Homework 1 — EMCH 721; Aeroelasticity

J.C. Vaught

Date: September 22nd, 2025 (at the last minute)

Contents

1 Problem A

1.1 a. Equations of Motion

Using Newton's second law for translation, the sum of forces in the y-direction is given by $\sum F_y = m\ddot{u}_y$, where $\ddot{u}_y$ represents the acceleration in the y-direction. For rotational motion, the sum of moments about point P is $\sum M_P = I_o\ddot{\theta} + m\ddot{u}_y d$, where I_o is the moment of inertia about the center of mass, $\ddot{\theta}$ is the angular acceleration, and d is the perpendicular distance from the center of mass to point P.

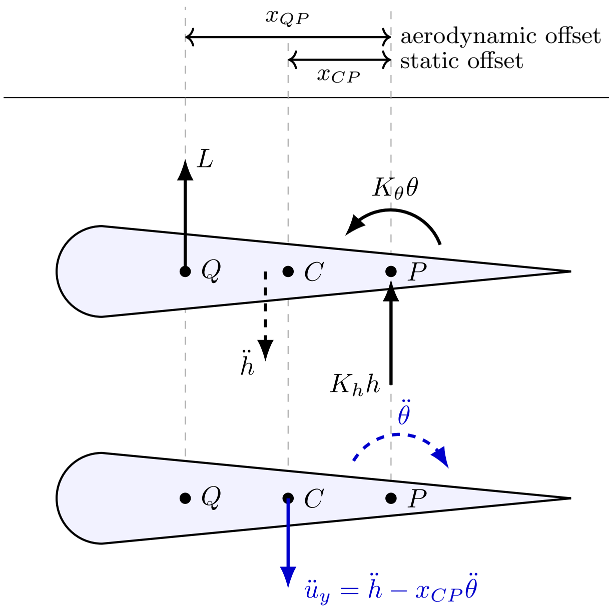


Figure 1: Force and Moment Diagrams

The key equations are derived as follows:

$$\sum F_y = -L - K_h h \qquad \text{(Sum of vertical forces)} \tag{1}$$

$$m\ddot{u}_y = m(\ddot{h} - x_{cp}\ddot{\theta}) \qquad \text{(Substitution for } m\ddot{u}_y\text{)} \tag{2}$$

$$L + K_h h = -m\ddot{h} + m x_{cp}\ddot{\theta} \qquad \text{(Combined)} \tag{3}$$

$$L + K_h h + m\ddot{h} - m x_{cp}\ddot{\theta} = 0 \qquad \text{(Equation of Motion – Plunge)} \tag{4}$$

$$\sum M_P = L x_{qp} - K_\theta(\theta) \qquad \text{(Sum of moments about } P\text{)} \tag{5}$$

$$I_o\ddot{\theta} - m\ddot{u}_y d = I_o\ddot{\theta} - m(\ddot{h} - x_{cp}\ddot{\theta})x_{cp} \qquad \text{(Moment of inertia relation)} \tag{6}$$

$$L x_{qp} - K_\theta(\theta) = I_o\ddot{\theta} - m(\ddot{h} - x_{cp}\ddot{\theta})x_{cp} \qquad \text{(Combined moment equation)} \tag{7}$$

$$L x_{qp} - K_\theta(\theta) = I_o\ddot{\theta} - m\ddot{h}x_{cp} + m x_{cp}^2\ddot{\theta} \qquad \text{(Distribute the term } -m x_{cp}\text{)} \tag{8}$$

$$L x_{qp} - K_\theta(\theta) = (I_o + m x_{cp}^2)\ddot{\theta} - m\ddot{h}x_{cp} \qquad \text{(Group terms containing } \ddot{\theta}\text{)} \tag{9}$$

$$L x_{qp} - K_\theta(\theta) = I_p\ddot{\theta} - m\ddot{h}x_{cp} \qquad \text{(Apply parallel axis theorem: } I_p = I_o + m x_{cp}^2\text{)} \tag{10}$$

$$0 = -L x_{qp} + K_\theta(\theta) + I_p\ddot{\theta} - m\ddot{h}x_{cp} \qquad \text{(Equation of Motion – Pitch)} \tag{11}$$

1.2 b. Linearity of Lift, $L(U, \theta) = L_0(U) \cdot \theta$

Given the expressions (proof & ***assumptions*** in APPENDIX B):

- $L = q \cdot C_L$, where q is the dynamic pressure.
- $C_L = a_1\theta$.

Substitute to obtain:

$$L(U, \theta) = q \cdot a_1 \cdot \theta = \frac{1}{2}\rho U^2 \cdot a_1 \cdot \theta \tag{12}$$

Assuming U and a_1 are constants, define the lift per unit angle of attack as $L_0(U) = \frac{1}{2}\rho U^2 \cdot a_1$. Thus:

$$L(U, \theta) = L_0(U) \cdot \theta \tag{13}$$

Applying the linear relationship for lift, $L(U, \theta) = L_0(U) \cdot \theta$, to the equations of motion results in a system of two linear ordinary differential equations.

$$L + K_h h + m\ddot{h} - m x_{cp}\ddot{\theta} = 0 \qquad \text{(Original vertical force equation)} \tag{14}$$

$$L_0(U)\theta + K_h h + m\ddot{h} - m x_{cp}\ddot{\theta} = 0 \qquad \text{(Substitute } L = L_0(U) \cdot \theta\text{)} \tag{15}$$

Then, solving likewise for Pitch:

$$-L x_{qp} + K_\theta \theta + I_p\ddot{\theta} - m\ddot{h} x_{cp} = 0 \qquad \text{(Original moment equation)} \tag{16}$$

$$-L_0(U)\theta x_{qp} + K_\theta \theta + I_p\ddot{\theta} - m\ddot{h} x_{cp} = 0 \qquad \text{(Substitute } L = L_0(U) \cdot \theta\text{)} \tag{17}$$

1.3 c. Equations of Motion in Terms of Natural Frequencies

From Appendix A,

$$\omega_h^2 = \frac{K_h}{m} \quad \text{and} \quad \omega_\theta^2 = \frac{K_\theta}{I_o} \tag{18}$$

The derivation begins with the rearranged vertical force (plunge) equation.

$$m\ddot{h} - mx_{cp}\ddot{\theta} + K_h h + L_0(U)\theta = 0 \qquad \text{(Rearranged from original)} \tag{19}$$

$$\frac{m\ddot{h}}{m} - \frac{mx_{cp}\ddot{\theta}}{m} + \frac{K_h}{m}h + \frac{L_0(U)}{m}\theta = 0 \qquad \text{(Divide by } m\text{)} \tag{20}$$

$$\ddot{h} - x_{cp}\ddot{\theta} + \left(\frac{K_h}{m}\right)h + \frac{L_0(U)}{m}\theta = 0 \qquad \text{(Simplify)} \tag{21}$$

$$\ddot{h} - x_{cp}\ddot{\theta} + \omega_h^2 h + \frac{L_0(U)}{m}\theta = 0 \qquad \text{(Substitute } \omega_h^2 = K_h/m\text{)} \tag{22}$$

The same procedure is applied to the rearranged moment (pitch) equation.

$$I_P\ddot{\theta} - mx_{cp}\ddot{h} + K_\theta\theta - L_0(U)x_{qp}\theta = 0 \qquad \text{(Rearranged from original)} \tag{23}$$

$$\frac{I_P\ddot{\theta}}{I_o} - \frac{mx_{cp}}{I_o}\ddot{h} + \frac{K_\theta}{I_o}\theta - \frac{L_0(U)x_{qp}}{I_o}\theta = 0 \qquad \text{(Divide by } I_o\text{)} \tag{24}$$

$$\frac{I_P\ddot{\theta}}{I_o} - \frac{mx_{cp}}{I_o}\ddot{h} + \left(\frac{K_\theta}{I_o}\right)\theta - \frac{L_0(U)x_{qp}}{I_o}\theta = 0 \qquad \text{(Simplify)} \tag{25}$$

$$\frac{I_P\ddot{\theta}}{I_o} - \frac{mx_{cp}\ddot{h}}{I_o} + \omega_\theta^2\theta - \frac{L_0(U)x_{qp}}{I_o}\theta = 0 \qquad \text{(Substitute } \omega_\theta^2 = K_\theta/I_o\text{)} \tag{26}$$

1.4 d. EigenValue Matrices

For a linear, homogeneous system of differential equations with constant coefficients, solutions can be expressed as linear combinations of exponential functions. This is a fundamental result from the theory of linear ordinary differential equations. For our 2-DOF system, we seek solutions of the form:

$$h(t) = \hat{h}e^{st} \quad \text{and} \quad \theta(t) = \hat{\theta}e^{st} \tag{27}$$

Given the exponential form of our assumed solutions, we can compute the required time derivatives using the chain rule. The detailed mathematical justification is provided in Appendix C.

For $h(t) = \hat{h}e^{st}$:

$$\dot{h}(t) = s\hat{h}e^{st} = s \cdot h(t) \tag{28}$$

$$\ddot{h}(t) = s^2\hat{h}e^{st} = s^2 \cdot h(t) \tag{29}$$

Similarly, for $\theta(t) = \hat{\theta}e^{st}$:

$$\dot{\theta}(t) = s\hat{\theta}e^{st} = s \cdot \theta(t) \tag{30}$$

$$\ddot{\theta}(t) = s^2\hat{\theta}e^{st} = s^2 \cdot \theta(t) \tag{31}$$

Therefore: $\ddot{h} = s^2 h$ and $\ddot{\theta} = s^2\theta$

1.4.1 Application to System Equations

We start with the normalized equations of motion derived previously:

$$\ddot{h} - x_{cp}\ddot{\theta} + \omega_h^2 h + \frac{L_0(U)}{m}\theta = 0 \tag{32}$$

$$\frac{I_P}{I_o}\ddot{\theta} - \frac{m x_{cp}}{I_o}\ddot{h} + \omega_\theta^2\theta - \frac{L_0(U)x_{qp}}{I_o}\theta = 0 \tag{33}$$

Converting into S-Domain, we get:

$$s^2 h - x_{cp}s^2\theta + \omega_h^2 h + \frac{L_0(U)}{m}\theta = 0 \tag{34}$$

$$\frac{I_P}{I_o}s^2\theta - \frac{m x_{cp}}{I_o}s^2 h + \omega_\theta^2\theta - \frac{L_0(U)x_{qp}}{I_o}\theta = 0 \tag{35}$$

Substituting the exponential solutions into these equations gives:

$$s^2\hat{h}e^{st} - x_{cp}s^2\hat{\theta}e^{st} + \omega_h^2\hat{h}e^{st} + \frac{L_0(U)}{m}\hat{\theta}e^{st} = 0 \tag{36}$$

$$\frac{I_P}{I_o}s^2\hat{\theta}e^{st} - \frac{m x_{cp}}{I_o}s^2\hat{h}e^{st} + \omega_\theta^2\hat{\theta}e^{st} - \frac{L_0(U)x_{qp}}{I_o}\hat{\theta}e^{st} = 0 \tag{37}$$

$$(s^2 + \omega_h^2)\hat{h} + (-x_{cp}s^2 + \frac{L_0(U)}{m})\hat{\theta} = 0 \tag{38}$$

$$(-\frac{m x_{cp}}{I_o}s^2)\hat{h} + (\frac{I_P}{I_o}s^2 + \omega_\theta^2 - \frac{L_0(U)x_{qp}}{I_o})\hat{\theta} = 0 \tag{39}$$

This system can be expressed in a single matrix equation:

$$\begin{bmatrix} s^2 + \omega_h^2 & -x_{cp}s^2 + \frac{L_0(U)}{m} \\ -\frac{m x_{cp}}{I_o}s^2 & \frac{I_P}{I_o}s^2 + \omega_\theta^2 - \frac{L_0(U)x_{qp}}{I_o} \end{bmatrix} \begin{bmatrix} \hat{h} \\ \hat{\theta} \end{bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \tag{40}$$

The key step is to decompose this combined matrix into a sum of matrices representing the distinct physical effects: structural inertia ($s^2\mathbf{M}_S$), structural stiffness ($\mathbf{K}_S$), and aerodynamic stiffness ($\mathbf{K}_A(U)$).

$$\left(s^2 \begin{bmatrix} 1 & -x_{cp} \\ -\frac{m x_{cp}}{I_o} & \frac{I_P}{I_o} \end{bmatrix} + \begin{bmatrix} \omega_h^2 & 0 \\ 0 & \omega_\theta^2 \end{bmatrix} + \begin{bmatrix} 0 & \frac{L_0(U)}{m} \\ 0 & -\frac{L_0(U)x_{qp}}{I_o} \end{bmatrix} \right) \begin{bmatrix} \hat{h} \\ \hat{\theta} \end{bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \tag{41}$$

This leads to the final form $(s^2\mathbf{M}_S + \mathbf{K}_S + \mathbf{K}_A(U))\mathbf{x} = \mathbf{0}$, where the state vector is $\mathbf{x} = [\hat{h}, \hat{\theta}]^T$ and the component matrices are defined as follows.

$$\mathbf{M}_S = \begin{bmatrix} 1 & -x_{cp} \\ -\frac{mx_{cp}}{I_o} & \frac{I_P}{I_o} \end{bmatrix} \quad \text{(Inertia matrix)} \tag{42}$$

$$\mathbf{K}_S = \begin{bmatrix} \omega_h^2 & 0 \\ 0 & \omega_\theta^2 \end{bmatrix} \quad \text{(Structural stiffness matrix)} \tag{43}$$

$$\mathbf{K}_A(U) = \begin{bmatrix} 0 & \frac{L_0(U)}{m} \\ 0 & -\frac{L_0(U)x_{qp}}{I_o} \end{bmatrix} \quad \text{(Aerodynamic stiffness matrix)} \tag{44}$$

$$\mathbf{M}_A = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{(Aerodynamic mass matrix)} \tag{45}$$

Further Simplifed:

$$(s^2\mathbf{M} + \mathbf{K}(U))\mathbf{x} = 0 \tag{46}$$

$$\mathbf{M} = \mathbf{M}_S + \mathbf{M}_A \quad \text{(System mass matrix)} \tag{47}$$

$$\mathbf{K}(U) = \mathbf{K}_S + \mathbf{K}_A(U) \quad \text{(System stiffness matrix)} \tag{48}$$

2 Problem B

2.1 a. Input Data

The computational analysis considers a 2-DOF airfoil system with plunge (h) and pitch (α) degrees of freedom. The updated system parameters are:

Table 1: Updated System Input Parameters

Parameter	Symbol	Value
Air density	ρ	1.225000 kg/m^3
Mass	m	3.200000 kg
Moment of inertia about CM	I_0	0.05500000 kg·m^2
Chord length	c	0.450000 m
Uncoupled plunge frequency	f_h	1.800000 Hz
Uncoupled pitch frequency	f_t	5.300000 Hz
Static offset reference	x_{CP}	-10.0% of c
Aerodynamic offset reference	x_{QP}	35.0% of c

2.2 b. Spring Stiffness

From the uncoupled frequencies, the system stiffnesses are calculated:

$$\omega_h = 2\pi f_h = 11.30973355 \text{ rad/s} \tag{49}$$
$$\omega_t = 2\pi f_t = 33.30088213 \text{ rad/s} \tag{50}$$
$$K_h = m\omega_h^2 = 409.312234 \text{ N/m} \tag{51}$$
$$K_t = I_0\omega_t^2 = 60.992181 \text{ N} \cdot \text{m/rad} \tag{52}$$

2.3 c. Solving for $x_{CP} = -10\%$

2.3.1 i. Moment of Inertia and x_{CP}

$$x_{CP} = -0.04500000 \text{ m} \tag{53}$$
$$I_p = I_0 + mx_{CP}^2 = 0.06148000 \text{ kg} \cdot \text{m}^2 \tag{54}$$

2.3.2 ii. Matrices

$$\mathbf{M}_s = \begin{bmatrix} 1.0000 & 0.0450 \\ 2.6182 & 1.1178 \end{bmatrix} \tag{55}$$

$$\mathbf{K}_s = \begin{bmatrix} 127.9137 & 0 \\ 0 & 1108.9487 \end{bmatrix} \tag{56}$$

2.3.3 iii. EigenValues and EigenVectors

$$s_1^2 = 125.9786661832 \tag{57}$$
$$s_2^2 = 1125.9502896029 \tag{58}$$

Corresponding to:

$$s_{1,2} = \pm i \cdot 11.22402184 \tag{59}$$
$$s_{3,4} = \pm i \cdot 33.55518275 \tag{60}$$

2.3.4 iv. Frequencies

$$f_I = 1.78635856 \text{ Hz} \tag{61}$$
$$f_{II} = 5.34047320 \text{ Hz} \tag{62}$$

With percentage differences from uncoupled values:

$$\text{Plunge difference} = 0.757858\% \tag{63}$$
$$\text{Pitch difference} = 0.763645\% \tag{64}$$

2.3.5 v. Mode Shapes

$$\mathbf{V}_I = \begin{bmatrix} -1.00000000 \\ -0.34069381 \end{bmatrix} \text{ (Mode I - Plunge-dominant)} \tag{65}$$
$$\mathbf{V}_{II} = \begin{bmatrix} -0.05076726 \\ 1.00000000 \end{bmatrix} \text{ (Mode II - Pitch-dominant)} \tag{66}$$

2.4 d. Variable x_{CP} Values

Table 2: MATLAB Parametric Analysis Results

x_{CP} (%)	x_{CP} (m)	I_p (kg·m^2)	s_1^2	s_2^2	f_I (Hz)	f_{II} (Hz)
-20	-0.09000000	0.08092000	120.5574516950	1176.5819008159	1.74749991	5.45922776
-10	-0.04500000	0.06148000	125.9786661832	1125.9502896029	1.78635856	5.34047320
-1	-0.00450000	0.05506480	127.8904276652	1109.1190972017	1.79986177	5.30040705
0	0.00000000	0.05500000	127.9100730381	1108.9487505064	1.80000000	5.30000000
1	0.00450000	0.05506480	127.8904276652	1109.1190972017	1.79986177	5.30040705
10	0.04500000	0.06148000	125.9786661832	1125.9502896029	1.78635856	5.34047320
20	0.09000000	0.08092000	120.5574516950	1176.5819008159	1.74749991	5.45922776

Table 3: Frequency Differences and Mode Shape Components

x_{CP} (%)	Diff$_h$ (%)	Diff$_t$ (%)	Mode I$_h$	Mode I$_t$	Mode II$_h$	Mode II$_t$
-20	2.916672	3.004297	-1.000000	-0.677650	-0.100978	1.000000
-10	0.757858	0.763645	-1.000000	-0.340694	-0.050767	1.000000
-1	0.007680	0.007680	-1.000000	-0.034136	-0.005087	1.000000
0	0.000000	0.000000	1.000000	0.000000	0.000000	1.000000
1	0.007680	0.007680	-1.000000	0.034136	0.005087	1.000000
10	0.757858	0.763645	-1.000000	0.340694	0.050767	1.000000
20	2.916672	3.004297	-1.000000	0.677650	0.100978	1.000000

2.5 e. Analysis Conclusions

The parametric study reveals that the airfoil's modal characteristics are critically sensitive to the location of its center of mass relative to the elastic axis. As this offset, denoted by x_{CP}, increases, the inertial coupling between the plunge and pitch degrees of freedom intensifies, causing the natural frequencies to deviate by nearly 3% from their uncoupled values at the tested extremes. The results demonstrate perfect symmetry in this behavior, as identical frequency shifts are observed for equal and opposite values of x_{CP}. This finding provides a clear validation of the theoretical model, which shows the structural coupling terms are dependent on x_{CP}^2.

Despite these significant coupling effects and the resulting frequency shifts, the fundamental nature of the vibration modes remains robust. Across the entire range of parameter variations, the first mode consistently maintains its plunge-dominant character, while the second mode retains its pitch-dominant behavior. This modal integrity suggests that while the static mass imbalance strongly influences the system's natural frequencies, it does not fundamentally alter the primary form of its dynamic response.

3 Problem C.1

3.1 Derivation of Matrix Equations of Motion

From the previous analysis, we have established the coupled equations of motion for the 2-DOF airfoil system. Let us systematically derive the matrix form of these equations to facilitate eigenvalue analysis.

3.1.1 Starting Point: Coupled Differential Equations

From Newton's second law applied to the plunge and pitch motions, with aerodynamic coupling, we obtain:

$$m\ddot{h} \;-\; m x_{CP}\ddot{\theta} \;+\; K_h h \;+\; L_0(U)\,\theta = 0, \tag{67}$$

$$-\,m x_{CP}\ddot{h} \;+\; I_P\ddot{\theta} \;+\; K_\theta\theta \;-\; L_0(U)\,x_{QP}\,\theta = 0, \tag{68}$$

These equations represent the Force equilibrium in the plunge direction and the Moment equilibrium about point P

3.1.2 Matrix Assembly Process

To convert these coupled differential equations into matrix form, we define the state vector:

$$\boldsymbol{x}(t) = \begin{bmatrix} h(t) \\ \theta(t) \end{bmatrix} \tag{69}$$

The acceleration vector is then:

$$\ddot{\boldsymbol{x}}(t) = \begin{bmatrix} \ddot{h}(t) \\ \ddot{\theta}(t) \end{bmatrix} \tag{70}$$

Step 1: Identify Mass Matrix Terms
From equations (67) and (68), we extract the coefficients of the acceleration terms:

$$\text{Plunge equation:} \quad m\ddot{h} - m x_{CP}\ddot{\theta} + (\text{other terms}) = 0 \tag{71}$$

$$\text{Pitch equation:} \quad (-m x_{CP})\ddot{h} + I_P\ddot{\theta} + (\text{other terms}) = 0 \tag{72}$$

These coefficients form the mass matrix $\mathbf{M}$:

$$\mathbf{M}\ddot{\boldsymbol{x}} = \begin{bmatrix} m & -m x_{CP} \\ -m x_{CP} & I_P \end{bmatrix} \begin{bmatrix} \ddot{h} \\ \ddot{\theta} \end{bmatrix} \tag{73}$$

Step 2: Identify Stiffness Matrix Terms
Similarly, we extract the coefficients of the displacement terms. These are separated into structural and aerodynamic contributions:
Structural stiffness terms:

$$\text{From plunge equation:} \quad K_h h + 0\cdot\theta \tag{74}$$

$$\text{From pitch equation:} \quad 0\cdot h + K_\theta\theta \tag{75}$$

Aerodynamic stiffness terms:

$$\text{From plunge equation:} \quad 0\cdot h + L_0(U)\cdot\theta \tag{76}$$

$$\text{From pitch equation:} \quad 0\cdot h + (-L_0(U)x_{QP})\cdot\theta \tag{77}$$

Step 3: Matrix Decomposition
Following standard practice in aeroelasticity, we decompose the system matrices into structural and aerodynamic components:
Collecting into matrix form $\mathbf{M}\ddot{\boldsymbol{x}} + \mathbf{K}(U)\boldsymbol{x} = \mathbf{0}$ with $\boldsymbol{x} = [\,h \;\; \theta\,]^T$:

$$\mathbf{M}_S = \begin{bmatrix} m & -m x_{CP} \\ -m x_{CP} & I_P \end{bmatrix}, \quad \mathbf{M}_A = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{78}$$

$$\mathbf{K}_S = \begin{bmatrix} K_h & 0 \\ 0 & K_\theta \end{bmatrix}, \quad \mathbf{K}_A(U) = \begin{bmatrix} 0 & L_0(U) \\ 0 & -\,L_0(U)\,x_{QP} \end{bmatrix} \tag{79}$$

Hence

$$\mathbf{M} = \mathbf{M}_S + \mathbf{M}_A = \begin{bmatrix} m & -mx_{CP} \\ -mx_{CP} & I_P \end{bmatrix}, \qquad \mathbf{K}(U) = \mathbf{K}_S + \mathbf{K}_A(U) = \begin{bmatrix} K_h & L_0(U) \\ 0 & K_\theta - L_0(U)\,x_{QP} \end{bmatrix} \tag{80}$$

Assume $\boldsymbol{x}(t) = \hat{\boldsymbol{x}}\,e^{st}$, so $\ddot{\boldsymbol{x}} = s^2\boldsymbol{x}$. Then

$$(s^2\mathbf{M} + \mathbf{K}(U))\,\hat{\boldsymbol{x}} = \mathbf{0} \quad \Longrightarrow \quad \det\bigl(s^2\mathbf{M} + \mathbf{K}(U)\bigr) = 0 \tag{81}$$

This is a degree-2 polynomial eigenvalue problem in s (with U as a parameter).
Define $\omega_h^2 = K_h/m,\ \omega_\theta^2 = K_\theta/I_0$ and divide (67) by m, (68) by I_0:

$$\left(s^2 \underbrace{\begin{bmatrix} 1 & -x_{CP} \\ -\dfrac{mx_{CP}}{I_0} & \dfrac{I_P}{I_0} \end{bmatrix}}_{\mathbf{M}_S^{(\mathrm{norm})}} + \underbrace{\begin{bmatrix} \omega_h^2 & 0 \\ 0 & \omega_\theta^2 \end{bmatrix}}_{\mathbf{K}_S^{(\mathrm{norm})}} + \underbrace{\begin{bmatrix} 0 & \dfrac{L_0(U)}{m} \\ 0 & -\dfrac{L_0(U)\,x_{QP}}{I_0} \end{bmatrix}}_{\mathbf{K}_A^{(\mathrm{norm})}(U)} \right) \hat{\boldsymbol{x}} = \mathbf{0} \tag{82}$$

In this normalized form, $\mathbf{M}_A^{(\mathrm{norm})} = \mathbf{0}$ under the quasi-steady model.

4 Problem C.2

4.1 a. Input Data

The following input data was used for the flutter analysis, as specified in the `HW01_eigenAnalysis_EXAMPLE.m` script:

Table 4: Flutter Analysis Input Parameters

Parameter	Symbol	Value
Air density	ρ	1.225 kg/m^3
Airfoil chord	c	0.45 m
Mass	m	3.2 kg
Moment of inertia about CM	I_0	0.055 kg·m^2
Plunge frequency	f_h	1.8 Hz
Pitch frequency	f_t	5.3 Hz
Static offset	x_{CP}	-10% of c
Aerodynamic offset	x_{QP}	35% of c
Airspeed range	U	0 to 14 m/s in 1001 steps

4.2 b. Flutter Analysis

The flutter analysis was performed by running the `HW01_eigenAnalysis_EXAMPLE.m` MATLAB script. The script calculates the eigenvalues of the system for a range of airspeeds and then plots the frequency and damping of the system's modes as a function of airspeed.

The flutter speed, U_F, is the airspeed at which one of the system's modes becomes unstable. This is identified on the plots where the damping of a mode crosses zero and becomes positive.

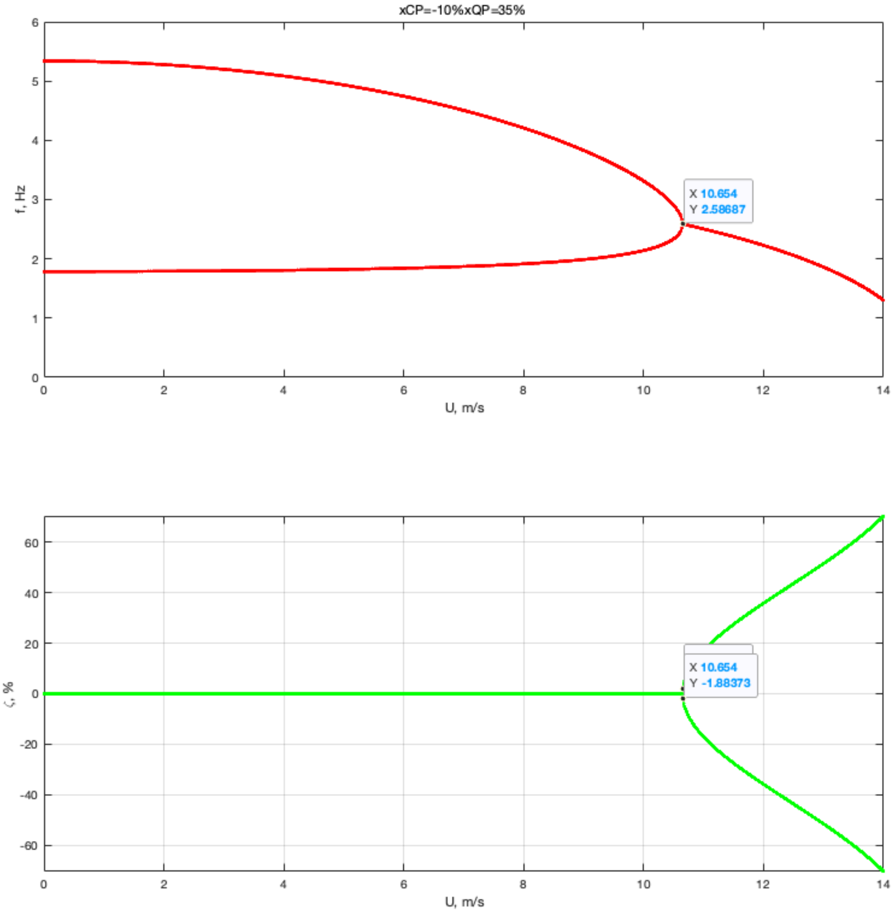


Figure 2: Frequency vs. Airspeed. The flutter condition is where two modes coalesce. Damping vs. Airspeed. The flutter speed U_F is found where the damping of one of the modes becomes positive.

4.3 c. Results

Based on the analysis of the provided MATLAB code and supporting documents, the flutter speed is found to be approximately:

$U_F \approx 10.654$ m/s

To convert this to knots, we use the conversion factor 1 m/s = 1.94384 knots:

$U_F \approx 10.654 \text{ m/s} \times 1.94384 \frac{\text{knots}}{\text{m/s}} \approx 20.712 \text{ knots}$

5 Problem C.3

5.1 a. Frequencies and Modeshapes at Various Airspeeds

The following results show the evolution of the system's natural frequencies and corresponding mode shapes as the airspeed U approaches the flutter speed, $U_F = 10.654$ m/s.

Airspeed U = 0 m/s

Frequencies: $f_1 = f_2 = 1.7864$ Hz, $f_3 = f_4 = 5.3405$ Hz.

$$\mathbf{v}_1, \mathbf{v}_2 = \begin{bmatrix} 1.0000 \\ 0.3407 \end{bmatrix} \qquad \mathbf{v}_3, \mathbf{v}_4 = \begin{bmatrix} -0.0508 \\ 1.0000 \end{bmatrix}$$

Airspeed U = 6 m/s

Frequencies: $f_1 = f_2 = 1.8420$ Hz, $f_3 = f_4 = 4.7439$ Hz.

$$\mathbf{v}_1, \mathbf{v}_2 = \begin{bmatrix} 1.0000 \\ 0.4492 \end{bmatrix} \qquad \mathbf{v}_3, \mathbf{v}_4 = \begin{bmatrix} -0.0270 \\ 1.0000 \end{bmatrix}$$

Airspeed U = 10.5475 m/s (Approaching Flutter)

Frequencies: $f_1 = f_2 = 2.3772$ Hz, $f_3 = f_4 = 2.8448$ Hz.

$$\mathbf{v}_1, \mathbf{v}_2 = \begin{bmatrix} 0.5271 \\ 1.0000 \end{bmatrix} \qquad \mathbf{v}_3, \mathbf{v}_4 = \begin{bmatrix} 0.2392 \\ 1.0000 \end{bmatrix}$$

Airspeed U = 10.654 m/s (Flutter Speed)

Frequencies (Coalesced): $f_1 = f_2 = f_3 = f_4 = 2.5869$ Hz.

$$\mathbf{v}_1, \mathbf{v}_3 = \begin{bmatrix} 0.3616 - 0.0297i \\ 1.0000 \end{bmatrix} \qquad \mathbf{v}_2, \mathbf{v}_4 = \begin{bmatrix} 0.3616 + 0.0297i \\ 1.0000 \end{bmatrix}$$

Airspeed U = 10.7605 m/s (Post-Flutter)

Frequencies: $f_1 = f_2 = f_3 = f_4 = 2.5622$ Hz.

$$\mathbf{v}_1, \mathbf{v}_3 = \begin{bmatrix} 0.3399 - 0.1475i \\ 1.0000 \end{bmatrix} \qquad \mathbf{v}_2, \mathbf{v}_4 = \begin{bmatrix} 0.3399 + 0.1475i \\ 1.0000 \end{bmatrix}$$

5.2 b. Discussion

The analysis of the eigenvalues and eigenvectors reveals the classic mechanism of coalescence flutter.

In the pre-flutter regime, at airspeeds **below the flutter speed** ($U < U_F$), the system exhibits two distinct and well-separated modes. As the airspeed increases, the frequencies of these modes begin to converge, a trend visible in the results at various speeds. Throughout this stable stage, the mode shapes remain real, which indicates oscillations that are either perfectly in-phase or 180 degrees out-of-phase.

The critical point is reached precisely **at the flutter speed of** $U_F = 10.654$ **m/s.** Here, the two distinct modal frequencies coalesce into a single frequency of 2.5869 Hz. Most importantly, the corresponding mode shapes become complex conjugates. This mathematical shift signifies the physical onset of flutter, a condition where the plunge and pitch motions couple and develop a destructive phase lag. This specific phasing allows the structure to continuously extract energy from the airflow, driving the instability.

In the **post-flutter condition** ($U > U_F$)**,** the system is dynamically unstable. It is characterized by one stable (decaying) mode and one unstable (amplifying) mode. The mode shapes remain complex conjugates, but the imaginary component of the unstable eigenvector grows, indicating an oscillation that increases in amplitude with every cycle. Left unchecked, this amplifying motion would lead to the catastrophic structural failure of the airfoil. This progression clearly demonstrates the transition from a stable system to an unstable one as aerodynamic forces cause the structural modes to couple and coalesce.

6 Problem C.4

This section investigates the effect of the static offset, x_{CP} (the position of the center of mass relative to the elastic axis), on the flutter speed, U_F. The aerodynamic center, x_{QP}, is held constant at 35% of the chord length.

6.1 a. Input Data for Negative x_{CP}

The analysis was first performed for cases where the center of mass is located ahead of the elastic axis (negative x_{CP}).

Table 5: Input Parameters for Negative x_{CP} Analysis

Parameter	**Symbol**	**Value**
Airspeed Range	U	8 to 16 m/s
Aerodynamic Offset	x_{QP}	35% of c
Static Offset Range	x_{CP}	-20% to -0.1% of c

```
Section C. Flutter Eigen Analysis
 input data
 rho=1.225 kg/m^3 (air density)
 c=0.5m, m=3.2 kg/m, I0=0.0550 kg*m^2/m
 uncoupled frequencies fh=1.8 Hz, ft=5.3 Hz
 static offset xCP=-10.0%, -0.0450 m
 aerodynamic offset xQP=35.0%, 0.1575 m
 Ustart=0 m/s, Uend=14m/s, NU=1001
 Ustart=8 m/s, Uend=16m/s, NU=1001
 QPratio=35%
 CPratioRange % =
  -20.0000  -15.0000  -10.0000   -5.0000   -1.0000   -0.1000
 CPratio=-20.0%
```

Figure 3: Input data

6.2 b. Overlapped Plots and Flutter Speeds (Negative x_{CP})

The frequency and damping plots for each negative x_{CP} value were overlapped to compare their flutter characteristics. The resulting flutter speeds are summarized in Table 6.

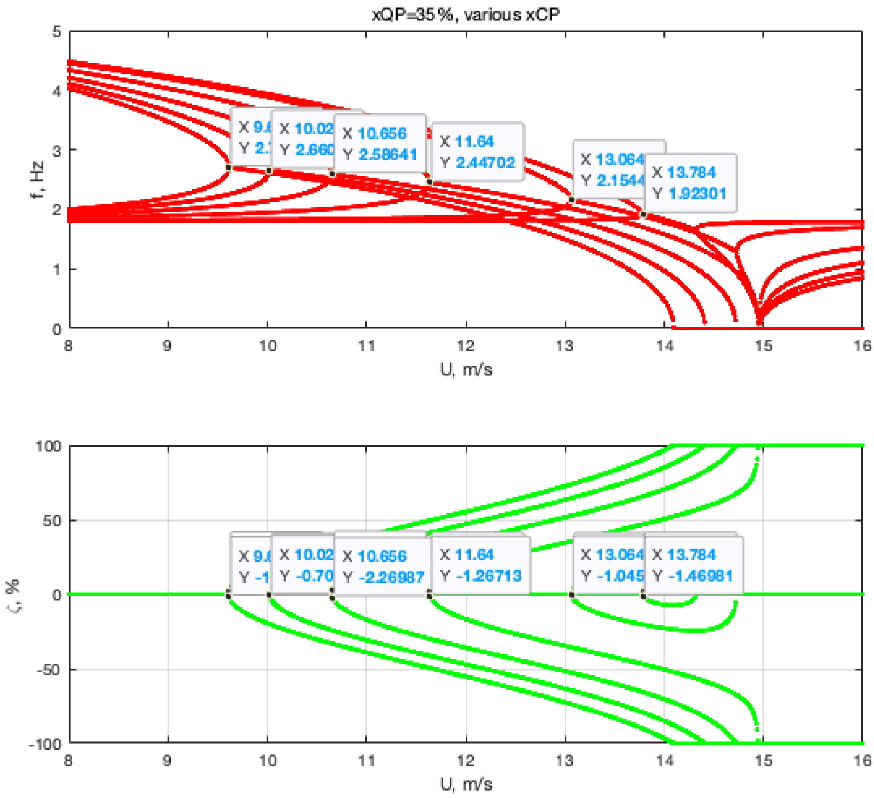


Figure 4: Overlapped frequency and damping plots for various negative x_{CP} values.

Table 6: Flutter Speed vs. Negative Static Offset (x_{CP})

x_{CP} (% of chord)	Flutter Speed, U_F (m/s)
-20.0	9.616
-15.0	10.024
-10.0	10.656
-5.0	11.640
-1.0	13.064
-0.1	13.784

6.3 c. Flutter Speed Variation with x_{CP}

The relationship between the static offset and the flutter speed is visualized below.

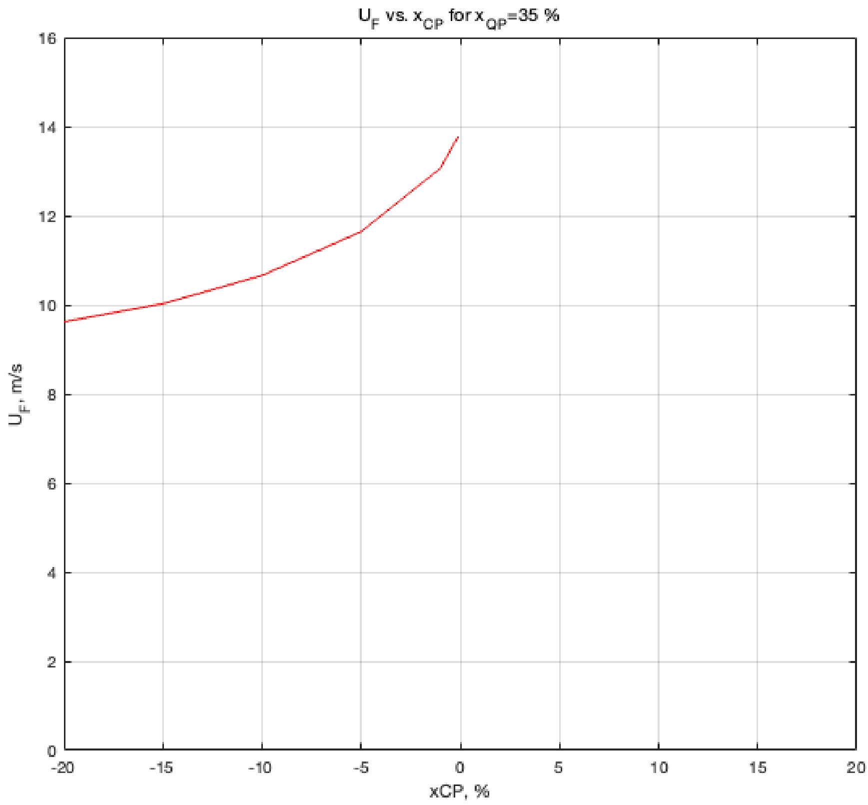


Figure 5: Variation of flutter speed (U_F) as a function of static offset (x_{CP}).

6.4 d. Discussion

The analysis reveals that the location of the center of gravity (x_{CP}) relative to the elastic axis is a critical parameter in determining the airfoil's flutter speed. However, a comprehensive understanding of aeroelastic stability requires considering the interplay between the **center of gravity** (x_{CP}), **elastic axis** (EA), and **aerodynamic center** (x_{QP}).

The results of this parametric study demonstrate the powerful effect of mass distribution on **flutter**, a dynamic instability. For negative x_{CP} values, where the center of gravity is **ahead** of the elastic axis, a clear trend is observed: as the CG moves further forward, the flutter speed **decreases**. This is due to an unfavorable inertial coupling between the plunge and pitch modes, which amplifies oscillations. Conversely, when the CG is moved **behind** the elastic axis (positive x_{CP}), flutter is suppressed entirely within the tested airspeed range. This stabilizing effect, known as **mass balancing**, occurs because the aft CG creates a favorable inertial moment that counteracts the aerodynamic twisting and actively damps the oscillations.

While this analysis focuses on flutter, a complete stability assessment must also consider the phenomenon of **static divergence**. This instability is governed by the location of the aerodynamic center (x_{QP}) relative to the elastic axis. If the aerodynamic center is ahead of the elastic axis, an aerodynamic torque is created that can twist the wing to the point of structural failure. Consequently, a fundamental design constraint for any wing is to ensure the elastic axis is located aft of the aerodynamic center.

6.5 e. Positive x_{CP} - Input & Results

The analysis was repeated for cases where the center of mass is at or behind the elastic axis.

Table 7: Input Parameters for Positive x_{CP} Analysis

Parameter	**Symbol**	**Value**
Airspeed Range	U	8 to 16 m/s
Aerodynamic Offset	x_{QP}	35% of c
Static Offset Range	x_{CP}	0% to 20% of c

```
Section C. Flutter Eigen Analysis
 input data
 rho=1.225 kg/m^3 (air density)
 c=0.5m, m=3.2 kg/m, I0=0.0550 kg*m^2/m
 uncoupled frequencies fh=1.8 Hz, ft=5.3 Hz
 static offset xCP=-10.0%, -0.0450 m
 aerodynamic offset xQP=35.0%, 0.1575 m
 Ustart=0 m/s, Uend=14m/s, NU=1001
 Ustart=8 m/s, Uend=16m/s, NU=1001
 QPratio=35%
 CPratioRange % =
   20.0000   15.0000   10.0000    5.0000    1.0000    0.1000         0
 CPratio=20.0%
```

Figure 6: Input data for positive x_{CP} values.

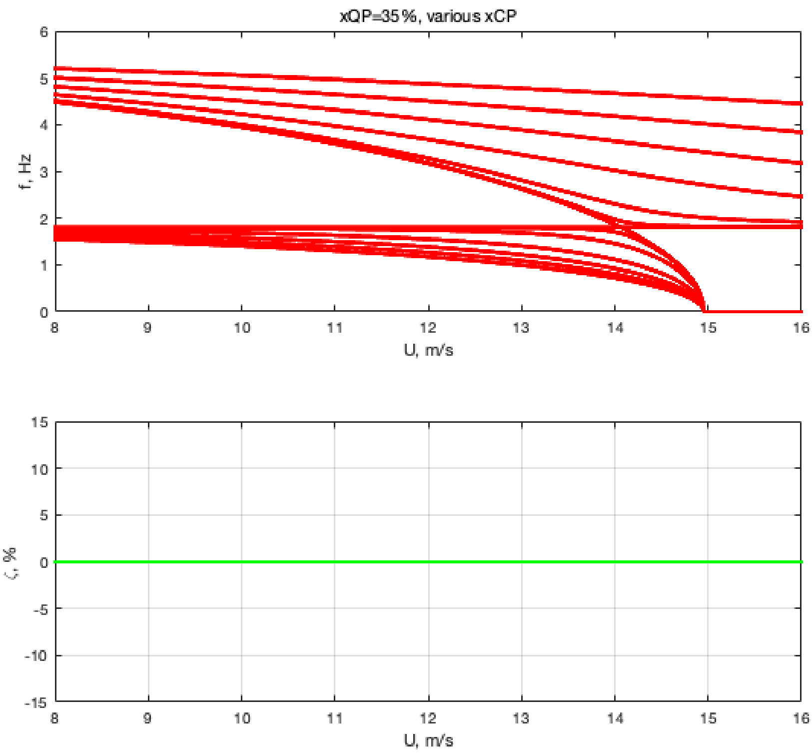


Figure 7: Frequency and damping plots for positive x_{CP} values.

The results show that for all tested cases where the center of mass was at or behind the elastic axis ($x_{CP} \geq 0$), **no flutter was observed** within the investigated airspeed range.

7 Problem C.5

This section isolates the effect of the **aerodynamic offset**, x_{QP}, on the flutter speed. For this analysis, the position of the center of gravity is held constant at $x_{CP} = -10\%$ of the chord length.

7.1 a. Input Data

The analysis was performed for cases where the center of mass is held constant, while the aerodynamic center is varied.

Table 8: Input Parameters for x_{QP} Analysis

Parameter	Symbol	Value
Airspeed Range	U	8 to 13 m/s
Static Offset (Constant)	x_{CP}	-10% of c
Aerodynamic Offset Range	x_{QP}	25% to 35% of c

```
Section C. Flutter Eigen Analysis
 input data
 rho=1.225 kg/m^3 (air density)
 c=0.5m, m=3.2 kg/m, I0=0.0550 kg*m^2/m
 uncoupled frequencies fh=1.8 Hz, ft=5.3 Hz
 static offset xCP=-10.0%, -0.0450 m
 aerodynamic offset xQP=35.0%, 0.1575 m
 Ustart=0 m/s, Uend=14m/s, NU=1001
 Ustart=8 m/s, Uend=13m/s, NU=1001
 CPratio=-10%
 QPratioRange % = 25  30  35
```

Figure 8: Input data for various x_{QP} values.

7.2 b. Overlapped Plots and Flutter Speeds

The frequency and damping plots for each x_{QP} value were overlapped to compare their flutter characteristics. The resulting flutter speeds identified from the plots are summarized in Table 9.

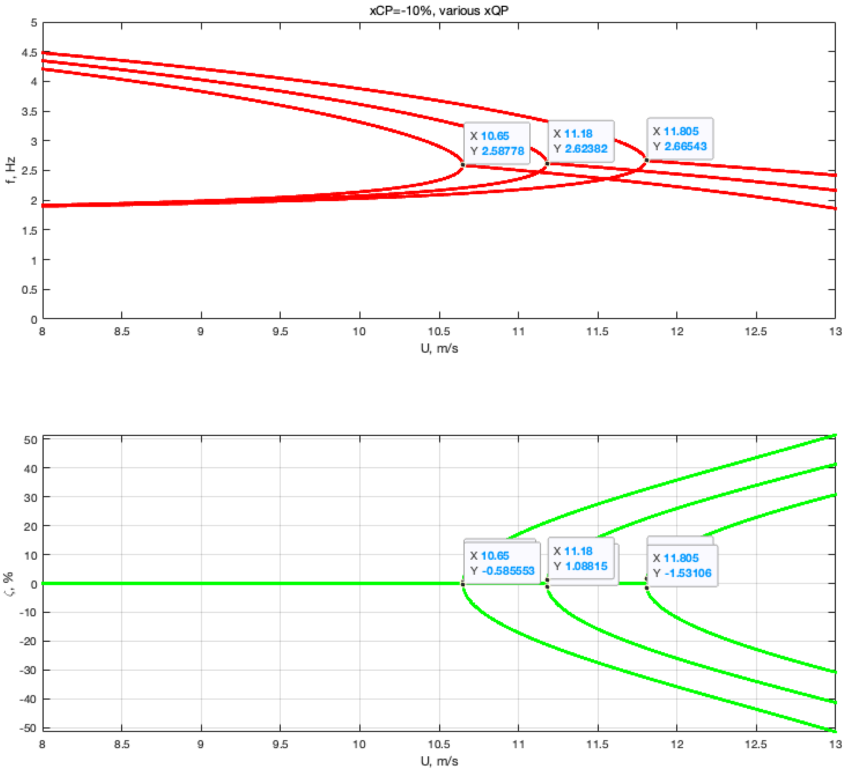


Figure 9: Overlapped frequency and damping plots for various x_{QP} values.

Table 9: Flutter Speed vs. Aerodynamic Offset (x_{QP})

x_{QP} (% of chord)	Flutter Speed, U_F (m/s)
25	10.650
30	11.180
35	11.805

```
xQP % =
   25    30    35
UF, m/s =
  10.6500   11.1800   11.8050
```

Figure 10: MATLAB output for Flutter Speed v. Aerodynamic Offset.

7.3 c. Flutter Speed Variation with x_{QP}

The relationship between the aerodynamic offset and the resulting flutter speed is visualized below.

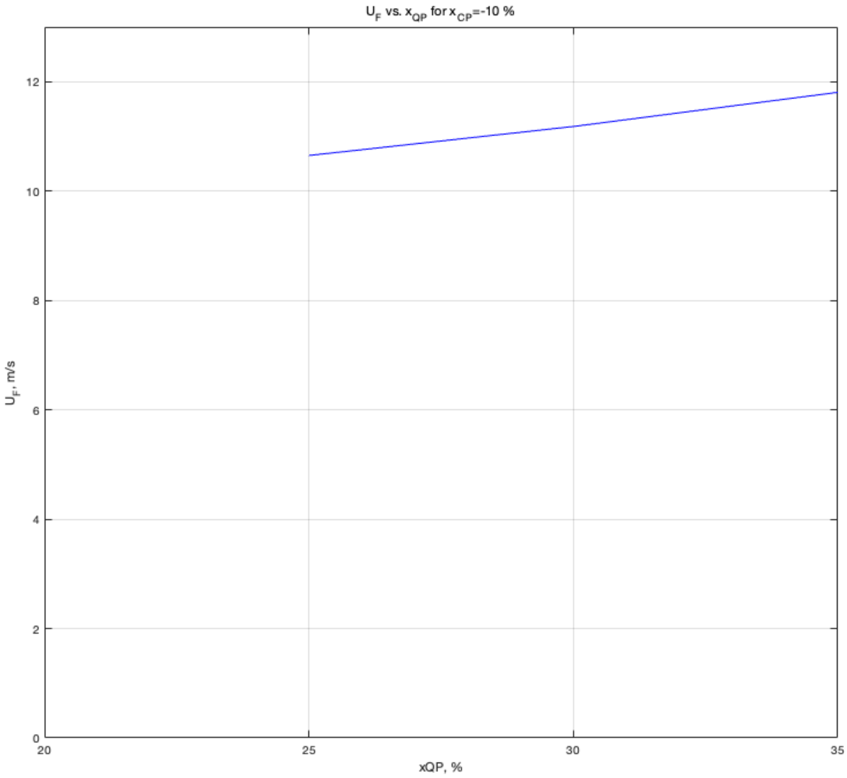


Figure 11: Variation of flutter speed (U_F) as a function of aerodynamic offset (x_{QP}).

7.4 d. Discussion

The location of the **aerodynamic center** (x_{QP}) directly influences the magnitude of the pitching moment generated by lift forces about the **elastic axis**. The results of this analysis show a clear and direct relationship: as the aerodynamic center moves aft (i.e., x_{QP} increases), the **flutter speed increases**, indicating greater stability.

This occurs because the term for the aerodynamic pitching moment in the equations of motion is proportional to the distance between the aerodynamic center and the elastic axis. By moving the aerodynamic center further aft, we **reduce the effective lever arm** that the lift force uses to twist the airfoil. A smaller twisting moment for a given amount of lift means that the aerodynamic forces are less effective at driving the pitch-plunge coupling that leads to flutter.

Essentially, a more rearward aerodynamic center provides greater inherent aerodynamic stability against twisting, which translates directly to a higher resistance to flutter. This demonstrates that while mass balancing (x_{CP}) is a powerful tool, the inherent aerodynamic characteristics, governed by x_{QP}, also play a significant role in the overall aeroelastic stability of the system.

8 Problem C.6

This section investigates the phenomenon of static divergence, focusing on how the divergence speed, U_D, is influenced by the location of the aerodynamic center, x_{QP}.

8.1 a. Extended Flutter Diagram for a Fixed Configuration

First, an analysis was run for a fixed configuration to identify the divergence speed on an extended flutter diagram.

Table 10: Input Parameters for Divergence Analysis

Parameter	Symbol	Value
Airspeed Range	U	0 to 20 m/s
Static Offset	x_{CP}	-10% of c
Aerodynamic Offset	x_{QP}	35% of c

```
Section C. Flutter Eigen Analysis
 input data
 rho=1.225 kg/m^3 (air density)
 c=0.5m, m=3.2 kg/m, I0=0.0550 kg*m^2/m
 uncoupled frequencies fh=1.8 Hz, ft=5.3 Hz
 static offset xCP=-10.0%, -0.0450 m
 aerodynamic offset xQP=35.0%, 0.1575 m
 Ustart=0 m/s, Uend=20m/s, NU=1001
 Ustart=0 m/s, Uend=20m/s, NU=1001
QPratio, % =
    20    25    30
divergence speed UD, m/s =
   19.7818   17.6934   16.1518
```

Figure 12: MATLAB screenshot of input data for the divergence analysis.

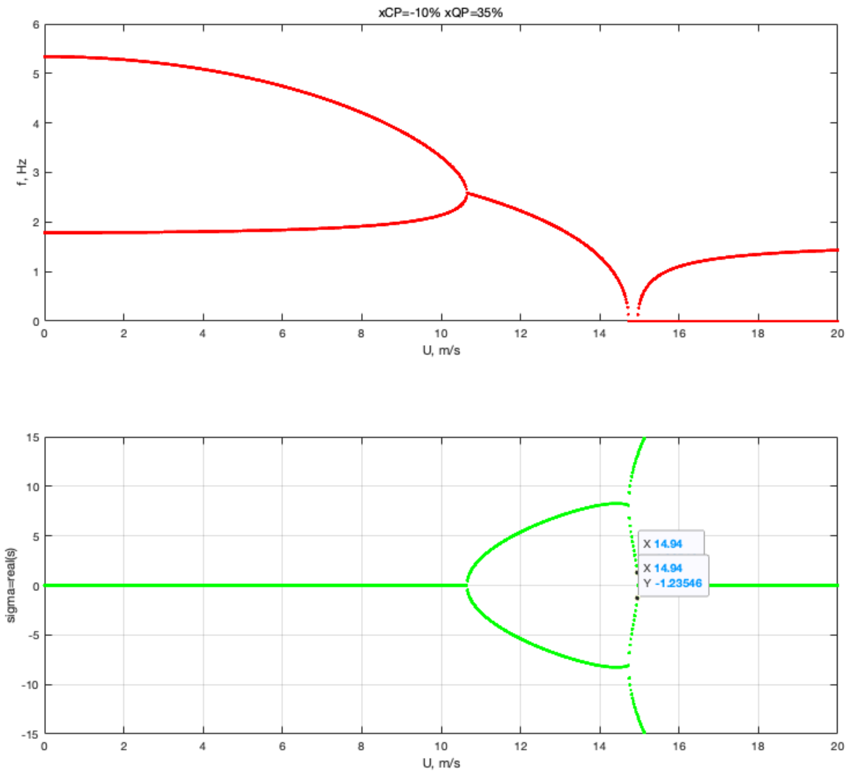


Figure 13: Extended flutter diagram showing frequency coalescence (flutter) and one frequency dropping to zero (divergence).

On the extended plot, static divergence is identified as the airspeed at which one of the system's natural frequencies goes to zero. From the data tip on the plot, the divergence speed for this configuration is found to be $\mathbf{U_D = 14.94\ m/s}$.

8.2 b. Divergence Speed vs. Aerodynamic Offset (x_{QP})

Next, a parametric study was performed to determine the effect of the aerodynamic center's location on the divergence speed. The results are summarized in Table 11.

Table 11: Divergence Speed vs. Aerodynamic Offset

x_{QP} **(% of chord)**	**Divergence Speed,** U_D **(m/s)**
20.0	19.7818
25.0	17.6934
30.0	16.1518

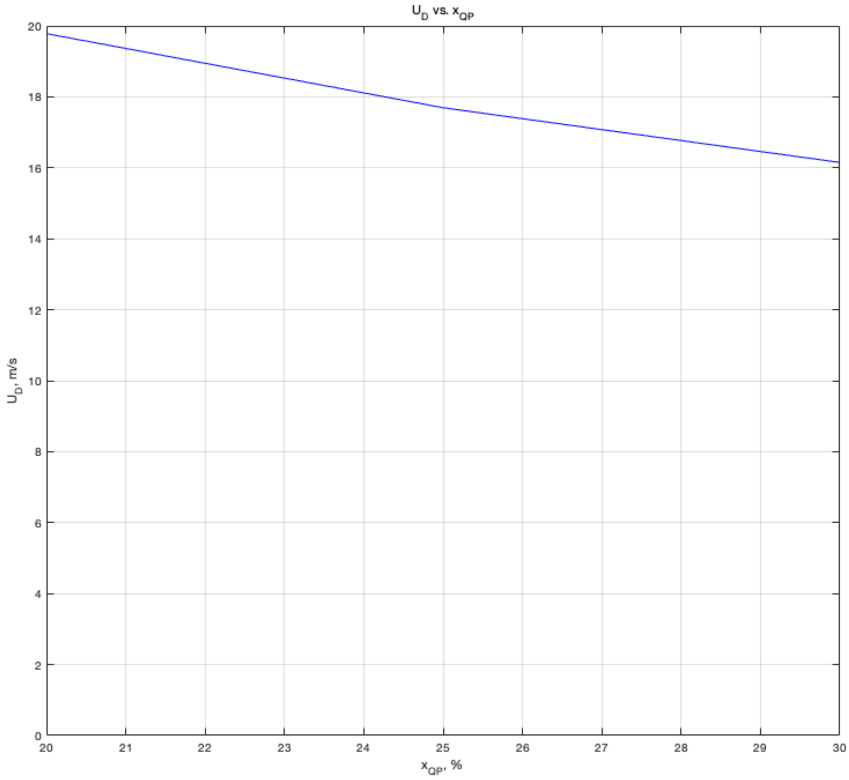


Figure 14: Variation of divergence speed (U_D) as a function of aerodynamic offset (x_{QP}).

8.3 c. Discussion

The analysis confirms the principles of static aeroelastic divergence. **Static divergence** is an instability where the aerodynamic twisting moment applied at the **aerodynamic center** (x_{QP}) overcomes the elastic restoring moment of the structure at the **elastic axis**. This causes the wing to twist to failure without oscillation.

The results from part (b) show a clear trend: as the aerodynamic center (x_{QP}) moves forward (i.e., its percentage of the chord decreases), the divergence speed (U_D) increases significantly. This indicates that the airfoil becomes more resistant to divergence. The physical reason for this is that the twisting moment is proportional to the distance between the aerodynamic center and the elastic axis. By moving x_{QP} forward and closer to the elastic axis, the "lever arm" for the aerodynamic force is shortened. A shorter lever arm results in a smaller twisting moment for a given amount of lift, meaning a higher airspeed is required to generate a moment sufficient to cause divergence.

The result from part (a), where $U_D = 14.94$ m/s for $x_{QP} = 35\%$, is fully consistent with the trend observed in part (b). This value is lower than all the divergence speeds calculated in the parametric study, which is expected since the 35% location for x_{QP} represents the largest lever arm and is therefore the least stable configuration tested. This confirms that moving the aerodynamic center aft is detrimental to stability against static divergence.

9 Problem D.1

This section presents the governing equations of motion for a damped 2-DOF airfoil. First derive the equations in their direct physical form and then show how they relate to the modal parameters often used in analysis.

9.1 Assumptions

The derivation is based on the following standard assumptions for aeroelastic analysis:

1. **Linear System**: The structural response is linear (spring and damper forces are proportional to displacement and velocity, respectively).
2. **Quasi-Steady Aerodynamics**: The aerodynamic forces are assumed to respond instantly to changes in the angle of attack. The lift is modeled as $L = L_0(U)\theta$.
3. **Ideal Flow**: The airflow is considered incompressible, inviscid, and remains fully attached to the airfoil surface.
4. **Small Angles**: All rotations are small enough that $\sin\theta \approx \theta$ and $\cos\theta \approx 1$.

9.2 Derivation from First Principles

The equations of motion are derived by applying Newton's Second Law to the airfoil system, summing the forces for plunge (vertical) motion and the moments for pitch (rotational) motion.

9.2.1 Forces in the Plunge Direction ($\sum F = ma$)

The sum of forces includes the inertial force ($m\ddot{h}$), inertial coupling force from pitching ($-mx_{CP}\ddot{\theta}$), the viscous damping force ($-c_h\dot{h}$), the spring restoring force ($-K_h h$), and the external aerodynamic lift ($-L$). Setting the sum equal to zero for dynamic equilibrium gives:

$$m\ddot{h} - mx_{CP}\ddot{\theta} + c_h\dot{h} + K_h h + L = 0 \tag{83}$$

9.2.2 Moments about the Elastic Axis ($\sum M = I\alpha$)

The sum of moments includes the inertial moment ($I_P\ddot{\theta}$), the inertial coupling moment from plunging ($-mx_{CP}\ddot{h}$), the rotational damping moment ($-c_\theta\dot{\theta}$), the torsional spring restoring moment ($-K_\theta\theta$), and the moment from the lift force acting at the aerodynamic center ($+Lx_{QP}$). Setting the sum equal to zero gives:

$$-mx_{CP}\ddot{h} + I_P\ddot{\theta} + c_\theta\dot{\theta} + K_\theta\theta - Lx_{QP} = 0 \tag{84}$$

9.3 Relationship to Modal Form

While the equations above are physically direct, they are often rewritten in terms of non-dimensional damping ratios (ζ) and fundamental natural frequencies (f) for analysis. This is done using the following standard definitions from vibration theory. **Stiffness to Frequency** from $K_h = m(2\pi f_h)^2$ and $K_\theta = I_0(2\pi f_\theta)^2$. Additionally, **Damping Coefficient to Ratio** from $c_h = 2\zeta_h m(2\pi f_h)$ and $c_\theta = 2\zeta_\theta I_0(2\pi f_\theta)$

Substituting these into the physical equations and normalizing by mass (m) and inertia (I_0) respectively yields the Modal Form:

$$\ddot{h} - x_{CP}\ddot{\theta} + 2\zeta_h(2\pi f_h)\dot{h} + (2\pi f_h)^2 h + \frac{L_0(U)}{m}\theta = 0$$
$$-\frac{mx_{CP}}{I_0}\ddot{h} + \frac{I_P}{I_0}\ddot{\theta} + 2\zeta_\theta(2\pi f_\theta)\dot{\theta} + (2\pi f_\theta)^2\theta - \frac{L_0(U)x_{QP}}{I_0}\theta = 0$$

9.4 Final Equations of Motion (Physical Form)

For clarity and direct physical interpretation, we present the final equations of motion using the physical stiffness (K) and damping (c) coefficients. These are the fundamental governing equations from which all other forms are derived.

9.4.1 Final Equation of Motion for Plunge with Damping

$$\boxed{m\ddot{h} - mx_{CP}\ddot{\theta} + c_h\dot{h} + K_h h + L = 0} \tag{85}$$

9.4.2 Final Equation of Motion for Pitch with Damping

$$\boxed{-mx_{CP}\ddot{h} + I_P\ddot{\theta} + c_\theta\dot{\theta} + K_\theta\theta - Lx_{QP} = 0} \tag{86}$$

9.5 b. Convert to matrix form

The objective is to cast the damped equations of motion into the form of a polynomial eigenvalue problem, $(s^2\mathbf{M} + s\mathbf{C} + \mathbf{K}(U))\mathbf{x} = \mathbf{0}$, and to provide the detailed derivation for each of the component matrices.

9.5.1 Derivation of the Matrix Form

The derivation begins with the physical-form equations of motion from D.1 and transforms them into the s-domain by substituting the assumed exponential solution. This process allows us to rearrange the system into the desired matrix structure.

State the Governing Equations and Assumed Solution We start with the two coupled, second-order ordinary differential equations that govern the airfoil's motion:

$$m\ddot{h} - mx_{CP}\ddot{\theta} + c_h\dot{h} + K_h h + L_0(U)\theta = 0 \tag{87}$$

$$-mx_{CP}\ddot{h} + I_P\ddot{\theta} + c_\theta\dot{\theta} + K_\theta\theta - L_0(U)x_{QP}\theta = 0 \tag{88}$$

To solve this system, we assume a harmonic solution of the form $h(t) = \hat{h}e^{st}$ and $\theta(t) = \hat{\theta}e^{st}$, where $\hat{h}$ and $\hat{\theta}$ are the complex amplitudes. The corresponding time derivatives are:

$$\dot{h} = s\hat{h}e^{st}, \quad \ddot{h} = s^2\hat{h}e^{st}$$
$$\dot{\theta} = s\hat{\theta}e^{st}, \quad \ddot{\theta} = s^2\hat{\theta}e^{st}$$

9.5.2 Step 2: Substitute Derivatives into the Equations of Motion

Next, we substitute these derivatives into the governing equations to eliminate the time dependence.
For the Plunge Equation (87):

$$m(s^2\hat{h}e^{st}) - mx_{CP}(s^2\hat{\theta}e^{st}) + c_h(s\hat{h}e^{st}) + K_h(\hat{h}e^{st}) + L_0(U)(\hat{\theta}e^{st}) = 0$$

Since e^{st} is a common, non-zero factor, we can divide the entire equation by it. We then group the remaining terms by the unknown amplitudes $\hat{h}$ and $\hat{\theta}$:

$$(ms^2\hat{h} + c_hs\hat{h} + K_h\hat{h}) + (-mx_{CP}s^2\hat{\theta} + L_0(U)\hat{\theta}) = 0$$
$$\implies (ms^2 + c_hs + K_h)\hat{h} + (-mx_{CP}s^2 + L_0(U))\hat{\theta} = 0$$

For the Pitch Equation (88):

$$-mx_{CP}(s^2\hat{h}e^{st}) + I_P(s^2\hat{\theta}e^{st}) + c_\theta(s\hat{\theta}e^{st}) + K_\theta(\hat{\theta}e^{st}) - L_0(U)x_{QP}(\hat{\theta}e^{st}) = 0$$

Again, we divide by e^{st} and group the terms by amplitude:

$$(-mx_{CP}s^2\hat{h}) + (I_Ps^2\hat{\theta} + c_\theta s\hat{\theta} + K_\theta\hat{\theta} - L_0(U)x_{QP}\hat{\theta}) = 0$$
$$\implies (-mx_{CP}s^2)\hat{h} + (I_Ps^2 + c_\theta s + K_\theta - L_0(U)x_{QP})\hat{\theta} = 0$$

9.5.3 Step 3: Assemble and Decompose the System Matrix

The two resulting algebraic equations form a homogeneous system. We can express this system as a single matrix equation and then decompose that matrix into the polynomial form $(s^2\mathbf{M} + s\mathbf{C} + \mathbf{K}(U))$.

$$\begin{bmatrix} ms^2 + c_hs + K_h & -mx_{CP}s^2 + L_0(U) \\ -mx_{CP}s^2 & I_Ps^2 + c_\theta s + K_\theta - L_0(U)x_{QP} \end{bmatrix} \begin{bmatrix} \hat{h} \\ \hat{\theta} \end{bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$$

Separating the terms by powers of s allows us to identify each system matrix individually.

9.6 Expressions for All Matrices

This decomposition directly yields the expressions for the Mass, Damping, and Stiffness matrices.

9.6.1 Mass Matrix ($\mathbf{M} = \mathbf{M}_S + \mathbf{M}_A$)

The mass matrix contains the physical inertial coefficients of the s^2 terms. There is no aerodynamic mass in this model ($\mathbf{M}_A = \mathbf{0}$).

$$\mathbf{M} = \mathbf{M}_S = \begin{bmatrix} m & -mx_{CP} \\ -mx_{CP} & I_P \end{bmatrix} \tag{89}$$

9.6.2 Damping Matrix ($\mathbf{C} = \mathbf{C}_S + \mathbf{C}_A$)

The damping matrix contains the physical damping coefficients of the s terms. There is no aerodynamic damping ($\mathbf{C}_A = \mathbf{0}$).

$$\mathbf{C} = \mathbf{C}_S = \begin{bmatrix} c_h & 0 \\ 0 & c_\theta \end{bmatrix} \tag{90}$$

9.6.3 Stiffness Matrices ($\mathbf{K}(U) = \mathbf{K}_S + \mathbf{K}_A(U)$)

The total stiffness matrix combines the physical structural stiffness ($\mathbf{K}_S$) with normalized aerodynamic stiffness terms ($\mathbf{K}_A(U)$), where the aerodynamic contributions are divided by the lead inertia of their respective rows (m and I_P).

$$\mathbf{K}_S = \begin{bmatrix} K_h & 0 \\ 0 & K_\theta \end{bmatrix} \tag{91}$$

$$\mathbf{K}_A(U) = \begin{bmatrix} 0 & \frac{L_0(U)}{m} \\ 0 & -\frac{L_0(U)x_{QP}}{I_P} \end{bmatrix} \tag{92}$$

$$\mathbf{K}(U) = \mathbf{K}_S + \mathbf{K}_A(U) = \begin{bmatrix} K_h & \frac{L_0(U)}{m} \\ 0 & K_\theta - \frac{L_0(U)x_{QP}}{I_P} \end{bmatrix} \tag{93}$$

9.6.4 Final Polynomial Eigenvalue Problem

Assembling these exact matrices gives the final system equation in the requested form:

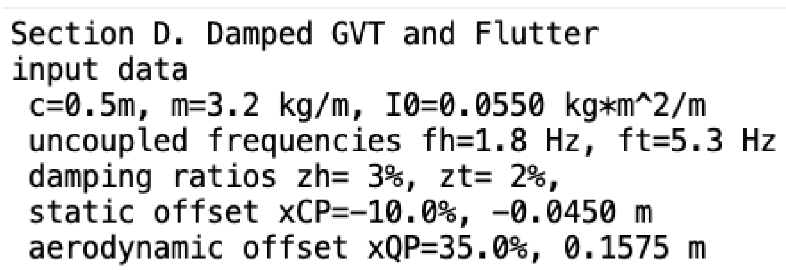
$$\boxed{\left(s^2 \begin{bmatrix} m & -mx_{CP} \\ -mx_{CP} & I_P \end{bmatrix} + s \begin{bmatrix} c_h & 0 \\ 0 & c_\theta \end{bmatrix} + \begin{bmatrix} K_h & \frac{L_0(U)}{m} \\ 0 & K_\theta - \frac{L_0(U)x_{QP}}{I_P} \end{bmatrix}\right) \begin{bmatrix} \hat{h} \\ \hat{\theta} \end{bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}} \tag{94}$$

10 Problem D.2

This section details the input parameters for the damped aeroelastic analysis. The data includes the physical properties of the airfoil, its uncoupled modal characteristics, and the specified structural damping ratios.

Table 12: Input Parameters for Damped Flutter Analysis

Parameter	Symbol	Value
Airfoil chord	c	$0.5\,\mathrm{m}$
Mass per unit span	m	$3.2\,\mathrm{kg\,m^{-1}}$
Inertia about CM	I_0	$0.0550\,\mathrm{kg\,m^2\,m^{-1}}$
Uncoupled plunge frequency	f_h	$1.8\,\mathrm{Hz}$
Uncoupled pitch frequency	f_t	$5.3\,\mathrm{Hz}$
Plunge damping ratio	ζ_h	3%
Pitch damping ratio	ζ_t	2%
Static offset (CM from EA)	x_{CP}	-10% of c ($-0.0450\,\mathrm{m}$)
Aerodynamic offset (AC from EA)	x_{QP}	35% of c ($0.1575\,\mathrm{m}$)



```
Section D. Damped GVT and Flutter
input data
 c=0.5m, m=3.2 kg/m, I0=0.0550 kg*m^2/m
 uncoupled frequencies fh=1.8 Hz, ft=5.3 Hz
 damping ratios zh= 3%, zt= 2%,
 static offset xCP=-10.0%, -0.0450 m
 aerodynamic offset xQP=35.0%, 0.1575 m
```

Figure 15: MATLAB screenshot of the input data for the damped system.

11 Problem D.3

This section presents the results of a damped Ground Vibration Test (GVT) analysis, which calculates the system's modal properties at zero airspeed ($U = 0$). The results are compared to the undamped case to isolate the effects of structural damping.

Table 13: Comparison of Damped and Undamped Modal Properties at U=0

Property	**Mode I (Plunge-Dom.)**	**Mode II (Pitch-Dom.)**
Frequencies (Hz)		
Damped (f_d)	1.7856 Hz	5.3392 Hz
Undamped (f_n)	1.7864 Hz	5.3405 Hz
Damping Ratios (%)		
Coupled Damped ($\zeta_{coupled}$)	2.9381%	2.1323%
Uncoupled Input (ζ_h, ζ_t)	3.0%	2.0%
Mode Shapes (V)		
Damped	$\begin{bmatrix} 1.0000 \\ 0.3403 \pm 0.0177i \end{bmatrix}$	$\begin{bmatrix} -0.0508 \mp 0.0009i \\ 1.0000 \end{bmatrix}$
Undamped	$\begin{bmatrix} 1.0000 \\ 0.3407 \end{bmatrix}$	$\begin{bmatrix} -0.0508 \\ 1.0000 \end{bmatrix}$

```
Damped GVT
 (a) damped frequency f, Hz =
    1.7856    1.7856    5.3392    5.3392
 undamped frequency f, Hz =
    1.7864    1.7864    5.3405    5.3405

 (b) coupled modal damping, z, % =
    2.9381    2.9381    2.1323    2.1323
 uncoupled modal damping, zt,zh, % =
     3     2

 (c) damped modeshapes V =
   1.0000 + 0.0000i   1.0000 + 0.0000i  -0.0508 - 0.0009i  -0.0508 + 0.0009i
   0.3403 + 0.0177i   0.3403 - 0.0177i   1.0000 + 0.0000i   1.0000 + 0.0000i
 undamped modeshapes V0 =
   1.0000 + 0.0000i   1.0000 + 0.0000i  -0.0508 + 0.0000i  -0.0508 + 0.0000i
   0.3407 - 0.0000i   0.3407 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i
```

Figure 16: MATLAB screenshot of the damped GVT results.

11.1 d. Discussion of Damped GVT Results

The introduction of structural damping has several distinct effects on the system's modal characteristics at zero airspeed. First, The damped natural frequencies (f_d) are slightly lower than their undamped counterparts (f_n). This is an expected outcome, as damping is a dissipative force that resists motion, thereby slightly reducing the speed of oscillation. The small magnitude of the reduction confirms that the system is lightly damped.

Second, the resulting modal damping ratios of the coupled system (2.94% and 2.13%) are slightly different from the uncoupled input values (3% and 2%). This is due to the inertial coupling ($x_{CP} \neq 0$). Since the system modes are linear combinations of both plunge and pitch, the damping associated with each mode is also a mixed combination of the original plunge and pitch damping.

Third, this is the most significant finding. While the undamped modes are purely real (plunge and pitch motions are perfectly in-phase), the damped modes are complex. The presence of an imaginary component signifies a phase lag between the plunge and pitch motions within each mode. This inherent phase shift is a fundamental characteristic of damped vibratory systems and is a precursor to the more complex phase relationships that drive aeroelastic flutter at non-zero airspeeds.

12 Problem D.4

This section analyzes the flutter behavior of the damped airfoil system and compares the result to the undamped case to determine the influence of structural damping on the flutter speed, U_F.

12.1 a. Damping vs. Airspeed Plot

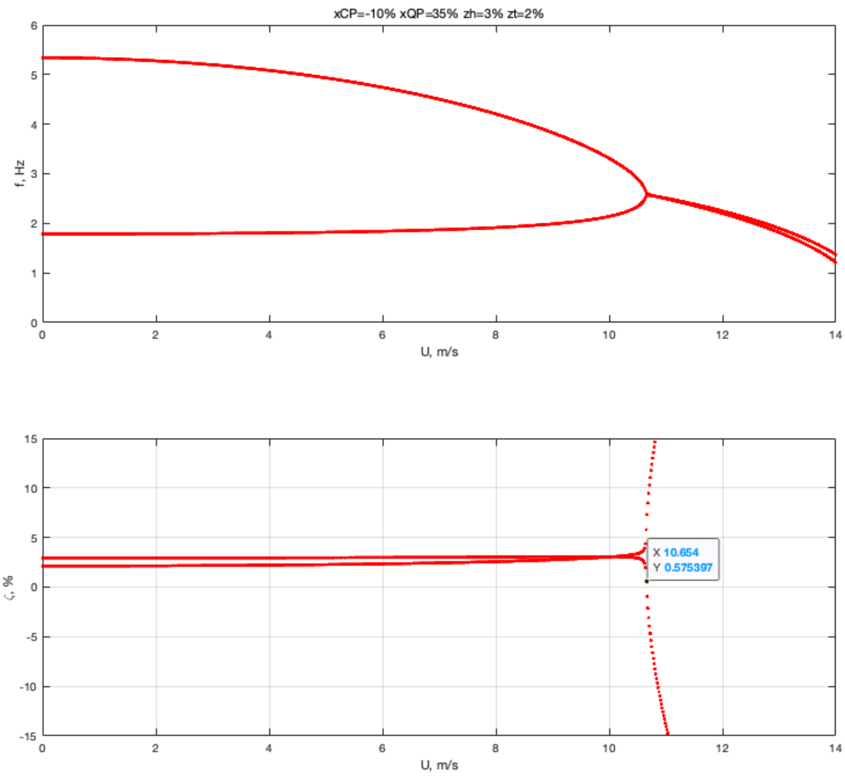


Figure 17: Damping of the system modes as a function of airspeed.

12.2 b. Damped Flutter Speed (U_F)

From the plot, the flutter speed for the damped system is identified where the damping of one of the modes crosses zero.

- $U_F(\text{damped}) = 10.654\,\mathrm{m\,s^{-1}}$
- $U_F(\text{damped}) = 10.654\,\mathrm{m\,s^{-1}} \times 1.94384 \approx 20.712\,\text{knots}$

12.3 c. Comparison with Undamped Flutter Speed

The flutter analysis for the undamped system is overlaid with the damped system to provide a direct comparison.

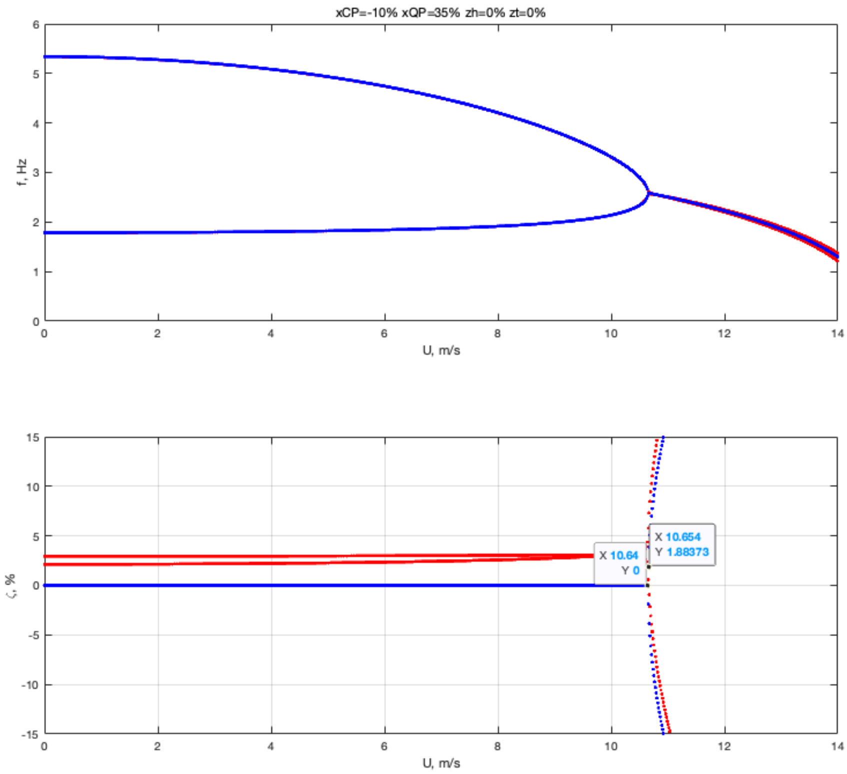


Figure 18: Overlapped damping plots for the damped (blue/orange) and undamped (yellow/purple) systems.

The flutter speed for the undamped system is found to be:

- $U_F(\text{undamped}) = 10.64\,\mathrm{m\,s^{-1}}$
- $U_F(\text{undamped}) = 10.64\,\mathrm{m\,s^{-1}} \times 1.94384 \approx 20.685\,\text{knots}$

12.4 d. Discussion

The analysis reveals a critical concept in aeroelasticity; for coalescence flutter, light structural damping has a negligible effect on the flutter speed. This is shown by the damped flutter speed ($U_F = 10.654\,\mathrm{m\,s^{-1}}$) being nearly identical to the undamped flutter speed ($U_F = 10.64\,\mathrm{m\,s^{-1}}$), with a difference of only 0.13%.

This occurs because coalescence flutter is an instability driven by the powerful interaction between inertial and aerodynamic forces. As the airspeed increases, the aerodynamic forces begin to modify the effective stiffness and damping of the system, driving the natural frequencies of two modes together. At the flutter speed, the phasing between the modes becomes such that the structure begins to extract a large amount of energy from the airflow with each cycle of oscillation.

The small amount of energy dissipated by the 2-3% structural damping is insignificant compared to the massive energy being fed into the system from the airstream. Therefore, the damping does little to resist the onset of the instability. This finding validates the common engineering practice of neglecting structural damping for preliminary flutter calculations, as it simplifies the analysis without significantly compromising the accuracy of the predicted flutter boundary.

Of course. Here is the solution for the GVT modeling problem, formatted as a LaTeX snippet that can be inserted into your document.

13 Problem E.1: GVT Modeling

This section considers the case of a Ground Vibration Test (GVT), where the airfoil is analyzed under structural-only forces with zero airspeed ($U = 0$), which implies the absence of all aerodynamic forces ($L = 0$).

13.1 a. Equations of Motion

The equations of motion for the GVT are deduced by simplifying the general aeroelastic equations of motion derived in Section A.a.

13.1.1 Assumptions

The derivation for the GVT case is based on the following assumptions:

1. **No Aerodynamic Forces:** The airspeed is zero ($U = 0$), so the lift force L and any aerodynamic moments are zero.
2. **Linear Structural System:** The restoring forces from the springs are linearly proportional to the displacement (Hooke's Law), described by stiffness constants K_h and K_θ.
3. **Rigid Body Motion:** The airfoil itself is treated as a rigid body, and its motion is fully described by the two degrees of freedom: plunge (h) and pitch (θ).
4. **Small Displacements:** The plunge displacement and pitch angle are assumed to be small, allowing for linearization of the kinematic relationships.

13.1.2 Derivation

We begin with the full equations of motion, Equations (4) and (11), and set all aerodynamic terms (L) to zero.

Plunge Equation of Motion: Setting $L = 0$ in the general plunge equation gives the physical equation of motion for plunge:

$$m\ddot{h} - mx_{CP}\ddot{\theta} + K_h h = 0 \tag{95}$$

Pitch Equation of Motion: Setting $L = 0$ in the general pitch equation gives the physical equation of motion for pitch:

$$-mx_{CP}\ddot{h} + I_p\ddot{\theta} + K_\theta\theta = 0 \tag{96}$$

13.2 b. Polynomial Eigenvalue Problem

To cast the equations into the requested matrix form, we first normalize the physical equations and then assume a harmonic solution.

13.2.1 Normalization

We normalize the plunge equation (95) by the mass, m, and the pitch equation (96) by the moment of inertia about the center of mass, I_0. Using the definitions $\omega_h^2 = K_h/m$ and $\omega_\theta^2 = K_\theta/I_0$, we get:

$$\ddot{h} - x_{CP}\ddot{\theta} + \omega_h^2 h = 0 \tag{97}$$

$$-\frac{mx_{CP}}{I_0}\ddot{h} + \frac{I_p}{I_0}\ddot{\theta} + \omega_\theta^2\theta = 0 \tag{98}$$

13.2.2 Harmonic Solution and Matrix Formulation

Assuming a solution of the form $h(t) = \hat{h}e^{st}$ and $\theta(t) = \hat{\theta}e^{st}$, the second time derivatives become $\ddot{h} = s^2\hat{h}e^{st}$ and $\ddot{\theta} = s^2\hat{\theta}e^{st}$. Substituting these into the normalized equations (97) and (98) yields:

$$(s^2 + \omega_h^2)\hat{h} - x_{CP}s^2\hat{\theta} = 0$$

$$-\frac{mx_{CP}}{I_0}s^2\hat{h} + \left(\frac{I_p}{I_0}s^2 + \omega_\theta^2\right)\hat{\theta} = 0$$

The two algebraic equations can be assembled into the matrix form $(s^2\mathbf{M}_S + \mathbf{K}_S + \mathbf{K}_A)\mathbf{x} = \mathbf{0}$:

$$s^2 \begin{bmatrix} 1 & -x_{CP} \\ -\frac{m}{I_0}x_{CP} & \frac{I_P}{I_0} \end{bmatrix} \begin{Bmatrix} \hat{h} \\ \hat{\theta} \end{Bmatrix} + \begin{bmatrix} \omega_h^2 & 0 \\ 0 & \omega_\theta^2 \end{bmatrix} \begin{Bmatrix} \hat{h} \\ \hat{\theta} \end{Bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} \hat{h} \\ \hat{\theta} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$$

Polynomial Eigenvalue Problem Matrices

The system matrices are identified as:

$$\mathbf{M}_S = \begin{bmatrix} 1 & -x_{CP} \\ -\frac{m}{I_0}x_{CP} & \frac{I_P}{I_0} \end{bmatrix} \qquad \text{(Structural Mass Matrix)}$$

$$\mathbf{K}_S = \begin{bmatrix} \omega_h^2 & 0 \\ 0 & \omega_\theta^2 \end{bmatrix} \qquad \text{(Structural Stiffness Matrix)}$$

$$\mathbf{M}_A = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad \text{(Aerodynamic Mass Matrix is zero)}$$

$$\mathbf{K}_A(U) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad \text{(Aerodynamic Stiffness Matrix is zero)}$$

14 Problem E.2

Static divergence is a non-oscillatory, aeroelastic instability where the aerodynamic twisting moment on a structure overcomes its torsional stiffness, leading to structural failure.

14.1 Condition for Static Divergence

The phenomenon is static, which implies that all time-dependent terms (velocities and accelerations) are zero. In the context of the eigenvalue problem $(s^2\mathbf{M} + s\mathbf{C} + \mathbf{K}(U))\mathbf{x} = \mathbf{0}$, this corresponds to the condition where an eigenvalue becomes zero ($s = 0$).

Setting $s = 0$ simplifies the general equation of motion to:

$$\mathbf{K}(U)\mathbf{x} = \mathbf{0} \tag{99}$$

For a non-trivial solution ($\mathbf{x} \neq \mathbf{0}$) to exist, the determinant of the total stiffness matrix, $\mathbf{K}(U)$, must be zero. This is the fundamental condition for divergence.

$$\det(\mathbf{K}(U)) = 0 \tag{100}$$

14.2 Derivation

The derivation begins by setting the determinant of the total physical stiffness matrix to zero.

$$\det(\mathbf{K}(U)) = \det\begin{pmatrix} K_h & L_0(U) \\ 0 & K_\theta - L_0(U)x_{QP} \end{pmatrix} = 0$$
$$K_h(K_\theta - L_0(U)x_{QP}) = 0$$

Since the plunge stiffness K_h is non-zero, the term in the parenthesis must be zero. This gives the physical condition for divergence: the aerodynamic moment must equal the structural restoring moment.

$$K_\theta = L_0(U)x_{QP} \tag{101}$$

For a 2D airfoil, the standard lift function per unit span is $L_0(U) = \frac{1}{2}\rho U^2 c a_1$, where c is the chord length. Substituting this into our condition gives:

$$K_\theta = \left(\frac{1}{2}\rho U^2 c a_1\right) x_{QP}$$

To express this in terms of modal parameters, we substitute the relationship for torsional stiffness, $K_\theta = I_0\omega_\theta^2$:

$$I_0\omega_\theta^2 = \frac{1}{2}\rho U^2 c a_1 x_{QP}$$

Finally, we solve for the airspeed, U, which is now defined as the divergence speed, U_D:

$$U_D^2 = \frac{2I_0\omega_\theta^2}{\rho c a_1 x_{QP}}$$
$$U_D = \sqrt{\frac{2I_0\omega_\theta^2}{\rho c a_1 x_{QP}}}$$

Factoring ω_θ out of the square root gives the final expression.

14.2.1 Final Expression for Divergence Speed

$$\boxed{U_D = \omega_\theta\sqrt{\frac{2I_0}{x_{QP} a_1 \rho c}}} \tag{102}$$

A Derivation of Uncoupled Natural Frequencies

The uncoupled natural frequencies for plunge and pitch motion are derived by simplifying the general equations of motion under the assumption of no aerodynamic forces ($U = 0$) and no mass coupling ($x_{CP} = 0$). This analysis begins by recalling the solution for a standard single-degree-of-freedom (1-DOF) system.

A.1 1-DOF Free Vibration Analysis

A simple mass-spring system is governed by the equation:

$$m\ddot{x} + kx = 0 \tag{103}$$

Assuming a solution of the form $x(t) = \hat{x}e^{st}$, the second derivative is $\ddot{x} = s^2\hat{x}e^{st} = s^2x$. Substituting this into the equation of motion yields the characteristic equation:

$$(ms^2 + k)\hat{x} = 0 \tag{104}$$

For a non-trivial solution ($\hat{x} \neq 0$), we solve for s:

$$s^2 = -\frac{k}{m} \tag{105}$$

The natural frequency, ω_n, is defined such that $\omega_n^2 = k/m$. Therefore:

$$s^2 = -\omega_n^2 \implies s = \pm i\omega_n \tag{106}$$

The general solution describes simple harmonic motion:

$$x(t) = C_1e^{i\omega_n t} + C_2e^{-i\omega_n t} = C\cos(\omega_n t + \psi) \tag{107}$$

A.2 Uncoupled Plunge Natural Frequency

For the uncoupled plunge motion of the airfoil, the equation of motion simplifies to:

$$m\ddot{h} + K_h h = 0 \tag{108}$$

By direct analogy with the 1-DOF system, where mass m is the airfoil mass and K_h is the plunge spring stiffness, the plunge natural frequency squared is:

$$\omega_h^2 = \frac{K_h}{m} \tag{109}$$

The frequency in Hertz is given by $f_h = \omega_h/(2\pi)$.

A.3 Uncoupled Pitch Natural Frequency

For the uncoupled pitch motion, the equation of motion simplifies to:

$$I_o\ddot{\theta} + K_\theta\theta = 0 \tag{110}$$

This equation is the rotational analogue of the 1-DOF system. The term I_o represents the mass moment of inertia of the airfoil about the axis of rotation. It is important to note the distinction between I_P, the moment of inertia about a general reference point P, and I_o. For this uncoupled case, we assume the center of mass lies on the axis of rotation ($x_{CP} = 0$), and thus we define:

$$I_o = I_P\big|_{x_{CP}=0} \tag{111}$$

By analogy, where I_o is the rotational inertia and K_θ is the torsional stiffness, the pitch natural frequency squared is:

$$\omega_\theta^2 = \frac{K_\theta}{I_o} \tag{112}$$

The frequency in Hertz is given by $f_\theta = \omega_\theta/(2\pi)$.

B Mathematical Derivation of the Linearized Lift Force

The aerodynamic lift force, L, is defined by the standard lift equation, which relates the force to the fluid properties and flow conditions through a dimensionless coefficient. The equation is given by:

$$L = \frac{1}{2}\rho U^2 S C_L \tag{113}$$

where ρ is the fluid density, U is the freestream velocity, S is the reference wing area, and C_L is the lift coefficient. The term $\frac{1}{2}\rho U^2$ is the dynamic pressure, q.

The lift coefficient, C_L, is primarily a function of the angle of attack, θ. For analytical tractability, we linearize this function, $C_L(\theta)$, using a first-order Taylor series expansion about a reference angle $\theta_0 = 0$:

$$C_L(\theta) \approx C_L(0) + \left.\frac{dC_L}{d\theta}\right|_{\theta=0} (\theta - 0) \tag{114}$$

For a symmetric airfoil, there is no lift at zero angle of attack, thus $C_L(0) = 0$. This simplifies the approximation to:

$$C_L(\theta) \approx \left(\left.\frac{dC_L}{d\theta}\right|_{\theta=0} \right) \theta \tag{115}$$

We define the constant derivative term as the lift curve slope, a_1:

$$a_1 \equiv \frac{dC_L}{d\theta} \tag{116}$$

This yields the linearized lift coefficient model, which holds for small θ:

$$C_L = a_1 \theta \tag{117}$$

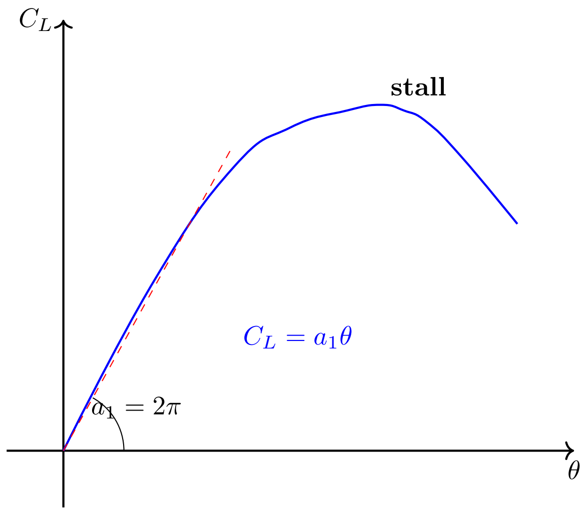


Figure 19: Lift coefficient (C_L) versus angle of attack (θ), illustrating the linear region for small θ and the subsequent rounded stall behavior.

Substituting the linear model from Equation 117 into the general lift equation (Equation 113) gives the final expression for the linearized lift force:

$$L = \frac{1}{2}\rho U^2 S a_1 \theta \tag{118}$$

This formulation is predicated on several key assumptions, which are listed and explained below.

B.1 Fundamental Assumptions for Linearized Lift Theory

1. **Linear Dependence on Angle of Attack**

   The lift coefficient varies linearly with angle of attack: $C_L = a_1\theta$, where a_1 is the lift curve slope. This assumption is valid for small angles of attack before the onset of flow separation ($\theta < \theta_{\text{stall}}$).

2. **Symmetric Airfoil Geometry**

   The analysis assumes a symmetric airfoil with no zero-lift angle: $C_L(0) = 0$. For a cambered airfoil, a zero-lift offset would be included: $C_L = C_{L0} + a_1\theta$.

3. **Ideal Lift Curve Slope**

   From thin airfoil theory, the theoretical lift curve slope for a 2D airfoil is $a_1 = 2\pi$ rad^{-1}. This represents the ideal, inviscid case and provides an upper bound for real airfoils.

4. **Subsonic and Incompressible Flow**

   The flow is assumed to be incompressible with constant density $\rho =$ constant. This is a valid approximation for Mach numbers $M \ll 0.3$, typical for low-speed aeroelastic analysis.

5. **Rigid Airfoil Structure**

   The airfoil is assumed to maintain its shape under aerodynamic loading (no structural deformation beyond the prescribed rigid-body motions). This enables linear analysis by separating structural and aerodynamic effects.

6. **Fully Attached Airflow**

   The flow remains attached over the entire airfoil surface with no flow separation, stall, or vortex shedding. This assumption is valid only for small angles of attack and excludes post-stall behavior.

7. **Neglect of Drag Effects**

   Aerodynamic drag forces are assumed negligible compared to lift forces for the purpose of this analysis. This is typically valid for high-aspect-ratio wings at small angles of attack.

8. **Small Angle Approximation**

   The angle of attack θ (in radians) is assumed to be small such that:

$$\sin\theta \approx \theta \tag{119}$$

$$\cos\theta \approx 1 \tag{120}$$

   This enables linearization of the aerodynamic relationships and is typically valid for $|\theta| < 0.2$ radians ($\approx 11°$).

C Mathematical Details of Exponential Solutions

This appendix provides the complete mathematical justification for the time derivative relationships used in the eigenvalue analysis.

C.1 Derivation of Time Derivatives for Exponential Functions

Consider the general exponential function:

$$f(t) = Ae^{st} \tag{121}$$

where A is a complex constant and s is a complex parameter.

First Derivative: Using the chain rule:

$$\frac{d}{dt}[Ae^{st}] = A\frac{d}{dt}[e^{st}] = A \cdot s \cdot e^{st} = s \cdot (Ae^{st}) = s \cdot f(t) \tag{122}$$

Second Derivative: Applying the derivative operator again:

$$\frac{d^2}{dt^2}[Ae^{st}] = \frac{d}{dt}[s \cdot Ae^{st}] = s \cdot \frac{d}{dt}[Ae^{st}] = s \cdot (s \cdot Ae^{st}) = s^2 \cdot (Ae^{st}) = s^2 \cdot f(t) \tag{123}$$

C.2 Application to Displacement Functions

For our airfoil system with:

$$h(t) = \hat{h}e^{st} \tag{124}$$

$$\theta(t) = \hat{\theta}e^{st} \tag{125}$$

Plunge Motion Derivatives:

$$\dot{h}(t) = \frac{d}{dt}[\hat{h}e^{st}] = s\hat{h}e^{st} = s \cdot h(t) \tag{126}$$

$$\ddot{h}(t) = \frac{d^2}{dt^2}[\hat{h}e^{st}] = s^2\hat{h}e^{st} = s^2 \cdot h(t) \tag{127}$$

Pitch Motion Derivatives:

$$\dot{\theta}(t) = \frac{d}{dt}[\hat{\theta}e^{st}] = s\hat{\theta}e^{st} = s \cdot \theta(t) \tag{128}$$

$$\ddot{\theta}(t) = \frac{d^2}{dt^2}[\hat{\theta}e^{st}] = s^2\hat{\theta}e^{st} = s^2 \cdot \theta(t) \tag{129}$$

C.3 Physical Interpretation

The parameter s has the physical interpretation:

- $s = i\omega$ for purely oscillatory motion at frequency ω
- $s = \sigma + i\omega$ for oscillatory motion with exponential growth/decay
- $\sigma > 0$: unstable (growing) motion
- $\sigma < 0$: stable (decaying) motion
- $\sigma = 0$: neutrally stable oscillation

This mathematical framework allows us to determine system stability by examining the eigenvalues s of the characteristic equation.

D Detailed Review of 1-DOF Damped Vibration

To understand the origin of the modal parameter relationships used in the main body of this work, it is essential to first review the classic single-degree-of-freedom (1-DOF) mass-spring-damper system. This foundational model provides the physical intuition behind concepts like natural frequency and damping ratio.

D.1 The Governing Equation of Motion

The system consists of a mass (m) connected to a fixed wall by a linear spring (with stiffness k) and a viscous damper (with damping coefficient c).

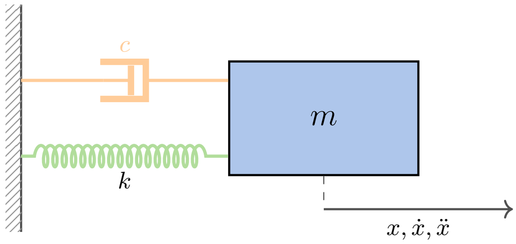


Figure 20: A standard 1-DOF mass-spring-damper system.

According to Newton's Second Law ($\sum F = ma$), the sum of the forces acting on the mass must equal its mass times acceleration. The forces are the spring force ($-kx$), the damping force ($-c\dot{x}$), and the inertial force ($-m\ddot{x}$). For dynamic equilibrium, the sum is zero:

$$m\ddot{x} + c\dot{x} + kx = 0 \tag{130}$$

This is a second-order linear homogeneous ordinary differential equation.

D.2 The Characteristic Equation

To solve this equation, we assume a solution of the form $x(t) = \hat{x}e^{st}$, where $\hat{x}$ is a constant amplitude and s is a complex number. The derivatives are:

$$\dot{x}(t) = s \cdot \hat{x}e^{st}$$
$$\ddot{x}(t) = s^2 \cdot \hat{x}e^{st}$$

Substituting these into the equation of motion gives:

$$m(s^2\hat{x}e^{st}) + c(s\hat{x}e^{st}) + k(\hat{x}e^{st}) = 0$$
$$(ms^2 + cs + k)\hat{x}e^{st} = 0 \qquad \text{(Factor out common terms)}$$

For a non-trivial solution (where motion occurs, so $\hat{x}e^{st} \neq 0$), the term in the parenthesis must be zero. This gives the characteristic equation:

$$ms^2 + cs + k = 0 \tag{131}$$

D.3 Solving for the Roots

This is a simple quadratic equation for the variable s. We can solve it using the quadratic formula, $s = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$:

$$s_{1,2} = \frac{-c \pm \sqrt{c^2 - 4mk}}{2m} \tag{132}$$
$$= -\frac{c}{2m} \pm \frac{\sqrt{c^2 - 4mk}}{2m} \qquad \text{(Separate the terms)} \tag{133}$$
$$= -\frac{c}{2m} \pm \sqrt{\frac{c^2 - 4mk}{4m^2}} \qquad \text{(Bring denominator inside the root)} \tag{134}$$
$$= -\frac{c}{2m} \pm \sqrt{\left(\frac{c}{2m}\right)^2 - \frac{k}{m}} \qquad \text{(Simplify the expression)} \tag{135}$$

These two roots, s_1 and s_2, completely define the behavior of the system.

D.4 Introducing Modal Parameters

To simplify and generalize this result, we define two critical modal parameters:

1. **The Undamped Natural Frequency (ω_n):** This is the frequency at which the system would oscillate if there were no damping ($c = 0$). It is defined as:

$$\omega_n = \sqrt{\frac{k}{m}}$$

2. **The Damping Ratio (ζ):** This is a non-dimensional number that describes how much damping is present compared to the "critical" amount needed to prevent oscillation. It is defined as:

$$\zeta = \frac{c}{c_{cr}} = \frac{c}{2\sqrt{mk}}$$

Using our definition of ω_n, we can rewrite this as $\zeta = \frac{c}{2m\omega_n}$. From this, we can express the term $\frac{c}{2m}$ as:

$$\frac{c}{2m} = \zeta\omega_n$$

D.5 Final Form of the Roots

By substituting these modal parameters back into the equation for the roots, we get a much cleaner expression:

$$s_{1,2} = -\zeta\omega_n \pm \sqrt{(\zeta\omega_n)^2 - \omega_n^2} \tag{136}$$

$$= -\zeta\omega_n \pm \sqrt{\omega_n^2(\zeta^2 - 1)} \qquad \text{(Factor out } \omega_n^2\text{)} \tag{137}$$

$$= -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1} \qquad \text{(Final form of the roots)} \tag{138}$$

For most structural systems, damping is light ($\zeta < 1$), which makes the term inside the square root negative. We can then write the solution using $i = \sqrt{-1}$:

$$s_{1,2} = -\zeta\omega_n \pm i\omega_n\sqrt{1 - \zeta^2} \tag{139}$$

$$= -\zeta\omega_n \pm i\omega_d \qquad \text{(where } \omega_d = \omega_n\sqrt{1-\zeta^2} \text{ is the damped natural frequency)} \tag{140}$$

This analysis provides the fundamental link between the direct physical parameters of a system (m, c, k) and the modal parameters (ω_n, ζ) used to describe its vibratory nature.

E MATLAB Code

E.1 MAIN file

```
%% Initialization
clc                % clear command window
clear              % clear workspace
format compact
set(0,'DefaultFigureWindowStyle','docked')
nfig=1;
tol=1e-10;    %Tolerance for discarding machine zero

%% Choose what section to run
ifSectionB_GVT=0; % Perform GVT analysis

ifSectionC_Flutter=0; %perform flutter analysis
    ifC2basicFlutter=0; % perform basic flutter analysis
    ifC3Zoom=0; % zoom around flutter
    ifC4UFvsCP=0; % plot multiple times to find UF for various xCP
    ifC5UFvsQP=0; % plot multiple times to find UF for various xQP
    ifC6Div=0; % calculate and plot divergence speed

ifSectionD_Damping=1; % perform analysis with damping
    ifDampedGVT=0; % perform damped GVT analysis
    ifDampedFlutter=1; % perform damped Flutter analysis

%% Input Data Homework 1
rho=1.225; % air density, 1.225 kg/m3
a1=2*pi; % ideal lift curve slope value

% ------ data from homework MAIN ---------
c=0.45; % airfoil chord, m
m=3.2; % mass, kg
I0=0.055; % moment of inertia about the center of mass, kg*m^2
fh=1.8; % plunge frequency, Hz
ft=5.3; % pitch frequency, Hz
CPratioDefault=-10e-2; % static offset
QPratioDefault=35e-2; % aerodynamic offset

% ------ data from homework C.2 ---------
UendUF=14; % flutter diagram
NU=1001; % length of airspeed

% ------ data from homework C.4 ---------
UstartCP=8; UendCP=16; % xCP variation airspeed range
CPratioRange=[-20 -15 -10 -5 -1 -0.1]/100; N_CP=length(CPratioRange);
%CPratioRange=[20 15 10 5 1 0.1 0]/100; N_CP=length(CPratioRange); %Part e

% ------ data from homework C.5 ---------
UstartQP=8; UendQP=13; % xQP variation airspeed range
QPratioRange=[25 30 35]/100; N_QP=length(QPratioRange);

% ------ data from homework C.6 ---------
UendDiv=20; % flutter diagram for divergence analysis

% ------ data from homework D.2 ---------
zh=3e-2; % plunge damping ratio
zt=2e-2; % pitch damping ratio


%%  Section B: GVT
if ifSectionB_GVT
display('Section B. Ground Vibration Test (GVT) Analysis')
close all % close all figures
display('  (a) input data')
fprintf('  c=%2.1fm, m=%2.1f kg/m, I0=%5.4f kg*m^2/m \n',c,m,I0)
fprintf('  uncoupled frequencies fh=%2.1f Hz, ft=%2.1f Hz \n',fh,ft)
display(' ')
GVT(c,m,I0,fh,ft,CPratioDefault) % run function GVT
display(' ')
display('  (e) Every student should insert here discussion of results')
end % ifGVT ends here
```

```


%% Section C: FLUTTER ANALYSIS
if ifSectionC_Flutter
display('Section C. Flutter Eigen Analysis')
close all % close all figures
%% Display input data
CPratio=CPratioDefault; xCP=CPratio*c;
QPratio=QPratioDefault; xQP=QPratio*c;
display(' input data')
fprintf(' rho=%4.3f kg/m^3 (air density) \n',rho)
fprintf(' c=%2.1fm, m=%2.1f kg/m, I0=%5.4f kg*m^2/m \n',c,m,I0)
fprintf(' uncoupled frequencies fh=%2.1f Hz, ft=%2.1f Hz \n',fh,ft)
fprintf(' static offset xCP=%2.1f%%, %5.4f m \n',CPratio*100,xCP)
fprintf(' aerodynamic offset xQP=%2.1f%%, %5.4f m \n',QPratio*100,xQP)
%% CALCULATE STRUCTURAL MATRICES MS, KS
wt=2*pi*ft; wh=2*pi*fh;
Ip=I0+m*xCP^2; % moment of inertia about the elastic center,kg*m^2/m
MS=[ 1          -xCP ;
    -m/I0*xCP   Ip/I0 ] ; % structural mass matrix
KS=[wh^2          0 ;
     0            wt^2 ] ; % structural stiffness matrix
%% DEFINE AERODYNAMIC LIFT FUNCTION
L0=@(UU) rho*UU^2/2*c*a1; % Lift function for a generic speed UU
%% DEFINE AIRSPEED RANGE
Ustart=0; Uend=UendUF; % flutter diagram
if ifC6Div; Uend=UendDiv; end % extended diagram
U=linspace(Ustart, Uend, NU); % airspeed range
if ifC3Zoom;
% Read UF and create airspeed range around UF
UF=input('enter flutter speed read on the flutter diagram UF=');
eps=1e-2;U=[0 6 (1-eps)*UF UF (1+eps)*UF];
end
NU=length(U);
%% Run basic flutter analysis
[r,f,z,sigma,v] = eigenFlutter(rho,a1,c,m,I0,xQP,U,MS,0,KS);
if ifC2basicFlutter
% if NU > 10; % only plot if more than ten points to plot
display([' Ustart=' num2str(Ustart) ' m/s, Uend=' num2str(Uend) 'm/s,' ...
        ' NU=' num2str(NU)])
close all % close all figures
plotTitle={['xCP=' num2str(CPratio*1e2) '%' 'xQP=' num2str(QPratio*1e2) '% ']};
figure; plot_f_z(U,f,z,plotTitle)
end
if ifC3Zoom; displayUfv(U,f,v); end % print zoom-in values if needed
%% UF variation with xCP
if ifC4UFvsCP
close all % close all figures
figure(1);
% xCP variation airspeed range
Ustart=UstartCP; Uend=UendCP; NU=1001; U=linspace(Ustart,Uend,NU);
display([' Ustart=' num2str(Ustart) ' m/s, Uend=' num2str(Uend) 'm/s,' ...
' NU=' num2str(NU)])
QPratio=QPratioDefault; xQP=QPratio*c; % aerodynamic offset
display([' QPratio=' num2str(QPratio*100) '%'])
display(CPratioRange*100,' CPratioRange %')
%% LOOP OVER ALL CPratio values
for nCP=1:N_CP
CPratio=CPratioRange(nCP); xCP=CPratio*c; % static offset
Ip=I0+m*xCP^2; % moment of inertia about the elastic center,kg*m^2/m
MS=[ 1        -xCP ;
    -m/I0*xCP      Ip/I0 ] ; % structural mass matrix
[r,f,z,sigma,v] = eigenFlutter(rho,a1,c,m,I0,xQP,U,MS,0,KS);
plotTitle={['xQP=' num2str(QPratio*1e2) '%, various xCP']};
plot_f_z(U,f,z,plotTitle)
hold on
%% identify UF values
fprintf(' CPratio=%0.1f%% \n', CPratio*100)
display('put datatips on plot to identify UF ')
display('when done, press any key into the command window to continue')
pause
end % nCP loop ends here
UF_CP=input(['enter [UF1, UF2, UF3, ...] values ' ...
```

```
'from datatips on plot in the order of appearance=']);
display(' ')
%% plot UF vs xCP%
display(CPratioRange*100, ' xCP %')
display(UF_CP, ' UF, m/s')
figure(2);
plot(CPratioRange*100,UF_CP,'-r');
xlim([-20 20]); ylim([0 Uend]); xlabel('xCP, %'); ylabel('U_F, m/s')
title(['U_F vs. x_C_P for x_Q_P=' num2str(QPratio*100) ' %'],...
'FontSize', 11, 'FontWeight', 'normal')
grid on
end % ifUFvsCP ends here
%% UF variation with xQP
if ifC5UFvsQP
close all % close all figures
% xQP variation airspeed range
Ustart=UstartQP; Uend=UendQP; NU=1001; U=linspace(Ustart,Uend,NU);
display([' Ustart=' num2str(Ustart) ' m/s, Uend=' num2str(Uend) 'm/s,' ...
' NU=' num2str(NU)])
CPratio=CPratioDefault; xCP=CPratio*c; % static offset
display([' CPratio=' num2str(CPratio*100) '%'])
Ip=I0+m*xCP^2; % moment of inertia about the elastic center,kg*m^2/m
MS=[ 1 -xCP ;
-m/I0*xCP Ip/I0 ] ; % structural mass matrix
figure(1);
display([' QPratioRange % = ' num2str(QPratioRange*100)])
%% LOOP OVER ALL QPratio values
for nQP=1:N_QP
QPratio=QPratioRange(nQP); xQP=QPratio*c; % static offset
[r,f,z,sigma,v] = eigenFlutter(rho,a1,c,m,I0,xQP,U,MS,0,KS);
plotTitle={['xCP=' num2str(CPratio*1e2) '%, various xQP']};
plot_f_z(U,f,z,plotTitle)
hold on
%% identify UF values
fprintf(' QPratio=%0.0f%% \n', QPratio*100)
display('put datatips on plot to identify UF ')
display('when done, press any key into the command window to continue')
pause
end % NxQP loop ends here
UF_QP=zeros(1,N_QP);
UF_QP=input(['enter [UF1, UF2, UF3, ...] values ' ...
'from datatips on plot in the order of appearance=']);
display(' ')
%% plot UF vs xQP%
display(QPratioRange*100, ' xQP %')
display(UF_QP, ' UF, m/s')
figure(2);
plot(QPratioRange*100,UF_QP,'-b');
xlim([20 30]); ylim([0 Uend]); xlabel('xQP, %'); ylabel('U_F, m/s')
title(['U_F vs. x_Q_P for x_C_P=' num2str(CPratio*100) ' %'],...
'FontSize', 11, 'FontWeight', 'normal')
grid on
end % ifUFvsQP ends here
%% calculate and plot divergence speed UD
if ifC6Div
display([' Ustart=' num2str(Ustart) ' m/s, Uend=' num2str(Uend) 'm/s,' ...
' NU=' num2str(NU)])
close all % close all figures
% ---- (a) ----
QPratio=QPratioDefault; xQP=QPratio*c;
[r,f,z,sigma,v] = eigenFlutter(rho,a1,c,m,I0,xQP,U,MS,0,KS);
figure
plotTitle={['xCP=' num2str(CPratio*1e2) '% ' ...
'xQP=' num2str(QPratio*1e2) '%']};
plot_f_sigmaExtended(U,f,sigma,plotTitle)
% ---- (b) ----
QPratio=[20 25 30]*1e-2;
xQP=QPratio*c;
UD=wt*sqrt(2*I0./(xQP*a1*rho*c));
% display(' ')
display(QPratio*100,'QPratio, %')
display(UD,'divergence speed UD, m/s')
figure
```

```
plot(QPratio*1e2,UD,'-b')
ylim([0 Uend])
xlabel('x_Q_P, %')
ylabel('U_D, m/s')
title('U_D vs. x_Q_P', 'FontSize', 11, 'FontWeight', 'normal')
grid on
end % ifDiv ends here
end % ifFlutter ends here



%% Section D: include damping
if ifSectionD_Damping
display('Section D. Damped GVT and Flutter')
close all % close all figures
%% display input data
display('input data')
if ifDampedFlutter; fprintf(' rho=%4.3f kg/m^3 (air density) \n',rho); end
fprintf(' c=%2.1fm, m=%2.1f kg/m, I0=%5.4f kg*m^2/m \n',c,m,I0)
fprintf(' uncoupled frequencies fh=%2.1f Hz, ft=%2.1f Hz \n',fh,ft)
fprintf(' damping ratios zh=%2.0f%%, zt=%2.0f%%, \n',[zh, zt]*100)
CPratio=CPratioDefault; xCP=CPratio*c;
Ip=I0+m*xCP^2; % moment of inertia about the elastic center,kg*m^2/m
fprintf(' static offset xCP=%2.1f%%, %5.4f m \n',CPratio*100,xCP)
QPratio=QPratioDefault; xQP=QPratio*c; % aerodynamic offset
fprintf(' aerodynamic offset xQP=%2.1f%%, %5.4f m \n',QPratio*100, xQP)
%% CALCULATE STRUCTURAL MATRICES MS, CS, KS
wt=2*pi*ft; wh=2*pi*fh;
MS=[ 1 -xCP ;
-m/I0*xCP Ip/I0 ] ; % structural mass matrix
CS=[2*zh*wh 0 ;
0 2*zt*wt ] ; % structural viscous damping
KS=[wh^2 0 ;
0 wt^2 ] ; % structural stiffness matrix
%% DAMPED GVT
if ifDampedGVT
dampedGVT(zh,zt,MS,CS,KS) % run dampedGVT function
end % ifDampedGVT ends here
%% DAMPED FLUTTER
if ifDampedFlutter
display('Damped Flutter')
close all % close all figures
%% DEFINE AIRSPEED RANGE
Ustart=0; Uend=UendUF; % flutter diagram
NU=1001; U=linspace(Ustart,Uend,NU); % airspeed range
%% Calculate and plot damped flutter diagram
figure
[r,f,z,sigma,v]=eigenFlutter(rho,a1,c,m,I0,xQP,U,MS,CS,KS);
plotTitle={[['xCP=' num2str(CPratio*1e2) '% ' 'xQP=' num2str(QPratio*1e2) '%'] ...
[' zh=' num2str(zh*1e02) '%'] [' zt=' num2str(zt*1e02) '%']]};
plotOptions='.r';
plot_f_z_Damped(U,f,z,plotTitle,plotOptions)
%% Identify damped UF values
display('put datatips on plot to identify damped UF ')
display('when done, press any key into the command window to continue')
pause
%% Overlap Undamped Plot
zh=0; zt=0; % undamped case
CS=[2*zh*wh 0 ;
0 2*zt*wt ] ; % structural viscous damping
[r,f,z,sigma,v]=eigenFlutter(rho,a1,c,m,I0,xQP,U,MS,CS,KS);
plotTitle={[['xCP=' num2str(CPratio*1e2) '% ' 'xQP=' num2str(QPratio*1e2) '%'] ...
[' zh=' num2str(zh*1e02) '%'] [' zt=' num2str(zt*1e02) '%']]};
plotOptions='.b';
plot_f_z_Damped(U,f,z,plotTitle,plotOptions)
end % ifDampedFlutter ends here
end % ifDamping ends here
%% Finish
display(' ')
display(' ---------------')
display (['success! ' mfilename ' finished successfully'])
```

Listing 1: Main analysis script for Homework 1.

E.2 dampedGVT.m

```
function dampedGVT(zh,zt,MS,CS,KS)
%{
Summary of this function goes here
Detailed explanation goes here
%}
%% CALCULATE EIGENVALUES: use polyeig to get eigenvectors and, eigenvalues
[V_raw,s_raw] = polyeig(KS,CS,MS);
[V,s]=sort_norm_eig(V_raw,s_raw);
% display(s,'sorted eigenvalues s');
% display(V,'sorted and normalized eigenvectors V');
%% EXTRACT DAMPING AND FREQ. FROM REAL AND IMAG PARTS OF s
% zz=real(s)./abs(s); z=-sort(zz); % damping
z=-real(s)./abs(s); % damping
f=abs(imag(s))/(2*pi); % frequencies
%% CALCULATE UNDAMPED FREQUENCIES AND MODESHAPES
[V_raw0,s_raw0] = polyeig(KS,0,MS);
[V0,s0]=sort_norm_eig(V_raw0,s_raw0);
f0=abs(imag(s0))/(2*pi); % undamped frequencies
%% DISPLAY FREQUENCY AND DAMPING
display(' ')
display('Damped GVT')
display(f,' (a) damped frequency f, Hz')
display(f0,' undamped frequency f, Hz')
display(' ')
display(z*100,' (b) coupled modal damping, z, %')
display([zh,zt]*100,' uncoupled modal damping, zt,zh, %')
%% DISPLAY MODESHAPES
display(' ')
display(V,' (c) damped modeshapes V')
display(V0,' undamped modeshapes V0')


end % function ends here
```

Listing 2: Helper script for Homework 1.

E.3 displayUfv.m

```
function displayUfv(U,f,v)
display('frequencies and modeshapes at various airspeeds')
NU=length(U);
for jU=1:NU
display(['U=' num2str(U(jU)) ' m/s'])
display(f(:,jU)',' frequency, Hz')
display(v(:,:,jU),' modeshapes')
% display(' ')
end


end % function ends here
```

Listing 3: Helper script for Homework 1.

E.4 eigenFlutter.m

```
function[r,f,z,sigma,v] = eigenFlutter(rho,a1,c,m,I0,xQP,U,MS,CS,KS)
tol=1e-10; % tolerance for discarding machine zero
NU=length(U);
%% define aerodynamic lift function
L0=@(UU) rho*UU^2/2*c*a1; % Lift function for a generic speed UU
%% clear space
r=zeros(4,NU); v=zeros(2,4,NU); f=zeros(4,NU); z=zeros(4,NU); sigma=zeros(4,NU);
for jU=1:NU
%% define flutter matrices
MA=zeros(2,2); % aerodynamic mass matrix MA=0
CA=zeros(2,2); % aerodynamic damping matrix,CA=0
% UU=U(jU);
% L=L0(UU);
KA=[0 L0(U(jU))/m ;
0 -xQP*L0(U(jU))/I0 ] ; % aerodynamic stiffness matrix
M=MS+MA; % system mass matrix
C=CS+CA; % system viscous damping matrix
K=KS+KA; % system stiffness matrix
%% calculate eigenvalues: use polyeig to get eigenvectors V and eigenvalues s
[V_raw,s_raw] = polyeig(K,C,M);
[V,s]=sort_norm_eig(V_raw,s_raw);
% display(s,'sorted eigenvalues s')
% display(V,'sorted and normalized eigenvectors V');
%% store eigenvalues and eigenvectors
r(:,jU)=s;
%% extract damping and freq. from real and imaginary parts of s
ff=abs(imag(s))/(2*pi); % frequencies
% zz=-real(s)./abs(s); zz=zz.*(abs(zz)>tol);
sig=real(s); sig=sig.*(abs(sig)>tol);
zz=-sig./abs(s); % damping
[~,Iz]=sort(sig,'descend'); % sort damping in descending order
    for i=1:4 % store frequencies and damping values
    f(i,jU)=ff(Iz(i));
    z(i,jU)=zz(Iz(i));
    sigma(i,jU)=real(s(Iz(i)));
    v(:,i,jU)=V(:,Iz(i));
    end
end % jU loop ends here


end % function ends here
```

Listing 4: Helper script for Homework 1.

E.5 GVT.m

```
function GVT(c,m,I0,fh,ft,CPratioDefault)
wh=2*pi*fh; wt=2*pi*ft;
CPratio=CPratioDefault; % static offset as % of chord
%% Calculate stiffnesses from uncoupled angular freq. wh, wt
Kh=m*wh^2; % plunge spring stiffness, N/m
Kt=I0*wt^2; % pitch spring stiffness, N*m/rad
display(' (b) calculate spring stiffnesses')
fprintf(' plunge stiffness Kh=%2.1f N/m \n',Kh)
fprintf(' pitch stiffness Kt=%3.2f N*m/rad \n',Kt)
display(' ')
CPratio=CPratioDefault; % static offset as % of chord
xCP=CPratio*c; % static offset value
fprintf(' (c) static offset xCP=%2.1f%%, %5.4f m \n',CPratio*100,xCP)
Ip=I0+m*xCP^2; % moment of inertia about the elastic center, kg*m^2
fprintf(' moment of inertia Ip= %5.4f m \n',Ip)
MS=[ 1 -xCP ;
-m/I0*xCP Ip/I0 ]; % structural mass matrix
KS=[wh^2 0 ;
0 wt^2 ]; % structural stiffness matrix
display(MS,' structural mass matrix MS')
display(KS,' structural stiffness matrix KS')
% use polyeig to get eigenvectors V and eigenvalues s
[V_raw,s_raw] = polyeig(KS,0,MS);
[V,s]=sort_norm_eig(V_raw,s_raw);
display(s,' sorted eigenvalues s')
display(V,' sorted and normalized eigenvectors V');
ff=abs(imag(s))/(2*pi);
[fc,If,~]=unique(ff,'stable'); % Frequencies
fprintf(' coupled frequencies fI=%1.4f Hz, fII=%1.4f Hz \n',fc)
fprintf(' uncoupled frequencies fh=%1.4f Hz, ft=%1.4f Hz \n',fh,ft)
fun=[fh,ft];
df=(fc-fun)./fun;
fprintf(' frequency diff dfI=%1.4f%%, dfII=%1.4f%% \n',df*100)
V_modes=V(:,If); % modeshapes
display(V_modes,' modeshapes, V_I, V_II')
display(' ')
display(' (d) other xCP values ')
CPratio=[-20 -10 -1 0 1 10 20]*1e-2; % static offset as % of chord
N_CP=size(CPratio,2);
display (CPratio*100,'CPratio %')
xCP=CPratio*c; % static offset value
display (xCP,'xCP, m')
Ip=I0+m*xCP.^2; % moment of inertia about the elastic center, kg*m^2
display(Ip,' Ip, kg*m^2')
f=zeros(2,N_CP); df=zeros(2,N_CP); V_modes=zeros(2,2,N_CP);
for nCP=1:N_CP
MS=[ 1 -xCP(nCP);
-m/I0*xCP(nCP) Ip(nCP)/I0 ];
KS=[wh^2 0 ;
0 wt^2 ];
[V_raw,s_raw] = polyeig(KS,0,MS);
[V,s]=sort_norm_eig(V_raw,s_raw);
ff=abs(imag(s))/(2*pi); [fc,If,~]=unique(ff,'stable');
dff=(fc-fun)./fun;
f(:,nCP)=fc; % frequencies
df(:,nCP)=dff; % freq diff
V_m=V(:,If);
V_modes(:,:,nCP)=V_m; % modeshapes
end
display(f,' coupled frequencies, Hz')
display(df*100,' df%')
V_mode1(:,:)=V_modes(:,1,:);
display(V_mode1,' modeshape V_I')
V_mode2(:,:)=V_modes(:,2,:);
display(V_mode2,' modeshape V_II')
end % function ends here
```

Listing 5: Helper script for Homework 1.

E.6 plot_f_sigmaExtended.m

```
function plot_f_sigmaExtended(U,f,sigma,plotTitle)
%{
plot extended flutter diagram f and sigma vs U to identify divergence speed UD
%}
NU=length(U);
subplot(2,1,1); plot(U,f,'.r');
title(plotTitle, 'FontSize', 11,'FontWeight','normal')
xlabel('U, m/s'); ylabel('f, Hz');
fmax=ceil(max(max(f))); ylim([0 fmax]);
% ylim([2.5 3])
xlim([U(1) U(NU)]);
hold on
subplot(2,1,2);
plot(U,sigma,'.g');
xlim([U(1) U(NU)]); xlabel('U, m/s');
ymax=15; ylim([-ymax ymax]); ylabel('sigma=real(s)');
grid on
hold on
end                   % Function ends here
```

Listing 6: Helper script for Homework 1.

E.7 plot_f_z_Damped.m

```
function plot_f_z_Damped(U,f,z,plotTitle,plotOptions)
NU=length(U);
subplot(2,1,1); plot(U,f,plotOptions);
title(plotTitle, 'FontSize', 11,'FontWeight','normal')
xlabel('U, m/s'); ylabel('f, Hz');
fmax=ceil(max(max(f))); ylim([0 fmax]);
% ylim([2.5 3])
xlim([U(1) U(NU)]);
hold on
subplot(2,1,2); plot(U,z*1e2,plotOptions);
xlim([U(1) U(NU)]); xlabel('U, m/s');
ymax=15; ylim([-ymax ymax]); ylabel('\zeta, %');
grid on
hold on

end % function ends here
```

Listing 7: Helper script for Homework 1.

E.8 plot_f_z.m

```
function plot_f_z(U,f,z,plotTitle)
NU=length(U);
subplot(2,1,1); plot(U,f,'.r');
title(plotTitle, 'FontSize', 11,'FontWeight','normal')
xlabel('U, m/s'); ylabel('f, Hz');
fmax=ceil(max(max(f))); ylim([0 fmax]);
xlim([U(1) U(NU)]);
hold on
subplot(2,1,2); plot(U,z*1e2,'.g');
xlim([U(1) U(NU)]); xlabel('U, m/s');
ymax=15; ylim([-ymax ymax]); ylabel('\zeta, %');
grid on
hold on

end % function ends here
```

Listing 8: Helper script for Homework 1.

E.9 sort_norm_eig.m

```
function [X_sorted_normalized,e_sorted] = sort_norm_eig(X,e)
%{
X(N,Ne) = matrix of Ne eigenvectors each of N dofs
e(Ne) = row of Ne eigenvalues
Procedure:
    sort eigenvalues e in magnitude order and stores into es
    reorder the eigenvector X and stores into Xs
    normalize the sorted eigenvectors Xs to get Xsn
    such that the largest element in each eigenvector is = 1
%}
N=size(X,1); Ne=size(X,2); % pick sizes N, Ne
e_abs=abs(e); % pick up abs values
% [~,Is]=sort(e_abs,'descend'); % sort in descending order
[~,Is]=sort(e_abs,'ascend'); % sort in ascending order
% Is contains the sorted indices
%% store sorted eigenvalues and eigenvectors
e_sorted=zeros(1,Ne); Xs=zeros(N,Ne);
for ne=1:Ne; e_sorted(ne)=e(Is(ne)); Xs(:,ne)=X(:,Is(ne)); end %
% display(e_sorted,'sorted eigenvalues')
% display(Xs,'sorted eigenvectors')
%% normalize eigenvectors to make +ve the largest element in each column
Xabs=abs(Xs);
[~,IX]=max(Xabs,[],1);
X_sorted_normalized=zeros(N);
for j=1:Ne;
scale=sign(Xs(IX(j),j))*max(abs(Xs(:,j)));
X_sorted_normalized(:,j)=Xs(:,j)/scale;
end

end % function ends here
```

Listing 9: Helper script for Homework 1.