STRUCTURAL CRACK ANALYSIS USING BILINEAR DYNAMICS

By:

Yashraj - 2101ME84

Vinay - 2101ME79

Key Discussions

01
OBJECTIVES

02
METHODOLOGY

03
RESULTS ANALYSIS

04
CONCLUSIONS

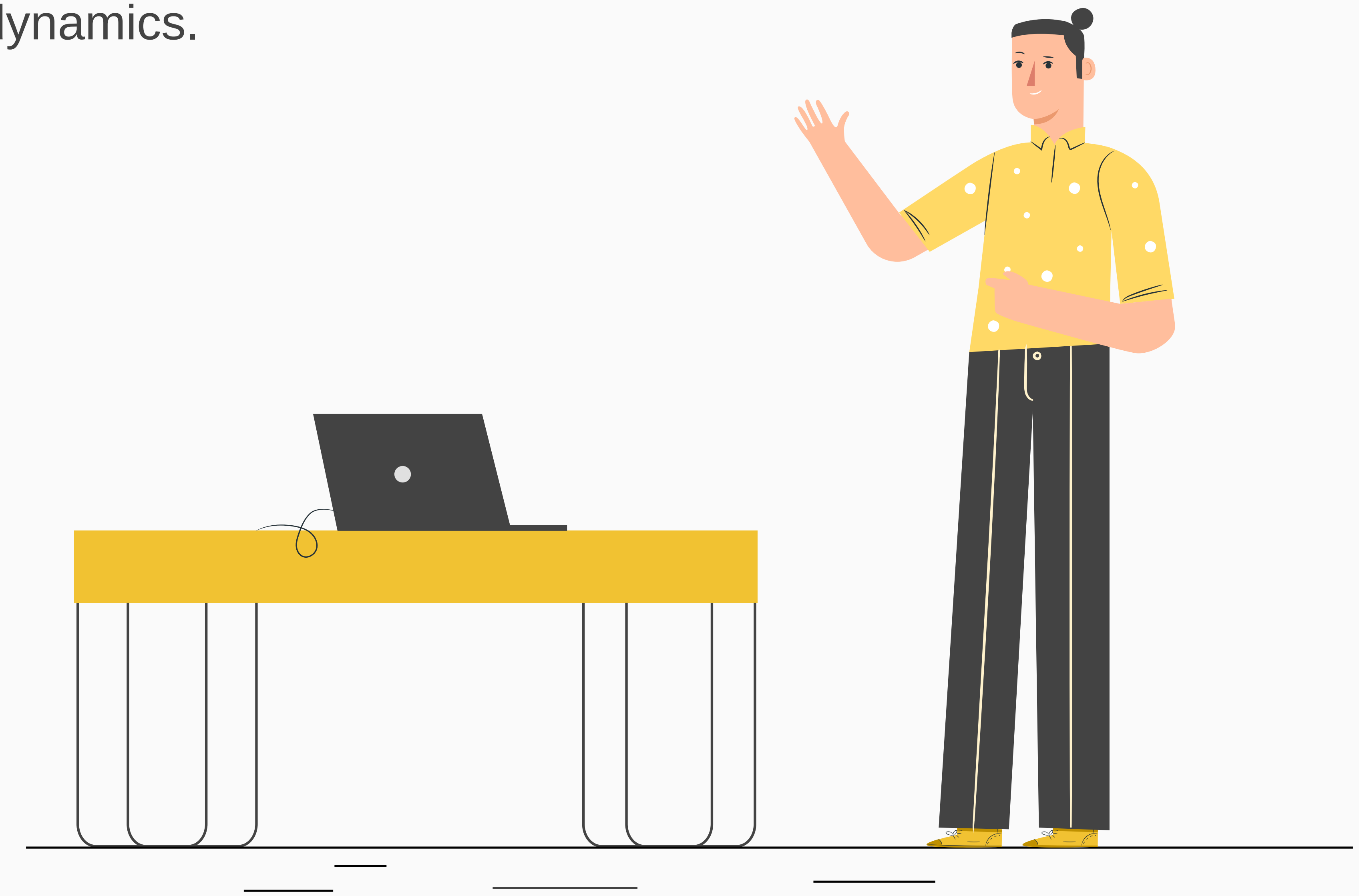
Motivation

Studying cracks with bilinear dynamics is vital for addressing gear backlash, loose parts, and structural damage, particularly in cantilever beams. Cracks introduce nonlinearity, leading to chaotic oscillations and increased failure risk. Early detection is critical, and the challenge is to model these effects accurately and create cost-effective solutions for real-time damage detection.

1. Structural Health Monitoring
2. Predicting issues like gear backlash and loose components
3. Modeling how cracks affect stiffness and dynamic response
4. Real-Time Damage Detection

OBJECTIVE

To analyze the bilinear dynamics of cracks in structures using a nonlinear spring-mass system. The system models crack-induced nonlinearity by engaging certain springs based on displacement thresholds. Using differential equations and numerical simulations, the study explores frequency and displacement responses under different initial conditions. The analysis also focuses on energy conservation in scenarios where stiffness increases, simulating the effects of crack propagation. Results, including frequency vs. displacement plots, highlight how crack-induced nonlinearities influence structural dynamics.

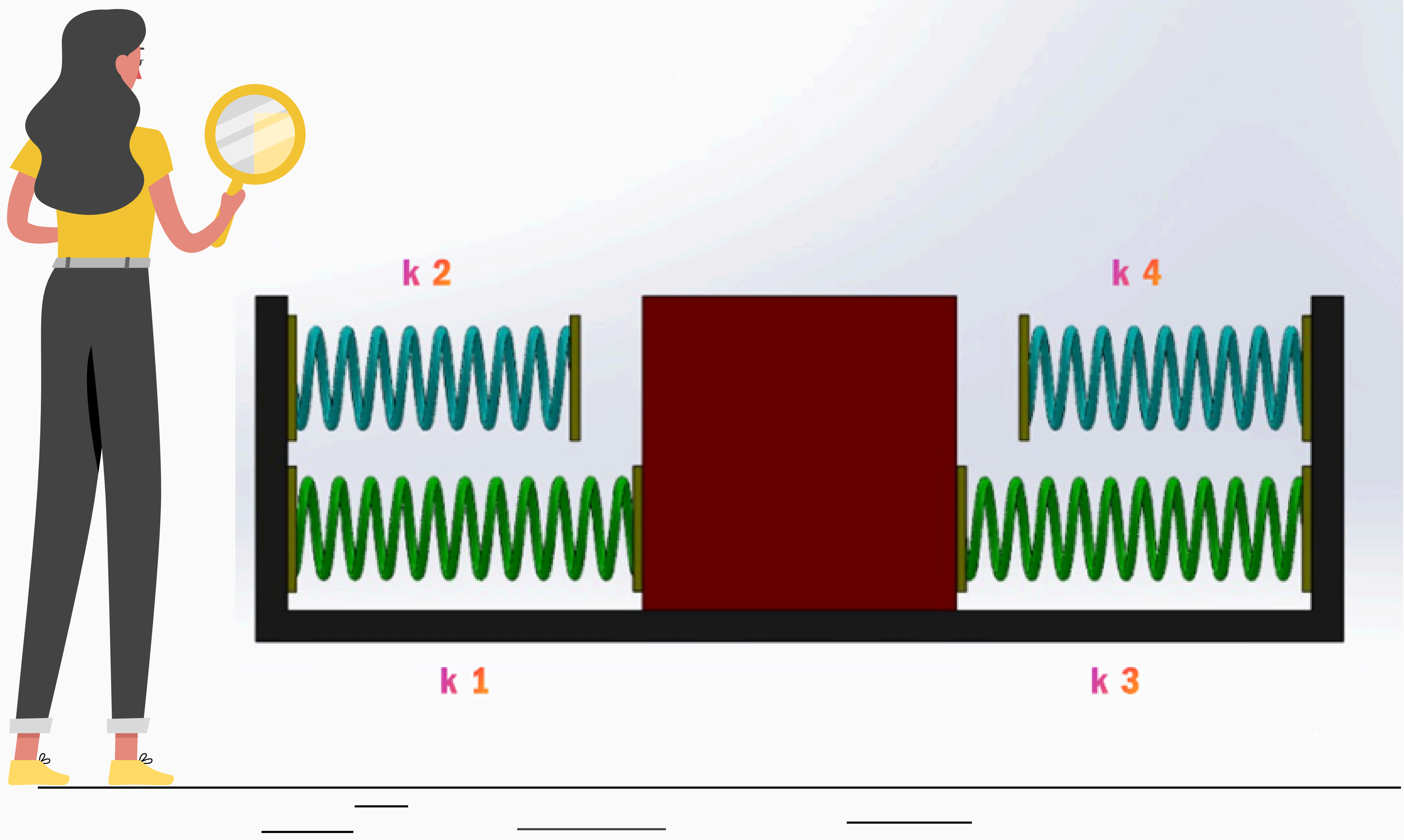


METHODOLOGY

To replicate the bilinear dynamics of cracks in a cantilever beam, a spring-mass system was created incorporating two non-engaging springs to introduce nonlinearity. Forced harmonic vibration was then applied to simulate effects analogous to crack breathing in beams. This approach enables the examination of how nonlinearity influences the system’s dynamic behavior and provides insights into the structural response under conditions similar to those caused by cracks in cantilever beams.

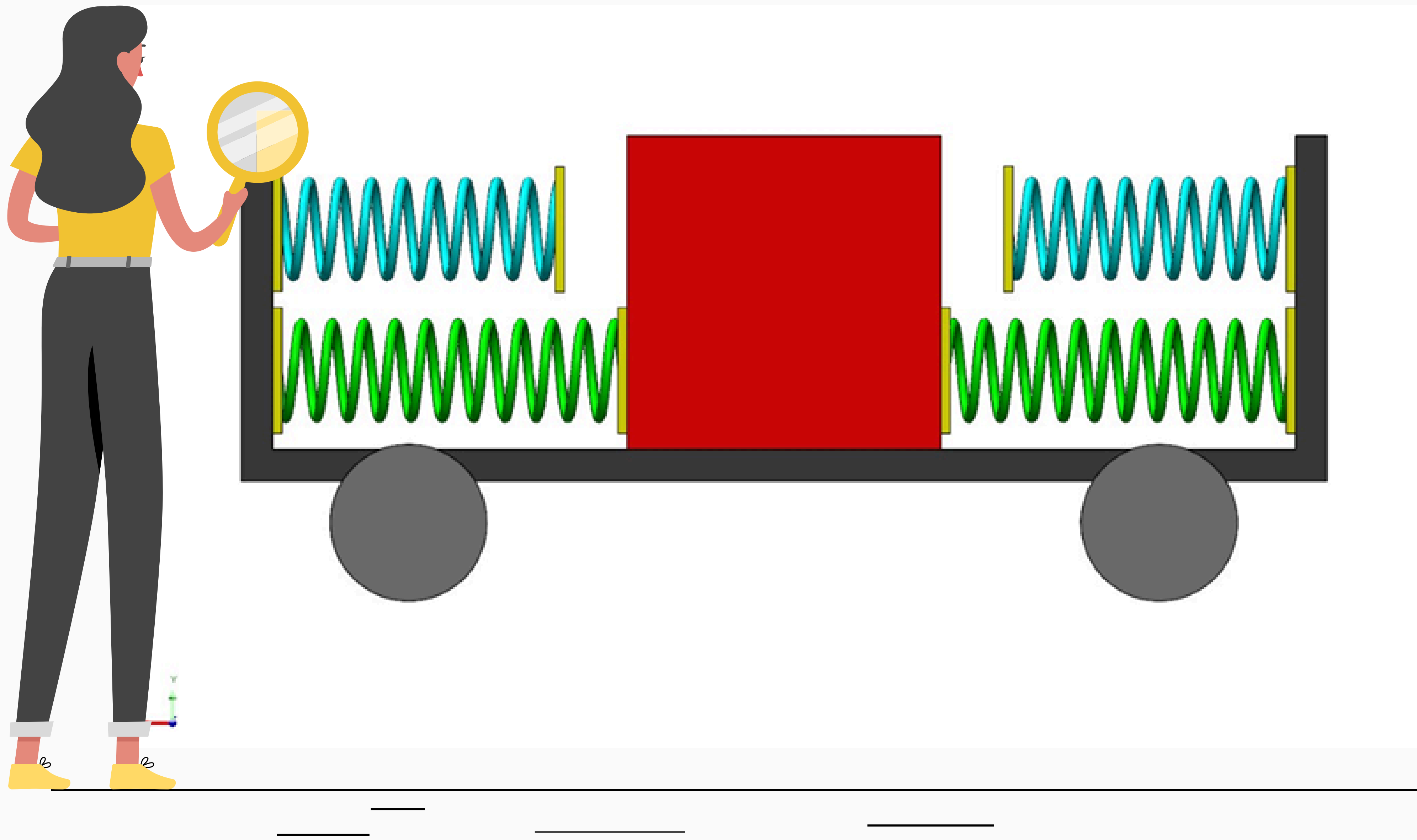
FREE BILINEAR VIBRATION

The setup involves a nonlinear mechanical system consisting of four springs connected to a single block. Each spring has distinct characteristics, and the engagement of some springs is conditional upon the displacement of the block. The springs are labelled as Spring 1, Spring 2, Spring 3, and Spring 4, with each having its spring constant denoted as k1 , k2, k3, and k4 respectively.



FORCED BILINEAR VIBRATION

The setup involves a nonlinear mechanical system consisting of four springs connected to a single block similar to the Free Bilinear Vibration Setup with the changes made to the base of system by attaching wheels and subjecting the system to a harmonic vibration.



Derived Equations

Bilinear Free Vibration

$$x1 = id * \cos\left(\sqrt{\frac{Keff1}{m}} * t\right) + id(initial\ position)$$

$$x2 = A2 * \cos\left(\sqrt{\frac{Keff2}{m}} * t + \Delta t1\right)$$

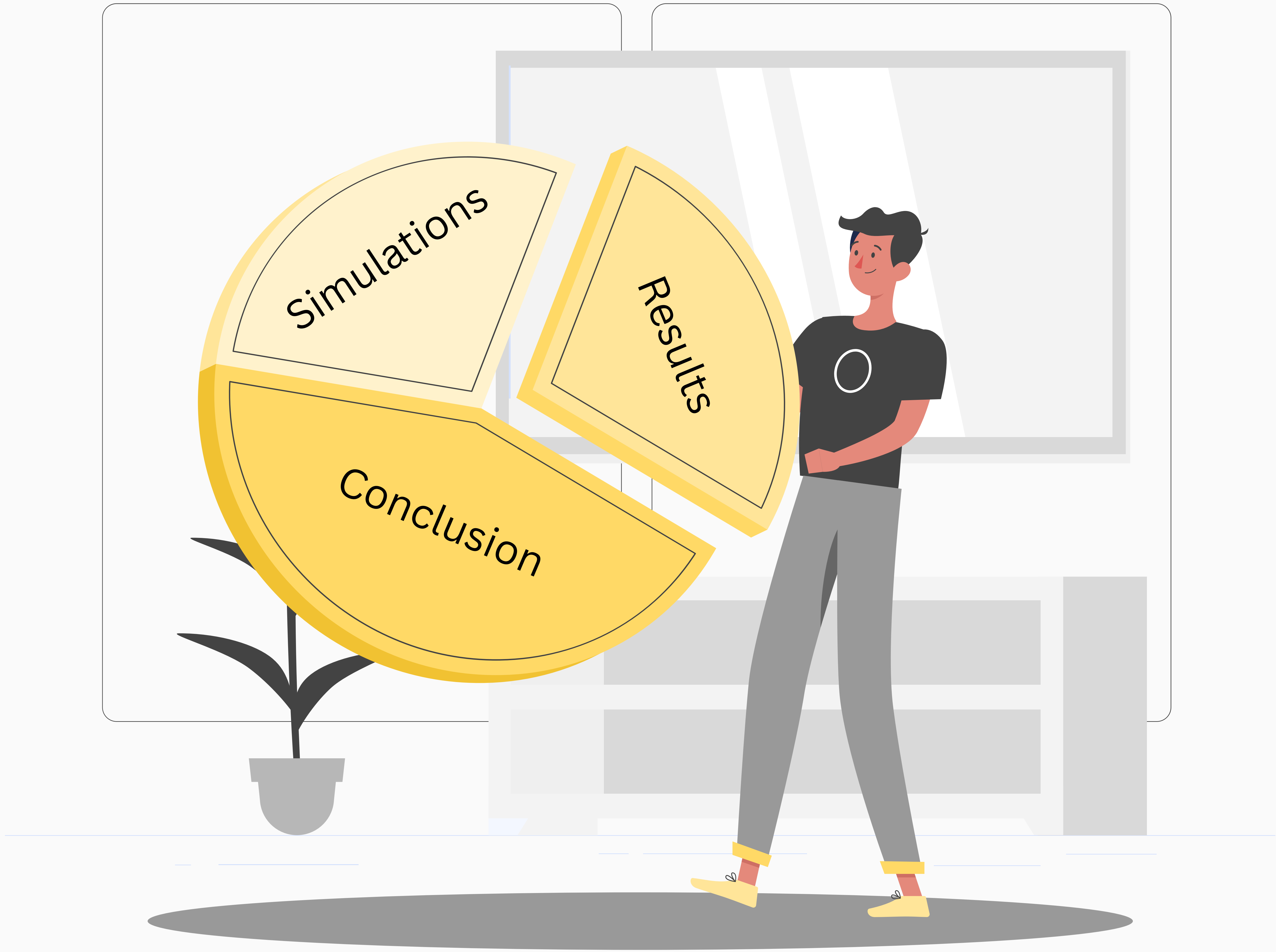
$$x3 = A3 * \cos\left(\sqrt{\frac{Keff3}{m}} * t + \Delta t1 + \Delta t2\right)$$

Bilinear Forced Vibration

$$x(t) = \left(id - \frac{Fe}{m} * \frac{1}{(wn)^2 - (wo)^2)}\right) * \cos(wn3 * t) + \left(\frac{Fe}{m} * \frac{1}{(wn)^2 - (w)^2)}\right) * \cos(wo * t)$$

$$x(t) = \frac{Vo}{wn2} * (\sin(wn2 * t)) + \left(d - \frac{Fe}{m} * \frac{1}{(wn)^2 - (wo)^2)}\right) * \cos(wn2 * t) + \left(\frac{Fe}{m} * \frac{1}{(wn)^2 - (wo)^2)}\right) * \cos(wo * t)$$

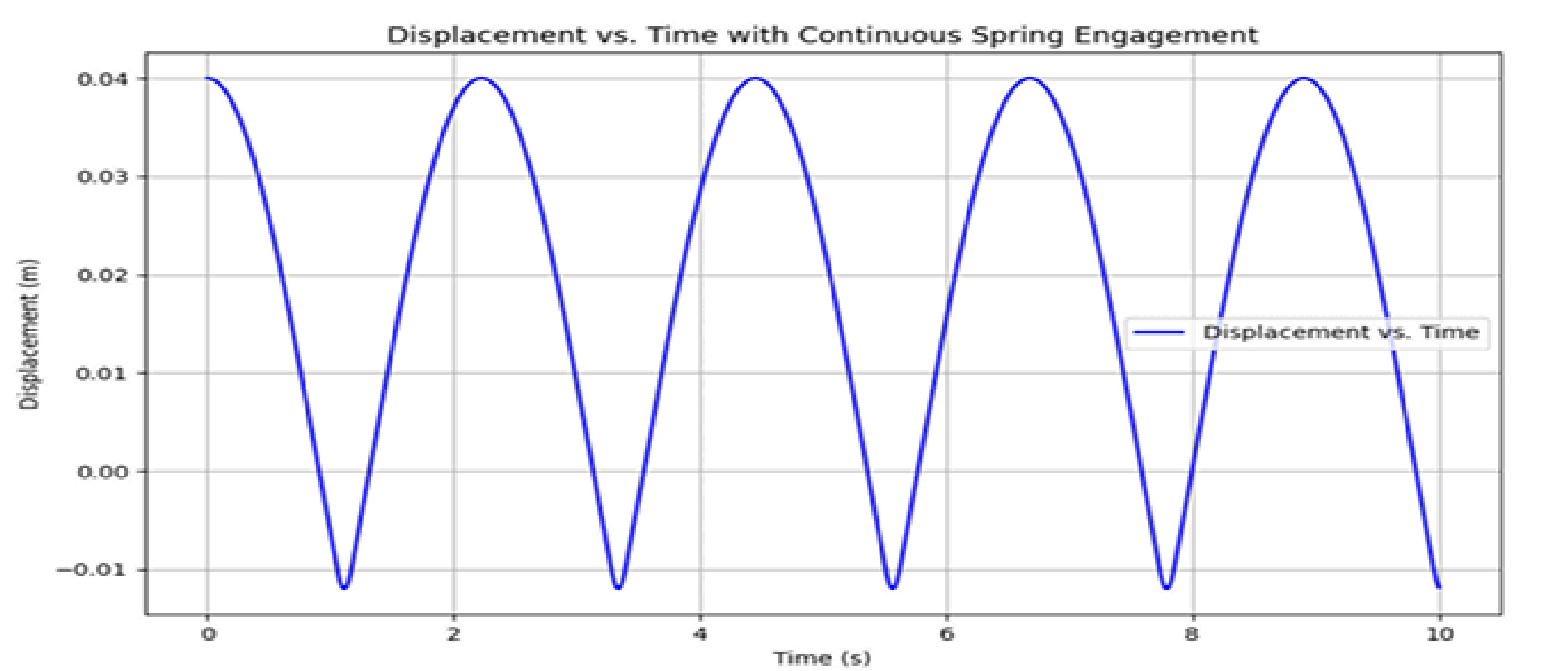
$$x(t) = \left(id - \frac{Fe}{m} * \frac{1}{(wn)^2 - (wo)^2)}\right) * \cos(wn1 * t) + \left(\frac{Fe}{m} * \frac{1}{(wn)^2 - (w)^2)}\right) * \cos(wo * t)$$


Simulations
Results
Conclusion

```
m_val= 1, d_val= 0.01
k1_val= 1, k2_val= 1, k3_val= 1, k4_val= 100
# Define the time span for simulation
tspan = [0, 10], t_eval = np.linspace(0, 10, 1000)# points for smoother plot
# Define the ODE function that dynamically changes based on displacement
def dynamic_ode(t, y):
    x, v = y  # Displacement and velocity
    # Determine which springs are engaged based on the current displacement x
    if abs(x) < d_val:  # Case 1: Not in contact with Spring 2 or Spring 4
        k_eff = k1_val + k3_val
    else:
        if d_val <= x:  # Case 2: Spring 2 engages
            k_eff = k1_val + k2_val + k3_val
        else:  # Case 3: Spring 4 engages
            k_eff = k1_val + k3_val + k4_val

    dxdt = v
    dvdt = -k_eff / m_val * x
    return [dxdt, dvdt]
# Initial conditions: [initial displacement, initial velocity]
initial_conditions_40mm = [0.04, 0]  # 40 mm
# Solve the ODE for the initial condition with increased solver accuracy
sol = solve_ivp(dynamic_ode, tspan, initial_conditions_40mm, t_eval=t_eval, rtol=1e-9, atol=1e-12)
# sol = solve_ivp(dynamic_ode, tspan, initial_conditions_40mm, t_eval=t_eval)
```

Code For Free Bilinear Vibrations

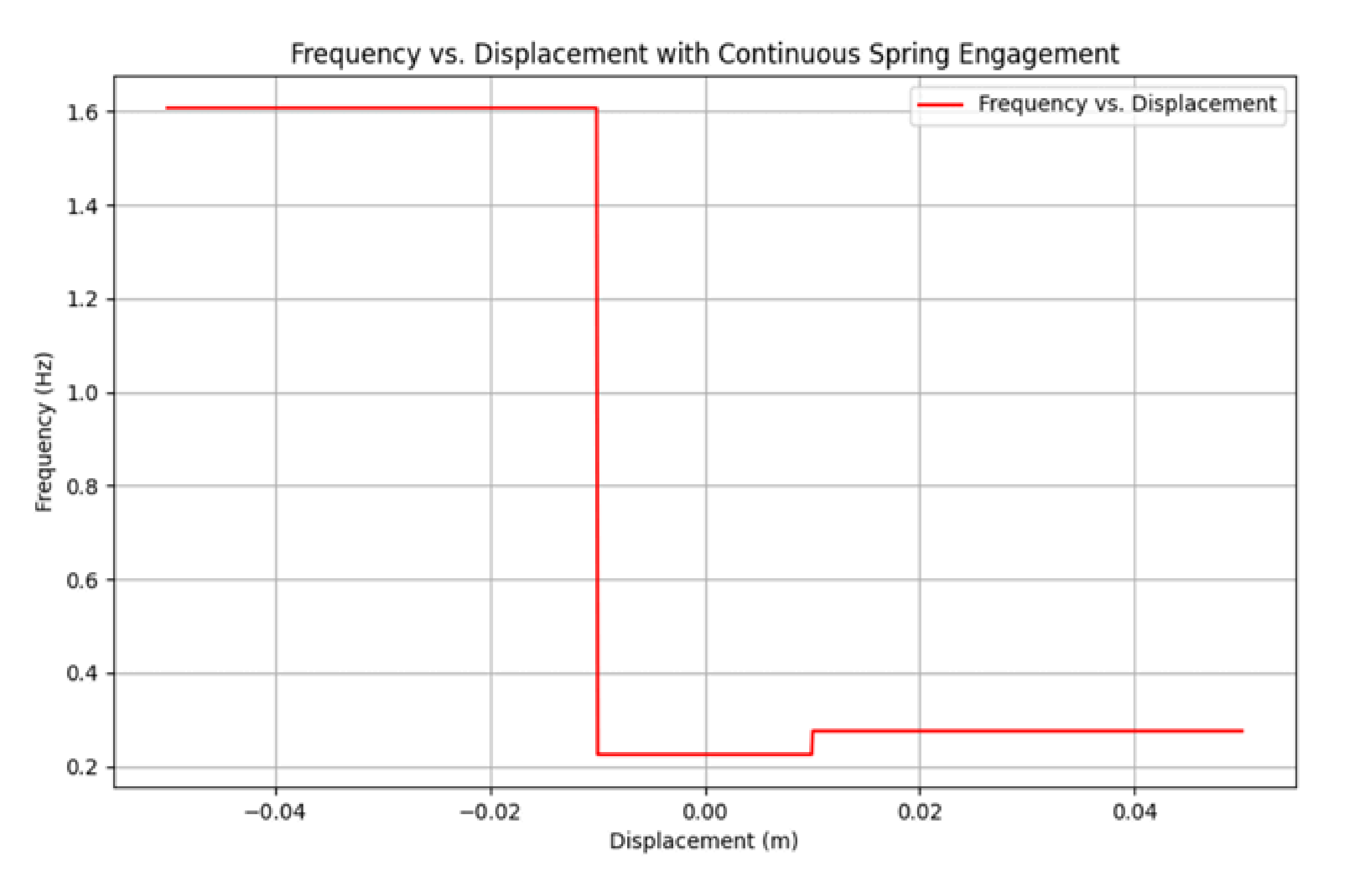


DISPLACEMENT V/S TIME GRAPH

```
m_val= 1, d_val= 0.01
k1_val= 1, k2_val= 1, k3_val= 1, k4_val= 100
# Define a range of displacements to test
displacements = np.linspace(-0.05, 0.05, 1000)  # From -50 mm to 50 mm
# Initialize an array to store frequencies
frequencies = np.zeros_like(displacements)
# Calculate the frequency for each displacement
for i, x in enumerate(displacements):
    # Determine which springs are engaged based on the current displacement x
    if abs(x) < d_val:  # Case 1: Not in contact with Spring 2 or Spring 4
        k_eff = k1_val + k3_val
    else:
        if d_val <= x:  # Case 2: Spring 2 engages
            k_eff = k1_val + k2_val + k3_val
        else:  # Case 3: Spring 4 engages
            k_eff = k1_val + k3_val + k4_val

    # Calculate the natural frequency
    frequency = (1 / (2 * np.pi)) * np.sqrt(k_eff / m_val)
    frequencies[i] = frequency
```

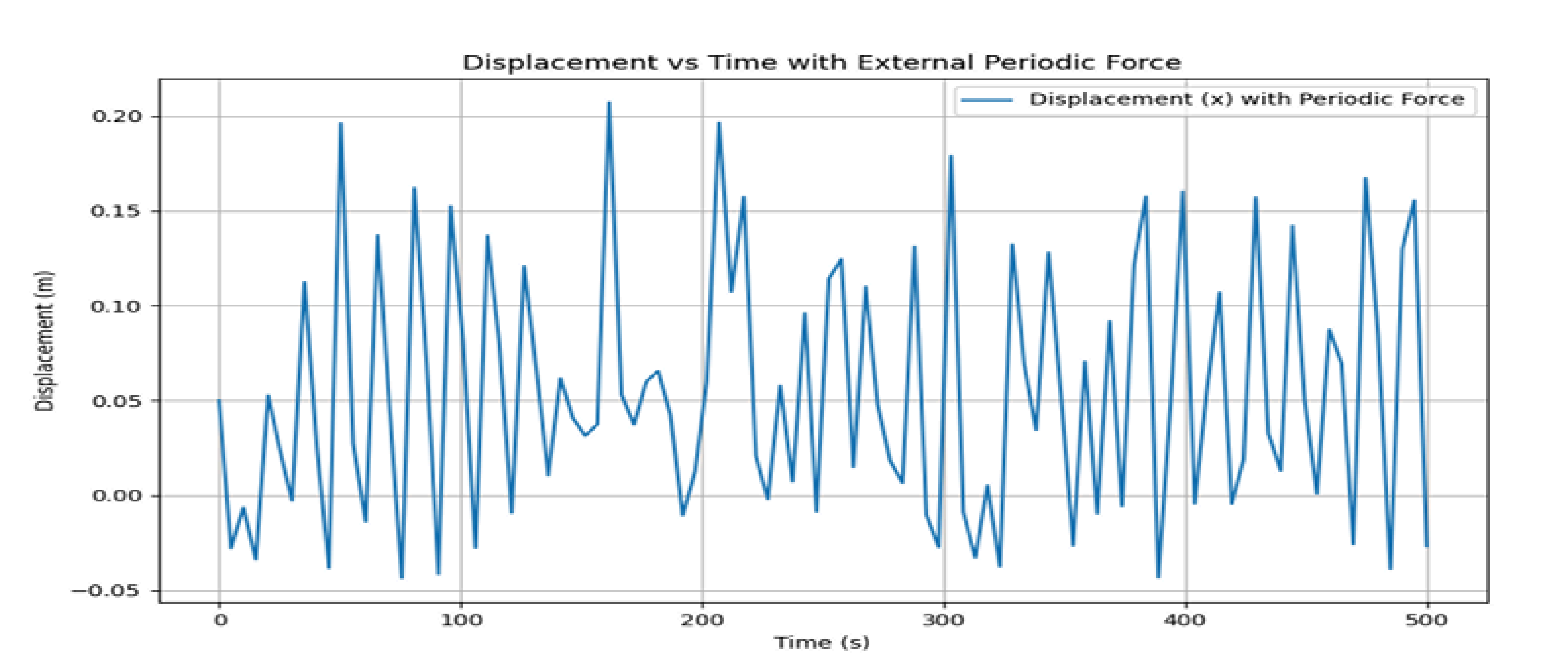
Code For Free Bilinear Vibrations



FREQUENCY V/S DISPLACEMENT GRAPH

Code For Forced Bilinear Vibrations

```
# Re-define constants and initial conditions
m = 1.0   # Mass of the block
k1 = 1.0, k2 = 3.0, k3 = 1.0, k4 = 100.0 # Spring constants
x_threshold_k3 = 0.04 # Threshold for k3 to engage
x_threshold_k4 = 0.04 # Threshold for k4 to engage
F0 = 1   # Amplitude of the periodic force
omega = 10 # Angular frequency of the periodic force (adjust as needed)
# Initial conditions
x0 = 0.05 # Initial displacement (m)
v0 = 0.0  # Initial velocity (m/s)
initial_conditions = [x0, v0]
# Time span
t_span = (0, 500)
t_eval = np.linspace(*t_span, 100)
# Modified system of differential equations with external periodic force
def spring_mass_system_with_force(t, y):
    x, v = y
    # Forces from the springs
    F1 = -k1 * x
    F2 = -k2 * x
    F3 = -k3 * (x - x_threshold_k3) if x > x_threshold_k3 else 0
    F4 = -k4 * (x - x_threshold_k4) if x < -x_threshold_k4 else 0
    # External periodic force F(t) = F0 * sin(omega * t)
    F_external = F0 * np.sin(omega * t)
    # Net force including the external periodic force
    F_net = F1 + F2 + F3 + F4 + F_external
    a = F_net / m  # Accelerationw
    return [v, a]
# Solve the modified system with external periodic force
sol_with_force = solve_ivp(spring_mass_system_with_force, t_span, initial_conditions, t_eval=t_eval,
rtol=1e-9, atol=1e-12)
# Extract results
t = sol_with_force.t
x = sol_with_force.y[0]
v = sol_with_force.y[1]
# Calculate energy
kinetic_energy = 0.5 * m * v**2
potential_energy = 0.5 * k1 * x**2 + 0.5 * k2 * x**2 + \
                0.5 * k3 * np.where(x > x_threshold_k3, (x - x_threshold_k3)**2, 0) + \
                0.5 * k4 * np.where(x < -x_threshold_k4, (x - (-x_threshold_k4))**2, 0)
total_energy = kinetic_energy + potential_energy
```

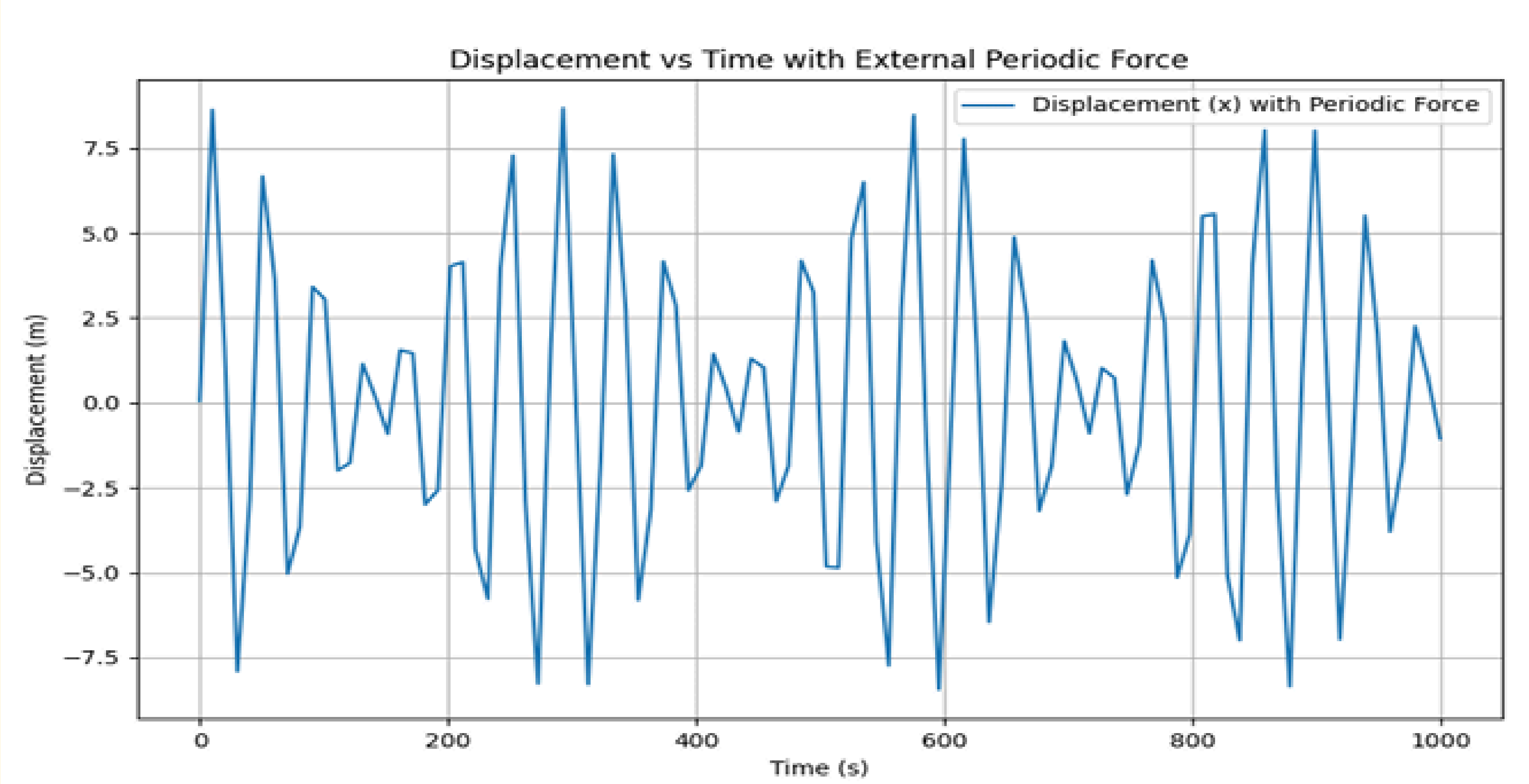


DISPLACEMENT V/S TIME GRAPH

Code For Forced Bilinear Vibrations

```
# Re-define constants and initial conditions
m = 1.0   # Mass of the block
k1 = 1.0, k2 = 1.0, k3 = 1.0, k4 = 1.0 # Spring constants
x_threshold_k3 = 0.04 # Threshold for k3 to engage
x_threshold_k4 = 0.04 # Threshold for k4 to engage
F0 = 1   # Amplitude of the periodic force
omega = 1.4 # Angular frequency of the periodic force (adjust as needed)
# Initial conditions
```

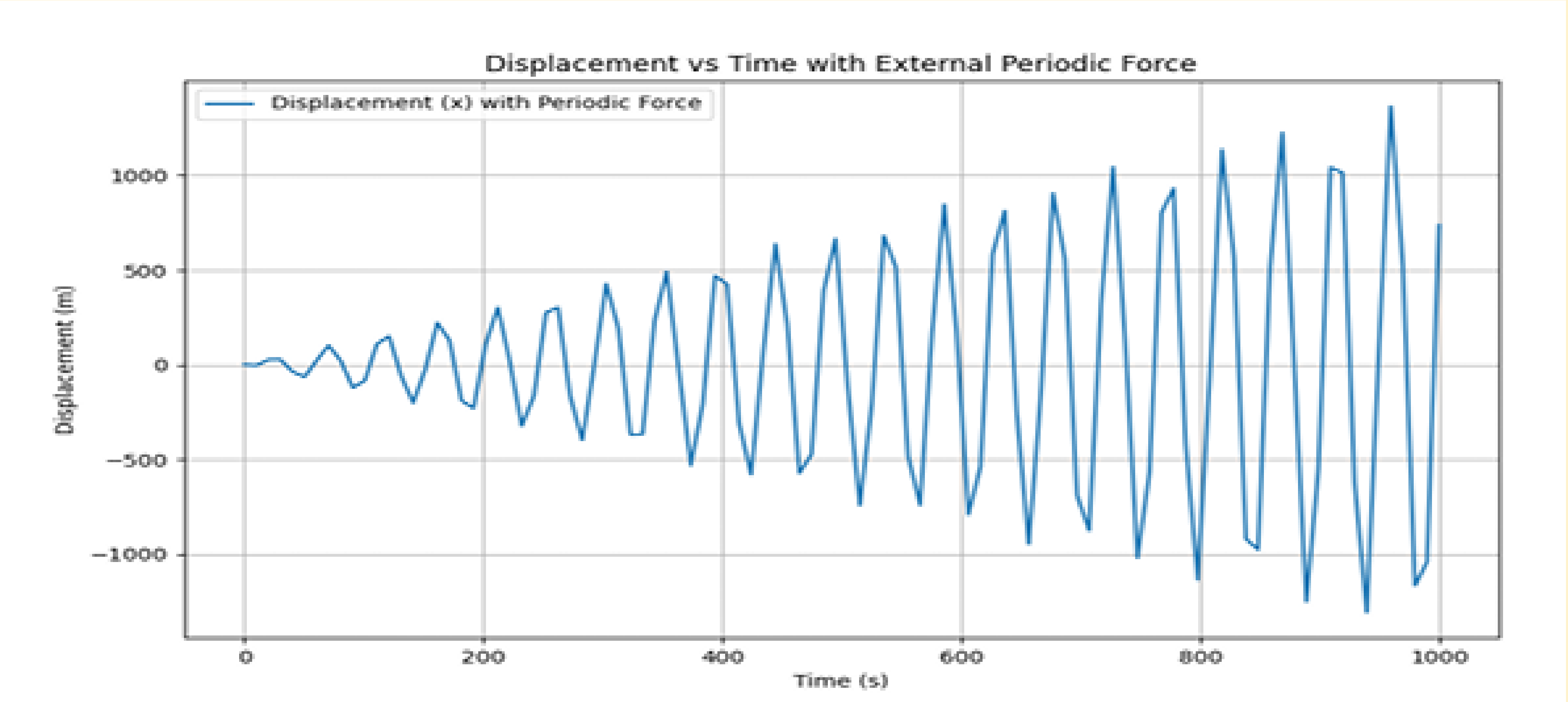
If we change the frequency for the applied force and bring it to closer proximity to the natural frequencies of the system, then we witness a beating frequency.



DISPLACEMENT V/S TIME GRAPH

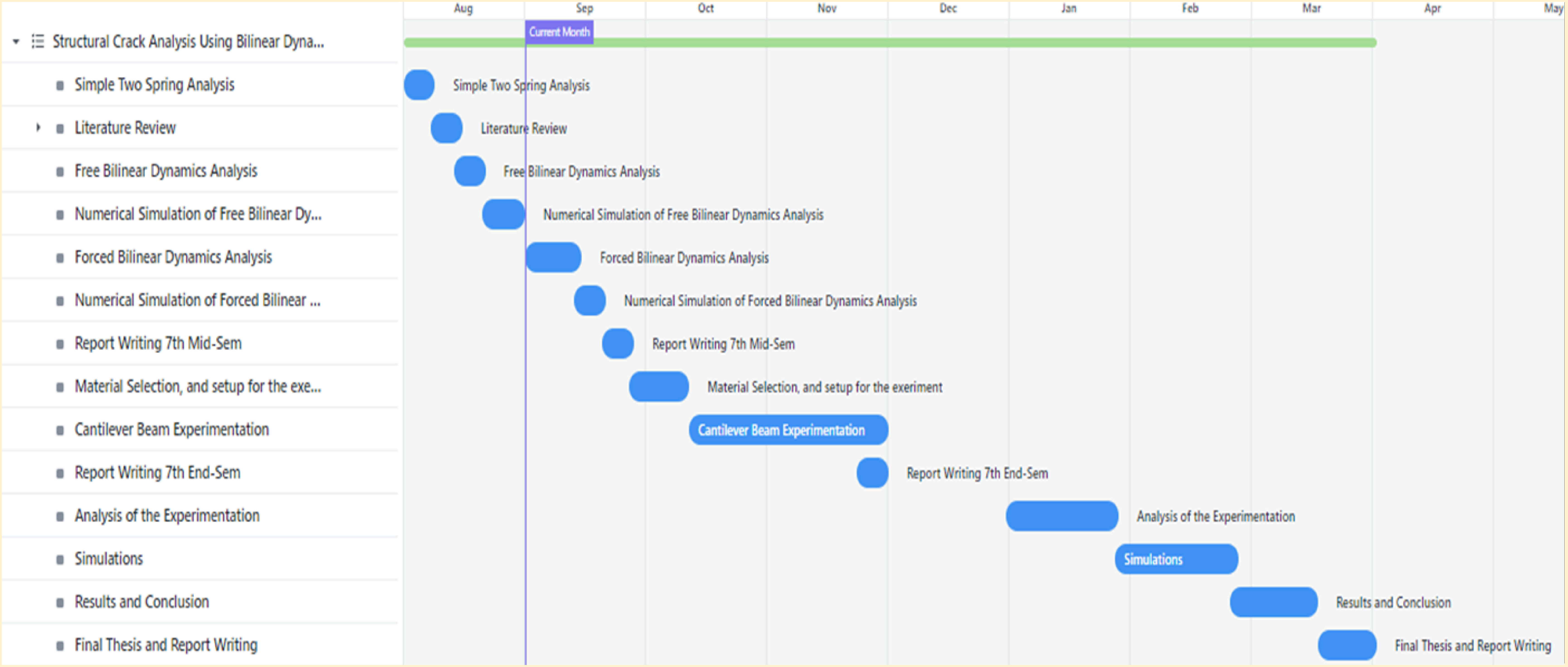
Code For Forced Bilinear Vibrations

```
# Re-define constants and initial conditions
m = 1.0   # Mass of the block
k1 = 1.0, k2 = 1.0, k3 = 1.0, k4 = 1.0
x_threshold_k3 = 0.04 # Threshold for k3 to engage
x_threshold_k4 = 0.04 # Threshold for k4 to engage
F0 = 5   # Amplitude of the periodic force
omega = 1.732050 # Angular frequency of the periodic force (Natural Freq)
```



Changing the frequency of the oscillating force to natural frequency results in resonance and uncontrolled oscillation.

DISPLACEMENT V/S TIME GRAPH

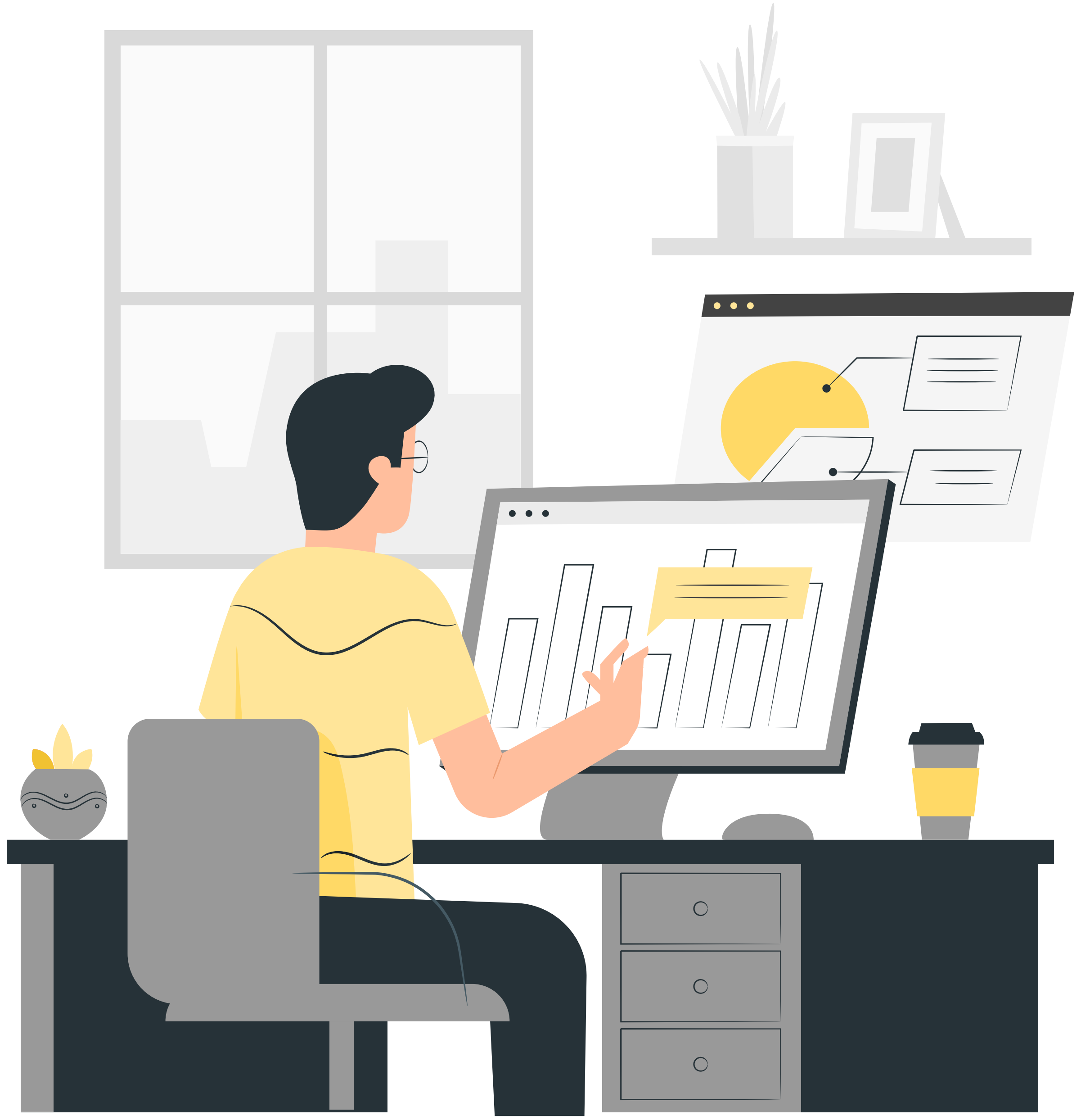

Aug
Sep
Oct
Nov
Dec
Jan
Feb
Mar
Apr
May
Current Month
Structural Crack Analysis Using Bilinear Dyna...
Simple Two Spring Analysis
Literature Review
Free Bilinear Dynamics Analysis
Numerical Simulation of Free Bilinear Dy...
Forced Bilinear Dynamics Analysis
Numerical Simulation of Forced Bilinear ...
Report Writing 7th Mid-Sem
Material Selection, and setup for the exe...
Cantilever Beam Experimentation
Report Writing 7th End-Sem
Analysis of the Experimentation
Simulations
Results and Conclusion
Final Thesis and Report Writing
Simple Two Spring Analysis
Literature Review
Free Bilinear Dynamics Analysis
Numerical Simulation of Free Bilinear Dynamics Analysis
Forced Bilinear Dynamics Analysis
Numerical Simulation of Forced Bilinear Dynamics Analysis
Report Writing 7th Mid-Sem
Material Selection, and setup for the exeriment
Cantilever Beam Experimentation
Report Writing 7th End-Sem
Analysis of the Experimentation
Simulations
Results and Conclusion
Final Thesis and Report Writing

GANTT CHART

CONCLUSION

The numerical simulations and resulting graphs confirm that the system's behavior matches the theoretical equations for bilinear free and forced vibrations. The plots accurately depict displacement and frequency responses, validating that nonlinear spring engagement due to displacement thresholds creates the expected bilinear dynamics. These results highlight the strong correlation between numerical and analytical methods, demonstrating the bilinear model's effectiveness in predicting structural behavior under various vibration scenarios.

- Validation: Numerical simulations align with theoretical equations for bilinear vibrations.
- Results: Plots accurately reflect displacement and frequency responses.
- Findings: Nonlinear spring engagement produces anticipated bilinear dynamics.
- Correlation: Strong agreement between numerical and analytical methods.
- Conclusion: Bilinear model effectively predicts structural behavior under free and forced vibrations.



FUTURE WORK

This report outlines our planned experimental investigation into the nonlinear dynamics of cantilever beams with 250-micrometer grooves. We will conduct experiments to replicate bilinear dynamics, where small vibrations result in linear behavior and larger displacements engage the grooves, introducing nonlinearity by altering the effective spring constant.

- System Setup: We will prepare cantilever beams with grooves and apply vibrational input to observe the transition from linear to nonlinear dynamics.
- Data Collection: Displacement sensors and frequency analysis will be used to record system behavior and track the shift from linear to nonlinear motion.
- Graphical Analysis: We will generate displacement-time and frequency-displacement plots to visualize system behavior and resonance changes.
- Comparison with Models: Experimental results will be compared with theoretical and numerical models to validate predictions of nonlinear effects.

REFERENCES

AUTHOR	Year	Title of the publication.	Publisher
S. W. SHAW	1985	Forced Vibrations of a Beam with One-Sided Amplitude Constraint: Theory and Experiment	Journal of Sound and Vibration.
R. M. LIN AND D. J. EWINS	1993	Mechanical Systems and Signal Processing.	Chaotic Vibration of Mechanical Systems with Backlash


THANKS