ME351: Lab Report

Experiment 1: 3 DOF System

<u>Group 6</u>

Harshil Safi | 20110073

Hrishikesh CP | 20110078

Kamal Vaishnav | 20110089

Kareena Beniwal | 20110095

Under the Guidance of
Prof. Jayaprakash KR

1. Objective

The objective of this experiment is to find the natural frequencies of a 3DOF system, its mode shapes and how they manifest in the physical system.
A 3 degree of freedom (DOF) pendulum system can have three different natural frequencies and corresponding mode shapes, each corresponding to a different mode of oscillation. The first mode has a single frequency and the pendulum moves back and forth in one direction. The second mode has two frequencies and the pendulum moves back and forth in two perpendicular directions. The third mode has three frequencies and the pendulum moves in a complex pattern, with different amplitudes and phases in each of the three directions. The exact natural frequencies and mode shapes of a 3 DOF pendulum system depend on the specific parameters of the system, such as the lengths and masses of the pendulums, and can be calculated using mathematical equations.

2. Experimental Design

Fig: The experimental setup

3. Fabrication details

The set up consists of three pendulum bobs connected with the help of nylon cords. The bobs were made out of MDF sheet and each weighed about 30 gm. They were wrapped with colored papers so as to facilitate their detection by the camera during open CV analysis.

A webcam was used to record the angular displacement of the pendulum.

Open CV code for the detection of the displacement of the pendulum:

```
import cv2
import numpy as np
import math
#ref
C1_hsv_lowerbound = np.array([89, 83, 135])  # replace THIS LINE w/ your hsv lowerb
C1_hsv_upperbound = np.array([100, 128, 255])  # replace THIS LINE w/ your hsv upperb

#p1
C2_hsv_lowerbound = np.array([44, 95, 0])  # replace THIS LINE w/ your hsv lowerb
C2_hsv_upperbound = np.array([88, 255, 255])  # replace THIS LINE w/ your hsv upperb

#p2
C3_hsv_lowerbound = np.array([9, 95, 0])  # replace THIS LINE w/ your hsv lowerb
C3_hsv_upperbound = np.array([28, 255, 255])  # replace THIS LINE w/ your hsv upperb

#p3
C4_hsv_lowerbound = np.array([156, 50, 0])  # replace THIS LINE w/ your hsv lowerb
C4_hsv_upperbound = np.array([179, 255, 255])  # replace THIS LINE w/ your hsv upperb

# Define the codec and create a VideoWriter object to write the output file
fourcc = cv2.VideoWriter_fourcc(*'XVID')
out = cv2.VideoWriter('output.avi', fourcc, 20.0, (640, 480))

def distance(x1, y1, x2, y2):
    """
    Calculate distance between two points
```

```
    """
    dist = math.sqrt(math.fabs(x2 - x1) * 2 + math.fabs(y2 - y1) * 2)
    return dist
def anglefind(c1x,c1y,c2x,c2y):
    hypotenuse = distance(c1x, c1y, c2x, c2y)
    horizontal = c2x-c1x
    vertical = c2y-c1y
    cv2.line(copy_frame, (c1x, c1y), (c2x, c2y), (0, 0, 255), 2)
    cv2.line(copy_frame, (c1x, c1y), (c1x, c2y), (0, 0, 255), 2)
    #cv2.line(copy_frame, (c2x, c2y), (c2x, c1y), (0, 0, 255), 2)
    if vertical == 0:
        return 90
    return np.arctan(horizontal / vertical) * 180.0 / math.pi

    # draw all 3 lines



def find_color(frame,hsv_lowerbound,hsv_upperbound):#green
    """
    Filter "frame" for HSV bounds for color1 (inplace, modifies frame) & return
coordinates of the object with that color
    """
    hsv_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

    mask = cv2.inRange(hsv_frame, hsv_lowerbound, hsv_upperbound)
    res = cv2.bitwise_and(frame, frame, mask=mask)  # filter inplace
    cnts, hir = cv2.findContours(mask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
    if len(cnts) > 0:
        maxcontour = max(cnts, key=cv2.contourArea)

        # Find center of the contour
        M = cv2.moments(maxcontour)
        if M['m00'] > 0 and cv2.contourArea(maxcontour) > 1000:
            cx = int(M['m10'] / M['m00'])
            cy = int(M['m01'] / M['m00'])
            return (cx, cy), True
        else:
```

```
        return (700, 700), False  # faraway point
    else:
        return (700, 700), False  # faraway point
cap = cv2.VideoCapture(3)

while (1):
    _, orig_frame = cap.read()
    orig_frame=cv2.rotate(orig_frame, cv2.ROTATE_90_CLOCKWISE)
    # we'll be inplace modifying frames, so save a copy
    copy_frame = orig_frame.copy()
    #(co0lor1_x, color1_y), found_color1 =
find_color(copy_frame,C1_hsv_lowerbound,C1_hsv_upperbound)
    (color2_x, color2_y), found_color2 =
find_color(copy_frame,C2_hsv_lowerbound,C2_hsv_upperbound)
    (color3_x, color3_y), found_color3 =
find_color(copy_frame,C3_hsv_lowerbound,C3_hsv_upperbound)
    (color4_x, color4_y), found_color4 =
find_color(copy_frame,C4_hsv_lowerbound,C4_hsv_upperbound)
    #print(color2_x,color2_y)
    print(distance(color2_x,color2_y,color3_x,color3_y))
    found_color1=True;
    color1_x=275
    color1_y=20
    # draw circles around these objects
    cv2.circle(copy_frame, (color1_x, color1_y), 20, (255, 0, 0), -1)
    cv2.circle(copy_frame, (color2_x, color2_y), 20, (0, 128, 255), -1)
    cv2.circle(copy_frame, (color3_x, color3_y), 20, (0, 255, 0), -1)
    cv2.circle(copy_frame, (color4_x, color4_y), 20, (255, 0, 255), -1)
    angle1=0
    angle2=0
    angle3=0
    if found_color1 and found_color2:
        angle1 = anglefind(color1_x,color1_y,color2_x,color2_y)
        cv2.putText(copy_frame, str(angle1), (color1_x - 30, color1_y),
cv2.FONT_HERSHEY_COMPLEX, 1, (0, 128, 229), 2)
    if found_color3 and found_color2:
        angle2 = anglefind(color2_x, color2_y, color3_x, color3_y)
```

```
        cv2.putText(copy_frame, str(angle2), (color2_x - 30, color2_y),
cv2.FONT_HERSHEY_COMPLEX, 1, (0, 128, 229), 2)
    if found_color4 and found_color3:
        angle3 = anglefind(color3_x, color3_y, color4_x, color4_y)
        cv2.putText(copy_frame, str(angle3), (color3_x - 30, color3_y),
cv2.FONT_HERSHEY_COMPLEX, 1, (0, 128, 229), 2)

    with open("output.txt", "a") as f:
            f.write(str(time.time()) + "," + str(angle1) + "\n")

     # Write the output frame to the output video file
    out.write(copy_frame)

    cv2.imshow('AngleCalc', copy_frame)

    print()
    cv2.waitKey(5)
    if cv2.waitKey(5) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

---

4. Mathematical Modelling and Analysis

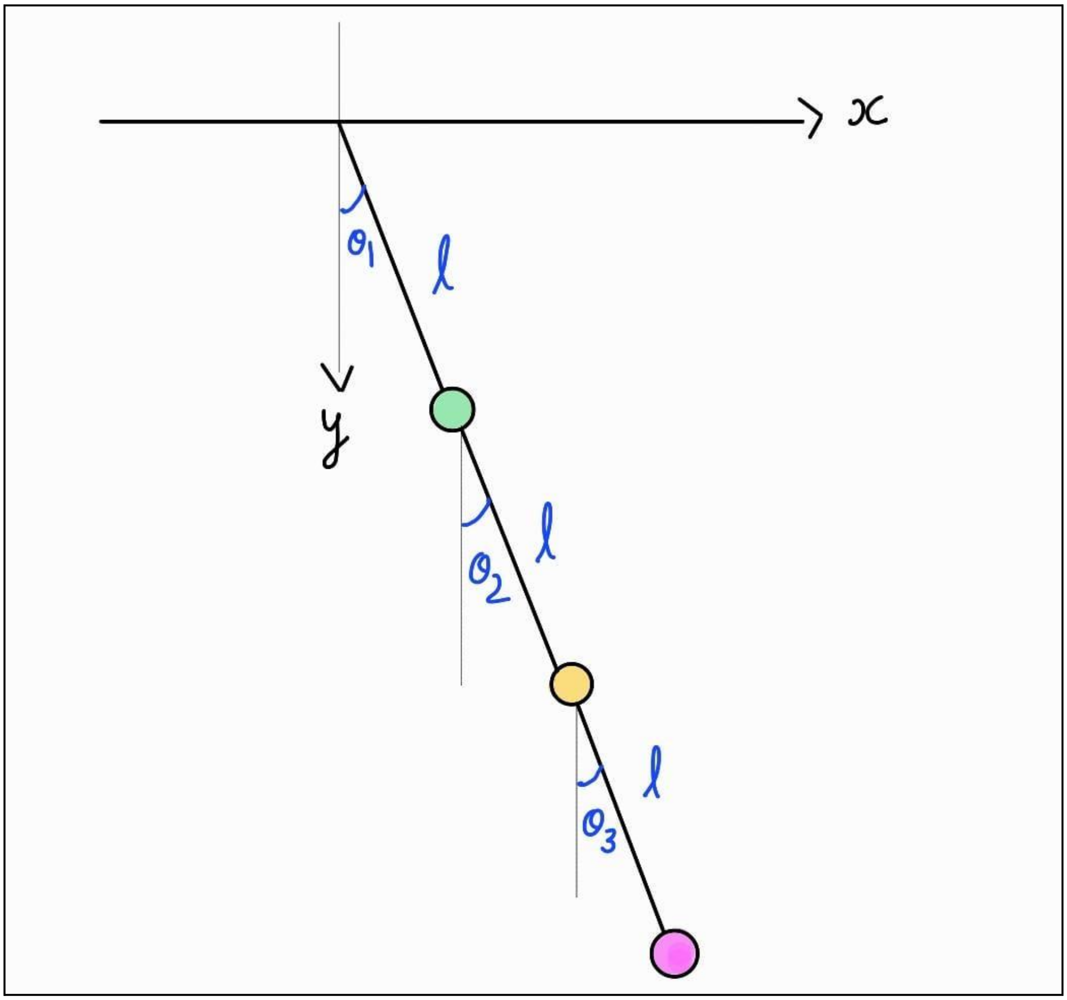


Fig: A triple pendulum

Equation of motion for a 3DOF pendulum

The triple pendulum is a holonomic system (a smooth function that is a solution of a linear homogeneous differential equation with polynomial coefficients) with 3 degrees of freedom. It consists of 3 identical point masses of mass m joined by massless cords of length l. The cords have frictionless pivots allowing them freely to rotate in the plane. It is one of the simplest systems that exhibits chaos.

$x_1 = l \sin\theta_1$
$y_1 = -l \cos \theta_1$
$x_2 = l \sin\theta_1 + l \sin\theta_2$
$y_2 = -l \cos\theta_1 + l \cos\theta_2$
$x_3 = l \sin\theta_1 + l \sin\theta_2 + l \sin\theta_3$

$y_3 = -(l\cos\theta_1 + l\cos\theta_2 + l\cos\theta_3)$

The driving term for the system is taken to be A sinωt and for simplicity, we take A=1.
The *Lagrangian* is a function in classical mechanics that summarizes the dynamics of a physical system. It is used to describe the motion of a system of particles or fields, and to determine the equations of motion of those particles or fields. The Lagrangian is defined as the difference between the kinetic energy, T, and the potential energy, U, of the system:

$$L = T - U$$

Assuming small oscillations, therefore $\sin\theta \simeq \theta$ and $\cos\theta \simeq 1 - \frac{\theta^2}{2}$

Hence Kinetic Energy and Potential Energy can be written as:

$$T = \frac{1}{2} m L^2 [3\dot{\theta}_1{}^2 + 2\dot{\theta}_2{}^2 + \dot{\theta}_3{}^2 + 4\dot{\theta}_1\dot{\theta}_2 + 2\dot{\theta}_1\dot{\theta}_2 + +2\dot{\theta}_2\dot{\theta}_3]$$

$$U = mgL[(\frac{\theta_1^2}{2}) + (\frac{\theta_1^2}{2} + \frac{\theta_2^2}{2}) + (\frac{\theta_1^2}{2} + \frac{\theta_2^2}{2} + \frac{\theta_3^2}{2})] = mgl[3(\frac{\theta_1^2}{2}) + 2(\frac{\theta_2^2}{2}) + 1(\frac{\theta_3^2}{2})]$$

which gives Lagrangian as:

$$L = \frac{1}{2} m L^2 [3\dot{\theta}_1{}^2 + 2\dot{\theta}_2{}^2 + \dot{\theta}_3{}^2 + 4\dot{\theta}_1\dot{\theta}_2 + 2\dot{\theta}_1\dot{\theta}_2 + +2\dot{\theta}_2\dot{\theta}_3] - mgL[3(\frac{\theta_1^2}{2}) + 2(\frac{\theta_2^2}{2}) + 1(\frac{\theta_3^2}{2})]$$

If we want to write equation of motion of triple pendulum, we will write it with respect to θ_1 ,θ_2 and θ_3 by use of Euler-Lagrange equation

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = 0.$$

The 3 Euler-Lagrange equations are:

$$\frac{\partial L}{\partial \theta_1} = \frac{\partial}{\partial t}\frac{\partial L}{\partial \dot{\theta}_1} \qquad \frac{\partial L}{\partial \theta_2} = \frac{\partial}{\partial t}\frac{\partial L}{\partial \dot{\theta}_2} \qquad \frac{\partial L}{\partial \theta_3} = \frac{\partial}{\partial t}\frac{\partial L}{\partial \dot{\theta}_3}$$

Our next step is to construct the M and K matrices. They are the generalized mass and generalized spring coefficient matrices.
The matrices must satisfy this generalized harmonic matrix equation: $M\ddot{\Phi}_2 = -K\Phi$

We get:

$$M = mL^2 \begin{pmatrix} 3 & 2 & 1 \\ 2 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix} \qquad K = mgl \begin{pmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Let $\Phi = \vec{B}e^{(i\omega t)}$ where $\vec{B}$ is a 3-n column vector, $\vec{B} =< b_1, b_2, b_3 >$ containing complex elements containing whose phase in the complex phase is the phase shift and amplitude the magnitude.

$\Rightarrow \quad K - \omega^2 M = 0$

We define :

$\omega_0^2 \equiv \frac{g}{L}$ and $\omega_n = \frac{\omega^2}{\omega_0^2}$. $\qquad \frac{1}{mgL}K - \frac{1}{m}\omega_n M = 0$

Since these are square, non-singular matrices, the determinant of the left side must be zero for there to be a nontrivial solution. We are assured there is a solution since K is a diagonal matrix.

$$\Rightarrow \qquad \det(\frac{1}{gL}K - \omega_n M) = 0$$

$$= \det \begin{pmatrix} 3 - 3\omega_n & -2\omega_n & -\omega_n \\ -2\omega_n & 2 - 2\omega_n & -\omega_n \\ -\omega_n & -\omega_n & 1 - \omega_n \end{pmatrix}$$

Using MATLAB, we get three eigen frequencies as :

$$\omega_n = .415775, 2.29428, 6.28995$$

Expressing ω^2 terms of ω_0^2:

$$\omega^2 = .415775\omega_0^2, 2.29428\omega_0^2, 6.28995\omega_0^2$$

To find the eigenvectors we plug the eigenfrequencies into our matrix-vector equation:

$$(K - \omega^2 M)\vec{B} = 0$$

$$= \begin{pmatrix} 3-3\omega^2 & -2\omega^2 & -\omega^2 \\ -2\omega^2 & 2-2\omega^2 & -\omega^2 \\ -\omega^2 & -\omega^2 & 1-\omega^2 \end{pmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Eigen vectors contain information about each *normal mode behavior** and all modes can be re-written as a linear combination of these 3 eigenvectors. In the first normal mode all the masses are oscillating in phase. The bottom has the largest amplitude, the middle swings less than it and the top the least. In the second normal mode, the top two masses are in phase and the bottom is out of phase. The top mass is oscillating with the largest amplitude, a lot less amplitude for the middle mass and even less for the bottom mass. In the third normal mode the middle is oscillating with the highest amplitude and the top and bottom are oscillating out of phase from it. The top mass has a lower amplitude and the bottom mass the lowest amplitude.

***normal mode behavior:** Type of motion in which all parts of a system oscillate with the same angular frequency.

All the units of angular frequency are in rad/s.

We have L = 32 cm = 0.32 m
w_0^2 = 9.8/0.32 = 30.625

Therefore,

w^2 = 12.733109375, 70.262325, 192.62971875

w = 3.5683482699, 8.3822625227, 13.8791108775 are the natural frequencies of the system.
Eigen vectors: (using MATLAB)

19.0227	20.0690	-22.0086
-20.8612	17.2876	-5.1773
4.8628	10.9123	17.5931

Therefore, eigen vector corresponding to:

w = 3.56 is <19.022, -20.86, 4.86>...............................Eigenvector e_1

w = 8.38 is <20.06, 17.28, 10.91> ....Eigenvector e_2

w = 13.87 is <-22, -5.17, 17.59> ....Eigenvector e_3

5. Experimental Procedure

For the OPEN CV model to work, the bobs were wrapped with colored paper for ease of detection and the camera was calibrated to properly detect color and hence the bobs.

The pendulum was left off from the angles obtained as eigen vectors and hence angular frequency w was measured.

w = $\frac{2\Pi}{T}$ where T is the time period of a *beat**

A Lagrangian approach to the triple pendulum is utilized to derive the equations of motion. Linearization of the Lagrangian near its stable equilibrium yields tractable equations of motion. ***<u>The system's behavior is characterized by solving the matrix equation for its eigen frequencies and eigen vectors</u>*****.**

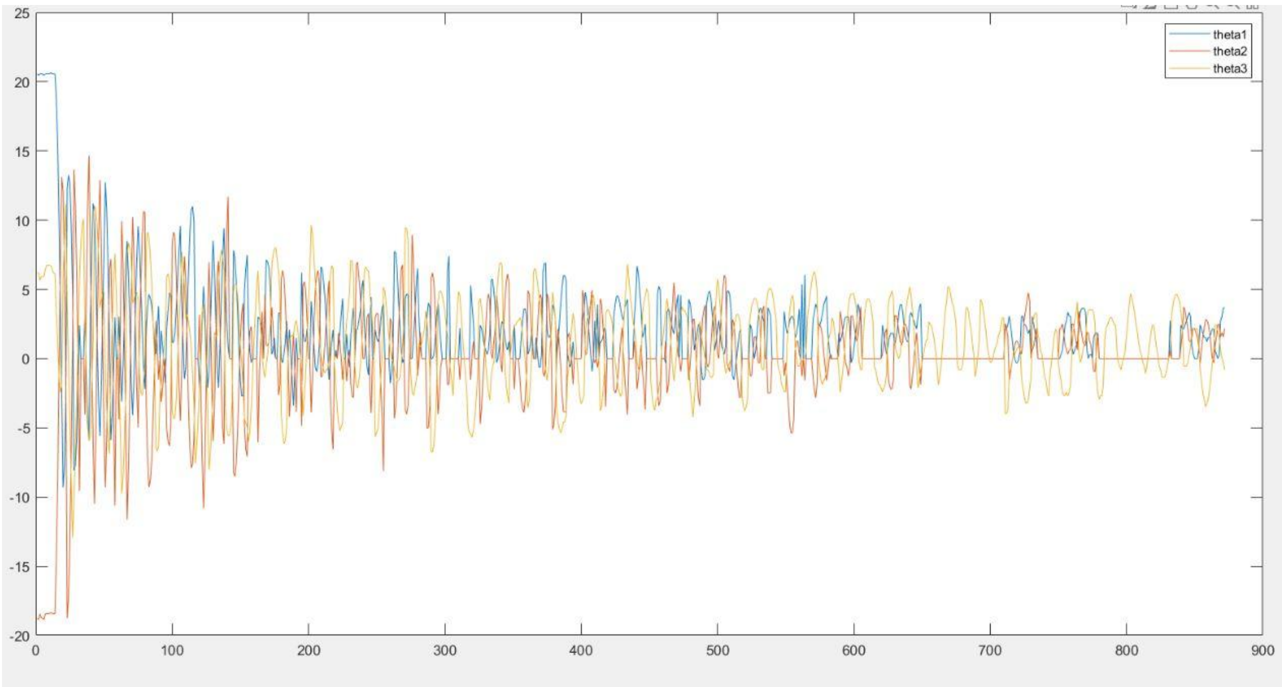
6. Comparison of experimental and theoretical results

Experimentally we got following values of frequency for eigen vectors e_1 , e_2 , e_3 :

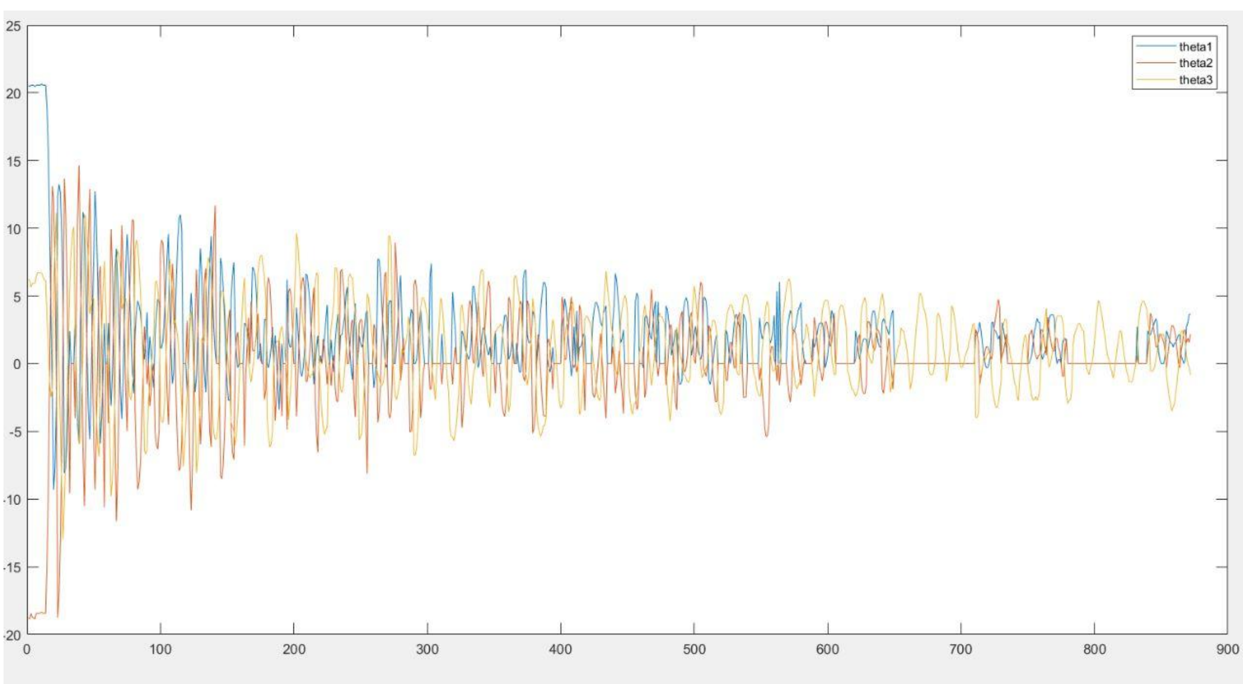
w_1 = 4
w_2 = 7.12
w_3 = 6.28

<u>The angular displacement vs time graph for the three modes are as follows:</u>

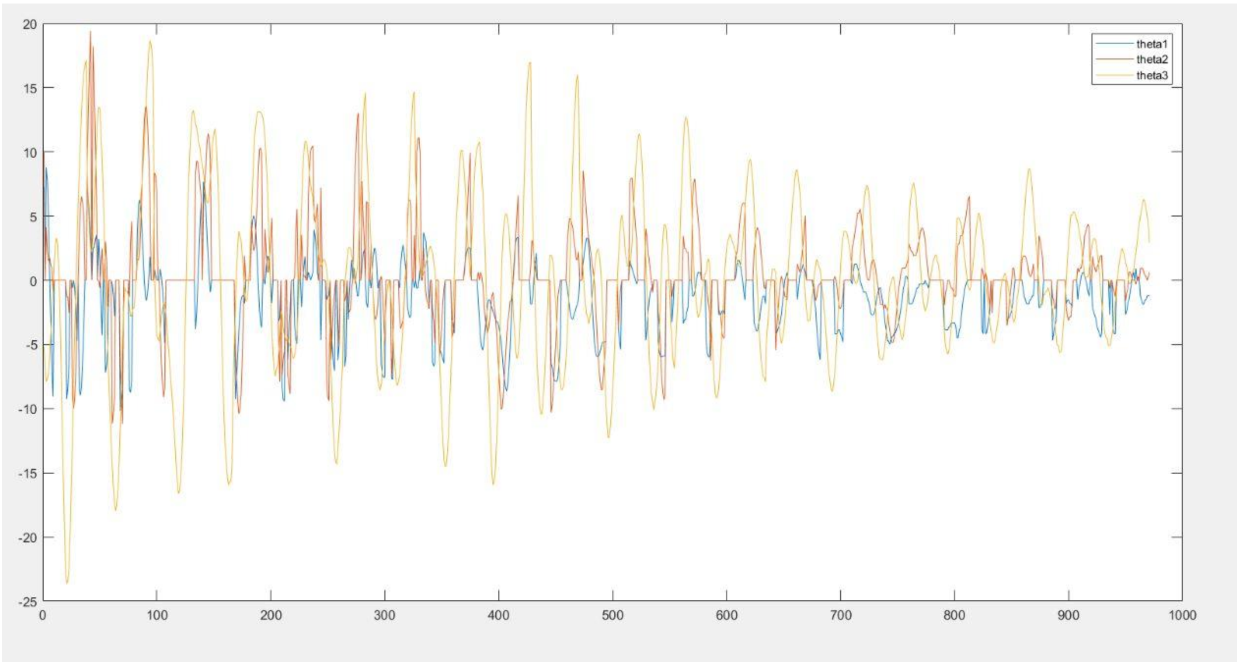
1. **For mode 1:**



2. **For mode 2:**

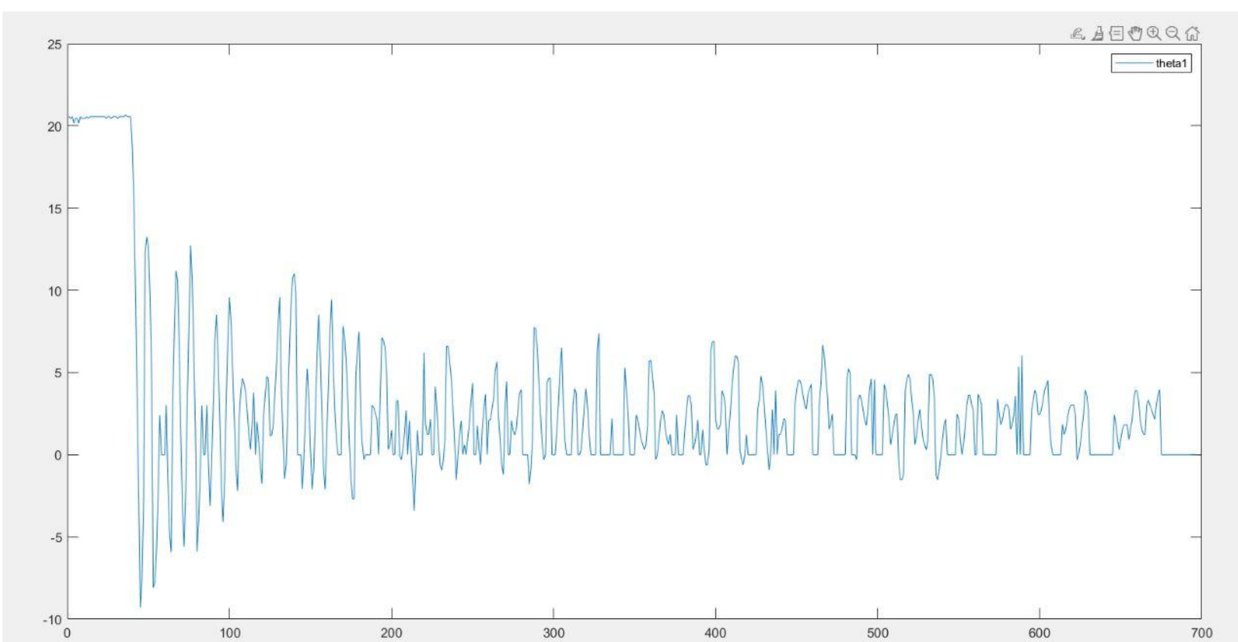


3. For mode 3:

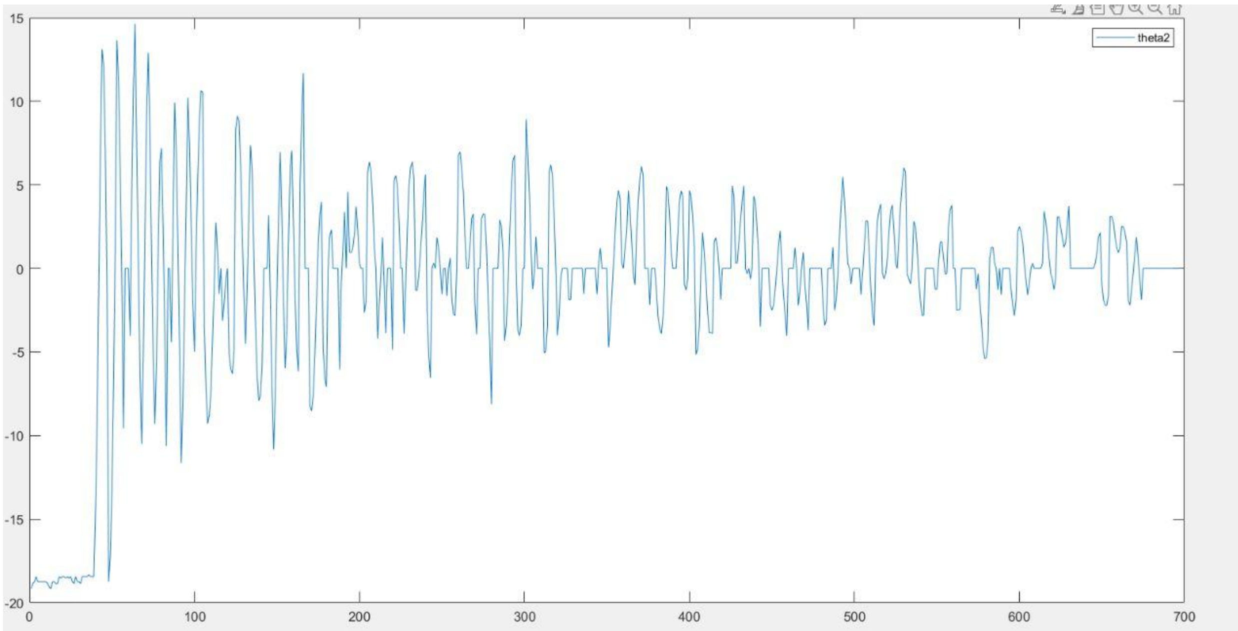


The displacement vs time graph of individual bob during all three modes are as follows:

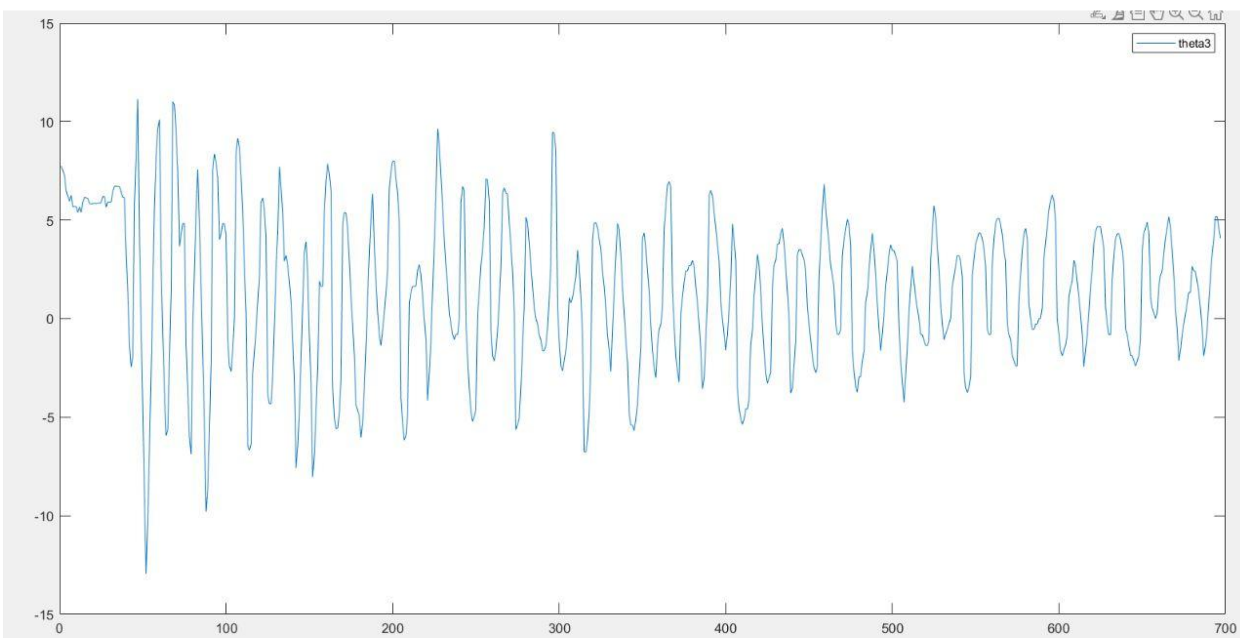
1. θ_1 vs t curve for mode 1:



2. θ_2 vs t curve for mode 1:



3. θ_3 vs t curve for mode 1:



And similarly all other curves for displacement vs t for individual bobs for mode 2 and 3 were obtained.

Comparison of experimental and theoretical results:

Frequencies	Theoretical	Experimental	Error (%)
w_1	3.56	4	12.35
w_2	8.38	7.12	15.03
w_3	13.87	6.28	54.72

7. Results and discussions

<u>*Beat period</u> refers to the time interval between two peaks or valleys in the graph, which correspond to the times when the pendulum reaches its maximum displacement from the vertical axis. The beat period is a measure of the oscillation frequency of the pendulum, and it depends on the characteristics of the system, such as its mass, length, and damping.

A driving frequency near the eigen frequency results in the system rapidly increasing in amplitude because of resonance. The steady-state solution of the system near its eigenfrequency results in beats. The beat frequency increases as the driving frequency deviates from the eigen frequency slightly.

It can be seen that at normal modes the system's amplitude quickly increases because it is experiencing resonance. The system's behavior is also more smooth and the oscillation is more sinusoidal than when the system is driven further from its normal mode.

In conclusion, the Langrangian approach combined with linearizing T and U, was utilized to find the M and K matrices. Matlab was used to solve the resulting eigenvalue problem. Setting the driving frequency near the eigen frequencies leads the system to experience resonance. Investigating the long-term steady state solution of our system near ω_1 uncovers a beat behavior. Choosing ω_1 and increasing the driving amplitude also decreases the beat period.

8. Source of discrepancy/mismatch

There can be several sources of discrepancy or mismatch in an experiment to measure the natural frequency of a 3DOF pendulum system. Here are some potential sources of error:

1. ***Uncertainty in the measurement of parameters*: The natural frequency of a 3DOF pendulum system depends on several parameters such as the masses, lengths, and angles of the pendulums. If these parameters are not measured accurately or there is uncertainty in their measurement, it can lead to errors in the calculated natural frequency.

2. ***Non-ideal boundary conditions*: In an experiment, the 3 DOF pendulum system may be subjected to non-ideal boundary conditions, such as friction, air resistance, or vibrations from other equipment in the laboratory. These boundary conditions can influence the behavior of the pendulum and lead to errors in the natural frequency measurement.

3. ***Nonlinear behavior*: A 3DOF pendulum system may exhibit nonlinear behavior if the amplitude of the motion is large or if the pendulums are displaced far from their equilibrium positions. Nonlinear behavior can lead to discrepancies between the measured and calculated natural frequencies.

4. ***Transient effects*: The initial conditions of the pendulum, such as the initial displacements or velocities, can influence its motion during the experiment. Transient effects can cause the pendulum to oscillate with a frequency that is different from its natural frequency.

5. ***Instrumentation errors*: The sensors used to measure the motion of the pendulum, such as accelerometers or position sensors, may have errors or noise that can lead to inaccuracies in the measured data.

6. ***Model inaccuracies*: The mathematical model used to describe the behavior of the 3 DOF pendulum system may not be accurate, particularly if the system exhibits complex or nonlinear behavior.

9. Scope for improvement

There are several possible areas for improvement in a 3DOF pendulum experiment where the natural frequency of vibration is measured. Here are some potential suggestions:

1. ***Accurate measurement of parameters*: The natural frequency of a 3DOF pendulum system depends on several parameters such as the masses, lengths, and angles of the pendulums. Improving the accuracy of measuring these parameters can lead to more precise calculation of the natural frequency.

2. ***Reducing external disturbances*: The natural frequency of a pendulum system can be influenced by external factors such as air currents or vibrations from nearby equipment. Reducing the impact of such disturbances by placing the apparatus in a vibration-free location or using shielding can improve the accuracy of the natural frequency measurement.

3. ***Improving instrumentation*: Using high-quality sensors and data acquisition systems can lead to more accurate and precise measurements of the pendulum's motion. In particular, high-speed cameras or other optical methods can allow for more precise measurements of the pendulum's position and angle.

4. ***Testing different configurations*: The natural frequency of a 3DOF pendulum system can be influenced by the relative masses and lengths of the pendulums. Testing different configurations, such as changing the mass or length of one of the pendulums, can help to better understand the relationship between the parameters and the natural frequency.

5. ***Using advanced analysis techniques*: Applying advanced signal processing techniques such as spectral analysis or wavelet transform can help to identify the natural frequency more accurately and to distinguish between closely spaced frequencies.

By improving these areas, the accuracy and precision of the natural frequency measurement of a 3DOF pendulum system can be improved, leading to a better understanding of its behavior and potential applications in engineering and physics.

10. Acknowledgements

We would like to express our sincerest gratitude to Mr. Jayprakash KR, Assistant Professor of Mechanical Engineering at IIT Gandhinagar, for his invaluable guidance and support throughout the completion of our experiment on '3 DOF System'. We also extend our appreciation to our TAs NagaVishnu, Vaibhav Tandel, Harsh Gupta, and Ishan for their dedicated support and motivation.

11. References

https://www.researchgate.net/publication/336868500_Equations_of_Motion_Formulation_of_a_Pendulum_Containing_N-point_Masses

https://www.youtube.com/watch?v=x8pGtspZrk0