x_{ac,i} - x_{np}) \right] = L_{\alpha,a}(x_{ac,a} - x_{np}) + L_{\alpha,b}(x_{ac,b} - x_{np})$$

$$-L_{\alpha,a}(x_{ac,a} - x_{np}) = L_{\alpha,b}(x_{ac,b} - x_{np})$$

$$L_{\alpha,a}(x_{np} - x_{ac,a}) = L_{\alpha,b}(x_{ac,b} - x_{np})$$

The lift slope of a wing in terms of the non-dimensional parameters is shown below

$$L_\alpha = \frac{1}{2}\rho v_\infty^2 A_w c_{L,\alpha}$$

Substituting this lift slope in terms of non-dimensional parameters,

$$\frac{1}{2}\rho v_\infty^2 A_{w,a} c_{L\alpha,a}(x_{np} - x_{ac,a}) = \frac{1}{2}\rho v_\infty^2 A_{w,b} c_{L\alpha,b}(x_{ac,b} - x_{np})$$

$$A_{w,a} c_{L\alpha,a}(x_{np} - x_{ac,a}) = A_{w,b} c_{L\alpha,b}(x_{ac,b} - x_{np}) \tag{4}$$

0.4.4 Tail Downforce

For the moments about the center of gravity to remain 0 for level and steady flight, the moments from all the wings about the center of gravity must add up to zero. Therefore,

$$0 = \sum_{i=1}^{n_w} \left[L_{0,i}(x_{cg,o} - x_{ac,i}) + M_{0,i} \right]$$

wherein n_w represents the number of wings on the aircraft. Expanding just for the case of 2 wings,

$$0 = L_{0,a}(x_{cg,o} - x_{ac,a}) + M_{0,a} + L_{0,b}(x_{cg,o} - x_{ac,b}) + M_{0,b}$$

$$-L_{0,b}(x_{cg,o} - x_{ac,b}) - M_{0,b} = L_{0,a}(x_{cg,o} - x_{ac,a}) + M_{0,a}$$

$$L_{0,b}(x_{ac,b} - x_{cg,o}) - M_{0,b} = L_{0,a}(x_{cg,o} - x_{ac,a}) + M_{0,a}$$

We can use this equation to determine the downforce that the tail should produce for trim flight providing we know enough of the unknowns. Manipulating,

$$L_{0,b}(x_{ac,b} - x_{cg,o}) = L_{0,a}(x_{cg,o} - x_{ac,a}) + M_{0,a} + M_{0,b}$$

$$L_{0,b} = \frac{L_{0,a}(x_{cg,o} - x_{ac,a}) + M_{0,a} + M_{0,b}}{(x_{ac,b} - x_{cg,o})} \tag{5}$$

For level and steady flight, the combined lift of wing A and wing B must be the total weight of the aircraft,

$$m_{tot}g = L_{0,a} + L_{0,b}$$

We can substitute the expression for $L_{0,b}$ to obtain an expression purely in terms of $L_{0,a}$,

$$m_{tot}g = L_{0,a} + \frac{L_{0,a}(x_{cg,o} - x_{ac,a}) + M_{0,a} + M_{0,b}}{(x_{ac,b} - x_{cg,o})}$$

If we define the dynamic pressure q to be $q = \frac{1}{2}\rho v^2$, then we can express the lift and moments to be,

$$L = qA_w c_L \quad , \quad M = qA_w c_{hr} c_M \tag{6}$$

Substituting these,

$$m_{tot}g = qA_{w,a}c_{L,0,a} + \frac{qA_{w,a}c_{L,0,a}(x_{cg,o} - x_{ac,a}) + qA_{w,a}c_{hr,a}c_{M,ac,a} + qA_{w,b}c_{hr,b}c_{M,ac,b}}{(x_{ac,b} - x_{cg,o})}$$

We can factor out the dynamic pressure from this expression,

$$m_{tot}g = q\left(A_{w,a}c_{L,0,a} + \frac{A_{w,a}c_{L,0,a}(x_{cg,o} - x_{ac,a}) + A_{w,a}c_{hr,a}c_{M,ac,a} + A_{w,b}c_{hr,b}c_{M,ac,b}}{(x_{ac,b} - x_{cg,o})}\right)$$

Making dynamic pressure the subject of the equation,

$$q = \frac{m_{tot}g}{A_{w,a}c_{L,0,a} + \frac{A_{w,a}c_{L,0,a}(x_{cg,o} - x_{ac,a}) + A_{w,a}c_{hr,a}c_{M,ac,a} + A_{w,b}c_{hr,b}c_{M,ac,b}}{(x_{ac,b} - x_{cg,o})}} \tag{7}$$

From the dynamic pressure, we can easily compute the downforce that the tail is producing. Just use the dynamic pressure and equation 6 to get the dimensionalized lift and pitching moments of wing A. Then use equation 5 for the dimensional "lift" the tail is producing. We can also have an idea of what the velocity of the aircraft would be at the "trim" conditions of 0 angle of attack.

0.4.5 Substitutions

Let us start by substituting x_{np} out of equation 4 using equation 3,

$$A_{w,a}c_{L\alpha,a}\left[s_m c_{hr,a} + x_{cg,o} - x_{ac,a}\right] = A_{w,b}c_{L\alpha,b}\left[x_{ac,b} - s_m c_{hr,a} - x_{cg,o}\right]$$

$$A_{w,a}c_{L\alpha,a}\left[(s_m c_{hr,a} - x_{ac,a}) + x_{cg,o}\right] = A_{w,b}c_{L\alpha,b}\left[(x_{ac,b} - s_m c_{hr,a}) - x_{cg,o}\right]$$

We are modelling the center of gravity to be a rational function of the wing B's area $A_{w,b}$, and it will get very complicated, so we are going to group the known variables together to make it more readable. Let,

$$k_{1,a} = s_m c_{hr,a} - x_{ac,a} \quad , \quad k_{1,b} = x_{ac,b} - s_m c_{hr,a} \tag{8}$$

Notationally simplifying,

$$A_{w,a}c_{L\alpha,a}\left[k_{1,a} + x_{cg,o}\right] = A_{w,b}c_{L\alpha,b}\left[k_{1,b} - x_{cg,o}\right] \tag{9}$$

Re-iterating the expression for center of gravity that includes the tail boom, wing B and miscellaneous masses at the tail (equation 2),

$$x_{cg,o} = \frac{m_f x_{cg,f} + m_{wb}x_{cg,wb} + m_{tb}x_{cg,tb} + m_{tm}x_{cg,tm}}{m_f + m_{wb} + m_{tb} + m_{tm}}$$

$$x_{cg,o} = \frac{m_f x_{cg,f} + m_{tb} x_{cg,tb} + m_{tm} x_{cg,tm} + m_{wb} x_{cg,wb}}{m_f + m_{tb} + m_{tm} + m_{wb}}$$

Let us group the known variables,

$$k_{2,u} = m_f x_{cg,f} + m_{tb} x_{cg,tb} + m_{tm} x_{cg,tm} \quad , \quad k_{2,l} = m_f + m_{tb} + m_{tm} \tag{10}$$

Notationally simplifying the expression for center of gravity for the aircraft,

$$x_{cg,o} = \frac{k_{2,u} + m_{wb} x_{cg,wb}}{k_{2,l} + m_{wb}}$$

Substituting the equation above into equation 9,

$$A_{w,a} c_{L\alpha,a} \left[k_{1,a} + x_{cg,o}\right] = A_{w,b} c_{L\alpha,b} \left[k_{1,b} - x_{cg,o}\right]$$

$$A_{w,a} c_{L\alpha,a} \left[k_{1,a} + \frac{k_{2,u} + m_{wb} x_{cg,wb}}{k_{2,l} + m_{wb}}\right] = A_{w,b} c_{L\alpha,b} \left[k_{1,b} - \frac{k_{2,u} + m_{wb} x_{cg,wb}}{k_{2,l} + m_{wb}}\right]$$

Multiplying both hand sides by $k_{2,l} + m_{wb}$,

$$A_{w,a} c_{L\alpha,a} \left[k_{1,a}(k_{2,l} + m_{wb}) + k_{2,u} + m_{wb} x_{cg,wb}\right] = A_{w,b} c_{L\alpha,b} \left[k_{1,b}(k_{2,l} + m_{wb}) - (k_{2,u} + m_{wb} x_{cg,wb})\right]$$

$$A_{w,a} c_{L\alpha,a} \left[k_{1,a}(k_{2,l} + m_{wb}) + k_{2,u} + m_{wb} x_{cg,wb}\right] = A_{w,b} c_{L\alpha,b} \left[k_{1,b}(k_{2,l} + m_{wb}) - k_{2,u} - m_{wb} x_{cg,wb}\right]$$

Let us expand and group m_{wb} terms together,

$$A_{w,a} c_{L\alpha,a} \left[k_{1,a} k_{2,l} + k_{1,a} m_{wb} + k_{2,u} + m_{wb} x_{cg,wb}\right] = A_{w,b} c_{L\alpha,b} \left[k_{1,b} k_{2,l} + k_{1,b} m_{wb} - k_{2,u} - m_{wb} x_{cg,wb}\right]$$

$$A_{w,a} c_{L\alpha,a} \left[k_{1,a} k_{2,l} + k_{2,u} + k_{1,a} m_{wb} + m_{wb} x_{cg,wb}\right] = A_{w,b} c_{L\alpha,b} \left[k_{1,b} k_{2,l} - k_{2,u} + k_{1,b} m_{wb} - m_{wb} x_{cg,wb}\right]$$

$$A_{w,a} c_{L\alpha,a} \left[(k_{1,a} k_{2,l} + k_{2,u}) + (k_{1,a} + x_{cg,wb}) m_{wb}\right] = A_{w,b} c_{L\alpha,b} \left[(k_{1,b} k_{2,l} - k_{2,u}) + (k_{1,b} - x_{cg,wb}) m_{wb}\right]$$

The expression is getting very long, so let us declare yet another set of notationally simplifying variables,

$$k_{3,lc} = k_{1,a} k_{2,l} + k_{2,u} \quad , \quad k_{3,ld} = k_{1,a} + x_{cg,wb} \tag{11}$$

$$k_{3,rc} = k_{1,b} k_{2,l} - k_{2,u} \quad , \quad k_{3,rd} = k_{1,b} - x_{cg,wb} \tag{12}$$

Substituting these notational simplifications again,

$$A_{w,a} c_{L\alpha,a} \left[k_{3,lc} + k_{3,ld} m_{wb}\right] = A_{w,b} c_{L\alpha,b} \left[k_{3,rc} + k_{3,rd} m_{wb}\right] \tag{13}$$

Without making any further assumptions, the equation above is as far as we can go with our tail sizing. To proceed further, we have to relate the area of the tail or wing B with the mass of wing B. For that, we are going to make a linear approximation, so we are going to assume the following,

$$m_{wb} = c_{bs} A_{w,b} + c_{bf}$$

wherein c_{bs} and c_{bf} are fixed constants and $A_{w,b}$ represents area of wing B (tail stabilizer). Substituting into equation 13,

$$A_{w,a} c_{L\alpha,a} \left[k_{3,lc} + k_{3,ld}(c_{bs} A_{w,b} + c_{bf})\right] = A_{w,b} c_{L\alpha,b} \left[k_{3,rc} + k_{3,rd}(c_{bs} A_{w,b} + c_{bf})\right]$$

It should be obvious at this point that the expression above is a quadratic in terms of $A_{w,b}$ and that we can solve for $A_{w,b}$. Manipulating the expression above to form a quadratic,

$$A_{w,a} c_{L\alpha,a} \left[k_{3,lc} + k_{3,ld} c_{bs} A_{w,b} + k_{3,ld} c_{bf}\right] = A_{w,b} c_{L\alpha,b} \left[k_{3,rc} + k_{3,rd} c_{bs} A_{w,b} + k_{3,rd} c_{bf}\right]$$

$$A_{w,a} c_{L\alpha,a} \left[(k_{3,lc} + k_{3,ld} c_{bf}) + k_{3,ld} c_{bs} A_{w,b}\right] = A_{w,b} c_{L\alpha,b} \left[(k_{3,rc} + k_{3,rd} c_{bf}) + k_{3,rd} c_{bs} A_{w,b}\right]$$

$$A_{w,a}c_{L\alpha,a}(k_{3,lc}+k_{3,ld}c_{bf})+A_{w,a}c_{L\alpha,a}k_{3,ld}c_{bs}A_{w,b}=A_{w,b}c_{L\alpha,b}(k_{3,rc}+k_{3,rd}c_{bf})+A_{w,b}c_{L\alpha,b}k_{3,rd}c_{bs}A_{w,b}$$

$$A_{w,a}c_{L\alpha,a}(k_{3,lc}+k_{3,ld}c_{bf})+A_{w,a}c_{L\alpha,a}k_{3,ld}c_{bs}A_{w,b}=c_{L\alpha,b}(k_{3,rc}+k_{3,rd}c_{bf})A_{w,b}+c_{L\alpha,b}k_{3,rd}c_{bs}A_{w,b}^2$$

$$A_{w,a}c_{L\alpha,a}(k_{3,lc}+k_{3,ld}c_{bf})=c_{L\alpha,b}k_{3,rd}c_{bs}A_{w,b}^2+c_{L\alpha,b}(k_{3,rc}+k_{3,rd}c_{bf})A_{w,b}-A_{w,a}c_{L\alpha,a}k_{3,ld}c_{bs}A_{w,b}$$

$$0=c_{L\alpha,b}k_{3,rd}c_{bs}A_{w,b}^2+[c_{L\alpha,b}(k_{3,rc}+k_{3,rd}c_{bf})-A_{w,a}c_{L\alpha,a}k_{3,ld}c_{bs}]A_{w,b}-A_{w,a}c_{L\alpha,a}(k_{3,lc}+k_{3,ld}c_{bf})$$

We can declare another set of constants, hopefully this is the last one to provide us an even greater shortening of the expression's notations.

$$k_{4,a}=c_{L\alpha,b}k_{3,rd}c_{bs} \tag{14}$$

$$k_{4,b}=c_{L\alpha,b}(k_{3,rc}+k_{3,rd}c_{bf})-A_{w,a}c_{L\alpha,a}k_{3,ld}c_{bs} \tag{15}$$

$$k_{4,c}=-A_{w,a}c_{L\alpha,a}(k_{3,lc}+k_{3,ld}c_{bf}) \tag{16}$$

Substituting these notational simplifications,

$$0=k_{4,a}A_{w,b}^2+k_{4,b}A_{w,b}+k_{4,c}$$

the constants above $k_{4,a}$ $k_{4,b}$ and $k_{4,c}$ are all known constants. We can solve the equation above,

$$A_{w,b}=\frac{1}{2k_{4,a}}\left(-k_{4,b}\pm\sqrt{k_{4,b}^2-4k_{4,a}k_{4,c}}\right)$$

Let us pause a moment and think of what we have done. For all of the workings just in this section, we have assumed nothing except that the mass of wing B (tail) is going to vary linearly with its surface area. We have a generalized method of determining the size of an aircraft's tail wing given that we know the size of the tail boom and everything else. We took into account the desired static margin, and also took into account the moving center of gravity due to a larger tail, which is very useful.

We can supplement additional equations for all the other unknowns and write an iteration algorithm to optimize a design for our aircraft. This will be discussed extensively in the next part of this study.

0.5 Tail Sizing Algorithm

0.5.1 Planning

We are going to claim that the aerodynamic properties of a wing can be completely described if we know the following things:

1. Lift Slope: How the lift coefficient of wing changes with angle of attack, this is affected by aspect ratio and oswald efficiency factor.

2. Location of aerodynamic center: The non-dimensional location of the wing's neutral point, this is not always quarter chord.

3. Moments about aerodynamic center: The Moment coefficient associated to the wing at its neutral point, affected by camber of wing, but independent to angle of attack

4. Value of lift at Trim AOA: This is affected by initial angle of attack, also due to camber of wing

Before we proceed with dynamics and stability, we have to assume the above are known for all of the lifting surfaces of the aircraft. Here is a list of equations we need and was derived from the previous section:

$$k_{1,a}=s_m c_{hr,a}-x_{ac,a} \quad , \quad k_{1,b}=x_{ac,b}-s_m c_{hr,a}$$

$$k_{2,u}=m_f x_{cg,f}+m_{tb}x_{cg,tb}+m_{tm}x_{cg,tm} \quad , \quad k_{2,l}=m_f+m_{tb}+m_{tm}$$

$$k_{3,lc} = k_{1,a}k_{2,l} + k_{2,u} \quad , \quad k_{3,ld} = k_{1,a} + x_{cg,wb}$$

$$k_{3,rc} = k_{1,b}k_{2,l} - k_{2,u} \quad , \quad k_{3,rd} = k_{1,b} - x_{cg,wb}$$

$$k_{4,a} = c_{L\alpha,b}k_{3,rd}c_{bs}$$

$$k_{4,b} = c_{L\alpha,b}(k_{3,rc} + k_{3,rd}c_{bf}) - A_{w,a}c_{L\alpha,a}k_{3,ld}c_{bs}$$

$$k_{4,c} = -A_{w,a}c_{L\alpha,a}(k_{3,lc} + k_{3,ld}c_{bf})$$

$$A_{wb} = \frac{1}{2k_{4,a}}\left(-k_{4,b} \pm \sqrt{k_{4,b}^2 - 4k_{4,a}k_{4,c}}\right)$$

Here is a list of variables that is referred to in the equations above,

1. s_m : Desired Static Margin of the aircraft
2. $c_{hr,a}$: Chord length of the main wing
3. $x_{ac,a}$: Aerodynamic Center of main wing
4. $x_{ac,b}$: Aerodynamic Center of wing B (tail wing)
5. m_f : The total mass of the front fuselage
6. $x_{cg,f}$: Center of Gravity of front fuselage
7. m_{tb} : Mass of the Tail Boom
8. $x_{cg,tb}$: Location of Center of Gravity of Tail Boom
9. m_{tm} : Mass of the other things on the tail
10. $x_{cg,tm}$: Location of the center of gravity of the other things on tail
11. $x_{cg,wb}$: Center of Gravity of wing B (tail wing)
12. $c_{L\alpha,b}$: Derivative of Lift coefficient with respect to Angle of attack for wing B (Tail Wing)
13. $c_{L\alpha,a}$: Derivative of Lift coefficient with respect to Angle of attack for wing A (Main Wing)
14. c_{bf} : Fixed Constant for relating Area to Mass of wing B
15. c_{bs} : Fixed Constant for relating Area to Mass of wing B
16. $A_{w,a}$: Area of wing A (Main Wing)
17. $A_{w,b}$: Area of wing B (Tail Wing)
18. k_{ijk} : These are all intermediate variables

All of the equations above and the variables above allows us to compute the area of wing B. After we have computed the area of wing B, we need to compute the mass of wing B and the compute the location of the overall center of gravity for the system. After we determine the center of gravity, we can proceed to determining the dynamic pressure experienced by the aircraft (equation 7). After that is done, we can finally compute the lift of both wing A and wing B. These are the relevant equations:

$$m_{wb} = c_{bs}A_{w,b} + c_{bf}$$

$$x_{cg,o} = \frac{k_{2,u} + m_{wb}x_{cg,wb}}{k_{2,l} + m_{wb}}$$

For our case, $m_{tot} = m_f + m_{wb} + m_{tb} + m_{tm}$

$$q = \frac{m_{tot}g}{A_{w,a}c_{L,0,a} + \frac{A_{w,a}c_{L,0,a}(x_{cg,o} - x_{ac,a}) + A_{w,a}c_{hr,a}c_{M,ac,a} + A_{w,b}c_{hr,b}c_{M,ac,b}}{(x_{ac,b} - x_{cg,o})}}$$

We might want to run iteration loops later. For the computer efficiency in generating an answer, the language of choice is `Rust`.

0.5.2 Common.rs

This is where common variables like the graviational acceleration and certain common functions like definitions of lift coefficient and dynamic presure get implemented.

```
/*List of Global Constants*/
pub const g:f64 = 9.81; //Gravitational constant (m/s^2)
pub const rho:f64 = 1.225; //Density of air (kg/m^3)

/*Given dynamic pressure, compute velocity*/
pub fn q2vel(q:f64)->f64 {((2.0*q)/rho).powf(0.5)}

/*Given Lift coeff and Dynamic pressure, and area, compute lift force*/
pub fn cL2L(c_L:f64, q:f64, A:f64)->f64 {q*A*c_L}
```

0.5.3 Main.rs

This is where the `Main` function of the program is implemented. The part that uses `Sizing_Tail_Area::default()` essentially uses the default values for the simulation variables. These default values can be overriden from within the main function but we have decided to implement the variables directly as default values. Here we created a `Sizing_Tail_Area` structure and then called methods to operate on it, using the object oriented programming approach.

```
//Library Import Definitions
mod Common; mod Tail_Sizing_Vars; mod Tail_Area;
mod IO; mod Specific_Tail_Optimization;
use Common::*; use Tail_Sizing_Vars::*;

/*Entry Point of the Program*/
fn main() {//Initialize based on default variables
let mut sim = Sizing_Tail_Area{..Sizing_Tail_Area::default()};
//Some Dynamic Linking going on
sim.Tail_Optimization_Var_Setup();
//Run Simulations
sim.Compute_Tail_Area();
sim.Tail_Downforce();
sim.Tail_Optimization_Cost_Function();
sim.print_essentials();}
```

0.5.4 Tail_Sizing_Vars.rs

This is where all of the data for the analysis on the aircraft tail is defined, and where the default initialization values are also defined. This and the `Common.rs` part are the only parts in the entire program where variables are initialized. These initial values dictate the entire running and output of the program. `DL` indicates variables which are later dynamically linked to other variables. Changing these `DL` variables at the default function is meaningless because these variables are overwritten before they are used later.

```
/*All the Variables Needed for Tail Sizing*/
pub struct Sizing_Tail_Area {
    /////////////////////////////////////////////////////////////////////
    // Start of Problem Specific Variables
    /////////////////////////////////////////////////////////////////////
        //////////////////////////////////////////////////////////////////
        // Needed for the tail boom of the aircraft
        //////////////////////////////////////////////////////////////////
        pub L_tb:f64, //Length of tail boom (m)
        pub m_tb_L:f64, //Mass per unit length of tail boom (kg/m)
        pub m_tb_f:f64, //Mass of tail boom that is fixed
        //////////////////////////////////////////////////////////////////
        // Needed to size the tail of the aircraft
        //////////////////////////////////////////////////////////////////
        pub s_m:f64, //Desired Static Margin of the aircraft (dimless)
        pub c_hr_a:f64, //Chord length of the main wing (m)
        pub x_ac_a:f64, //Aerodynamic Center of main wing (m)
        pub x_ac_b:f64, //Aerodynamic Center of wing B (tail wing) (m)
        pub m_f:f64, //The total mass of the front fuselage (kg)
        pub x_cg_f:f64, //Center of Gravity of front fuselage (m)
        pub m_tb:f64, //Mass of the Tail Boom (kg)
        pub x_cg_tb:f64, //Location of Center of Gravity of Tail Boom (m)
        pub m_tm:f64, //Mass of the other things on the tail (kg)
        pub x_cg_tm:f64, //Location of the center of gravity of the other things on tail (m)
        pub x_cg_wb:f64, //Center of Gravity of wing B (tail wing) (m)
        pub c_L_alpha_b:f64, //Derivative of Lift coefficient  for wing B (Tail Wing) (dimless)
        pub c_L_alpha_a:f64, //Derivative of Lift coefficient  for wing A (Main Wing) (dimless)
        pub c_bf:f64, //Fixed Constant for relating Area to Mass of wing B
        pub c_bs:f64, //Fixed Constant for relating Area to Mass of wing B
        pub A_w_a:f64, //Area of wing A (Main Wing) (m^2)
        //////////////////////////////////////////////////////////////////
        // Needed to Figure out lift force of tail
        //////////////////////////////////////////////////////////////////
        pub c_L_0_a:f64, //Lift coefficient at trim conditions for main wing
        pub c_M_ac_a:f64, //Moments about the aerodynamic center of main wing
        pub c_hr_b:f64, //chord length of tail wing
        pub c_M_ac_b:f64, //Moments about the aerodynamic center of tail wing
    /////////////////////////////////////////////////////////////////////
    // Start of Problem-Derived Variables
    /////////////////////////////////////////////////////////////////////
    //Communicable intermediate k-constants
    pub k_2_u:f64, pub k_2_l:f64,
    pub m_wb:f64, //Weight of wing B (kg)
    pub A_w_b:f64, //Area of wing B (m^2)
    pub x_cg_o:f64, //Overall Center of Gravity Location (m)
    pub m_tot:f64, //Total Weight of the aircraft (kg)
    pub q:f64, //Dynamic Pressure of aircaft (Pa)
    pub trim_vel:f64, //Trim velocity of aircraft (m/s)
    pub L_0_a:f64, //Lift force on tail wing at trim flight
    pub L_0_b:f64, //Lift force on tail wing at trim flight
    pub cost_val:f64, //Value of the cost function that needs to be minimized
}

impl Sizing_Tail_Area {
/*Default values that are used for the variables*/
pub fn default() -> Sizing_Tail_Area {
    Sizing_Tail_Area{//Default values for Aircraft Tail Sizing
    L_tb: 1.6, //Length of tail boom (m)
    m_tb_L: 0.10, //Mass per unit length of tail boom (kg/m)
    m_tb_f: 0.01, //Mass of tail boom that is fixed
    s_m: 0.17, //Desired Static Margin of the aircraft (dimless)
    c_hr_a: 0.3556, //Chord length of the main wing (m)
    x_ac_a: 0.08249, //Aerodynamic Center of main wing (m)
    x_ac_b: 1.9, //Aerodynamic Center of wing B (tail wing) (m)                     [WARNING!!!]DL
    m_f: 2.8, //The total mass of the front fuselage (kg)
    x_cg_f: 0.1, //Center of Gravity of front fuselage (m)
    m_tb: 0.2, //Mass of the Tail Boom (kg)                                         [WARNING!!!]DL
    x_cg_tb: 0.3, //Location of Center of Gravity of Tail Boom (m)                  [WARNING!!!]DL
    m_tm: 0.0, //Mass of the other things on the tail (kg)
    x_cg_tm: 0.0, //Location of the center of gravity of other things on tail (m)   [WARNING!!!]DL
    x_cg_wb: 0.9, //Center of Gravity of wing B (tail wing) (m)                     [WARNING!!!]DL
    c_L_alpha_b: 0.050, //Derivative of Lift coefficient  for wing B (Tail Wing) (dimless)
    c_L_alpha_a: 0.07340, //Derivative of Lift coefficient  for wing A (Main Wing) (dimless)
    c_bf: 0.0, //Fixed Constant for relating Area to Mass of wing B
    c_bs: 1.12, //Fixed Constant for relating Area to Mass of wing B (kg/m^2)
    A_w_a: 0.903224, //Area of wing A (Main Wing) (m^2)
    c_L_0_a: 0.8, //Lift coefficient at trim conditions for main wing
    c_M_ac_a: -0.13048, //Moments about the aerodynamic center of main wing
    c_M_ac_b: 0.0, //Moments about the aerodynamic center of tail wing
    //Garbage valus for intermediate variables that need to communicate
    k_2_u: 1e-3, k_2_l: 1e-3, m_wb: 1e-3, A_w_b: 1e-3, x_cg_o: 1e-3, m_tot: 1e-3, q : 1e-3,
    c_hr_b: 0.2, trim_vel: 1e-3, L_0_a: 1e-3, L_0_b: 1e-3, cost_val: 1e-3,
    }}
}
```

0.5.5 Tail_Area.rs

These are all the necessary functions needed to compute the tail area of an aircraft given enough information about the ”forward fusealge section” of the entire aircraft.

```
use g; use q2vel; use cL2L; use Sizing_Tail_Area;
impl Sizing_Tail_Area{

/*Sizes the area of wing B based on location of neutral point and center of gravity*/
pub fn Compute_Tail_Area(&mut self) {
    //Computation of all intermediate k-constants
    let k_1_a:f64 = self.s_m*self.c_hr_a - self.x_ac_a;
    let k_1_b:f64 = self.x_ac_b -self.s_m*self.c_hr_a;
    self.k_2_u = self.m_f*self.x_cg_f + self.m_tb*self.x_cg_tb + self.m_tm*self.x_cg_tm;
    self.k_2_l = self.m_f + self.m_tb + self.m_tm;
    let k_3_lc:f64 = k_1_a*self.k_2_l + self.k_2_u;
    let k_3_ld:f64 = k_1_a + self.x_cg_wb;
    let k_3_rc:f64 = k_1_b*self.k_2_l - self.k_2_u;
    let k_3_rd:f64 = k_1_b - self.x_cg_wb;
    let k_4_a:f64 = self.c_L_alpha_b*k_3_rd*self.c_bs;
    let k_4_b:f64 = self.c_L_alpha_b*(k_3_rc+k_3_rd*self.c_bf)-self.A_w_a*self.c_L_alpha_a*k_3_ld*
    self.c_bs;
    let k_4_c:f64 =  - self.A_w_a*self.c_L_alpha_a*(k_3_lc+k_3_ld*self.c_bf);
    //Computation of the Tail Wing Area and return this value
    self.A_w_b = (-k_4_b + (k_4_b.powf(2.0)-4.0*k_4_a*k_4_c).powf(0.5))/(2.0*k_4_a);}

/*Computes the mass of wing B, implements linear approximation with area*/
pub fn Tail_Wing_Mass(&mut self){self.m_wb = self.c_bs*self.A_w_b+self.c_bf;}

/*Based on wing B's area and for level steady flight, compute Lift at wing B*/
pub fn Tail_Downforce(&mut self) {
    self.Tail_Wing_Mass();
    self.x_cg_o = (self.k_2_u+self.m_wb*self.x_cg_wb)/(self.k_2_l+self.m_wb);
    self.m_tot = self.m_f + self.m_wb + self.m_tb + self.m_tm;
    self.q = (self.m_tot*g)/(self.A_w_a*self.c_L_0_a + (self.A_w_a*self.c_L_0_a*(self.x_cg_o-self.
    x_ac_a) + self.A_w_a*self.c_hr_a*self.c_M_ac_a + self.A_w_b*self.c_hr_b*self.c_M_ac_b)/(self.
    x_ac_b-self.x_cg_o));
    self.trim_vel = q2vel(self.q);
    self.L_0_a = cL2L(self.c_L_0_a, self.q, self.A_w_a);
    self.L_0_b = self.m_tot*g - self.L_0_a;}
}
```

0.5.6 Specific_Tail_Optimization.rs

This is the place where assumptions are made about the system and we start computing the ”cost” function which we would want to minimize in a later part.

```
use g;use Sizing_Tail_Area;
impl Sizing_Tail_Area {

//c_hr_b, gotta change it to link with aspect ratio!!!

/*Supplemental Equations, mass of tail boom*/
pub fn Tail_Boom_Mass(&mut self){self.m_tb = self.L_tb*self.m_tb_L +self.m_tb_f}

/*Supplemental Equations, Linking some geometric variables to each other*/
pub fn Tail_Optimization_Var_Setup(&mut self){
    //Approximate mass of tail boom
    self.Tail_Boom_Mass();
    //Set location of center of gravity of tail boom to half its total length
    self.x_cg_tb = 0.5*self.L_tb;
    //Set aerodynamic center of wing B to edge of tail boom
    self.x_ac_b = self.L_tb;
    //set the center of mass of wing B to edge of tail boom
    self.x_cg_wb = self.L_tb;
    //Set the miscellaneous mass at the end of the tail
    self.x_cg_tm = self.L_tb;
}

/*Cost function for the tail, minimize this function*/
pub fn Tail_Optimization_Cost_Function(&mut self){
    self.cost_val = g*(self.m_tb + self.m_wb) - self.L_0_b;}

//Bisection solving to determine the length of the optimum tail
//fn Tail_Optimization_Run(&mut self){}
}
```

0.5.7 IO.rs

This is a file that handles all of the inputs and outputs of the program. This shows all of the variables and data generated which we are interested in.

```
use Sizing_Tail_Area;

impl Sizing_Tail_Area {
/*Print all variables we are interested in*/
pub fn print_essentials(&mut self){
//println!("", );
    println!("####################################################################");
    println!("#␣Global␣Information␣about␣Aircraft:");
    println!("####################################################################");
    println!("Total␣Aircraft␣Mass␣=␣{}␣kg", self.m_tot);
    println!("Location␣of␣Overall␣Center␣of␣Gravity␣=␣{}␣m", self.x_cg_o);
    println!("Aircraft␣Trim␣Velocity␣=␣{}␣m/s", self.trim_vel);
    println!("####################################################################");
    println!("#␣Print␣Detail␣about␣Tail␣Boom");
    println!("####################################################################");
    println!("Tail␣Boom␣Length:␣{}␣m", self.L_tb);
    println!("Tail␣Boom␣Mass:␣{}␣kg", self.m_tb);
    println!("####################################################################");
    println!("#␣Print␣Details␣about␣wing␣B");
    println!("####################################################################");
    println!("Area␣of␣tail␣wing␣(wing␣B)␣=␣{}␣m^2", self.A_w_b);
    println!("Mass␣of␣tail␣wing␣(wing␣B)␣=␣{}␣kg", self.m_wb);
    println!("####################################################################");
    println!("#␣Lift␣Distribution");
    println!("####################################################################");
    println!("Lift␣produced␣at␣main␣wing␣(wing␣A)␣=␣{}␣N", self.L_0_a);
    println!("Lift␣produced␣at␣tail␣wing␣(wing␣B)␣=␣{}␣N", self.L_0_b);
    println!("Cost␣Function␣(Down␣Force␣of␣entire␣tail␣section)␣=␣{}␣N", self.cost_val);
    println!("####################################################################");
    println!("#␣End␣of␣Outputs");
    println!("####################################################################");}
}
```

0.5.8 Results

Compiling, running and then redirecting to a file can be done with the following command,

```
rustc Main.rs && ./Main > Algo_Res.tex
```

Here is the result of running the algorithm,

```
####################################################################
# Global Information about Aircraft:
####################################################################
Total Aircraft Mass = 3.259920892049323 kg
Location of Overall Center of Gravity = 0.269906373932219 m
Aircraft Trim Velocity = 8.114832959948883 m/s
####################################################################
# Print Detail about Tail Boom
####################################################################
Tail Boom Length: 1.6 m
Tail Boom Mass: 0.17000000000000004 kg
####################################################################
# Print Details about wing B
####################################################################
Area of tail wing (wing B) = 0.258857939329753 m^2
Mass of tail wing (wing B) = 0.28992089204932336 kg
####################################################################
# Lift Distribution
####################################################################
Lift produced at main wing (wing A) = 29.14410466777778 N
Lift produced at tail wing (wing B) = 2.8357192832260836 N
Cost Function (Down Force of entire tail section) = 1.6761046677777793 N
####################################################################
# End of Outputs
####################################################################
```