AE352 HW2 richiem2

1. (25 points) The motion of the wingtip of your glider can be modeled as a damped, driven harmonic oscillator of the form

$$m\ddot{z} = -kz - c\dot{z} + d\sin(\omega_f t). \tag{1}$$

The mass of this system is $m = 4\,\text{kg}$, k=10 N/m and you know that c is about one tenth of the system's critical damping coefficient. The forcing amplitude $d = 0.5\,\text{m}$.

(a) (5 pts) What is the natural frequency of the system without damping ($c = 0$, $d = 0$)?

(b) (5 pts) What is the frequency of your system with damping, i.e. what is the frequency of the homogeneous solution ($c \neq 0$, $d = 0$)?

(c) (10 pts) What frequency ω_f would drive your system (1) to a maximum resonant response?

(d) (5 pts) What is the amplitude ($A = a \cdot d$, i.e. gain x forcing amplitude) of the solution of the above system when driven at the resonance frequency?

a) $W_n = \sqrt{\frac{K}{m}} = \sqrt{\frac{10}{4}} = 1.581$ rad/s

$W_d = \sqrt{W_n^2 - \left(\frac{c}{2m}\right)^2}$ $\qquad c = \frac{1}{10} \cdot 2\sqrt{Km}$

b) $W_d = \sqrt{(1.58)^2 - \left(\frac{\sqrt{Km}}{2m}\right)^2} = \sqrt{(1.58)^2 - \left(\frac{\sqrt{40}}{20}\right)^2}$

$W_d = 1.573$ rad/s

c) The frequency W_f that drives the system to a maximum resonant response is the same as the damped natural frequency

$W_f = W_d = 1.573$ rad/s

d) $c = \frac{1}{10} 2\sqrt{Km}$

from slides

$a = \dfrac{1}{\sqrt{m^2(K/m - W_f^2)^2 + c^2 W_f^2}} = 0.501$

$A = a \cdot d = (0.502)(0.5) = 0.251$ m

2. (25 points) A system of two coupled harmonic oscillators is shown in Figure 1. We will assume gravity and friction are negligible.

   (a) (10 pts) Find the equations of motion for both masses m_1 and m_2!

   (b) (10 pts) What would the eigensystem of the coupling matrix look like?

   (c) (5 pts) Find the eigenfrequencies of this system!

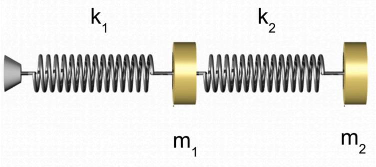


Figure 1: Two coupled springs with masses m_1 and m_2. The left spring connects to m_1 and is attached to a wall. The right spring connects masses m_1 and m_2. The mass m_2 is not connected to a wall. Both m_1 and m_2 can only move along the axis that connects the springs.

a) For m_1:

$$m_1\ddot{x}_1 = -k_1x_1 - k_2(x_2 - x_1)$$

For m_2:

$$m_2\ddot{x}_2 = -k_2(x_2 - x_1)$$

b)
$$\underbrace{\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix}}_{M}\begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{bmatrix} + \underbrace{\begin{bmatrix} k_1+k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix}}_{K}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

c) $\det(K - w^2M)$

$$\det\begin{vmatrix} k_1+k_2-w^2m_1 & -k_2 \\ -k_2 & k_2-w^2m_2 \end{vmatrix}$$

$$= (k_1+k_2-w^2m_1)(k_2-w^2m_2) - k_2^2$$

↳ plug into jupyter →

```python
import sympy as sym
import numpy as np

k1, k2, w, m1, m2 = sym.symbols('k1, k2, w, m1, m2')

eq = (k1 + k2 - w**2 * m1) * (k2 - w**2 * m2) - k2**2

ans = sym.solve(eq, w)

ans
```

```
[-sqrt(2)*sqrt(k1/m1 + k2/m2 + k2/m1 - sqrt(k1**2*m2**2 - 2*k1*k2*m1*m2 + 2*k1*k2*m2**2 + k2**2*m1**2 + 2*k2**2*m1*m2 + k2**2*m2**2)/(m1*m2))/2,
 sqrt(2)*sqrt(k1/m1 + k2/m2 + k2/m1 - sqrt(k1**2*m2**2 - 2*k1*k2*m1*m2 + 2*k1*k2*m2**2 + k2**2*m1**2 + 2*k2**2*m1*m2 + k2**2*m2**2)/(m1*m2))/2,
 -sqrt(2)*sqrt(k1/m1 + k2/m2 + k2/m1 + sqrt(k1**2*m2**2 - 2*k1*k2*m1*m2 + 2*k1*k2*m2**2 + k2**2*m1**2 + 2*k2**2*m1*m2 + k2**2*m2**2)/(m1*m2))/2,
 sqrt(2)*sqrt(k1/m1 + k2/m2 + k2/m1 + sqrt(k1**2*m2**2 - 2*k1*k2*m1*m2 + 2*k1*k2*m2**2 + k2**2*m1**2 + 2*k2**2*m1*m2 + k2**2*m2**2)/(m1*m2))/2]
```

```python
m = sym.Matrix([-sym.sqrt(2)*sym.sqrt(k1/m1 + k2/m2 + k2/m1 - sym.sqrt(k1**2*m2**2 - 2*k1*k2*m1*m2 + 2*
 sym.sqrt(2)*sym.sqrt(k1/m1 + k2/m2 + k2/m1 - sym.sqrt(k1**2*m2**2 - 2*k1*k2*m1*m2 + 2*k1*k2*m2**2 + k2
 -sym.sqrt(2)*sym.sqrt(k1/m1 + k2/m2 + k2/m1 + sym.sqrt(k1**2*m2**2 - 2*k1*k2*m1*m2 + 2*k1*k2*m2**2 + k
 sym.sqrt(2)*sym.sqrt(k1/m1 + k2/m2 + k2/m1 + sym.sqrt(k1**2*m2**2 - 2*k1*k2*m1*m2 + 2*k1*k2*m2**2 + k2

m
```

eigenfrequencies

$$-\frac{\sqrt{2}\sqrt{\frac{k_1}{m_1}+\frac{k_2}{m_2}+\frac{k_2}{m_1}-\frac{\sqrt{k_1^2m_2^2-2k_1k_2m_1m_2+2k_1k_2m_2^2+k_2^2m_1^2+2k_2^2m_1m_2+k_2^2m_2^2}}{m_1m_2}}}{2}$$

$$\frac{\sqrt{2}\sqrt{\frac{k_1}{m_1}+\frac{k_2}{m_2}+\frac{k_2}{m_1}-\frac{\sqrt{k_1^2m_2^2-2k_1k_2m_1m_2+2k_1k_2m_2^2+k_2^2m_1^2+2k_2^2m_1m_2+k_2^2m_2^2}}{m_1m_2}}}{2}$$

3. (25 points) Carbon monoxide (CO) is diatomic molecule consisting of a carbon atom with a mass of roughly 12 amu and an oxygen atom with a mass of 16 amu. In a simple spring-mass model (Figure 2) the constant of restitution for this setup is $k = 1860\,\text{N/m}$. One atomic mass unit (amu) $= 1.66 \times 10^{-27}$ kg. Damping is negligible.

(a) (10 pts) What are the equations of motion for this system?

(b) (5 pts) What is/are the eigenfrequency/frequencies of this system?

(c) (5 pts) What would the eigenmode(s), $q_i(t)$, look like?

(d) (5 pts) What wavelength would you tune your laser to in order to excite the molecules eigenfrequency into resonance? The following equation should help,

$$\lambda \nu = c, \tag{2}$$

where λ is the wavelength in meters, $\nu = \frac{\omega}{2\pi}$ is the frequency of the laser in [Hz] and c is the speed of light.

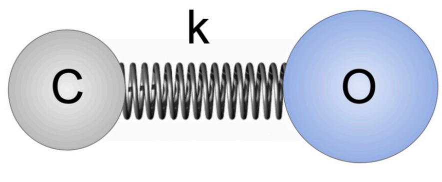


Figure 2: Schematic of a CO molecule.

C: 1.992×10^{-26}

O: 2.656×10^{-26}

a) C: $m_1 \ddot{x}_1 = -kx_1 + kx_2$ $\longrightarrow$ $(1.99 \times 10^{-26})\ddot{x}_1 = 1860(x_2 - x_1)$

O: $m_2 \ddot{x}_2 = -kx_2 + kx_1$ $\quad$ $(2.656 \times 10^{-26})\ddot{x}_2 = 1860(x_1 - x_2)$

b)
$$\begin{bmatrix} 1.992\times10^{-26} & 0 \\ 0 & 2.656\times10^{-26} \end{bmatrix}\begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{bmatrix} + \begin{bmatrix} 1860 & -1860 \\ -1860 & 1860 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

$\det|K - w^2 M| = 0$

$$\det\begin{vmatrix} 1860 - w^2(1.992\times10^{-26}) & -1860 \\ -1860 & 1860 - w^2(2.656\times10^{-26}) \end{vmatrix} = 0$$

$w^2 = 0,\ 1.634\times10^{29}$ $\quad \rightarrow \quad$ $w_1 = 0$ rad/s

$w = 0,\ 4.023\times10^{14}$ $\quad$ $w_2 = 4.023\times10^{14}$ rad/s

c) $(K - w^2 M)\phi_n = 0$

$$\begin{bmatrix} 1860 - w^2(1.992\times10^{-6}) & -1860 \\ -1860 & 1860 - w^2(2.656\times10^{-6}) \end{bmatrix}$$

$\lambda_1 = 0 \quad w_1 = 0$

$$\begin{bmatrix} 1860 & -1860 \\ -1860 & 1860 \end{bmatrix}\begin{bmatrix} \phi_{11} \\ \phi_{21} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \longrightarrow \phi_{11} = \phi_{21}$$

$\lambda_2 = 1.634\times10^{29} \quad w_2 = 4.023\times10^{14}$

$$\begin{bmatrix} 1860 - 3253 & -1860 \\ -1860 & 1860 - 4340 \end{bmatrix}\begin{bmatrix} \phi_{12} \\ \phi_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} -1394 & -1860 \\ -1860 & -2480 \end{bmatrix}\begin{bmatrix} \phi_{12} \\ \phi_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \longrightarrow \phi_{12} = -\frac{4}{3}\phi_{22}$$

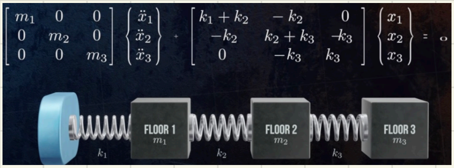
Eigenmodes:
$$\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} -4 \\ 3 \end{bmatrix}$$

d) $c = 3\times10^8$ m/s

$v = \frac{w}{2\pi} = \frac{4.023\times10^{14}}{2\pi}$ $\quad \rightarrow \quad$ $\lambda = \frac{c}{v} = \frac{3\times10^8}{\frac{4.023\times10^{14}}{2\pi}}$

$\lambda = 4.663\times10^{-6}$ m

4. (25 pts) The system of three coupled harmonic oscillators presented in Figure 3 has masses and constants of restitution $\{m_i\} = \{1, 2, 3\}$ metric tons and $\{k_i\} = \{3000, 2000, 1000\}$ N/m, $i \in \{1, 2, 3\}$, respectively. What are the eigenmodes / mode shapes of this system, if the first mass is displaced by $x_1 = 1$ m and then released? All other initial conditions are equal to zero. Analytic and numeric results are acceptable!



$$M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix} \times 1000 \quad \text{to get kg}$$

$$K = \begin{bmatrix} 3000+2000 & -2000 & 0 \\ -2000 & 2000+1000 & -1000 \\ 0 & -1000 & 1000 \end{bmatrix} = \begin{bmatrix} 5000 & -2000 & 0 \\ -2000 & 3000 & -1000 \\ 0 & -1000 & 1000 \end{bmatrix}$$

$$(K - \omega^2 M)\phi = 0$$

to find $\omega \rightarrow$ jupyter

```python
import numpy as np

# Given values
m = np.array([1, 2, 3]) * 1000  # converting metric tons to kg
k = np.array([3000, 2000, 1000])  # spring constants in N/m

# Mass matrix M
M = np.diag(m)

# Stiffness matrix K
K = np.array([
    [k[0] + k[1], -k[1], 0],
    [-k[1], k[1] + k[2], -k[2]],
    [0, -k[2], k[2]]
])

# We need to solve the generalized eigenvalue problem KX = omega^2 * MX
# To find omega^2 we can use the scipy function eigh which is for symmetric matrices
from scipy.linalg import eigh

# Calculate eigenvalues and eigenvectors
eigenvalues, eigenvectors = eigh(K, M)

# eigenvalues are omega^2, so we take the square root to get omega
omegas = np.sqrt(eigenvalues)

# Display eigenvalues (omegas) and eigenvectors (mode shapes)
omegas, eigenvectors
```

```
(array([0.39322223, 1.08402258, 2.34597627]),
 array([[ 0.0036761 , -0.00998231, -0.02977985],
        [ 0.00890604, -0.01909065,  0.00749864],
        [ 0.01661175,  0.00755971, -0.00048345]]))
```

```python
# Scaling factors for each mode to make the displacement of the first mass to be 1 meter
scaling_factors = 1 / eigenvectors[0, :]

# Scaled eigenvectors (mode shapes) for x1 = 1 meter
scaled_eigenvectors = eigenvectors * scaling_factors

scaled_eigenvectors
```

```
array([[ 1.        ,  1.        ,  1.        ],
       [ 2.42268814,  1.91244753, -0.25180233],
       [ 4.51885432, -0.75731053,  0.01623399]])
```

For ω_1 (0.393): $\begin{bmatrix} 1 \\ 2.423 \\ 4.519 \end{bmatrix}$

For ω_3 (2.346): $\begin{bmatrix} 1 \\ -0.252 \\ 0.016 \end{bmatrix}$

For ω_2 (1.08): $\begin{bmatrix} 1 \\ 1.912 \\ -0.757 \end{bmatrix}$