Final Report

AERO321

Dynamics of Aerospace Vehicles

# Fall 23 Project



| *Group Members* | | *Tasks* |
|---|---|---|
| Jirina Bredberg | Jirina Bredberg | Tables 1,2,5 |
| Ines Meyer | [illegible] | Tables 1,3,7 |
| Kanishka Kamal | [illegible] | Tables 3,4,6 |
| Benjamin Tollison | Benjamin Tollison | Author/Debugger,8 |



Monday 4th December, 2023

# Contents

# 1 Task 1a

Determine the trim AoA, tail incidence angle, equilibrium thrust, and the normal force with the following system of equations in the body-axis system:

$$\sum F_X = 0 = T_0 - W\sin\alpha_0 + L_{WF}\sin\alpha_0 - D_0\cos\alpha_0 + L_H\sin(\alpha_0 - \epsilon)$$

$$\sum F_Z = 0 = -F_{N_0} + W\cos\alpha_0 - L_{WF}\cos\alpha_0 - D_0\sin\alpha_0 - L_H\cos(\alpha_0 - \epsilon)$$

$$\sum M_{cg} = 0 = F_{N_0}(X_{cg} - X_{inlet}) - L_{WF}(X_{AC_{WF}} - X_{cg})\cos\alpha_0$$
$$-D_0(X_{AC_{WF}} - X_{cg})\sin\alpha_0 - L_H(X_{AC_H} - X_{cg})\cos(\alpha_0 - \epsilon)$$

A systematic approach to solving a system of equations involves breaking down the problem into manageable steps. Initially, one focuses on solving each equation by isolating the capitalized coefficients. This often requires employing linear approximations to better understand the behavior of each equation. The next step is to deconstruct the equations further, systematically breaking them down into simpler forms that are easier to solve. This iterative process allows for a methodical exploration of the system's intricacies, gradually unveiling solutions and facilitating a more comprehensive understanding of the relationships among the variables. Through this systematic method, one navigates the complexity of the system, step by step, until arriving at a point where the equations can be effectively solved.

## 1.1 Determining $L_{WF}$

It is stated that we can neglect the lift from the fuselage in the project assigment such that

$$L_{WF} \approx L_W = q_\infty S_W C_{L_W}$$

Find $C_{L_W} \approx C_{L_{W0}} + C_{L_{\alpha,W}}\alpha_0$
where $C_{L_{W0}} = 0$ due to symmetry, which leads to the problem that we need to now find $C_{L_{\alpha,W}}$

## 1.2 Finding $C_{L_{\alpha,W}}$

1. Find Taper ratio $\lambda = \frac{2S_w}{bc_r}$
2. Find SweepAngle of the LE: $\Lambda_{LE} = \tan^{-1}\left(\tan\Lambda_{c/4} + \frac{1}{AR}\frac{1-\lambda}{1+\lambda}\right)$
3. Find SweepAngle of the c/2: $\Lambda_{c/2} = \tan^{-1}\left(\tan\Lambda_{LE} - \frac{2}{AR}\frac{1-\lambda}{1+\lambda}\right)$
4. Find $C_{l_\alpha} = \frac{2\pi}{\sqrt{1-M_\infty^2}}$
5. Find beta/k constants: $\beta = \sqrt{1-M^2}$, $k = \frac{C_{l_\alpha}}{2\pi}$
6. plug into equation:

$$C_{L_\alpha} = \frac{2\pi AR}{2 + \sqrt{\frac{AR^2\beta^2}{k^2}\left(1 + \frac{\tan^2\Lambda_{c/2}}{\beta^2}\right) + 4}}$$

## 1.3 Find $q_\infty$

$$q_\infty = \frac{1}{2}\rho_\infty v_\infty^2$$

1. Get speed of sound from the tables: $a = 296.5338[m/s]$

2. Find $v_\infty$ from $M = \frac{v_\infty}{a}$

3. Get the $\rho_\infty$ with isentropic equation

    (a) $\frac{\rho_0}{\rho} = \left(1 + \frac{\gamma-1}{2}M^2\right)^{\frac{1}{\gamma-1}}$

    (b) $\rho_0 = 7.3654 \cdot 10^{-4}[sl/ft^3] = 0.3795971165326[kg/m^3]$

$$\therefore L_{WF} = q_\infty S_W C_{L_{\alpha,W}} \alpha_0$$

## 1.4 Finding $D_0$

$$\rightarrow D_0 \equiv D(\alpha_0)$$

$$D = q_\infty S_W C_D$$

1. Find $C_D$ from the given equation: $C_D = 0.0145 + 0.1C_{L_W}^2$

    (a) with $C_{L_W} = C_{L_{\alpha,W}}\alpha_0$

$$\therefore D_0 = q_\infty S_W (0.0145 + 0.1C_{L_{\alpha,W}}^2 \alpha_0^2)$$

## 1.5 Finding $L_H$

$$L_H = \eta_H \frac{S_H}{S_W} C_{L_H}$$

## Finding $C_{L_H}$

$$C_{L_H} \equiv C_{L_H}(\alpha, \epsilon, i_H)$$

1. $\epsilon$ is also a function of $\alpha$

2. $C_{L_H} \approx C_{L_{H0}} + C_{L_{\alpha,H}}\alpha_H + C_{L_{i_H}} i_H | C_{L_{H0}} = 0, symmetry$

3. set $\alpha_H = \alpha_0 + i_H - \frac{d\epsilon}{d\alpha_W}\alpha_0$

4. - Find $C_{L_{\alpha},H}$ with the same process as the wing

    (a) Find Taper ratio $\lambda = \frac{2S_w}{bc_r}$

    (b) Find SweepAngle of the LE: $\Lambda_{LE} = \tan^{-1}\left(\tan\Lambda_{c/4} + \frac{1}{AR}\frac{1-\lambda}{1+\lambda}\right)$

    (c) Find SweepAngle of the c/2: $\Lambda_{c/2} = \tan^{-1}\left(\tan\Lambda_{LE} - \frac{2}{AR}\frac{1-\lambda}{1+\lambda}\right)$

    (d) Find $C_{l_\alpha} = \frac{2\pi}{\sqrt{1-M_\infty^2}}$

    (e) Find Beta/k constants: $\beta = \sqrt{1-M^2}$, $k = \frac{C_{l_\alpha}}{2\pi}$

    (f) plug into equation:

$$C_{L_{\alpha,H}} = \frac{2\pi AR}{2 + \sqrt{\frac{AR^2\beta^2}{k^2}\left(1 + \frac{\tan^2\Lambda_{c/2}}{\beta^2}\right) + 4}}$$

5. Find $C_{L_{i_H}}$ from the given equation: $C_{L_{i_H}} = C_{L_{\alpha,H}}\eta_H \frac{S_H}{S_W}$

$$\therefore L_H = \eta_H \frac{S_H}{S_W} C_{L_{\alpha,H}} \left(\alpha_0 + i_H - \frac{d\epsilon}{d\alpha_W}\alpha_0\right) + C_{L_{\alpha,H}}\eta_H \frac{S_H}{S_W} i_H$$

## 1.6 Finding $F_{N_0}$

Using the given equation

$$F_{N_0} = 2q_\infty A_{inlet} \cos\alpha_0^2 \sin\alpha_0$$

$A_{inlet}$ is given as 15 $ft^2$ or 1.39355 $m^2$

### 1.7 Finding $\epsilon$

$\epsilon$ estimated purely as a function of $\alpha$

$$\epsilon = \frac{d\epsilon}{d\alpha_W}(\alpha_W - \alpha_0)$$

where $\alpha_0$ is 0 because it is a symmetrical airfoil

$$\therefore \epsilon = \frac{d\epsilon}{d\alpha}\alpha_0$$

Then with only 3 unknowns remaining of $T_0, \alpha_0, i_H$, we used the SciPy fsolve function to solve the system of nonlinear equations.

## 2 Task 1b

Determine the most forward c.g. location from the tail incidence angle limit condition.

### 2.1 Equations to consider

$$CL(a, ih) = CL_{\alpha_{wf}} \cdot a + n_H \cdot CL_{\alpha_H} \cdot (a \cdot (1 - \delta_{e\delta a}) + ih) \cdot \frac{S_H}{S_W} \tag{1}$$

$$CL_H(a, ih) = CL_{\alpha_H} \cdot (a \cdot (1 - \delta_{e\delta a}) + ih) \tag{2}$$

$$CL_{wf}(a) = CL_{\alpha_{wf}} \cdot a \tag{3}$$

$$CM(a, ih) = CL_{wf}(a) \cdot (X_{cg} - X_{ac_{wf}}) - n_H \cdot CL_H(a, ih) \cdot (X_{ac_H} - X_{cg}) \cdot \frac{S_H}{S_W} \tag{4}$$

$$CL_{\min} = \frac{W}{q_{\inf} \cdot S_{\mathrm{W}}} \tag{5}$$

## 2.2 Algorithm

The method to find the forward most positon of the center of gravity uses the following algorithm:

1. **Initialization:**

   - Set the $\alpha_0, i_H, T_0$ as the trim conditions
   - Set $X_cg$ as 27 ft
   - $$\begin{aligned} e &= \text{de\_da} \cdot a_0 \\ a_H &= a_0 + i_h - e \\ L_{wf} &= \text{CL\_a\_W} \cdot a_0 \cdot q_{\text{inf}} \cdot S_{\text{W}} \quad \text{(where CL\_a\_wf is the same as CL\_a\_w)} \\ L_h &= \text{CL\_a\_H} \cdot a_H \cdot q_H \cdot S_H \\ C_{D0} &= 0.0145 + 0.1 \cdot (\text{CL\_a\_W} \cdot a_0)^2 \\ D_0 &= q_{\text{inf}} \cdot S_{\text{W}} \cdot C_{D0} \\ F_{N0} &= 2 \cdot q_{\text{inf}} \cdot A_{\text{in}} \cdot (\cos(a_0))^2 \cdot \sin(a_0) \end{aligned} \tag{6}$$

2. **Iterative Process:**

   - Change the $X_{cg}$ to an increment smaller
   - Use the equations of motion to find the new trim values
   - $$\begin{aligned} X &= T_0 \cdot \cos(a_0) - F_{N0} \cdot \sin(a_0) - D_0 - L_h \cdot \sin(e) \\ Z &= W - T_0 \cdot \sin(a_0) - F_{N0} \cdot \cos(a_0) - L_{wf} - L_h \cdot \cos(e) \\ M &= F_{N0} \cdot (X_{\text{cg}} - 0) - L_{wf} \cdot (X_{\text{ac\_wf}} - X_{\text{cg}}) \cdot \cos(a_0) \\ &\quad - D_0 \cdot (X_{\text{ac\_w}} - X_{\text{cg}}) \cdot \sin(a_0) - L_h \cdot (X_{\text{ac\_h}} - X_{\text{cg}}) \cdot \cos(a_0 - e) \end{aligned}$$
   - If the new $i_H \neq \pm 20$ make $X_cg$ smaller
   - If new $i_H \approx 20$ return the $X_cg$

# 3 Task 1 Tables

**Givens**

| | Wing | Horizontal Tail (all moveable) | Vertical Tail | Fuselage |
|---|---|---|---|---|
| Area ($S$) | 400 ft² | 100 ft² | 70 ft² | Length = 48 ft |
| Root chord ($c_r$) | 17 ft | 10 ft | 10 ft | |
| Aspect Ratio ($AR$) | $AR_W$ = 3.6 | $AR_H$ = 3.4 | $AR_V$ = 1.5 | |
| Span ($b$) | b = $\sqrt{(AR)S_W}$ = 37.947 | b = $\sqrt{(AR)S_H}$ = 18.439 | b = $\sqrt{(AR)S_V}$ = 10.247 | |
| Quarter chord sweep ($\Lambda_{\frac{c}{4}}$) | 45° | 45° | 45° | Max fuselage width = 6 ft |
| Airfoil Lift curve slope | $c_{l_{\alpha_W}} = 5.6$ | $c_{l_{\alpha_H}} = 6$ | $c_{l_{\alpha_V}} = 6$ | |
| Incidence angle | $i_W = 0$ | $i_H$ = variable ± 20° | -- | |
| Tail efficiency factor | -- | $\eta_H = 0.5$ | $\eta_V = 1.0$ | Volume = 1000 ft^3 |
| Fuselage interference factor ($k$) | -- | -- | $k = 1.0$ | |
| X-apex | 16 ft | 36 ft | 35 ft | |
| Z-apex | 0 | -- | -- | |
| Downwash/Sidewash gradient | -- | $\frac{d\varepsilon}{d\alpha} = 0.75$ | $\frac{d\sigma}{d\beta} = 0.12$ | $X_{AC_{WF}} = 0.05\overline{c_w}$ (Its 0.05 in front of the wing) |
| $X_{AC}$ | $0.32\overline{c_w}$ $\overline{c_w} = 11.86$ | $0.32\overline{c_w}$ $\overline{c_H} = 6.71$ | $0.32\overline{c_w}$ $\overline{c_V} = 7.32$ | |

| Parameter/Non-dimensional Stability Derivative | Equations used or Reference (if applicable) | Value (units) |
|---|---|---|
| $\alpha_0$ | See Code | 10.6284° |
| $i_H$ | See Code | -4.9097° |
| $C_{L_0}(Trim)$ | $C_{L_W} + \eta \frac{S_H}{S_W} C_{L_H} \cos \varepsilon$ | 0.6572 |
| $C_{D_0}(Trim)$ | $C_D = 0.0145 + 0.1 C_{L_W}^{\,2}$ <br> $C_{D_0} = 0.0145 + 0.1\left(C_{L_{\alpha_W}}(\alpha_{trim} - \alpha_{zero-lift})\right)^2$ <br> $C_{D_0} = 0.0145 + 0.1\left(C_{L_{\alpha_W}}\alpha_{trim}\right)^2$ | 0.04836 |
| $Trim\ thrust$ | See Code | 1849.2467 lb |
| $Normal\ Force$ | $F_N = \dot{m}_p V_\infty \cos\alpha \sin\alpha$ <br> $= \rho A_{in}(V_\infty \cos\alpha)^2 \sin\alpha$ <br> $= 2q_\infty A_{in} \cos^2\alpha \sin\alpha$ | 499.7178 lb |
| $Propulsive\ C_{P_{M_0}}$ | $F_{N_0}(X_{CG} - 0)/(q_\infty \overline{c_W} S_W)$ | 0.03042 |
| $C_{M_0}(Trim, Aerodynamic)$ | $-C_{P_{M_0}}$ | -0.03042 |
| $C_{L_\alpha}$ | $C_{L_{\alpha_{W\&F}}} + C_{L_{\alpha_H}} \eta_H \frac{S_H}{S_W}(1 - \frac{d\varepsilon}{d\alpha})$ | 3.2397 |
| $C_{D_\alpha}$ | $C_D = 0.0145 + 0.1 C_{L_W}^{\,2}$ <br> $C_D = 0.0145 + 0.1\left(C_{L_{\alpha_W}}(\alpha - \alpha_{zero-lift})\right)^2$ <br> $C_{D\alpha} = 0.2 C_{L\alpha W}^2 \alpha$ | 0.3652 |
| $C_{L_{i_H}}$ | $C_{L_{i_H}} = C_{L_{\alpha_H}} \eta_H \left(\frac{S_H}{S_W}\right)$ | 0.4096 |
| $\bar{X}_{NP}$ | $\bar{X}_{NP} = \dfrac{C_{L\alpha_{WF}} X_{AC_{WF}} + C_{L_{\alpha_H}} n_H \left(\frac{S_H}{SW}\right) X_{ACH} \left(1 - \frac{d\varepsilon}{d\alpha}\right)}{C_{L_\alpha}}$ | 28.486 ft |
| $SM$ | $SM = \overline{X_{\mathrm{NP}}} - X_{\mathrm{CG}}$ | 1.486 ft |
| $C_{M_\alpha}$ | $C_{M_\alpha} = C_{L_{\alpha_{WF}}} \Delta\bar{X}_{AC_{WF}} - C_{L_{\alpha_H}} \eta_H V_H \left(1 - \frac{d\varepsilon}{d\alpha}\right)$ | -0.4058 |
| $C_{M_{i_H}}$ | $C_{M_{i_H}} = -C_{L_{\alpha_H}} \eta_H V_H$ | -0.5278 |
| $Forward\ CG\ limit$ | Current CG at 27 ft | 24.3239 ft |

| | | |
|---|---|---|
| $\lvert V_V \rvert$ | $\boldsymbol{M} = \frac{U_\infty}{a} \rightarrow$ $U_\infty = 0.52 * 660\, mph = 343.2\, mph$ | $503.36 \frac{ft}{s}$ |
| $q_\infty$ | $\frac{1}{2}\rho V_V^2 = \frac{1}{2}\left(0.000738 \frac{slugs}{ft^3}\right)\left(503.36 \frac{ft}{s}\right)^2$ | 93.494 |
| $\lambda$ | $\frac{2b - c_r AR}{c_r AR}$ | $\lambda_W = 0.24085$ $\lambda_H = 0.084647$ $\lambda_V = 0.366267$ |
| $\bar{c}$ | $\frac{2(1+\lambda+\lambda^2)}{3(1+\lambda)} c_r$ | $\bar{c}_W = 11.86$ $\bar{c}_H = 6.71$ $\bar{c}_V = 7.32$ |
| $x_{LE_{MAC}}$ | $\frac{b}{2}\left(\frac{1+2\lambda}{3(1+\lambda)}\right)\tan(\Lambda_{LE})$ | Wing = 8.83299 HT = 4.1353 VT = 2.8362 |
| $X_{AC_W}, \overline{X_{AC_W}}$ | $X_{APP_W} + x_{LE_{MAC}} + 0.32\bar{c}_W,$ $\frac{X_{APP_W} + x_{LE_{MAC}} + 0.32\bar{c}_W}{\bar{c}_W}$ | 28.628, 2.41383 |
| $X_{AC_{WF}}, \overline{X_{AC_{WF}}}$ | $X_{APP_W} + x_{LE_{MAC}} + 0.27\bar{c}_W,$ $\frac{X_{APP_W} + x_{LE_{MAC}} + 0.27\bar{c}_W}{\bar{c}_W}$ | 28.0352, 2.36284 |
| $X_{AC_H}, \overline{X_{AC_H}}$ | $X_{APP_H} + x_{LE_{MAC}} + 0.32\bar{c}_H,$ $\frac{X_{APP_H} + x_{LE_{MAC}} + 0.32\bar{c}_H}{\bar{c}_W}$ | 42.2825, 3.56513 |
| $C_{L_{\alpha_\#}}$ | $\frac{2\pi AR}{2 + \sqrt{\frac{AR^2\beta^2}{k^2}\left(1 + \frac{\tan^2(\Lambda_{\frac{c}{2}})}{\beta^2}\right) + 4}}$ | $C_{L_{\alpha_{WF}}} = C_{L_{\alpha_W}}$ =3.1373 $C_{L_{\alpha_H}}$ =3.277 |
| $\Delta\bar{X}_{AC_H}$ | $\frac{X_{AC_H} - X_{Ref(CG)}}{\bar{c}_W}$ | 1.288 |
| $\Delta\bar{X}_{AC_{WF}}$ | $\frac{X_{Ref(CG)} - X_{AC_{W\&F}}}{\bar{c}_W}$ | -0.087285 |
| $k$ | $\frac{c_{l\alpha}}{2\pi}$ | 1.17073 |
| $\Lambda_{LE}$ | $arctan\left(\frac{ARtan\left(\Lambda_{\frac{1}{4}}\right) + \frac{1-\lambda}{1+\lambda}}{AR}\right)$ | $\Lambda_{LE_W} = 49.47°$ $\Lambda_{LE_H} = 51.30°$ $\Lambda_{LE_V} = 52.63°$ |
| $\Lambda_{1/2}$ | $arctan\left(\frac{ARtan\left(\Lambda_{\frac{1}{4}}\right) - 4\left(\frac{1}{2} - \frac{1}{4}\right)\left(\frac{1-\lambda}{1+\lambda}\right)}{AR}\right)$ | $\Lambda_{w(1/2)} = 39.69°$ $\Lambda_{h(1/2)} = 36.94°$ $\Lambda_{v(1/2)} = 34.64°$ |

# 4 Task 2

## Summary of Longitudinal Stability and Control Derivatives

1. $C_{D_M}$ **and** $C_{L_M}$**: Drag and Lift Contributions due to Pitching Motion**

$$C_{D_M} = \frac{M\cos^2(q_{c_W})}{1 - M^2\cos^2(q_{c_W})} \cdot C_{D0}$$
$$C_{L_M} = \frac{M\cos^2(q_{c_W})}{1 - M^2\cos^2(q_{c_W})} \cdot C_{L0}$$

2. $C_{M_q}$**: Pitch Damping Derivative**

$$C_{M_q} = -n_H \cdot CL_{a_H} \cdot \frac{S_H}{S_W} \cdot (X_{ac_h} - X_{cg})^2 \cdot \cos^2(\alpha) \cdot \frac{1}{U_0 \cdot c_{w_w}}$$

3. $C_{M_{\dot\alpha}}$**: Rate of Change of Pitch Damping with Respect to Angle of Attack**

$$C_{M_{\dot\alpha}} = -n_H \cdot \frac{S_H}{S_W} \cdot CL_{a_H} \cdot (X_{ac_h} - X_{cg}) \cdot (X_{ac_h} - X_{ac_W}) \cdot \cos^2(\alpha) \cdot \delta_{e/\delta_a}$$

4. $X_u$, $Z_u$, $M_u$**: Longitudinal Stability Derivatives**

$$X_u = -\frac{(C_{D_u} + \frac{2}{U_0} \cdot C_{D0}) \cdot q_\infty \cdot S_W}{m}$$
$$Z_u = -\frac{(C_{L_u} + \frac{2}{U_0} \cdot C_{L0}) \cdot q_\infty \cdot S_W}{m}$$
$$M_u = -\frac{(C_{M_u} + \frac{2}{U_0} \cdot -\text{prop\_CP\_M0}) \cdot q_\infty \cdot S_W \cdot c_{w_w}}{I_{yy}}$$

5. $X_a$, $Z_a$, $M_a$**: Longitudinal Control Derivatives**

$$X_a = -\frac{(-C_{D_\alpha} + C_{L0}) \cdot q_\infty \cdot S_W}{m}$$
$$Z_a = -\frac{(-CL_\alpha + C_{D0}) \cdot q_\infty \cdot S_W}{m}$$
$$M_a = \frac{C_{M_\alpha} \cdot q_\infty \cdot S_W \cdot c_{w_w}}{I_{yy}}$$

6. $M_q$**: Pitch Control Derivative**

$$M_q = \frac{C_{M_q} \cdot q_\infty \cdot S_W \cdot c_{w_w}}{I_{yy}}$$

| Stability Derivative | Equations Used | Values (units) |
|---|---|---|
| $C_{D_M}$ | $\frac{M_\infty \cos^2 \Lambda_{W_{c/4}}}{1 - M_\infty^2 \cos^2 \Lambda_{W_{c/4}}} C_{D_0}$ | 0.01454 |
| $C_{L_M}$ | $\frac{M_\infty \cos^2 \Lambda_{W_{c/4}}}{1 - M_\infty^2 \cos^2 \Lambda_{W_{c/4}}} C_{L_0}$ | 0.1976 |
| $C_{M_u}$ | Neglect | 0 |
| $C_{M_q}$ | $-\eta_H \frac{S_H}{S_W} C_{L_{\alpha_H}} \left(X_{AC_H} - X_{CG}\right)^2 \cos^2 \alpha_0 \frac{1}{U_0 \bar{c}_w}$ | -0.01548 |
| $C_{M_{\dot{\alpha}}}$ | $-\eta_H \frac{S_H}{S_W} C_{L_{\alpha_H}} (X_{AC_H} - X_{CG})(X_{AC_H} - X_{AC_W}) \cos^2 \alpha_0 \frac{d\varepsilon}{d\alpha} \frac{1}{U_0 \bar{c}_w}$ | -0.01037 |
| $X_u$ | $\frac{-\left(C_{D_u} + \frac{2}{U_0} C_{D_0}\right) q_\infty S_W}{m}$ | -0.01134 |
| $Z_u$ | $\frac{-\left(C_{L_u} + \frac{2}{U_0} C_{L_0}\right) q_\infty S_W}{m}$ | -0.1541 |
| $M_u$ | $\frac{\left(\frac{2}{U_0} C_{M_0}\right) q_\infty S_W \bar{c}_w}{I_{yy}}$ | -0.001729 |
| $X_\alpha$ | $\frac{(-C_{D_\alpha} + C_{L_0}) q_\infty S_W}{m}$ | 15.9862 |
| $Z_\alpha$ | $\frac{-(C_{L_\alpha} + C_{D_0}) q_\infty S_W}{m}$ | -179.9777 |
| $M_\alpha$ | $\frac{C_{M_\alpha} q_\infty S_W \bar{c}_w}{I_{yy}}$ | -5.8060 |
| $M_q$ | $\frac{C_{M_q} q_\infty S_W \bar{c}_w}{I_{yy}}$ | -0.2215 |
| $M_{\dot{\alpha}}$ | $\frac{C_{M_{\dot{\alpha}}} q_\infty S_W \bar{c}_w}{I_{yy}}$ | -0.1484 |

| | | |
|---|---|---|
| $C_{D_u}$ | $\frac{C_{D_M}}{a_0}$ | 1.5022 e -05 |
| $C_{L_u}$ | $\frac{C_{L_M}}{a_0}$ | 0.0002041 |

## 4.1 Analysis of the Longitudinal Derivatives

$$\mathbf{A}_{\text{mat}} = \begin{bmatrix} X_u & X_a & 0 & -32.2 \\ \frac{Z_u}{U_0} & \frac{Z_a}{U_0} & 1 & 0 \\ \frac{M_u + M_a \dot{Z}_u}{U_0} & \frac{M_a + M_a \dot{Z}_a}{U_0} & M_q + M_a \dot{Z}_a & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\begin{aligned} A &= \text{eig.real} \\ B &= \text{eig.imag} \\ \text{natural\_freq} &= \sqrt{A^2 + B^2} \\ \text{damping\_ratio} &= -\frac{A}{\text{natural\_freq}} \\ \text{time\_to\_half} &= \frac{\log 2}{|A|} \\ \text{time\_constant} &= \frac{1}{|A|} \\ \text{cycle\_to\_half} &= \left| \frac{\text{time\_to\_half}}{2\pi/B} \right| \end{aligned}$$

## 4.2 Explanation of Dynamic Parameters from Eigenvalues

The eigenvalues of the longitudinal derivatives matrix provide valuable information about the dynamic behavior of the system. From these eigenvalues ($A$ and $B$), several key parameters are derived:

### Natural Frequency ($\omega_n$)

The natural frequency is determined from the real and imaginary parts of the eigenvalues using the formula:

$$\text{natural\_freq} = \sqrt{A^2 + B^2}$$

It represents the rate at which the system oscillates without damping when disturbed.

### Damping Ratio ($\zeta$)

The damping ratio is calculated as the negative ratio of the real part to the natural frequency:

$$\text{damping\_ratio} = -\frac{A}{\text{natural\_freq}}$$

It provides information about the rate at which the system's amplitude decreases over time.

### Time to Half Amplitude ($t_{1/2}$)

The time to half amplitude is computed as the natural logarithm of 2 divided by the absolute value of the real part of the eigenvalue:

$$\text{time\_to\_half} = \frac{\log 2}{|A|}$$

It indicates the time it takes for the system's amplitude to decrease to half of its initial value.

### Time Constant ($\tau$)

The time constant is the reciprocal of the absolute value of the real part of the eigenvalue:

$$\text{time\_constant} = \frac{1}{|A|}$$

It represents the time required for the system's response to reach $1 - \frac{1}{e}$ (approximately 63.2%) of its final value.

### Cycles-to-Half ($n_{1/2}$)

Cycles-to-half is calculated by taking the absolute value of the time to half amplitude and dividing it by the period of oscillation:

$$\text{cycle\_to\_half} = \left| \frac{\text{time\_to\_half}}{2\pi/B} \right|$$

It gives the number of oscillations required for the system's amplitude to decrease to half.

These parameters provide insights into the dynamic behavior of the system, crucial for stability and control analysis in aerospace engineering.

## 4.3 Categorzing the Aircraft

By leveraging the content presented in the lecture slides, we gain the capability to systematically classify aircraft according to their performance and stability characteristics. The comprehensive insights provided in these educational materials allow us to analyze and categorize various aircraft models, taking into consideration factors such as aerodynamic performance, longitudinal and lateral stability, and other crucial parameters. This categorization is instrumental in enhancing our understanding of different aircraft types, their design principles, and their operational capabilities, thereby contributing to a more profound comprehension of aeronautical engineering principles and the broader field of aviation.

## 4.4 Task 2 Tables

*Table 3. Modal Characteristics (4x4 system, no approximations needed)*

| Mode | Natural Frequency (rad/s) | Damping Ratio | Damped Frequency (rad/s) | Time-to-half (s) | Cycles-to-half | Time Constant (s) |
|---|---|---|---|---|---|---|
| Phugoid | 0.1145 | 0.06523 | 0.1145 | 92.7219 | 1.6875 | 133.7695 |
| Short Period | 2.4237 | 0.1493 | 2.3965 | 1.9151 | 0.7304 | 2.7630 |

*Table 4. Flying Qualities (Airplane Class, Flight Phase)*

| Parameter | Value (units) | FQ Level |
|---|---|---|
| Phugoid damping ratio | 0.06523 | Level 1 Cat. B |
| Short Period damping ratio | 0.1493 | Level 3 Cat. B |
| Short Period Natural Frequency | 2.4237 (rad/s) | Level 1 Cat. B |
| $n / \alpha = -\frac{Z_\alpha}{g}$ | 5.5894 | Level 1 Cat. B |
| $CAP = \frac{\omega_{sp}^2}{n / \alpha}$ | 1.0276 | Level 3 Cat. C |

# 5 Task 3

In the process of determining the lateral-directional coefficients and derivatives for the aircraft, we relied on specific equations provided in the pertinent tables. These equations encapsulate crucial aerodynamic parameters and stability derivatives, allowing us to meticulously characterize the lateral and directional behavior of the aircraft. By extracting values from these tables using the specified equations, we gained insights into how the aircraft responds to lateral and directional perturbations, its inherent stability characteristics, and the influence of control surfaces on these dynamic aspects. This systematic approach enabled a comprehensive understanding of the aircraft's lateral-directional stability and control attributes, providing essential information for assessing its flying qualities and informing design considerations.

1. **Stability and Control Analysis:**

   - **Predicting Stability:** Lateral-directional stability coefficients provide information about the stability of the aircraft in roll and yaw. Positive stability ensures that the aircraft tends to return to its original state after a disturbance, while negative stability could lead to undesirable oscillations.
   - **Control Effectiveness:** The derivatives indicate how control surfaces (such as ailerons and rudders) affect the aircraft's motion. Understanding these effects is crucial for effective and predictable control.

2. **Pilot Handling Qualities:**

   - **Aircraft Response:** The lateral-directional coefficients influence how the aircraft responds to pilot inputs. A well-designed aircraft should respond intuitively to control inputs, allowing the pilot to easily and precisely maneuver the aircraft.
   - **Control Harmony:** The relationship between roll and yaw responses, as indicated by these coefficients, contributes to control harmony. A well-harmonized aircraft minimizes the need for constant pilot corrections to maintain stable flight.

3. **Aircraft Performance:**

   - **Dynamic Response:** Lateral-directional coefficients are fundamental to understanding the dynamic response of the aircraft during maneuvers. This includes roll rates, sideslip angles, and yaw rates.
   - **Adverse Yaw:** Coefficients related to adverse yaw (undesirable yaw induced by aileron deflections) influence the efficiency of turns and coordination between roll and yaw.

4. **Safety and Handling:**

   - **Spin and Stall Characteristics:** Stability derivatives are crucial for predicting an aircraft's spin and stall characteristics. Proper lateral-directional stability is essential for preventing dangerous and unrecoverable spins.
   - **Controllability at Extremes:** Understanding how the aircraft behaves at the edges of its flight envelope (such as high angles of attack or during asymmetrical thrust conditions) is vital for safety.

5. **Regulatory Compliance:**

   - **Certification Requirements:** Regulatory authorities often set specific criteria for lateral-directional stability and control parameters. Meeting these requirements is essential for obtaining certification for the aircraft.

In summary, quantifying lateral-directional stability coefficients and derivatives is vital for assessing the flying qualities of an aircraft. This information informs designers, pilots, and regulators about the aircraft's behavior in various flight conditions, contributing to the overall safety, controllability, and performance of the aircraft.

## 5.1 Task 3 Tables

*Table 5 . Lateral-Directional Stability Coefficients (Stability Axis System)*

| | Wing-Fuselage (inlet) | Vertical Tail | Horizontal Tail | Total |
|---|---|---|---|---|
| $C_{Y_\beta}$ | $-2A_{in}/S_W=$ -0.075 | $-kC_{L_{\alpha V}}\left(1+\frac{d\sigma}{d\beta}\right)\eta_V\left(\frac{S_V}{S_W}\right)=$ -0.5460 | Neglect = 0 | -0.6210 |
| $C_{L_\beta}$ | $-\left(C_{L_{\alpha W}}\Gamma_W + C_{L_W}\sin\left(2\Lambda_{LE_W}\right)\right)\bar{Y}_{AC_W}=$ -0.3661 | $C_{Y_{\beta_V}}\left(\frac{\Delta Z_{ACV_S}}{b_W}\right)=$ 0.034973 | $-\eta_H\frac{S_H}{S_W}\left(C_{L_{\alpha_H}}\Gamma_H + C_{L_H}\sin\left(2\Lambda_{LE_W}\right)\right)\bar{Y}_{AC_H}=$ 0.007855 | -0.3233 |
| $C_{N_\beta}$ | $-1.3\frac{volume}{S_W b_W}-\frac{2A_{in}X_{cg}}{S_W b_W}=$ -0.13900 | $-C_{Y_{\beta_V}}\left(\frac{\Delta X_{ACV_S}}{b_W}\right)=$ 0.1864 | Neglect = 0 | 0.04736 |
| $C_{Y_P}$ | Neglect= 0 | $C_{Y_{\beta_V}}\left(\frac{\Delta Z_{AC_{V_S}}}{V_\infty}\right)$ | Neglect = 0 | 0.002637 |
| $C_{L_p}$ | $-C_{L_{\alpha W}}\tilde{Y}_{AC_W}/(b_W V_\infty)=$ -0.01368 | Neglect= 0 | $-\eta_H\frac{S_H}{S_W b_W V_\infty}C_{L_{\alpha_H}}\tilde{Y}_{AC_H}=$ -0.0003512 | -0.01403 |
| $C_{N_P}$ | $-C_{L_W}\tilde{Y}_{AC_W}/(b_W V_\infty)=$ -0.0025377 | $-C_{Y_{\beta_V}}\left(\frac{\Delta X_{ACV_S}\Delta Z_{ACV_S}}{b_W V_\infty}\right)=$ -0.00089992 | Neglect = 0 | -0.003438 |
| $C_{Y_r}$ | Neglect = 0 | $-C_{Y_{\beta_V}}\left(\frac{\Delta X_{AC_{V_S}}}{V_\infty}\right)$ | Neglect = 0 | 0.01405 |
| $C_{L_r}$ | $2C_{L_W}\tilde{Y}_{AC_W}/(b_W V_\infty)=$ 0.005075 | $-C_{Y_{\beta_V}}\left(\frac{\Delta X_{ACV_S}\Delta Z_{ACV_S}}{b_W V_\infty}\right)=$ -0.0008999 | Neglect = 0 | 0.004175 |
| $C_{N_r}$ | Neglect = 0 | $C_{Y_{\beta_V}}\left(\frac{\Delta X_{AC_{V_S}}^2}{b_W V_\infty}\right)$ | Neglect = 0 | -0.004796 |

Additional Calculations for above:

| Formulas | Value |
|---|---|
| $\Delta X_{AC_{V_S}} = \left(X_{AC_V} - X_{CG}\right)\cos\alpha_0 + (Z_{AC_V} - Z_{CG})\sin\alpha_0$ | 12.9525 |
| $\Delta Z_{AC_{V_S}} = -\left(X_{AC_V} - X_{CG}\right)\sin\alpha_0 + (Z_{AC_V} - Z_{CG})\cos\alpha_0$ | -2.4306 |
| $Y_{AC_W} = \frac{2}{S_W}\int_0^{b_W/2} yc(y)dy = \frac{b_W}{2}\left[\frac{1+2\lambda}{3(1+\lambda)}\right]$ (first moment) | 7.5521 |
| $Y_{AC_H} = \frac{2}{S_H}\int_0^{b_h/2} yc(y)dy = \frac{b_h}{2}\left[\frac{1+2\lambda}{3(1+\lambda)}\right]$ (first moment) | 3.3130 |
| $\tilde{Y}_{AC_W} = \frac{2}{S_W}\int_0^{b_W/2} y^2c(y)dy = \frac{b_W^2}{24}\left[\frac{1+3\lambda}{1+\lambda}\right]$ (first moment) | 83.2906 |
| $\tilde{Y}_{AC_H} = \frac{2}{S_H}\int_0^{b_h/2} y^2c(y)dy = \frac{b_h^2}{24}\left[\frac{1+3\lambda}{1+\lambda}\right]$ (first moment) | 16.3777 |
| $C_{LW} = C_{L_{\alpha_W}}\alpha$ | 0.5820 |
| $C_{LH} = C_{L_{\alpha_H}}\alpha$ | -0.12884 |

*Table 6. Dimensional Stability Derivatives (moment derivatives are primed)*

| Stability Derivative | Equation used | Value (units) |
|---|---|---|
| $Y_\beta$ | $Y_\beta = \frac{C_{Y_\beta} q_\infty S_W}{m}$ | -33.9911 ($\frac{ft}{rad-(sec)^2}$) |
| $Y_p$ | $Y_p = \frac{C_{Y_p} q_\infty S_W}{m}$ | 0.1443 ($\frac{ft}{rad-sec}$) |
| $Y_r$ | $Y_r = \frac{C_{Y_r} q_\infty S_W}{m}$ | 0.7690 ($\frac{ft}{rad-sec}$) |
| $L'_\beta$ | $\frac{(L_\beta + \frac{I_{xz}}{I_{xx}} N_\beta)}{D}$ | -63.8705 ($1/sec^2$) |
| $L'_p$ | $\frac{(L_p + \frac{I_{xz}}{I_{xx}} N_p)}{D}$ | -2.6414 (1/sec) |
| $L'_r$ | $\frac{(L_r + \frac{I_{xz}}{I_{xx}} N_r)}{D}$ | 0.9248 (1/sec) |
| $N'_\beta$ | $\frac{(N_\beta + \frac{I_{xz}}{I_{xx}} L_\beta)}{D}$ | 9.7000 ($1/sec^2$) |
| $N'_p$ | $\frac{(N_p + \frac{I_{xz}}{I_{xx}} L_p)}{D}$ | 0.1904 (1/sec) |
| $N'_r$ | $\frac{(N_r + \frac{I_{xz}}{I_{xx}} L_r)}{D}$ | -0.3009 (1/sec) |

Additional Calculations for above:

| Formulas | Value |
|---|---|
| $L_\beta = \frac{C_{L_\beta} q_\infty S_W b_W}{I_{XX}}$ | -58.3568 ($1/sec^2$) |
| $L_p = \frac{C_{L_p} q_\infty S_W b_W}{I_{XX}}$ | -2.5331 ($1/sec^2$) |
| $L_r = \frac{C_{L_r} q_\infty S_W b_W}{I_{XX}}$ | 0.7538 ($1/sec^2$) |
| $N_\beta = \frac{C_{N_\beta} q_\infty S_W b_W}{I_{ZZ}}$ | 1.8488 ($1/sec^2$) |
| $N_p = \frac{C_{N_p} q_\infty S_W b_W}{I_{ZZ}}$ | -0.1342 ($1/sec^2$) |
| $N_r = \frac{C_{N_r} q_\infty S_W b_W}{I_{zz}}$ | -0.1872 ($1/sec^2$) |
| $D = 1 - (\frac{I_{xz}^2}{I_{xx} I_{zz}})$ | 0.9301 |
| $I_{stab}$ | $\begin{bmatrix} 7860.83 & 0 & 4468.18 \\ 0 & 31000 & 0 \\ 4468.18 & 0 & 36349.17 \end{bmatrix}$ slugs-$ft^2$ |

*Table 7. Lateral-Directional Modal Characteristics*

| Mode | Natural Frequency (rad/s) | Damping Ratio | Damped Frequency (rad/s) | Time-to-half (s) | Cycles-to-half | Time Constant (s) |
|---|---|---|---|---|---|---|
| Spiral | 0.0381 | 1.0 | 0.0 | 18.1516 | 0.0 | 26.1873 |
| Roll | 2.0226 | 1.0 | 0.0 | 0.3426 | 0.0 | 0.4943 |
| Dutch Roll | 2.9134 | 0.1628 | -2.8745 | 1.4607 | 0.6682 | 2.1074 |

*Table 8. Flying Qualities (Airplane Class, Flight Phase)*

| Mode | Criterion | Criterion Value | Flying Quality Level |
|---|---|---|---|
| Spiral Mode | Time to double | 18.1516 (s) | Level 1 |
| Roll Mode | Time constant | 0.4943 (s) | Level 1 |
| Dutch Roll | Damping ratio | 0.1628 | Level 1 Cat. B |
| | Product $\varsigma\omega_n$ | 0.4738 | Level 1 |
| | Natural Frequency | 2.9134 (rad/s) | Level 1 Cat. A |

# 6 Disscusion

Upon meticulous examination of the aerodynamic tables, it becomes evident that the aircraft falls within Category C in terms of its flying characteristics. This classification signifies a commendable level of stability, suggesting that the aircraft possesses the desirable trait of returning to its original state after encountering disturbances. The data from parameters such as longitudinal or lateral stability derivatives aligns with expectations for an aircraft in this category, reinforcing its predictable and manageable flight behavior.

It is noteworthy that the characterization goes beyond a mere classification of stability and introduces a nuanced perspective. The aircraft's stability is described as judiciously balanced within Category C – a classification denoting stability that is not excessively robust. This nuanced stability implies a well-thought-out equilibrium, allowing the aircraft to remain responsive to control inputs without presenting undue challenges for the pilot. The deliberate balance struck by the aircraft, falling within the confines of Category C, highlights its aptitude for maintaining a harmonious relationship between stability and maneuverability. This characteristic ensures a flying experience that is both stable and responsive, contributing to an aircraft that is well-suited for a diverse range of flight conditions.

Improving flight characteristics involves meticulous considerations of the aircraft's balance, stability, and control. The following suggestions outline strategies for optimizing the center of gravity (CG) relative to the aerodynamic center (AC) and adjusting the surface area ratio between the horizontal tail and the wing:

### 1. Center of Gravity (CG) Adjustment:

- **Optimal CG Range:** Ensure the aircraft operates within the optimal CG range specified by the design for stable flight.
- **Weight Distribution:** Evaluate the distribution of fuel, payload, and components for a balanced weight distribution.
- **Load Management:** Consider redistributing loads or reorganizing equipment to optimize the CG position for both longitudinal and lateral stability.

### 2. Surface Area Ratio Adjustment:

- **Horizontal Tail Size:**
  - **Increase Tail Area:** Enlarging the horizontal tail surface area enhances stability, especially in pitch, improving pitch response.
  - **Adjust Tail Span:** Modifying the span of the horizontal tail can influence lateral stability.
- **Wing Area:**
  - **Optimal Wing Area:** Ensure the wing area is appropriately sized for the intended mission, avoiding extremes that affect stability.
  - **Winglets:** Consider adding or modifying winglets to impact aerodynamics and lateral stability.

### 3. Combined Adjustments:

- **Dynamic Modeling:** Utilize computational tools for dynamic modeling and simulation to predict the effects of CG and surface area ratio adjustments.
- **Wind Tunnel Testing:** Conduct wind tunnel testing to validate and refine proposed modifications, providing empirical data on aerodynamic forces.

# 7 Code

*italicized text*# Normal *italicized text*

```
######################################## UNCHANGED TASK 1 CODE ################################################
from scipy.optimize import fsolve
import numpy as np
from sympy import (symbols, sin, cos, solve, pi)
import math
a_0 = symbols('a_0')
T_0 = symbols('T_0')
i_h = symbols('i_h')
#Variables

W = 22000 #lb
A_in = 15 # sqft

cw_w=11.86
cw_h=6.71
cw_v=7.32

X_ac_wf = 28.0352 #ft
X_ac_h = 42.2825 #ft
X_ac_w = 28.628 #ft
X_cg = 27 #ft

S_H = 100 #ft^2
S_W = 400 #ft^2
A_in = 15 #ft^2

n_H = 0.5
rho=7.38*(10**-4) # slug/ft^3
q_inf = (1/2) * rho * (503.36 ** 2) #whatever units pressure is
#print(q_inf)
q_H = n_H * q_inf
vv=503.36 #ft/s

CL_a_wf = 3.1373
CL_a_H = 3.277
CL_a_W = 3.1373 #5.7875

de_da = 0.75

# Function representing the system of equations

def equations(variables):
    a_0, i_h, T_0 = variables

    e = de_da * a_0
    a_H = a_0 + i_h - e
    L_wf = CL_a_W * a_0 * q_inf * S_W #CL_a_wf is the same as CL_a_w
    L_h = CL_a_H * a_H * q_H * S_H
    C_D0 = 0.0145 + 0.1*((CL_a_W*a_0)**2)
    D_0 = q_inf * S_W * C_D0
    F_N0 = 2 * q_inf * A_in * (np.cos(a_0)**2) * np.sin(a_0)

    # Trim equations
    #X = T_0 - W*np.sin(a_0) + L_wf * np.sin(a_0) - D_0 *np.cos(a_0) + L_h*np.sin(a_0 - e)
    X = T_0 * np.cos(a_0) - F_N0 * np.sin(a_0)- D_0 - L_h * np.sin(e)

    #Z = W*np.cos(a_0) - F_N0 - L_wf * np.cos(a_0) - D_0 * np.sin(a_0) - L_h* np.cos(a_0 - e)
    Z = W - T_0 * np.sin(a_0) - F_N0 * np.cos(a_0) - L_wf - L_h* np.cos(e)

    #M = F_N0 * (X_cg - 0) - L_wf * (X_ac_wf - X_cg)*np.cos(a_0) - D_0 * (X_ac_w - X_cg) * np.sin(a_0) - L_h * (X_ac_h - X_cg) * np.cos(a_0 -
    M = F_N0 * (X_cg - 0) - L_wf * (X_ac_wf - X_cg)*np.cos(a_0) - D_0 * (X_ac_w - X_cg) * np.sin(a_0) - L_h * (X_ac_h - X_cg) * np.cos(a_0 -

    return [X, Z, M]

# Initial guess for the variables
initial_guess = [np.radians(10), np.radians(-5), 2000]  # Provide your initial guess here
radtodeg = 180/np.pi
# Solve the equations
solution = fsolve(equations, initial_guess)

print("Alpha trim is ", solution[0]* radtodeg, '\n')
print("Incidence Angle is ", (solution[1]) * radtodeg, "\n")
print("Trim Thrust is ", solution[2], "\n")
```

```
alpha=solution[0]
i_h=solution[1]
thrust=solution[2]
```

```
Alpha trim is  10.628368614640332

Incidence Angle is  -4.909762693703019

Trim Thrust is  1849.2467118310285
```

```
#Most Forward CG
while X_cg != 0.0:
   # print(f'i_h = {fsolve(equations,initial_guess)[1]*radtodeg}')
   if abs(abs(fsolve(equations,initial_guess)[1]*radtodeg) - 20) < 0.1:
      print(f'forward most xcg = {X_cg}')
      break
   X_cg -= .0001
X_cg = 27
```

```
forward most xcg = 24.323900000006237
```

*Task 1 of project* ✍

```
#Rest of Table 1
alpha = solution[0]
e = de_da * alpha
AR_w=3.6
M=0.52
beta=math.sqrt(1-M**2)

c_la=7.35593
k=c_la/(2*math.pi)

half_chord=39.69

Normal_force = 2 * q_inf * A_in * np.cos(alpha) ** 2 * np.sin(alpha)
print("The normal force is", Normal_force, "\n")

C_D0 = 0.0145 + 0.1*((CL_a_W * alpha)**2)
print("C_D0", C_D0, "\n")

D_0 = q_inf * S_W * C_D0

C_L0 = CL_a_wf*alpha + n_H*(S_H/S_W)*CL_a_H*alpha*np.cos(e)
print("C_L0", C_L0, "\n")

prop_CP_M0=(Normal_force*(X_cg-0))/(q_inf*cw_w*S_W)
print("Propulsive force", prop_CP_M0, "\n")

#work=math.sqrt(((3.6**2 * beta**2) /k**2) * (1+((math.tan(math.radians(half_chord))**2)/beta**2))+4)
#CL_alpha=(2*math.pi*3.6)/(2 + work)
CL_alpha = CL_a_wf + CL_a_H * n_H * (S_H / S_W) * (1 - de_da)
print("Big CL_a", CL_alpha, "\n")

deltaX_ach = (X_ac_h - X_cg)/ cw_w
deltaX_acwf = (X_cg - X_ac_wf)/ cw_w

CM_a = CL_a_wf*deltaX_acwf - CL_a_H * n_H * (S_H/S_W) * deltaX_ach * (1- de_da)
print("C_Ma = ", CM_a,"\n")

CM_ih = -CL_a_H*n_H * (S_H/S_W) * deltaX_ach
print("CM_ih = ", CM_ih,"\n")

C_Da=0.2*CL_a_W**2*alpha
print("C_Da", C_Da, "\n")

#X_bar_np=(CL_a_wf * (X_ac_wf/cw_w ) + CL_a_H*n_H*(S_H/S_W)*(X_ac_h/cw_w)*(1-de_da))/CL_alpha
X_bar_np=(CL_a_wf * X_ac_wf + CL_a_H*n_H*(S_H/S_W)*X_ac_h*(1-de_da))/CL_alpha
print("X_bar_np",X_bar_np, "\n")

SM=X_bar_np - X_cg#-CM_a/CL_alpha
print("SM", SM, "\n")

CM_ih=-CL_a_H*n_H*((S_H/S_W) * deltaX_ach)
print("CM_ih", CM_ih,"\n")

CL_ih=CL_a_H*n_H*(S_H/S_W)
print("CL_ih", CL_ih, "\n")
```

```
The normal force is 499.7177220566599

C_D0 0.0483688188104683

C_L0 0.6572204881969858

Propulsive force 0.03042006007273037

Big CL_a 3.2397062500000002

C_Ma =  -0.40579734195826306

CM_ih =  -0.5278325516441822

C_Da 0.3651624149091098

X_bar_np 28.485553350901796

SM 1.485553350901796

CM_ih -0.5278325516441822
```

```
CL_ih 0.409625
```

*Table 2 ;0*

```
# Task 2
import math
import numpy

q_c_W=np.radians(45)
q_c_H=np.radians(45)
q_c_V=np.radians(45)
M=0.52
U_0=503.36 # ft/s
C_Mu=0
I_xx=7210 # slug-ft^2
I_yy=31000 # slug-ft^2
I_zz=37000 # slug-ft^2
I_xz=1000 # slug-ft^2
m=22000/32.2
sos = 968 #speed of sound at cruising altitude (ft/s)

C_DM= ((M*math.cos(q_c_W)**2)/(1 - (M**2) * math.cos(q_c_W)**2)) * C_D0
print("C_DM", C_DM, "\n")

C_LM=((M*math.cos(q_c_W)**2)/(1 - (M**2) *math.cos(q_c_W)**2)) * C_L0
print("C_LM", C_LM, "\n")

C_Mu=0
print("C_Mu", C_Mu, "\n")

C_Mq= -n_H * CL_a_H* (S_H/S_W) * (X_ac_h-X_cg)**2 * (math.cos(alpha)**2) *(1/(U_0*cw_w))
print("C_Mq", C_Mq, "\n")

C_Ma_dot= -n_H * (S_H/S_W) * CL_a_H * (X_ac_h-X_cg) * (X_ac_h-X_ac_w) * (math.cos(alpha)**2) * de_da*(1/(U_0*cw_w))
print("C_Ma_dot", C_Ma_dot,"\n")


C_Du = C_DM / sos
print("C_Du", C_Du, "\n")
X_u=-((C_Du+(2/U_0)*C_D0)*q_inf*S_W)/m
#X_u = (-q_inf * S_W * (2*C_D0)) / (m*U_0)  #Maybe this equation
print("X_u", X_u, "\n")

C_Lu = C_LM / sos
print("C_Lu", C_Lu, "\n")

Z_u=-((C_Lu + (2/U_0) * C_L0)* q_inf * S_W)/m
#Z_u = (-q_inf * S_W * (2*C_L0)) / (m*U_0)
print("Z_u", Z_u, "\n")

M_u=((2/U_0)*-prop_CP_M0*q_inf*S_W*cw_w)/I_yy
print("M_u", M_u, "\n")

X_a=((-C_Da+C_L0)*q_inf*S_W)/m
print("X_a", X_a, "\n")

Z_a=-((CL_alpha+C_D0)*q_inf*S_W)/m
print("Z_a", Z_a, "\n")

M_a=(CM_a*q_inf*S_W*cw_w)/I_yy
print("M_a", M_a, "\n")

M_q=(C_Mq*q_inf*S_W*cw_w)/I_yy
print("M_q", M_q, "\n")

M_a_dot=(C_Ma_dot*q_inf*S_W*cw_w)/I_yy
print("M_a_dot", M_a_dot, "\n")
```

```
C_DM 0.014541966802407215

C_LM 0.19759172864386718

C_Mu 0

C_Mq -0.015480364668898882
```

```
C_Ma_dot -0.010373465043586445

C_Du 1.5022692977693404e-05

X_u -0.01134175628262406

C_Lu 0.00020412368661556528

Z_u -0.15410826198352734

M_u -0.001729330959888878

X_a 15.98623401381166

Z_a -179.97769044094454

M_a -5.805985576082417

M_q -0.2214868474652724

M_a_dot -0.14841937634784372
```

*Table 3* ✳

```
# Task 3
import numpy as np
import math
from numpy.linalg import eig

A_mat=np.array([[X_u, X_a, 0, -32.2],
                [(Z_u / U_0), (Z_a / U_0), 1, 0],
                [M_u + (M_a_dot * Z_u) / U_0, M_a + (M_a_dot * Z_a) / U_0, (M_q + M_a_dot), 0],
                [0, 0, 1, 0]])

def values(eig):
    A = eig.real
    B = eig.imag
    natural_freq = np.sqrt(A**2 + B**2)
    damping_ratio = -A / natural_freq
    time_to_half = np.log(2) / np.abs(A)
    time_constant = 1/ np.abs(A)
    cycle_to_half = abs(time_to_half / (2*np.pi / B))

    return     print("Real", A, "\nImaginary", B, '\nDamping Ratio =', damping_ratio , "\n Natural Frequency =", natural_freq,
            "\n Time to half", time_to_half,"\n Time Constant =", time_constant,
            "\n Cycles-to-half =", cycle_to_half)

w,v=eig(A_mat)
print("Exact Eigen values \n", w)
print("")
print("Exact Eigen vector \n", v)
print("")
print("Phugoid ", values(w[3]))
print('')
print("Short Period", values(w[1]))
```

```
Exact Eigen values
 [-0.36629098+2.39859519j -0.36629098-2.39859519j -0.00310932+0.07956431j
 -0.00310932-0.07956431j]

Exact Eigen vector
 [[ 9.23303364e-01+0.00000000e+00j  9.23303364e-01-0.00000000e+00j
   9.99996796e-01+0.00000000e+00j  9.99996796e-01-0.00000000e+00j]
 [ 5.89619121e-02-1.24817632e-01j  5.89619121e-02+1.24817632e-01j
  -3.05661653e-04-1.16034432e-06j -3.05661653e-04+1.16034432e-06j]
 [ 2.99154419e-01+1.42516459e-01j  2.99154419e-01-1.42516459e-01j
   1.97910749e-04-2.47310359e-05j  1.97910749e-04+2.47310359e-05j]
 [ 3.94504577e-02-1.30745183e-01j  3.94504577e-02+1.30745183e-01j
  -4.07415745e-04-2.47150957e-03j -4.07415745e-04+2.47150957e-03j]]

Real -0.003109321640247104
Imaginary -0.07956431356678743
Damping Ratio = 0.039049543015698296
 Natural Frequency = 0.07962504552222613
 Time to half 222.92553191919365
 Time Constant = 321.613559387355
```

```
 Cycles-to-half = 2.8229180036110377
Phugoid  None

Real -0.3662909820210849
Imaginary -2.3985951879302982
Damping Ratio = 0.15096053406814078
 Natural Frequency = 2.4264022665403924
 Time to half 1.8923402829503606
 Time Constant = 2.7300699418869034
 Cycles-to-half = 0.7223976493936727
Short Period None
```

*Table 4* ✈

```
###### Table 4 ########## added in portions
g=32.2

na=-(Z_a/g)
print("n/alpha is ", na, "\n")


CAP=(2.3965347938016235**2)/na
print("CAP is", CAP, "\n")
```

```
n/alpha is  5.589369268352314

CAP is 1.0275540480786443
```

*Table 5* ⛄

```
###### Table 5 ##########
# Lateral-Directional Stability Coefficients (Stability Axis System)
# b=beta

AR_V=1.5
do_db=0.12
S_V=70
n_V=1
chord_half_V=math.radians(34.64)
k=1

C_YbWF=-(2*A_in)/S_W
print("C_YbWF is", C_YbWF,"\n")

AR_Veff=1.55*AR_V
print("AR_Veff is", AR_Veff, "\n")

C_LaV=(2*math.pi*AR_Veff)/(2+math.sqrt(((AR_Veff**2*beta**2)/k**2)*(1+math.tan(chord_half_V)**2/beta**2)+4))
print("C_LaV is", C_LaV, "\n")

C_YbV=-k*C_LaV*(1+do_db)*n_V*(S_V/S_W)
print("C_YbV is", C_YbV, "\n")

C_Yb = C_YbWF + C_YbV
print("********C_Yb is***********", C_Yb, "\n")

# C_Lbeta
chord_LEW=math.radians(49.47)

b_w=37.947
lam_w=0.24085
b_h=18.439
lam_h=0.084647
Y_sq_ACW=((b_w**2)/24)*((1+3*lam_w)/((1+lam_w)))
Y_sq_ACH=((b_h**2)/24)*((1+3*lam_h)/((1+lam_h)))
print("Y_sqiggle_ACW", Y_sq_ACW, "\n")
print("Y_sqiggle_ACH", Y_sq_ACH, "\n")

Y_ACW=(b_w/2)*((1+2*lam_w)/(3*(1+lam_w)))
print("Y_ACW", Y_ACW, "\n")

Y_ACH=(b_h/2)*((1+2*lam_h)/(3*(1+lam_h)))
print("Y_ACH", Y_ACH, "\n")

X_LEmac=2.8362
X_APPV=35
X_ACV=X_APPV + X_LEmac + 0.32*7.32
Z_cg=0
Z_ACV=0
delta_Z_ACVS=-(X_ACV - X_cg)*math.sin(alpha)
print("Delta_Z_ACVs", delta_Z_ACVS, "\n")

delta_X_ACVs=(X_ACV - X_cg)*math.cos(alpha)
print("Delta_X_ACVs", delta_X_ACVs, "\n")

C_YP=C_YbV*(delta_Z_ACVS/U_0)
print("*************C_YP***************", C_YP, "\n")

C_Yr=-C_YbV*(delta_X_ACVs/U_0)
print("**************C_Yr**************", C_Yr, "\n")

C_Nr=C_YbV*((delta_X_ACVs**2)/(b_w*U_0))
print("**************C_Nr**************", C_Nr, "\n")

vol=1000
C_NbWF=-1.3*(vol/(S_W*b_w))-(2*A_in*X_cg)/(S_W*b_w)
print("C_NbWF", C_NbWF, "\n")

C_NbV=-C_YbV*(delta_X_ACVs/b_w)
print("C_NbV", C_NbV, "\n")

C_Nb=C_NbWF + C_NbV
print("**************C_Nb**************", C_Nb, "\n")

C_LW = CL_a_W * alpha
print("C_LW", C_LW, "\n")
```

```
C_LbWF = -C_LW*math.sin(2*chord_LEW)*(Y_ACW/cw_w)
print("C_LbWF", C_LbWF,"\n")

C_LbV = C_YbV*(delta_Z_ACVS/b_w)
print("C_LbV", C_LbV, "\n")

alpha_H = alpha + i_h - (de_da*alpha)
print("Alpha H", np.degrees(alpha_H), "\n")

C_LH=CL_a_H*alpha_H
print("C_LH", C_LH, "\n")

C_LbH = -n_H*(S_H/S_W)*(C_LH*math.sin(2*chord_LEW))*(Y_ACH/cw_h)
print("C_LbH", C_LbH, "\n")

C_Lb = C_LbWF + C_LbV + C_LbH
print("*************C_Lb**************", C_Lb, "\n")


C_LPWF = -((CL_a_W*Y_sq_ACW)/(b_w*U_0))
print("C_LPWF", C_LPWF,"\n")


C_LPV = -n_H*(S_H/(S_W*b_w*U_0))*CL_a_H*Y_sq_ACH
print("C_LPV", C_LPV, "\n")

C_LP = C_LPWF + C_LPV
print("*************C_LP**************", C_LP, "\n")

C_NPWF = -(C_LW * Y_sq_ACW)/(b_w*U_0)
print("C_NPWF", C_NPWF, "\n")

C_NPH = -C_YbV*((delta_X_ACVs*delta_Z_ACVS)/(b_w*U_0))
print("C_NPH", C_NPH, "\n")

C_NP = C_NPWF + C_NPH
print("*************C_NP**************", C_NP, "\n")

C_LrWF = (2*C_LW*Y_sq_ACW)/(b_w*U_0)
print("C_LrWF", C_LrWF, "\n")

C_LrV = -C_YbV*((delta_X_ACVs*delta_Z_ACVS)/(b_w*U_0))
print("C_LrV", C_LrV, "\n")

C_Lr = C_LrWF + C_LrV
print("*************C_Lr**************", C_Lr, "\n")
```

```
********C_Yb is*********** -0.6209943855396328

Y_sqiggle_ACW 83.29064106737823

Y_sqiggle_ACH 16.377672041666592

Y_ACW 7.552090623363019

Y_ACH 3.3129998463401766

Delta_Z_ACVs -2.4306346877779736

Delta_X_ACVs 12.95250991022861

*************C_YP*************** 0.0026365084488728786

**************C_Yr************** 0.014049582186966137

**************C_Nr************** -0.004795566250592938

C_NbWF -0.1390096713837721

C_NbV 0.18636513267534388
```

```
Alpha H -2.252670540042936

C_LH -0.12884022911382453

C_LbH 0.007855107122223752

**************C_Lb************** -0.3232506090584371

C_LPWF -0.013680315269068656

C_LPV -0.0003512226184870033

**************C_LP************** -0.01403153788755566

C_NPWF -0.00253769884413485

C_NPH -0.0008999236253834932

**************C_NP************** -0.003437622469518343

C_LrWF 0.0050753976882697

C_LrV -0.0008999236253834932

**************C_Lr************** 0.004175474062886207
```

*Table 7 ;D*

```
# Table 6
# b=beta

Y_b=(C_Yb*q_inf*S_W)/m
print("**************Y_b**************", Y_b, "\n")

Y_P=(C_YP*q_inf*S_W)/m
print("**************Y_P**************", Y_P, "\n")

Y_r=(C_Yr*q_inf*S_W)/m
print("**************Y_r**************", Y_r, "\n")



I_mat = np.array([[7210, 0, -1000],
                 [0, 31000, 0],
                 [-1000, 0, 37000]])

b_to_s = np.array([[np.cos(alpha), 0, np.sin(alpha)],
                  [0, 1, 0],
                  [-np.sin(alpha), 0, np.cos(alpha)]])
I_mat_stab = b_to_s.dot(I_mat.dot(np.transpose(b_to_s)))

print(I_mat_stab)
Is_xx = I_mat_stab[0, 0]
Is_yy = I_mat_stab[1, 1]
Is_zz = I_mat_stab[2, 2]
Is_xz = I_mat_stab[0, 2] * (-1)
```