Toronto Metropolitan University

Department of Aerospace Engineering
Faculty of Engineering & Architectural Science

Semester (Term, Year)	Fall, 2024
Course Code	AER715
Course Title	Aeroelasticity
Course Instructor	Hekmat Alighanbari
Title	Project 2
Submission Due Date	November 27, 2024
Submission Date	November 27, 2024

Submission by (Name):	Student ID (XXXX1234)	Signature
Sharvani Yadav	XXXXX8658	S.Y
Daniel Mielnik	XXXXX8927	D.M
Alexia Economou	XXXXX5924	A.E

By signing the above you attest that you have contributed to this submission and confirm that all work you contributed to this submission is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a "0" on the work, an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Academic Integrity Policy 60, which can be found at www.torontomu.ca/senate/policies/ Aerospace Assignment Cover as of May 2024

Abstract

In this report, a two-dimensional, two-degree-of-freedom aeroelastic model designed for wind tunnel testing will be analyzed under Newtonian and Lagrangian mechanics methods. Using a free-body diagram and these two methods, a 2x2 matrix was developed to express the equations of motion about the mid-chord. Computational programming was then used through MATLAB to determine varying properties of the airfoil.

Through MATLAB, the given set of parameters were used along with the assumption of incompressible flow to find the critical flutter speed at 72.038 m/s. Additional assumptions included a density of $1.225 kg/m^3$ and a maximum speed of 100 m/s. Using the equations of motion, the real and imaginary parts of eigenvalues were plotted, displaying the patterns of damping and frequency characteristics. Prior to critical speed, the eigenvalues were found to be stable, while past the critical speed the eigenvalues present imaginary components, proving instability. The relationship between the critical speed and spring constants was plotted for k_1, k_2 and $k_{\theta 1}$, where it was determined that varying spring constant had a large effect on critical speed.

Optimization was then used to determine the maximum critical speed of the airfoil while adhering to specific constants. The optimized configuration was found as $k_1 = 4500\, N/m$, $k_2 = 5500\, N/m$, and $k_{\theta 1} = 600\, Nm/rad$, with a critical speed of $U_{Max} = 198.94\, m/s$. The location of the point mass was changed from the leading edge to the trailing edge of the foil, and plotted against varying critical speeds. To maximize the flutter speed, the distance of the point mass was found as 0.15 m. Lastly, the effect of mechanical damping was observed through plotting the critical speed against the damping constants.

Table of Contents

1.0 Introduction

This project examines a two-dimensional, two-degree-of-freedom aeroelastic model designed for wind tunnel testing, focusing on the dynamic interplay between aerodynamic forces, structural stiffness, and damping. The system features a thin airfoil of unit span and chord, equipped with translational and torsional springs and dampers, enabling a comprehensive analysis of its dynamic behavior. Critical to this study are the governing equations of motion, derived using both Newtonian and Lagrangian mechanics, which offer insights into the system's response under various loading conditions.

To analyze the system, key parameters such as the airfoil's mass distribution, stiffness, and damping characteristics were incorporated into the mathematical model. Computational tools were then employed to determine critical aeroelastic phenomena, including flutter and divergence speeds. By systematically varying parameters such as spring constants and damping coefficients, the study aims to optimize the model for stability and performance, providing practical recommendations for real-world applications.

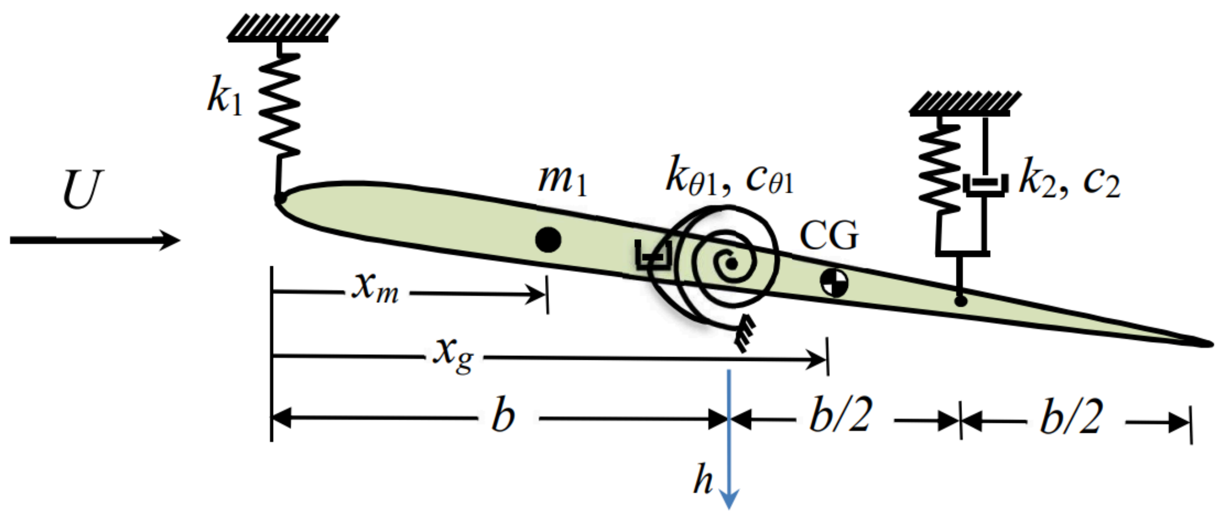

Figure 1: Aeroelasticity Schematic Model [1]

2.0 Computational Simulation and Programming

Given the two-dimensional, two-degree-of-freedom aeroelastic model, a free-body diagram was developed in which the equations of motion could be determined. Regarding the equations of motion, both methods considered distances measured from the elastic axis of the airfoil. It was also assumed that the moment about the aerodynamic center occurs at quarter chord of the airfoil.

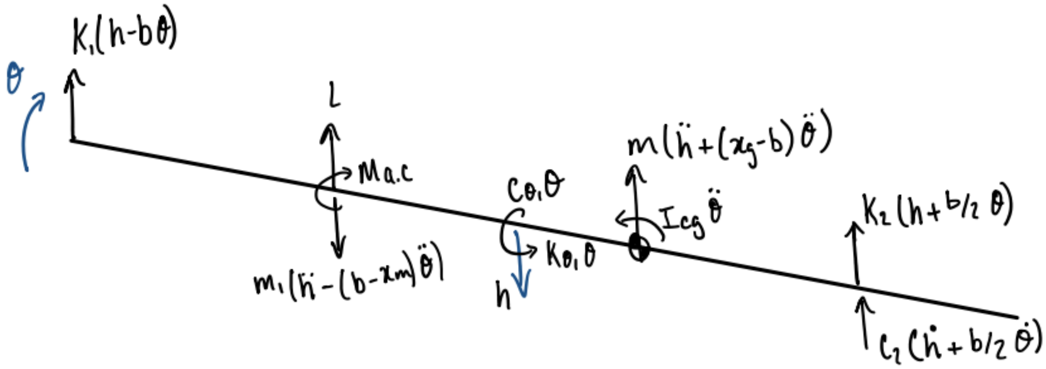

Figure 2: Aeroelastic System Free-Body Diagram

The equations of motion for the aeroelastic system are derived using two approaches: Newtonian and the Lagrangian mechanics method. The subsequent section provides the detailed derivation of the equations and the construction of the matrix, beginning with the Newtonian method. The final matrix will be presented in the following format:

$$[M]\begin{Bmatrix}\ddot{h}\\ \ddot{\theta}\end{Bmatrix}+\left([B_s]+f(U)[B_a]\right)\begin{Bmatrix}\dot{h}\\ \dot{\theta}\end{Bmatrix}+\left([E]+f(U)[K]\right)\begin{Bmatrix}h\\ \theta\end{Bmatrix}=\begin{Bmatrix}0\\ 0\end{Bmatrix} \qquad [1]$$

Quasi-Steady Incompressible Aerodynamic for Lift and Moment Equations

$$C_L = 2\pi qsc\left(\theta + \frac{\dot{h}}{V} + \frac{\theta b}{2V}\right) \qquad [2]$$

$$L = 2\pi qsc\left(\theta + \frac{\dot{h}}{V} + \frac{\theta b}{2V}\right) \qquad [3]$$

$$C_{M_{ac}} = -\frac{\pi b}{4V}\theta \qquad [4]$$

$$M_{ac} = 2\pi qsc\left(-\frac{\pi\theta b}{4V}\right) \qquad [5]$$

2.1 Newtonian Equations Method

The Newtonian equations method is used to find the equations of motion through taking the sum of forces acting in the vertical direction and the sum of moments about the quarter chord. These two equations can then be simplified and put in a 2x2 matrix form for the aeroelastic system.

Summation of forces acting in the vertical direction:

$$\Sigma F = 0$$

$$m\left(\ddot{h} + (x_g - b)\ddot{\theta}\right) + k_1(h - b\theta) + k_2\left(h + \frac{b}{2}\theta\right) + C_2\left(\dot{h} + \frac{b}{2}\theta\right) + m_1(\ddot{h} + (b - x_m)\ddot{\theta}) + L = 0 \qquad [6]$$

Expanding equation:

$$m\ddot{h} + m(x_g - b)\ddot{\theta} + k_1 h - k_1 b\theta + k_2 h + k_2\frac{b}{2}\theta + C_2\dot{h} + C_2\frac{b}{2}\theta - m_1(b - x_m)\ddot{\theta} + 2\pi qsc\theta + 2\pi qsc\frac{h}{V} + 2\pi qsc\frac{\theta b}{2V} = 0$$

Combining like terms:

$$m\ddot{h} + m(x_g - b)\ddot{\theta} + k_1 h - k_1 b\theta + k_2 h + k_2\frac{b}{2}\theta + C_2\dot{h} + C_2\frac{b}{2}\dot{\theta} - m_1\ddot{h} + m_1(b - x_m)\ddot{\theta} + 2\pi qsc\theta + 2\pi qsc\frac{h}{V} + 2\pi qsc\frac{\dot{\theta}b}{2V} = 0$$

Moment about quarter chord:

$$\curvearrowleft + \Sigma M = 0$$

$$k_2\left(\frac{b}{2}\right)\left(h + \frac{b}{2}\theta\right) + I_{cg}\ddot{\theta} + m(x_g - b)\left[\ddot{h} + \left(x_g - b\right)\ddot{\theta}\right] + C_2\left(\frac{b}{2}\right)\left(\dot{h} + \frac{b}{2}\theta\right) + k_{\theta_1}\theta - M_{ac} - m_1(b - x_m)\left(\ddot{h} - x_m\ddot{\theta}\right) - k_1 b(h - b\theta) + L\left(\frac{b}{2}\right) + C_{\theta_1}\theta = 0 \qquad [7]$$

Equation written in the form $a\ddot{h} + b\ddot{\theta} + c\dot{h} + d\theta + eh + f\theta = 0$:

$$\left[m(x_g - b) - m_1(b - x_m)\right]\ddot{h} + \left[m\left(x_g - b\right)^2 + I_{cg} + m_1(b - x_m)^2\right]\ddot{\theta} + \left[C_2\left(\frac{b}{2}\right) - 2\pi qsc\frac{b}{2V}\right]\dot{h} +$$

$$\left[C_2\left(\frac{b}{2}\right)^2 - 2\pi qsc\frac{\pi b}{4V} + C_{\theta_1} + \frac{2\pi qsc}{V}\right]\theta + \left[k_2\frac{b}{2} - k_1 b\right]h + \left[k_2\frac{b}{2}^2 + k_{\theta_1} - k_1 b^2 - 2\pi qsc\frac{b}{2}\right]\theta = 0$$

Equation of motion in the written form:

$$\begin{bmatrix} m + m_1 & m\,(x_g - b) - m_1(b - x_m) \\ m(x_g - b) - m_1(b - x_m) & m(x_g - b)^2 + m_1(b - x_m)^2 + I_{cg} \end{bmatrix} \begin{Bmatrix} \ddot{h} \\ \ddot{\theta} \end{Bmatrix} +$$

$$\left(\left(\begin{bmatrix} C_2 & C_2 \\ C_2\left(\frac{b}{2}\right) & C_2\left(\frac{b}{2}\right) \end{bmatrix}\right) + \frac{2\pi qsc}{U}\begin{bmatrix} 1 & \frac{b}{2} \\ \frac{-b}{2} & 0 \end{bmatrix}\right)\begin{Bmatrix} \dot{h} \\ \dot{\theta} \end{Bmatrix}$$

$$+\begin{bmatrix} k_1 + k_2 & k_2\left(\frac{b}{2}\right) - k_1 b \\ k_2\left(\frac{b}{2}\right) - k_1 b & k_2\left(\frac{b}{2}\right)^2 + k_1 b^2 + k_{\theta_1} \end{bmatrix}\begin{Bmatrix} h \\ \theta \end{Bmatrix} + 2\pi qsc\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$$

Lagrange Mechanics Method

The Lagrangian mechanics method takes the kinetic energy, potential energy, and work of the system to develop a 2x2 matrix for the aeroelastic system. For each equation, the partial derivatives are determined which are then combined for each Lagrange equation. .

Kinetic Energy of the system:

$$T = \frac{1}{2}\int_0^s \left\{ m\left(\frac{\partial H}{\partial t} + x_a b\frac{\partial \theta}{\partial t}\right)^2 + I_p\left(\frac{\partial \theta}{\partial t}\right)^2 \right\} dy \qquad [8]$$

$$T = \frac{1}{2}m\left(\dot{h} + (x_g - b)\dot{\theta}\right)^2 + \frac{1}{2}\left(\dot{h} + (b - x_m)\dot{\theta}\right)^2 + \frac{1}{2}I_{cg}\dot{\theta}^2$$

Expand and simplify equation:

$$T = \frac{1}{2}m\dot{h}^2 + \frac{1}{2}m2\dot{h}(x_g - b)\dot{\theta} + \frac{1}{2}m\left(x_g - b\right)^2\dot{\theta}^2 + \frac{1}{2}m_1\dot{h}^2 + \frac{1}{2}m_1 2\dot{h}\left(b - x_m\right)\dot{\theta}$$
$$+ \frac{1}{2}m_1\left[\left(b - x_m\right)\dot{\theta}\right]^2 + \frac{1}{2}I_{cg}\dot{\theta}^2$$

$$T = \frac{1}{2}\left(m + m_1\right)\dot{h}^2 + \left[m\left(x_g - b\right) + m_1\left(b - x_m\right)\right]\dot{h}\theta + \frac{1}{2}\left[m\left(x_g - b\right)^2 + m_1\left(b - x_m\right)^2 + I_{cg}\right]\theta^2$$

Partial derivatives of kinetic energy equation:

$$\frac{\partial T}{\partial \dot{h}} = \left(m + m_1\right)\dot{h} + \left[m\left(x_g - b\right) + m_1\left(b - x_m\right)\right]\theta$$

$$\frac{\partial}{\partial T}\left(\frac{\partial T}{\partial \dot{h}}\right) = \left(m + m_1\right)\ddot{h} + \left[m\left(x_g - b\right) + m_1\left(b - x_m\right)\right]\ddot{\theta}$$

$$\frac{\partial T}{\partial h} = 0$$

$$\frac{\partial T}{\partial \theta} = \left[m\left(x_g - b\right) - m_1\left(b - x_m\right)\right]\dot{h} + \left[m\left(x_g - b\right)^2 + m_1\left(b - x_m\right)^2 + I_{cg}\right]\dot{\theta}$$

$$\frac{\partial}{\partial T}\frac{\partial T}{\partial \theta} = \left[m\left(x_g - b\right) - m_1\left(b - x_m\right)\right]\ddot{h} + \left[m\left(x_g - b\right)^2 + m_1\left(b - x_m\right)^2 + I_{cg}\right]\ddot{\theta}$$

$$\frac{\partial T}{\partial \theta} = 0$$

Potential Energy of the system:

$$V = \frac{1}{2}K_2\left(h + \frac{b}{2}\theta\right)^2 + \frac{1}{2}K_1(h - b\theta)^2 + \frac{1}{2}K_{\theta 1}\theta^2 \qquad [9]$$

Expand and simplify equation:

$$V = \frac{1}{2}K_2\left[h^2 + 2h\theta\left(\frac{b}{2}\right) + \left(\frac{b}{2}\right)^2\theta^2\right] + \frac{1}{2}K_1\left[h^2 - 2h\theta b + b^2\theta^2\right] + \frac{1}{2}K_{\theta 1}\theta^2$$

$$V = \frac{1}{2}K_2h^2 + K_2h\theta\left(\frac{b}{2}\right) + \frac{1}{2}K_2\left(\frac{b}{2}\right)^2\theta^2 + \frac{1}{2}K_1h^2 - K_1h\theta b + \frac{1}{2}K_1b^2\theta^2 + \frac{1}{2}K_{\theta 1}\theta^2$$

$$V = \frac{1}{2}h^2\left[K_2 + K_1\right] + \frac{1}{2}\theta^2\left[K_2\left(\frac{b}{2}\right)^2 + K_1b^2\right] + h\theta\left[K_2\left(\frac{b}{2}\right) - K_1b\right] + \frac{1}{2}K_{\theta 1}\theta^2$$

Partial derivatives of potential energy equation:

$$\frac{\partial V}{\partial h} = K_2h + K_1h + K_2\left(\frac{b}{2}\right)\theta - K_1b\theta$$

$$\frac{\partial V}{\partial h} = \left(K_1 + K_2\right)h + \left(K_2\left(\frac{b}{2}\right) - K_1 b\right)\theta$$

$$\frac{\partial V}{\partial \theta} = \left[K_2\left(\frac{b}{2}\right)^2 + K_1 b^2 + K_{\theta 1}\right]\theta + \left[K_2\left(\frac{b}{2}\right) - K_1 b\right]h$$

Work of the system:

$$\Delta W = M_{ac}\theta - Lh - C_2\left(\dot{h} + \frac{b}{2}\dot{\theta}\right)\left(h + \frac{b}{2}\theta\right) - C_{\theta 1}\dot{\theta} \qquad [10]$$

$$\Delta W = M_{ac}\theta - Lh - C_2\dot{h}h - C_2\dot{h}\left(\frac{b}{2}\right)\theta - C_2 h\left(\frac{b}{2}\right)\dot{\theta} - C_2\left(\frac{b}{2}\right)^2\dot{\theta}\theta - C_{\theta 1}\dot{\theta}$$

Partial derivatives of the system work:

$$Q_h = \frac{\partial \Delta W}{\partial h} = - L - C_2\dot{h} - C_2\left(\frac{b}{2}\right)\dot{\theta}$$

$$Q_\theta = \frac{\partial \Delta W}{\partial \theta} = M_{ac} - C_2\dot{h}\left(\frac{b}{2}\right) - C_2\left(\frac{b}{2}\right)^2\dot{\theta} - C_{\theta 1}\dot{\theta}$$

Substituting the calculated parameters into final Lagrange equation:

$$\frac{\partial}{\partial t}\left(\frac{\partial T}{\partial \dot{h}}\right) - \frac{\partial T}{\partial h} + \frac{\partial V}{\partial h} = Q_h$$

$$\left(m + m_1\right)\ddot{h} + \left[m\left(x_g - b\right) + m_1\left(b - x_m\right)\right]\ddot{\theta} + \left[K_1 + K_2\right]h + \left[K_2\left(\frac{b}{2}\right) - K_1 b\right]\theta = - L - C_2\dot{h} - C_2\left(\frac{b}{2}\right)\dot{\theta}$$

$$\left(m + m_1\right)\ddot{h} + \left[m\left(x_g - b\right) + m_1\left(b - x_m\right)\right]\ddot{\theta} + C_2\dot{h} + \left[C_2\left(\frac{b}{2}\right)\right]\dot{\theta} + \left[K_1 + K_2\right]h + \left[K_2\left(\frac{b}{2}\right) - K_1 b\right]\theta = - L$$

$$\frac{\partial}{\partial t}\left(\frac{\partial T}{\partial \dot{\theta}}\right) - \frac{\partial T}{\partial \theta} + \frac{\partial V}{\partial \theta} = Q_\theta$$

$$\left[m\left(x_g - b\right) + m_1\left(b - x_m\right)\right]\ddot{h} + \left[m\left(x_g - b\right)^2 + m_1\left(b - x_m\right)^2 + I_{cg}\right]\ddot{\theta} + \left[K_2\left(\frac{b}{2}\right)^2 + K_1 b^2 + K_{\theta 1}\right]\theta$$
$$+ \left[K_2\left(\frac{b}{2}\right) - K_1 b\right]h = M_{ac} - C_2\dot{h}\left(\frac{b}{2}\right) - C_2\left(\frac{b}{2}\right)^2\dot{\theta} - C_{\theta 1}\dot{\theta}$$

$$\left[m\left(x_g - b\right) + m_1\left(b - x_m\right)\right]\ddot{h} + \left[m\left(x_g - b\right)^2 + m_1\left(b - x_m\right)^2 + I_{cg}\right]\ddot{\theta} + \left[C_2\left(\frac{b}{2}\right)\right]\dot{h}$$

$$+\left[C_2\left(\frac{b}{2}\right)^2 + C_{\theta 1}\right]\theta + \left[K_2\left(\frac{b}{2}\right) - K_1 b\right]h + \left[K_2\left(\frac{b}{2}\right)^2 + K_1 b^2 + K_{\theta 1}\right]\theta = M_{ac}$$

Equation of motion in matrix form:

$$\begin{bmatrix} m + m_1 & m\,(x_g - b) - m_1(b - x_m) \\ m(x_g - b) - m_1(b - x_m) & m(x_g - b)^2 + m_1(b - x_m)^2 + I_{cg} \end{bmatrix} \begin{Bmatrix} \ddot{h} \\ \ddot{\theta} \end{Bmatrix} +$$

$$\left(\left(\begin{bmatrix} C_2 & C_2 \\ C_2\left(\frac{b}{2}\right) & C_2\left(\frac{b}{2}\right) \end{bmatrix}\right) + \frac{2\pi q s c}{U}\begin{bmatrix} 1 & \frac{b}{2} \\ \frac{-b}{2} & 0 \end{bmatrix}\right) \begin{Bmatrix} \dot{h} \\ \dot{\theta} \end{Bmatrix}$$

$$+\begin{bmatrix} k_1 + k_2 & k_2\left(\frac{b}{2}\right) - k_1 b \\ k_2\left(\frac{b}{2}\right) - k_1 b & k_2\left(\frac{b}{2}\right)^2 + k_1 b^2 + k_{\theta_1} \end{bmatrix} \begin{Bmatrix} h \\ \theta \end{Bmatrix} + 2\pi q s c \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$$

Observing Newtonian and Lagrangian mechanics methods, it was determined that both 2x2 matrices match. Shown in the following sections, this matrix is used as the equations of motion for the aeroelastic system to determine a series of values.

3.0 Calculations and Solutions

3.1 Airfoil Critical Speeds

This task entailed creating a MATLAB program to solve the aeroelastic equations and find the critical speeds (flutter or divergence) for the airfoil with the following parameters:

Table 1: Airfoil Parameters [1]

Parameter	Value
c	0.5 m
k_1	$5 \frac{kN}{n}$
k_2	$1 \frac{kN}{n}$
$k_{\theta 1}$	$500 \frac{Nm}{rad}$
$c_{\theta 1}$	$0.0 \frac{Nms}{rad}$
c_1	$0.0 \frac{N.s}{n}$
I_{cg}	$0.05\ kg\ m^2$
m	5 kg
m_1	$2\ kg$
x_m	0.15 m
x_g	0.15 m

These were the following assumptions that were made:

- Flow is assumed to be incompressible.
- Maximum speed is $U_{Max} = 100 \frac{m}{s}$
- Density is $\rho = 1.225 \frac{kg}{m^3}$

For the given aeroelastic system, the equations of motion were derived using both Newtonian mechanics and Lagrangian mechanics, considering the translational and rotational dynamics of the airfoil. These equations were formulated in a matrix form to represent the aeroelastic interactions. The general form of the equations is:

$$\begin{bmatrix} m + m_1 & m\,(x_g - b) - m_1(b - x_m) \\ m(x_g - b) - m_1(b - x_m) & m(x_g - b)^2 + m_1(b - x_m)^2 + I_{cg} \end{bmatrix} \begin{Bmatrix} \ddot{h} \\ \ddot{\theta} \end{Bmatrix} +$$

$$\left(\left(\begin{bmatrix} C_2 & C_2 \\ C_2\left(\frac{b}{2}\right) & C_2\left(\frac{b}{2}\right) \end{bmatrix} \right) + \frac{2\pi qsc}{U} \begin{bmatrix} 1 & \frac{b}{2} \\ \frac{-b}{2} & 0 \end{bmatrix} \right) \begin{Bmatrix} \dot{h} \\ \dot{\theta} \end{Bmatrix}$$

$$+ \begin{bmatrix} k_1 + k_2 & k_2\left(\frac{b}{2}\right) - k_1 b \\ k_2\left(\frac{b}{2}\right) - k_1 b & k_2\left(\frac{b}{2}\right)^2 + k_1 b^2 + k_{\theta_1} \end{bmatrix} \begin{Bmatrix} h \\ \theta \end{Bmatrix} + 2\pi qsc \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$$

The determinant of the matrix leads to the following equation:

$$p_4\lambda^4 + p_3\lambda^3 + p_2\lambda^2 + p_1\lambda + p_0 = 0$$

Using the p values and Routh's Stability criteria can be found using the equation below:

$$T_3 = p_1 p_2 p_3 - p_1^2 p_4 - p_0 p_3^2 \geq 0$$

The critical speed $U_{Critical}$ was computed by solving the equation above for the coupled equations. At the critical speed, the real part of at least one eigenvalue transitions to zero, indicating the onset of flutter or divergence. The calculated critical speed is:

$$U_{Critical\ Flutter} = 72.038\ m/s$$

This value represents the minimum speed at which aeroelastic instability (flutter) occurs for the baseline configuration. This speed was obtained by analyzing the eigenvalues of the system matrices and ensuring that all calculations adhered to the assumptions of quasi-steady, incompressible aerodynamics.

3.2 Variation of Real and Imaginary Parts of Eigenvalues

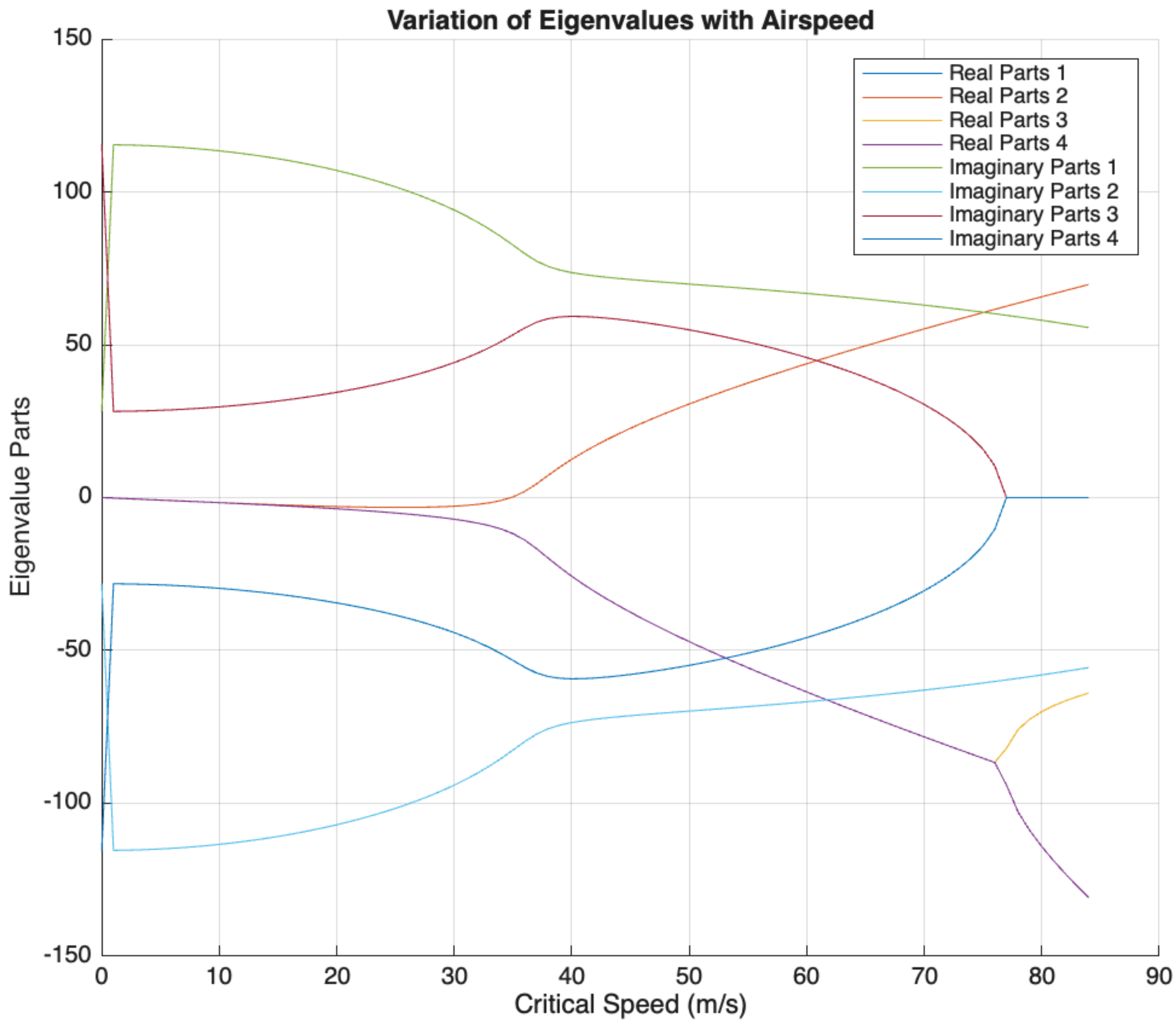

Figure 3: Variation of Eigenvalues versus Critical Speed

The dynamic stability of the aeroelastic system was analyzed by examining the variation of eigenvalues with airspeed. The system's eigenvalues were calculated using the equations of motion derived in Question 2, considering the effects of structural, aerodynamic, and inertial forces. The plot (Figure 3) shows the real and imaginary parts of the eigenvalues as functions of airspeed, up to 1.2 times the critical speed $U_{Critical} = 72.038\, m/s$.

The real parts of the eigenvalues provide information about the damping characteristics of the system, while the imaginary parts indicate the frequency of oscillatory modes. As airspeed increases, a noticeable separation occurs around $U_{Critical}$, where the eigenvalues transition from purely real to complex conjugate pairs. This transition signifies the onset of flutter, an aeroelastic instability characterized by oscillatory motion. Below the critical speed, the eigenvalues are stable (negative real parts), indicating that any disturbances in the system will decay over time.

Above the critical speed, the presence of imaginary components coupled with positive real parts indicates sustained oscillations, leading to dynamic instability.

The results align with theoretical expectations, demonstrating that the critical speed serves as a threshold for stable operation. The bifurcation of eigenvalues provides a clear visual marker of flutter onset, reinforcing the importance of accurately predicting this phenomenon for safe and effective system design.

3.3 Magnitude of Spring Constants

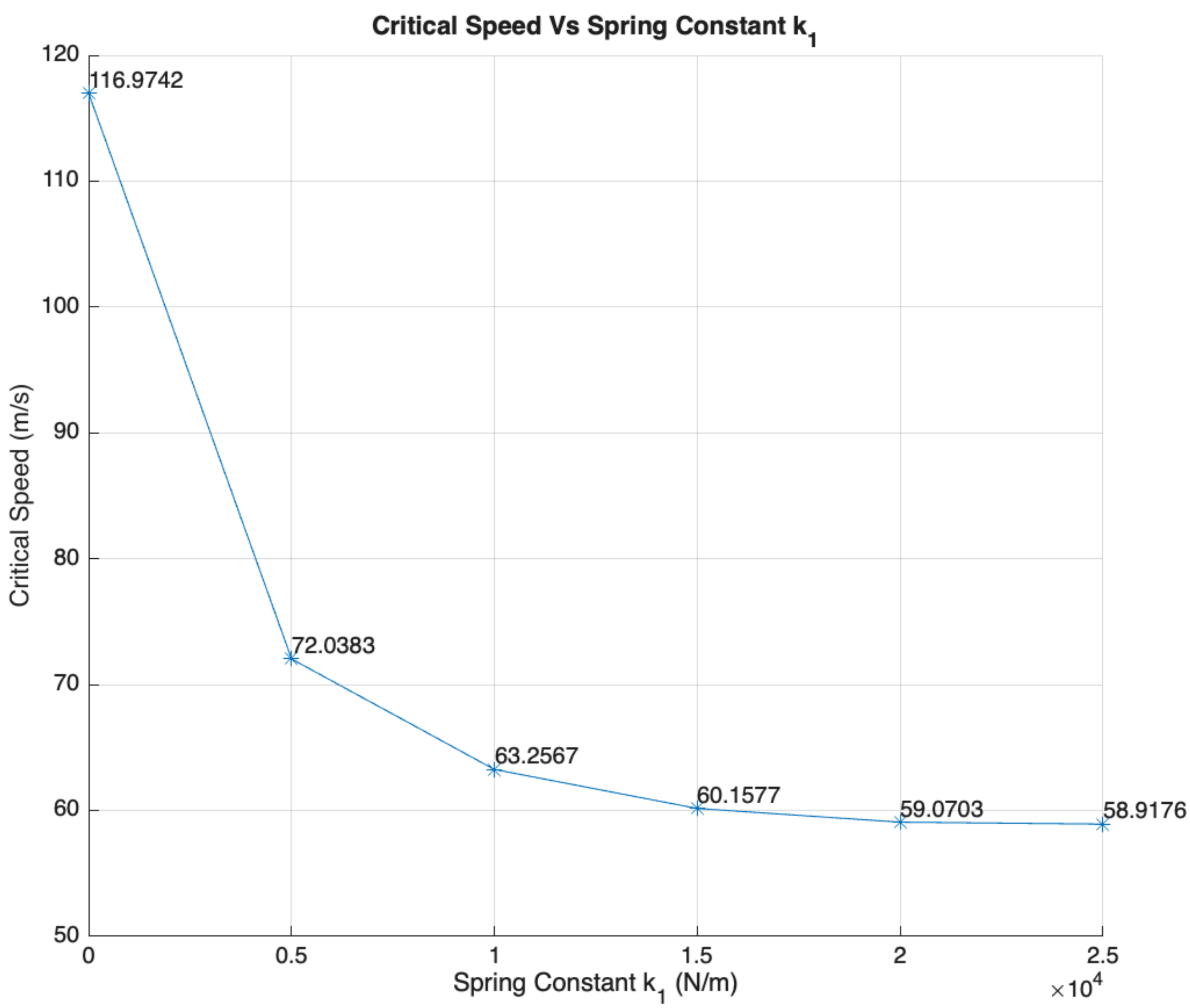

Figure 4: Varying Critical Speed versus Spring Constant k_1

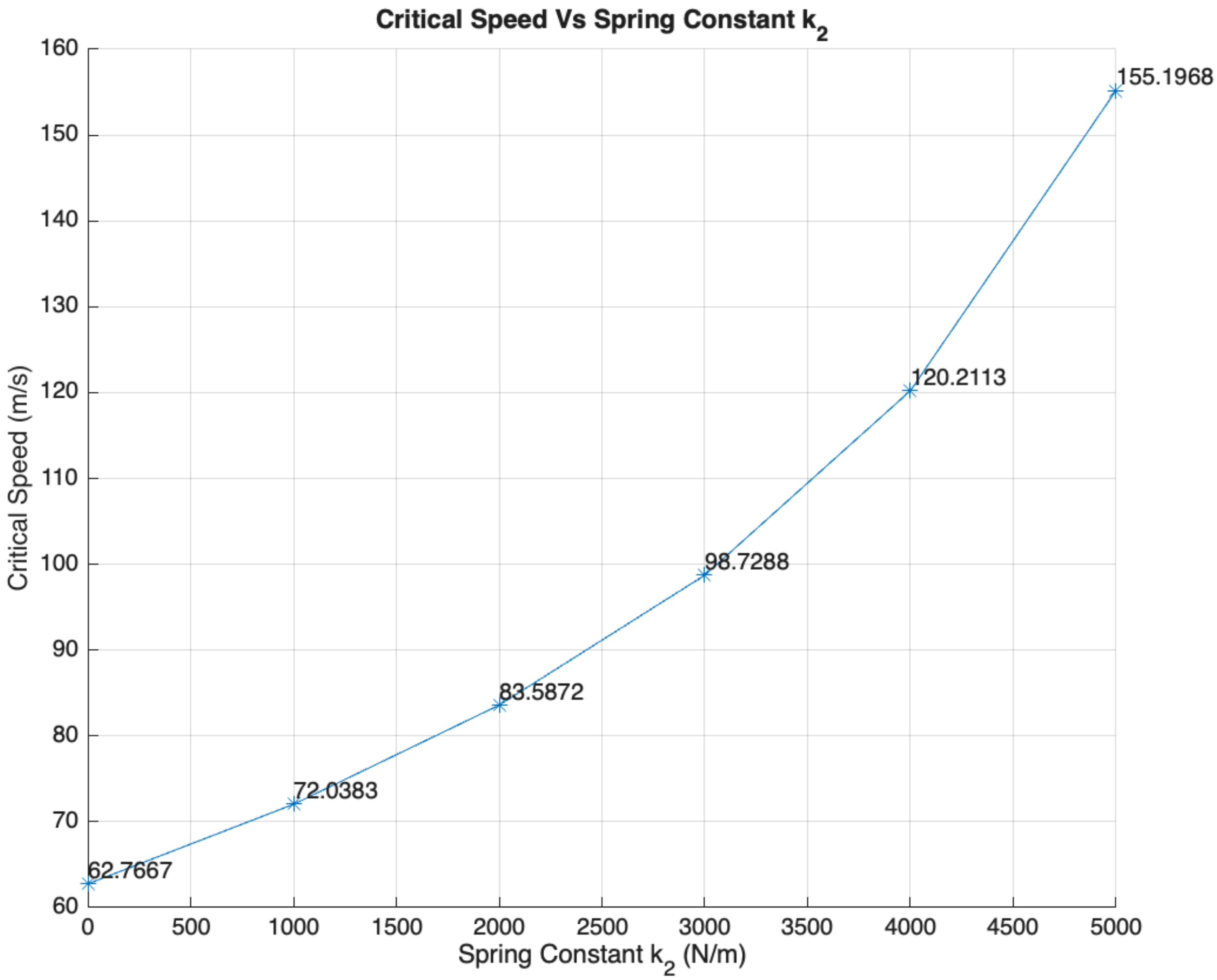

Figure 5: Varying Critical Speed versus Spring Constant k_2

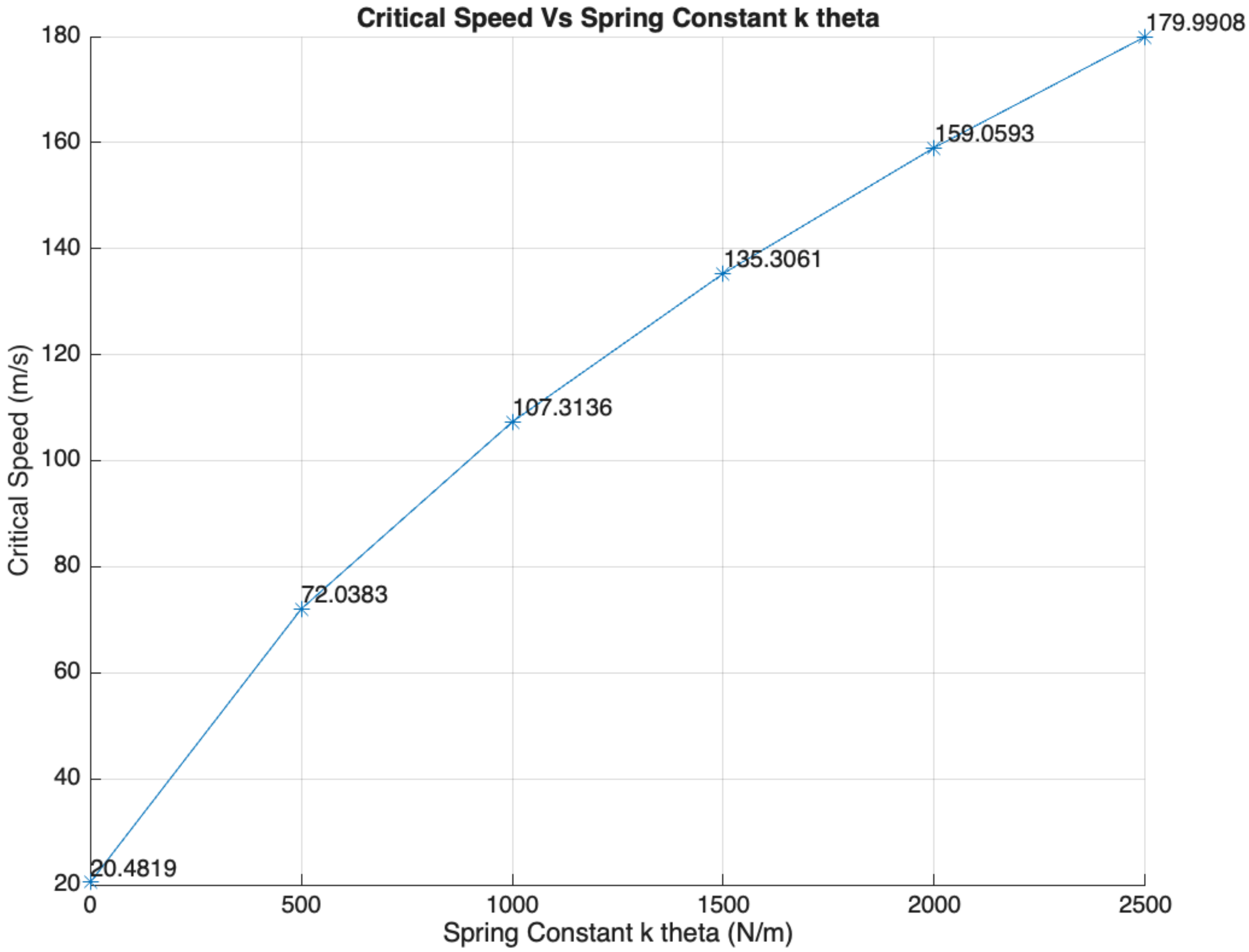

Figure 6: Varying Critical Speed versus Spring Constant k_3

The relationship between the critical speed and the stiffness parameters k_1, k_2 and $k_{\theta 1}$ was analyzed by varying each spring constant individually while keeping the other parameters constant. The results are presented in Figures 4, 5, and 6.

Figure 4 shows the effect of varying the leading-edge spring constant k_1 on the critical speed. It was observed that as k_1 increases, the critical speed decreases significantly, suggesting that increasing the stiffness at the leading edge destabilizes the system. This may be due to the higher spring force at the leading edge amplifying aerodynamic loads, making the system more susceptible to flutter.

Figure 5 illustrates the effect of varying the trailing-edge spring constant k_2. In contrast to k_1 increasing k_2 leads to a linear increase in the critical speed. This indicates that additional stiffness at the trailing edge improves the system's stability, likely due to enhanced structural support against torsional effects.

Figure 6 examines the impact of the torsional spring constant $k_{\theta 1}$ on the critical speed. The results demonstrate a positive correlation, with the critical speed increasing as $k_{\theta 1}$ rises. This trend reflects the stabilizing effect of torsional stiffness, which counters aerodynamic pitching moments and delays the onset of flutter.

In summary, the critical speed is highly sensitive to variations in the spring constants. Increasing k_1 and $k_{\theta 1}$ enhances the aeroelastic stability, while increasing k_1 reduces it. These insights are critical for optimizing the stiffness distribution of the system to achieve higher critical speeds.

3.4 Maximum Critical Speed and Minimum Stiffness

To maximize the critical speed of the aeroelastic system, the optimal combination of spring constants was determined while adhering to the specified constraints. The constraints required the sum of the translational spring constants: k_1 and k_2 to remain below 10,000 N/m (k_1 $+k_2 \leq 10,000\,N/m$), with each individual spring constant being greater than 1,000 N/m. Additionally, the torsional spring constant ($k_{\theta 1}$) was restricted values below 700 Nm/rad ($k_{\theta 1} < 700\,Nm/rad$).

Through an iterative computational analysis, the critical speed was evaluated for various combinations of spring constants within the given constraints. The results identified the optimal configuration as

$$k_1 = 4500\,N/m$$

$$k_2 = 5500\,N/m$$

$$k_{\theta 1} = 600\,Nm/rad$$

This combination resulted in a maximum critical speed of

$$U_{Max} = 198.94\,m/s$$

representing a significant improvement over the baseline critical speed of $U_{Max} = 72.038\,m/s$

This optimization process highlights the importance of balancing translational and torsional stiffness to improve system stability. By increasing k_2, which enhances stiffness at the trailing edge, and $k_{\theta 1}$, which strengthens torsional stability, the critical speed was significantly improved.

At the same time, the selected value of k_1, provided adequate stiffness at the leading edge without negatively impacting stability. These results show that achieving high stability depends on carefully distributing stiffness between the translational and torsional components while staying within the specified design constraints.

All tested combinations of spring constants adhered to the given constraints, ensuring that the final design was both feasible and consistent with aeroelastic principles. The optimized configuration represents a well-thought-out balance, improving stability while respecting the mechanical and structural limitations of the system.

3.5 Critical Speeds with Varying Point Mass Location

To study how the location of the point mass x_m affect the critical speed ($U_{Critical}$), the mass was moved along the airfoil chord from the leading edge ($x_m = 0$) to the training edge ($x_m = c$). Figure 7 illustrates the results, showing that the critical speed decreases significantly as the point mass is positioned farther from the leading edge. This happens because moving the mass away increases the rotational inertia, making the system less stable and more prone to flutter.

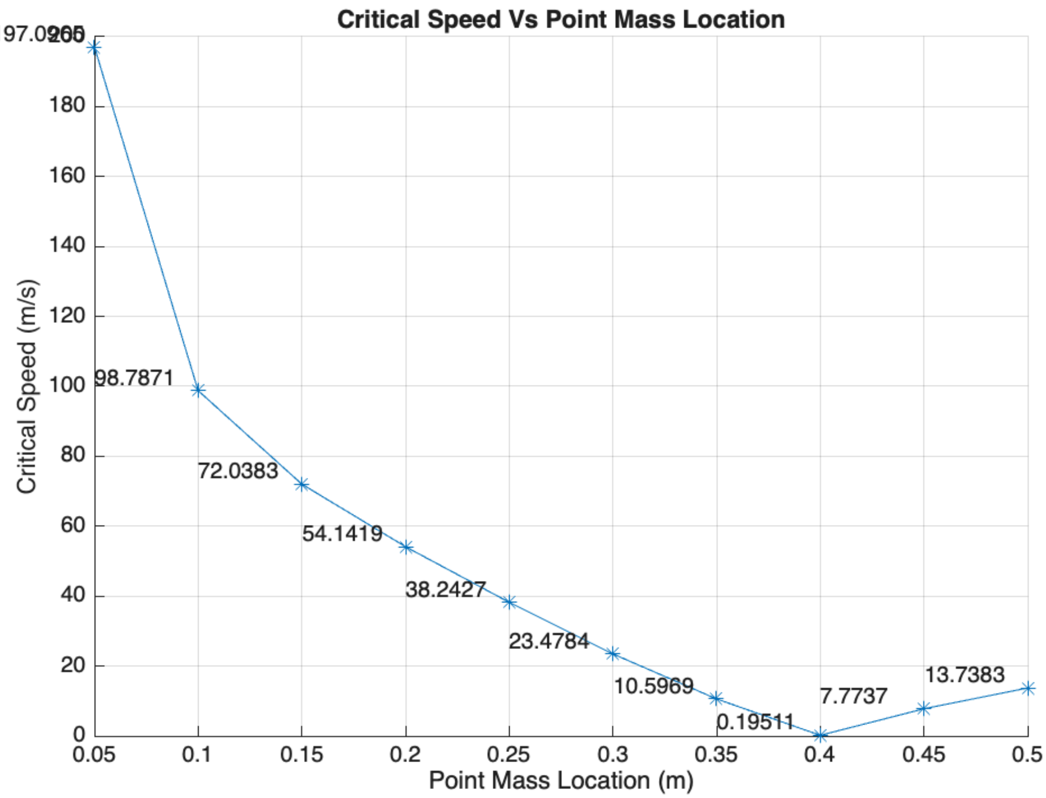

Figure 7: Variation of Critical Speeds versus Point Mass Location

At the baseline point location ($x_m = 0.15\ m$), the critical speed matches the previously calculated value of 72.038 m/s. As the point mass shifts closer to the trailing edge, the critical speed drops sharply hitting its lowest speed near $x_m = 0.4\ m$.

It is worth noting that a slight recovery in critical speed occurs as the mass approaches the chord's end, likely due to subtle changes in the balance of aerodynamic and inertial forces.

From these results, it's clear that placing the point mass closer to the leading edge significantly improves stability and increases the critical speed. This insight is useful for designing systems where the placement of additional mass can be adjusted to achieve better aeroelastic performance.

3.6 Effects of Mechanical Damping

This part of the project focused on how mechanical damping influences the critical speed of the system. Two damping parameters were studied: the translational damping constant c_2 and the torsional damping constant $c_{\theta 1}$. For c_2, its value varied from 0 to 20 Ns.m, while $c_{\theta 1}$ was varied from 0 to 0.7 Nms/rad. All other system parameters remained fixed during the analysis.

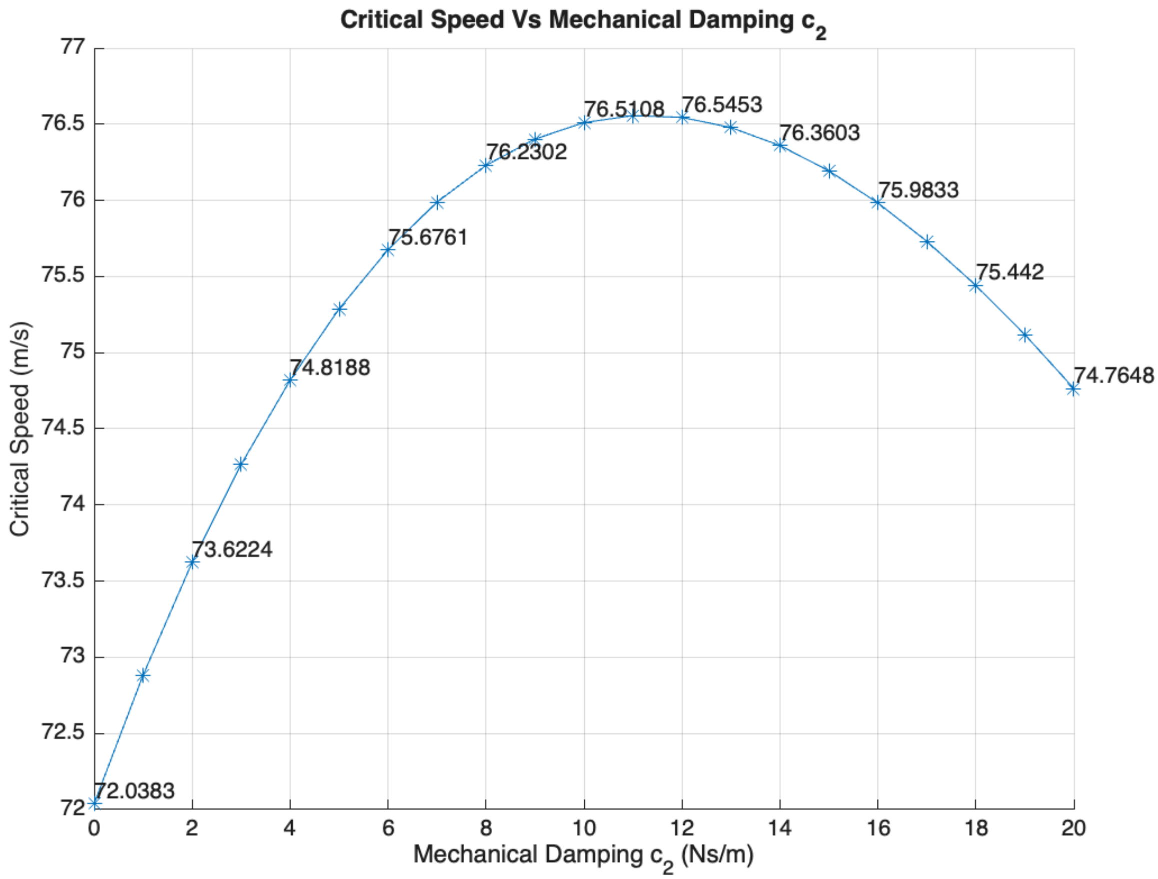

Figure 8: Critical Speed Vs Mechanical Damping c_2

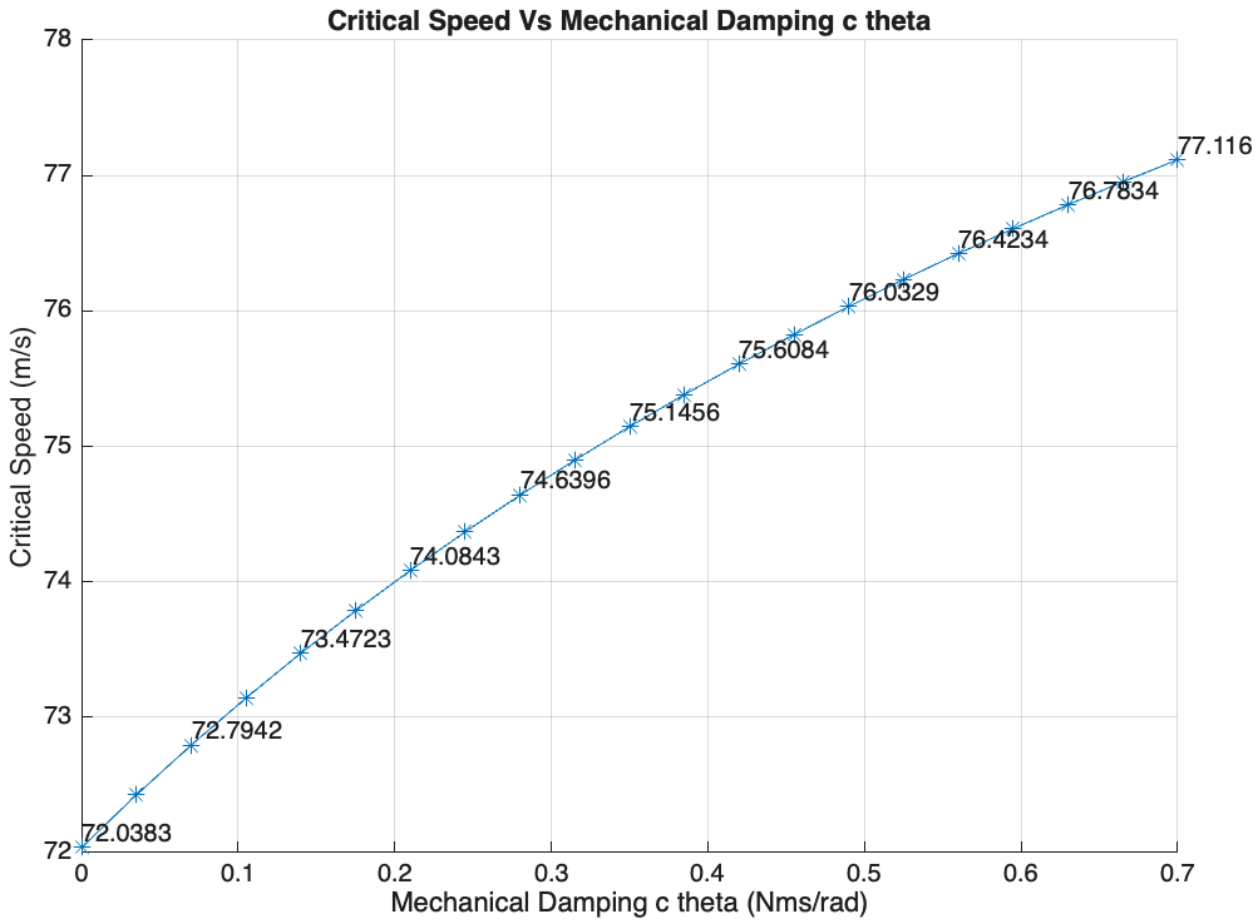

Figure 9: Critical Speed Vs Mechanical Damping c_{theta}

The results are shown in Figures 8 and 9. In Figure 8, increasing c_2 initially raises the critical speed, which means that adding translational damping improves the system's stability. However, after reaching an optimal damping level ($c_2 \approx 10\ Ns/m$), further increases in c_2 because the critical speed to drop. This indicates that too much damping can start to reduce stability, likely because it overcompensates for the system's oscillations.

On the other hand, Figure 9 shows the increasing $c_{\theta 1}$ consistently improves the critical speed. As the torsional damping increases, the system becomes more stable with no clear limit within the studied range. This is because torsional damping effectively reduces rotational instabilities in the airfoil.

In summary, the results show that while there is an optimal range for translational damping c_2 to achieve maximum stability, torsional damping $c_{\theta 1}$ continuously enhanced stability as it increases. These findings highlight the importance of balancing damping levels when designing aeroelastic systems.

4.0 Conclusion and Design Suggestions

In conclusion, the project successfully was able to analyze a two dimensional, 2 DOF aeroelasticity model. Both Netwonian and Lagrandian mechanics methods were utilized to derive the equations of motion. The model was solved computationally with the help of MATLAB. The results of the MATLAB code were as follows: critical speed was calculated to be 72.083 m/s, and the spring constants k_1, k_2 and $k_{\theta 1}$ were determined to be 4500 N/m, 5500 N/m, and 600 Nm/rad, respectively, to achieve maximum critical speed with minimal stiffness. The study also demonstrated the impact of spring constant, mass distribution, and mechanical damping on the aeroelastic stability of the system. Additionally, the influence of translational and torsional damping on stability was explored, highlighting their critical roles in suppressing oscillatory instabilities.

5.0 References

[1] *Torontomu.ca*, 2024. https://courses.torontomu.ca/d2l/le/content/927799/viewContent/6060311/View (accessed Nov. 22, 2024).

[2] *Torontomu.ca*, 2024. https://courses.torontomu.ca/d2l/le/content/927799/viewContent/5919600/View (accessed Nov. 22, 2024).

[2] D. H. Hodges and G Alvin Pierce, *Introduction to structural dynamics and aeroelasticity*. New York, Ny: Cambridge University Press, 2014.

[3] Y. C. Fung, *An introduction to the theory of aeroelasticity*. New York: Dover, 2008.

6.0 Appendix

```
%% AER 722 Project 2 | Sharvani Yadav, Alexia Economou, Daniel Mielnik
clear all;
clc;

%% Constants
S = 1; % m
c = 0.5; % m
b = 0.5*c; % m
k1 = 5000; % kN/m
k_theta1 = 500; % Nm/rad
k2 = 1000; % kN/m
c_theta1 = 0; % Nms/rad
c2 = 0; % Ns/m
m = 5; % kg
I_CG = 0.05; % kgm^2
x_g = 0.15; % m
m1 = 2; % kg
x_m = 0.15; % m
rho = 1.225; % kg/m^3
U_max = 100; % m/s
syms U lambda
q = 0.5*rho*U^2;

%% Question 2
M = [m+m1, m*(x_g-b)-m1*(b-x_m); m*(x_g-b)-m1*(b-x_m),
m*(x_g-b)^2+m1*(b-x_m)^2+I_CG];
B_s = [c2, c2*(b/2); c2*(b/2), c2*(b/2)^2+c_theta1];
B_a = [1, b/2; -b/2, 0];
B_bar_a = pi*rho*c*S*B_a;
B_bar = B_s + U*B_bar_a;
E = [(k1+k2), k2*(b/2)-k1*b; k2*(b/2)-k1*b, k2*(b/2)^2+k1*b^2+k_theta1];
K = [0, 1; 0, -b/2];
Kb = pi*c*S*rho*K;
K_bar = pi*rho*S*Kb;
J = [0, 0; 0, 0];

A = [M, B_bar; J, M];
C = [J, E+U^2*K_bar; -M, J];
```

```
Asub = double(subs(A, U, U_max));
Csub = double(subs(C, U, U_max));

eigenVal = eig(Csub, -Asub);

CharMatrix = [(M(1,1)*lambda^2+(B_s(1,1)+U*B_bar_a(1,1))*lambda+E(1,1)),
(M(1,2)*lambda^2+(B_s(1,2)+U*B_bar_a(1,2))*lambda+E(1,2)+U^2*Kb(1,2));
(M(2,1)*lambda^2+(B_s(2,1)+U*B_bar_a(2,1))*lambda+E(2,1)),
(M(2,2)*lambda^2+(B_s(2,2)+U*B_bar_a(2,2))*lambda+E(2,2)+U^2*Kb(2,2))];

CharEqn = det(CharMatrix);
eigCollect = collect(CharEqn, lambda);

Cf = vpa(fliplr(coeffs((CharEqn),lambda)),4);
p0 = Cf(1);
p1 = Cf(2);
p2 = Cf(3);
p3 = Cf(4);
p4 = Cf(5);

T3 = p1*p2*p3 - p1^2*p4 - p0*p3^2;
T3 = vpa(solve(T3==0,U),3);

U_Critical = vpa(min(T3(T3>0)), 5)


%% Question 3
U_Critical = 70;
realParts = [];
imagParts = [];

for U_new = 0:1:(1.2*U_Critical)
   Asub = double(subs(A, U, U_new));
   Csub = double(subs(C, U, U_new));

   eigenVal = eig(Csub, -Asub);
   realParts = [realParts; real(eigenVal(1)), real(eigenVal(2)), real(eigenVal(3)),
real(eigenVal(4))];
   imagParts = [imagParts; imag(eigenVal(1)), imag(eigenVal(2)), imag(eigenVal(3)),
imag(eigenVal(4))];
```

```
end

% Plotting the results
figure;
hold on;
plot(0:1:(1.2 * U_Critical), realParts);
plot(0:1:(1.2 * U_Critical), imagParts);
xlabel('Critical Speed (m/s)');
ylabel('Eigenvalue Parts');
title('Variation of Eigenvalues with Airspeed');
legend('Real Parts 1', 'Real Parts 2', 'Real Parts 3', 'Real Parts 4', 'Imaginary Parts 1', 'Imaginary
Parts 2', 'Imaginary Parts 3', 'Imaginary Parts 4');
grid on;
hold off;


%% Question 4
k1_og = 5000;
k_theta1_og = 500;
k2_og = 1000;

% Spring Constant k1
U_Crit_k1 = [];

for i = 0:1:5
   k1 = k1_og*i;
   k_theta1 = k_theta1_og;
   k2 = k2_og;

   E = [(k1+k2), k2*(b/2)-k1*b; k2*(b/2)-k1*b, k2*(b/2)^2+k1*b^2+k_theta1];

   CharMatrix = [(M(1,1)*lambda^2+(B_s(1,1)+U*B_bar_a(1,1))*lambda+E(1,1)),
(M(1,2)*lambda^2+(B_s(1,2)+U*B_bar_a(1,2))*lambda+E(1,2)+U^2*Kb(1,2));
(M(2,1)*lambda^2+(B_s(2,1)+U*B_bar_a(2,1))*lambda+E(2,1)),
(M(2,2)*lambda^2+(B_s(2,2)+U*B_bar_a(2,2))*lambda+E(2,2)+U^2*Kb(2,2))];

   CharEqn = det(CharMatrix);
   Cf = vpa(fliplr(coeffs((CharEqn),lambda)),4);
   p0 = Cf(1);
   p1 = Cf(2);
```

```
    p2 = Cf(3);
    p3 = Cf(4);
    p4 = Cf(5);

    T3 = p1*p2*p3 - p1^2*p4 - p0*p3^2;
    T3 = vpa(solve(T3==0,U),3);

    U_Crit_k1 = [U_Crit_k1, vpa(min(T3(T3>0)), 5);];
end

spring_const = 0:k1_og:5*k1_og;

figure;
hold on;
plot(spring_const, U_Crit_k1, '-*');
xlabel('Spring Constant k_1 (N/m)');
ylabel('Critical Speed (m/s)');
title('Critical Speed Vs Spring Constant k_1');
grid on;

for i = 1:length(U_Crit_k1)
    text(0 + (i-1)*k1_og, U_Crit_k1(i), num2str(double(U_Crit_k1(i))), 'VerticalAlignment',
'bottom', 'HorizontalAlignment', 'left');
end

hold off;

% Spring Constant k2
U_Crit_k2 = [];

for i = 0:1:5
    k1 = k1_og;
    k_theta1 = k_theta1_og;
    k2 = k2_og*i;

    E = [(k1+k2), k2*(b/2)-k1*b; k2*(b/2)-k1*b, k2*(b/2)^2+k1*b^2+k_theta1];

    CharMatrix = [(M(1,1)*lambda^2+(B_s(1,1)+U*B_bar_a(1,1))*lambda+E(1,1)),
(M(1,2)*lambda^2+(B_s(1,2)+U*B_bar_a(1,2))*lambda+E(1,2)+U^2*Kb(1,2));
```

```
(M(2,1)*lambda^2+(B_s(2,1)+U*B_bar_a(2,1))*lambda+E(2,1)),
(M(2,2)*lambda^2+(B_s(2,2)+U*B_bar_a(2,2))*lambda+E(2,2)+U^2*Kb(2,2))];

    CharEqn = det(CharMatrix);
    Cf = vpa(fliplr(coeffs((CharEqn),lambda)),4);
    p0 = Cf(1);
    p1 = Cf(2);
    p2 = Cf(3);
    p3 = Cf(4);
    p4 = Cf(5);

    T3 = p1*p2*p3 - p1^2*p4 - p0*p3^2;
    T3 = vpa(solve(T3==0,U),3);

    U_Crit_k2 = [U_Crit_k2, vpa(min(T3(T3>0)), 5);];
End

spring_const = 0:k2_og:5*k2_og;

figure;
hold on;
plot(spring_const, U_Crit_k2, '-*');
xlabel('Spring Constant k_2 (N/m)');
ylabel('Critical Speed (m/s)');
title('Critical Speed Vs Spring Constant k_2');
ax = gca;
chart = ax.Children(1);
grid on;

for i = 1:length(U_Crit_k2)
    text(0 + (i-1)*k2_og, U_Crit_k2(i), num2str(double(U_Crit_k2(i))), 'VerticalAlignment',
'bottom', 'HorizontalAlignment', 'left');
end

hold off;

% Spring Constant k theta
U_Crit_ktheta = [];

for i = 0:1:5
```

```
    k1 = k1_og;
    k_theta1 = k_theta1_og*i;
    k2 = k2_og;

    E = [(k1+k2), k2*(b/2)-k1*b; k2*(b/2)-k1*b, k2*(b/2)^2+k1*b^2+k_theta1];

    CharMatrix = [(M(1,1)*lambda^2+(B_s(1,1)+U*B_bar_a(1,1))*lambda+E(1,1)),
(M(1,2)*lambda^2+(B_s(1,2)+U*B_bar_a(1,2))*lambda+E(1,2)+U^2*Kb(1,2));
(M(2,1)*lambda^2+(B_s(2,1)+U*B_bar_a(2,1))*lambda+E(2,1)),
(M(2,2)*lambda^2+(B_s(2,2)+U*B_bar_a(2,2))*lambda+E(2,2)+U^2*Kb(2,2))];

    CharEqn = det(CharMatrix);
    Cf = vpa(fliplr(coeffs((CharEqn),lambda)),4);
    p0 = Cf(1);
    p1 = Cf(2);
    p2 = Cf(3);
    p3 = Cf(4);
    p4 = Cf(5);

    T3 = p1*p2*p3 - p1^2*p4 - p0*p3^2;
    T3 = vpa(solve(T3==0,U),3);

    U_Crit_ktheta = [U_Crit_ktheta, vpa(min(T3(T3>0)), 5);];
end

spring_const = 0:k_theta1_og:5*k_theta1_og;
figure;
hold on;
plot(spring_const, U_Crit_ktheta, '-*');
xlabel('Spring Constant k theta (N/m)');
ylabel('Critical Speed (m/s)');
title('Critical Speed Vs Spring Constant k theta');
grid on;

for i = 1:length(U_Crit_ktheta)
    text(0 + (i-1)*k_theta1_og, U_Crit_ktheta(i), num2str(double(U_Crit_ktheta(i))),
'VerticalAlignment', 'bottom', 'HorizontalAlignment', 'left');
end

hold off;
```

```
%% Question 5
k1_vals = 1000:500:9000;
k2_vals = 1000:500:9000;
k_theta1_vals = 0:50:700;

U_max = 0;

for k1 = k1_vals
    for k2 = k2_vals
        if k1 + k2 <= 10000
            for k_theta1 = k_theta1_vals
                E = [(k1+k2), k2*(b/2)-k1*b; k2*(b/2)-k1*b, k2*(b/2)^2+k1*b^2+k_theta1];

                CharMatrix = [(M(1,1)*lambda^2+(B_s(1,1)+U*B_bar_a(1,1))*lambda+E(1,1)),
(M(1,2)*lambda^2+(B_s(1,2)+U*B_bar_a(1,2))*lambda+E(1,2)+U^2*Kb(1,2));
(M(2,1)*lambda^2+(B_s(2,1)+U*B_bar_a(2,1))*lambda+E(2,1)),
(M(2,2)*lambda^2+(B_s(2,2)+U*B_bar_a(2,2))*lambda+E(2,2)+U^2*Kb(2,2))];
                CharEqn = det(CharMatrix);

                Cf = vpa(fliplr(coeffs((CharEqn),lambda)),4);
                p0 = Cf(1);
                p1 = Cf(2);
                p2 = Cf(3);
                p3 = Cf(4);
                p4 = Cf(5);

                T3 = p1*p2*p3 - p1^2*p4 - p0*p3^2;
                T3 = vpa(solve(T3==0,U),3);

                U_Crit = vpa(min(T3(T3>0)), 5);

                if any(U_Crit > U_max) && all(U_Crit < 200)
                    U_max = U_Crit;

                    k1_max = k1;
                    k2_max = k2;
                    k_theta1_max = k_theta1;
                end
```

```
            end
        end
    end
end

% Valid combinations
U_max
k1_max
k2_max
k_theta1_max


%% Question 6
k1 = k1_og;
k_theta1 = k_theta1_og;
k2 = k2_og;

U_Crit_xm = [];

for i = 0:0.05:c
    x_m = i;

    M = [m+m1, m*(x_g-b)-m1*(b-x_m); m*(x_g-b)-m1*(b-x_m),
m*(x_g-b)^2+m1*(b-x_m)^2+I_CG];
    E = [(k1+k2), k2*(b/2)-k1*b; k2*(b/2)-k1*b, k2*(b/2)^2+k1*b^2+k_theta1];

    CharMatrix = [(M(1,1)*lambda^2+(B_s(1,1)+U*B_bar_a(1,1))*lambda+E(1,1)),
(M(1,2)*lambda^2+(B_s(1,2)+U*B_bar_a(1,2))*lambda+E(1,2)+U^2*Kb(1,2));
(M(2,1)*lambda^2+(B_s(2,1)+U*B_bar_a(2,1))*lambda+E(2,1)),
(M(2,2)*lambda^2+(B_s(2,2)+U*B_bar_a(2,2))*lambda+E(2,2)+U^2*Kb(2,2))];

    CharEqn = det(CharMatrix);
    Cf = vpa(fliplr(coeffs((CharEqn),lambda)),4);
    p0 = Cf(1);
    p1 = Cf(2);
    p2 = Cf(3);
    p3 = Cf(4);
    p4 = Cf(5);

    T3 = p1*p2*p3 - p1^2*p4 - p0*p3^2;
```

```
    T3 = vpa(solve(T3==0,U),3);
    U_Crit_xm = [U_Crit_xm, vpa(min(T3(T3>0)), 5);];
end

figure;
hold on;
plot(0.05:0.05:c, U_Crit_xm, '-*');
xlabel('Point Mass Location (m)');
ylabel('Critical Speed (m/s)');
title('Critical Speed Vs Point Mass Location');
grid on;

for i = 1:length(U_Crit_xm)
    text(0 + (i-1)*0.05, U_Crit_xm(i), num2str(double(U_Crit_xm(i))), 'VerticalAlignment',
'bottom', 'HorizontalAlignment', 'left');
end

hold off;
%% Question 7
x_m = 0.15;

% Mechanical Damping c2
U_Crit_c2 = [];

for i = 0:1:20
    c2 = i;

    M = [m+m1, m*(x_g-b)-m1*(b-x_m); m*(x_g-b)-m1*(b-x_m),
m*(x_g-b)^2+m1*(b-x_m)^2+I_CG];

    E = [(k1+k2), k2*(b/2)-k1*b; k2*(b/2)-k1*b, k2*(b/2)^2+k1*b^2+k_theta1];

    B_s = [c2, c2*(b/2); c2*(b/2), c2*(b/2)^2+c_theta1];
    B_bar = B_s + U*B_bar_a;

    CharMatrix = [(M(1,1)*lambda^2+(B_s(1,1)+U*B_bar_a(1,1))*lambda+E(1,1)),
(M(1,2)*lambda^2+(B_s(1,2)+U*B_bar_a(1,2))*lambda+E(1,2)+U^2*Kb(1,2));
(M(2,1)*lambda^2+(B_s(2,1)+U*B_bar_a(2,1))*lambda+E(2,1)),
(M(2,2)*lambda^2+(B_s(2,2)+U*B_bar_a(2,2))*lambda+E(2,2)+U^2*Kb(2,2))];
```

```
    CharEqn = det(CharMatrix);
    Cf = vpa(fliplr(coeffs((CharEqn),lambda)),4);
    p0 = Cf(1);
    p1 = Cf(2);
    p2 = Cf(3);
    p3 = Cf(4);
    p4 = Cf(5);

    T3 = p1*p2*p3 - p1^2*p4 - p0*p3^2;
    T3 = vpa(solve(T3==0,U),3);

    U_Crit_c2 = [U_Crit_c2, vpa(min(T3(T3>0)), 5);];
end

figure;
hold on;
plot(0:1:20, U_Crit_c2, '-*');
xlabel('Mechanical Damping c_2 (Ns/m)');
ylabel('Critical Speed (m/s)');
title('Critical Speed Vs Mechanical Damping c_2');
grid on;

for i = 1:2:length(U_Crit_c2)
    text(0 + (i-1)*1, U_Crit_c2(i), num2str(double(U_Crit_c2(i))), 'VerticalAlignment', 'bottom',
'HorizontalAlignment', 'left');
end

hold off;

% Mechanical Damping c_theta1
c2 = 0;

U_Crit_ctheta1 = [];

for i = 0:0.035:0.7
    c_theta1 = i;

    M = [m+m1, m*(x_g-b)-m1*(b-x_m); m*(x_g-b)-m1*(b-x_m),
m*(x_g-b)^2+m1*(b-x_m)^2+I_CG];
```

```
    E = [(k1+k2), k2*(b/2)-k1*b; k2*(b/2)-k1*b, k2*(b/2)^2+k1*b^2+k_theta1];

    B_s = [c2, c2*(b/2); c2*(b/2), c2*(b/2)^2+c_theta1];
    B_bar = B_s + U*B_bar_a;

    CharMatrix = [(M(1,1)*lambda^2+(B_s(1,1)+U*B_bar_a(1,1))*lambda+E(1,1)),
(M(1,2)*lambda^2+(B_s(1,2)+U*B_bar_a(1,2))*lambda+E(1,2)+U^2*Kb(1,2));
(M(2,1)*lambda^2+(B_s(2,1)+U*B_bar_a(2,1))*lambda+E(2,1)),
(M(2,2)*lambda^2+(B_s(2,2)+U*B_bar_a(2,2))*lambda+E(2,2)+U^2*Kb(2,2))];

    CharEqn = det(CharMatrix);
    Cf = vpa(fliplr(coeffs((CharEqn),lambda)),4);
    p0 = Cf(1);
    p1 = Cf(2);
    p2 = Cf(3);
    p3 = Cf(4);
    p4 = Cf(5);

    T3 = p1*p2*p3 - p1^2*p4 - p0*p3^2;
    T3 = vpa(solve(T3==0,U),3);

    U_Crit_ctheta1 = [U_Crit_ctheta1, vpa(min(T3(T3>0)), 5);];
end

figure;
hold on;
plot(0:0.035:0.7, U_Crit_ctheta1, '-*');
xlabel('Mechanical Damping c theta (Nms/rad)');
ylabel('Critical Speed (m/s)');
title('Critical Speed Vs Mechanical Damping c theta');
grid on;

for i = 1:2:length(U_Crit_ctheta1)
    text(0 + (i-1)*0.035, U_Crit_ctheta1(i), num2str(double(U_Crit_ctheta1(i))),
'VerticalAlignment', 'bottom', 'HorizontalAlignment', 'left');
end

hold off;
```