Jeevith Gnanakumaran Selwyn College Lab Group 12

A1 Vibrations – Exploring the use of multiple vibration absorbers and positioning of dampers in buildings

05/02/2019

Abstract

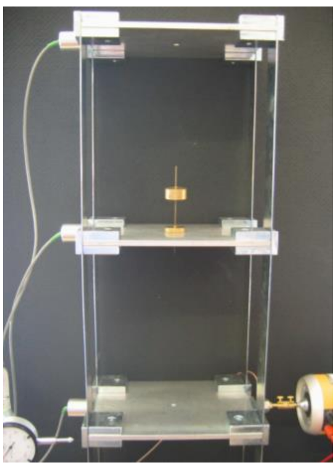
Buildings usually incorporate a single tuned mass damper which is tuned so that it absorbs kinetic energy at the resonant frequency of the building. This is useful in preventing buildings from collapsing during earthquakes or cyclic wind loads. However, when adding a tuned mass damper, it increases the number of Degrees of Freedom (DOF), hence the original 3DOF system (model of a 3 story building) becomes a 4DOF system with 4 resonant frequencies. Without any damping, this would imply the system is actually more dangerous than the previous system even though it now produces anti-resonance at one of the resonant frequencies in the 3DOF system.

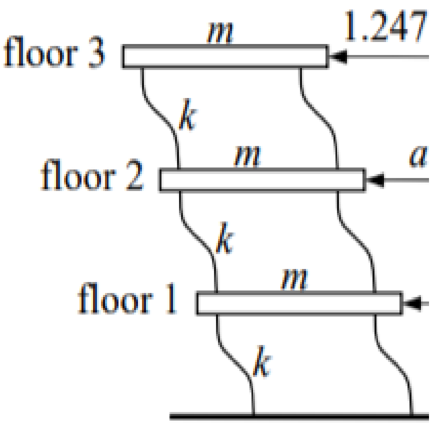
Motivation for experiment

The objective of this experiment is to see whether multiple vibration absorbers achieve a more resilient structure even though the DOF increases. This experiment can be experimentally done; however, computational simulation could provide answers (with assumptions) to adding 1, 10, 50, 100 or even more vibration absorbers with ease. Positioning of the vibration absorber also plays a key role in preventing large resonance (which is why swimming pools are placed on certain floors). Doing this experiment will provide suggestions to the location, size, and number of vibration absorbers required to optimise a structure's frequency response. Another consideration is to see the effect of the impulse response of the structure and see whether there is any effect from adding vibration absorbers to the impulse response. Earthquakes can cause harmonic inputs up to 20Hz, so measuring the frequency response up to 130rad/s is sufficient enough to measure the model's response to earthquakes.

Plan for experiment

This experiment will involve simulating a linear system, so matrix operations will be crucial. For data representation, a software package that is able to provide graphs is also required. Matlab does provide a useful resource, however, as I am more confident with using Python, (which includes Numpy, Scipy, and Matplotlib), this experiment can be done equally as well in Python.





Figures 1 (left) and 2 (right): Figure 1 represents the actual model we used in previous experiments, and figure 2 represents the mathematical model used. The value k can be calculated using compatibility constraints (rotation of each beam is 0 at each floor).

Measuring the frequency response of the model with one absorber

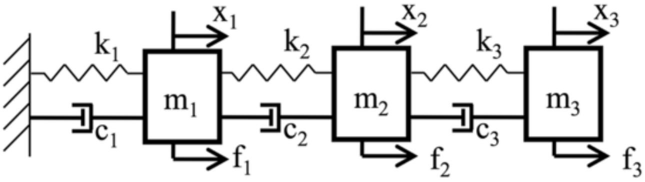
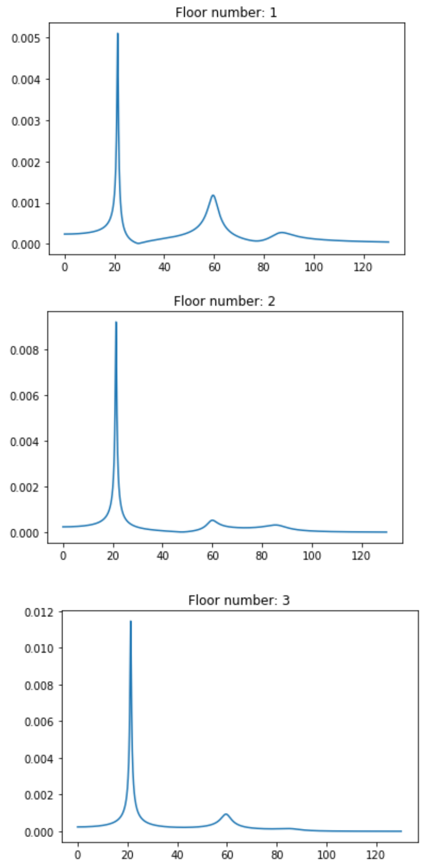


Figure 3



Figures 4, 5, and 6

Figure 3 shows the equivalent model of the structure (without any absorbers), where x represents the horizontal displacement of each mass and f is the force applied at each mass. The values of K are equal due to the fact each support has the same dimensions and material quantities. A copy of the Python Code is available in the appendix to see the parameters used and the calculations for the stiffness. This system below is a 3DOF system, so a matrix of size 3x3 is used to model the 3 individual displacements.

Figures 4-6 shows the response the angular frequencies from 0 to 130 rad/s. It is interesting to note that Floor number 1 resonates more at around 60 rad/s rather than the fundamental frequency. This suggests that it is not always appropriate to only consider the fundamental frequency. The responses were calculated using the equations of motion for each floor and representing it in matrix form:

$$MY'' + GY' + KY = F$$

$$MY'' + KY = F$$

Where M is the mass matrix, Y is the 3x1 vector representing the displacements of each floor, K is the stiffness matrix, and F is the forcing vector. G is the damping matrix, where I have assigned a coefficient such that the damping coefficient is 0.05 (light damping doesn't shift the resonant frequency from the natural frequency too much). To model an earthquake, F is a sinusoid only applied at floor 1. The natural angular frequencies are calculated to be 86.3254559, 59.73906903, and 21.32062689 rad/s. This has been achieved by noticing that equation two (neglecting damping) is a generalised eigenvalue problem. We will assume that the resonant frequency (function of damping) is equal to the natural frequency. We will next eliminate the latter term by adding a vibration absorber. We can achieve this be adding a mass in series with a spring to either one of the masses above (as shown below on figure 7).

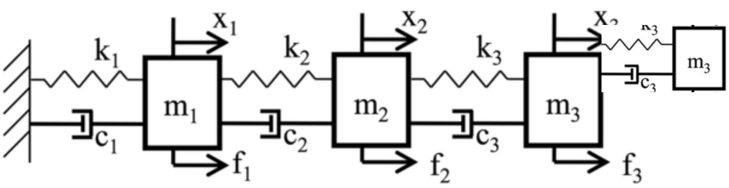


Figure 7: Adding a vibration absorber added to floor 3. The parameters of this absorber are determined so it has a natural frequency of 21.32 rad/s.

The equation remains exactly the same, although now the mass and stiffness matrix are 4x4 to account for the quantities associated with the vibration absorber. Due to damping, the resonance at 21rad/s hasn't been completely eliminated, but it is in a totally different order of magnitude of vibration. The largest peak is 10 times smaller than the resonant peak without a TMD.

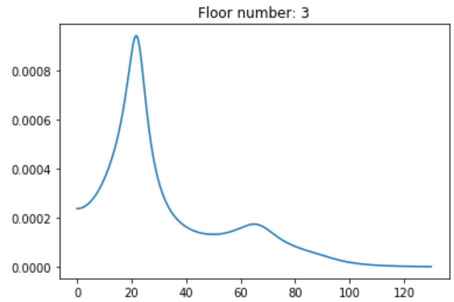


Figure 8

Figure 9 shows that adding the same absorber on floor 3 produces a similar result (4 resonant peaks), but they are all greatly reduced. The greatest overall amplitude occurs on floor 3 at a frequency of around 13 rad/s. It can be shown that in this case the vibration absorber has been more effective by positioning at floor 3. This could imply that positioning might depend on certain data (anti-nodes of the structure).

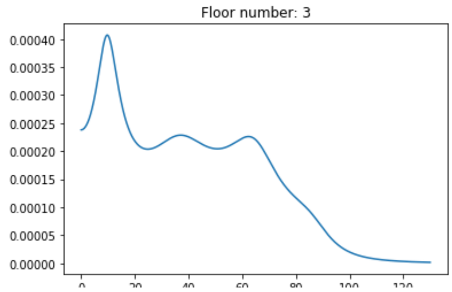
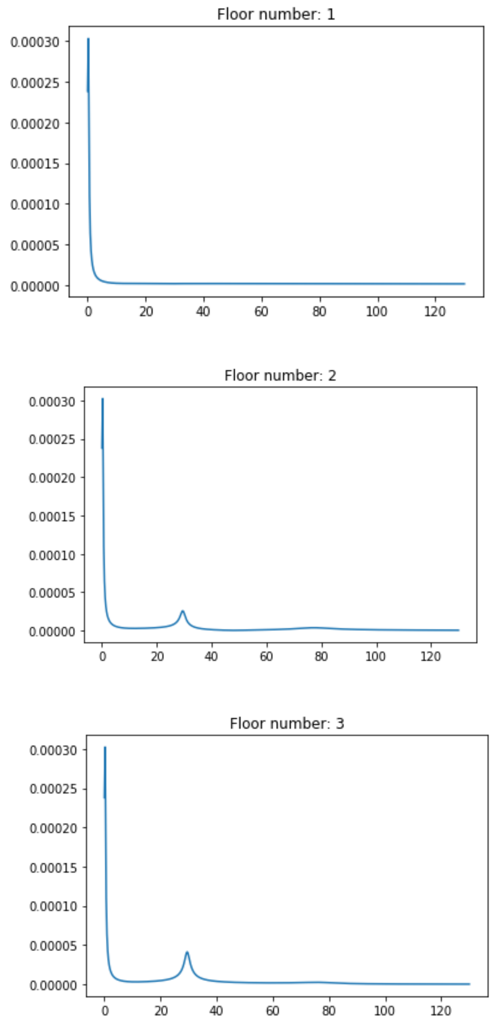


Figure 9

Adding multiple vibration absorbers

Adding more vibration absorbers also does not change the matrix equation of motion. All that changes is the size of the matrix and vectors (n dampers gives n+3 displacements to account for). As long as the matrices are calculated correctly, a frequency response due to the effect of absorbers tuned to frequencies equally spaced ranging from 1-130 rad/s can be placed on a certain floor. A separate optimisation code seems to suggest that increasing the number of absorbers and placing them on floor 1 provides the best solution. It must be noted that in our simulation, the total mass of vibration absorbers remains the same; limitations on mass and spring constants have not been taken into account on this small scale model. Figures 9-11 shows the frequency response to each floor when 100 dampers (tuned between 1 and 130 rad/s) on floor 1. Even though there should technically be 103 resonant frequencies (as our simulation calculated), a lot of them are so close together that it seems to cancel them out. There are, however, still some resonant frequencies, in this case very close to 0, which causes resonance. The amplitude of this vibration, however, is 4 times smaller than without any vibration absorbers. This suggests that the more absorbers the better. However, with 50 absorbers added on floor 1, the largest amplitude of vibration is of similar order of magnitude to this case, so it seems that the amplitude tends to level off to a certain value. From a manufacturing perspective, it is simpler to have fewer absorbers, as it involves less maintenance and less manufacturing.



Figures 10, 11, and 12: Adding 100 dampers to floor 1

Discussion of results

Our model has assumed a dashpot value of 8.77 for each connection between floors/absorbers. This is a very big assumption and the values can only be determined experimentally between each floor as well as the absorber's damping coefficient. Moreover, it is also subject to experimentation with how this value affects the system as a whole, and how it could also be altered to aid in providing a more resilient structure. With that aside, it has been shown that adding more absorbers on floor one provides the best stability for the structure. What is interesting to note is that the fundamental frequency isn't always the resonant frequency of highest amplitude. This suggests that it is not always crucial to eliminate the fundamental frequency.

There is a correlation that the highest natural frequency of the model increases as the number of absorbers increases. This can be a problem when considering the case of the impulse response. An impulse excites all the natural frequencies, such that the model will oscillate at a superposition of all the natural frequencies. At high frequencies, regardless of the amplitude being large, high cycle fatigue can be a problem. Materials will therefore need to be checked regularly to make sure fatigue does not occur within the structure.

This simulation has investigated whether the number of dampers affects the system. However, it fails to consider the cases where the dampers are distributed within the structure (20 absorbers on floor 1, 20 absorbers on floor 2, 15 absorbers on floor 3). This could be easily added to the code; however, for a total of 100 dampers there are 3 x 100 factorial possible cases to evaluate. The code also iterates through 1000 frequencies between 0 and 130. A computer will not be able to find the solution for the continuous case, hence a function of amplitude against frequency cannot be obtained albeit numerically.

Conclusion

- Adding more dampers improves the frequency response of a 3DOF building with light damping
- The effect of adding more dampers reduces each time – it 'saturates'. 50 dampers is just about as effective as adding 100 dampers.
- With higher number of dampers, it seems that floor 1 is the best floor to place the dampers
- With one damper, floor 3 provides the best vibrational absorber
- Adding more dampers adds more natural frequencies, so it makes the system more complex and unpredictable
- Adding more dampers increases the highest natural frequency. This can go into a frequency range where high cycle fatigue may need to be considered.
- Damping also needs to be investigated to see how its values can be altered to optimise the resilience of the structure.

Appendix (Python Code) – Simulating the 3DOF system with multiple tuned mass dampers

```python
import numpy as np
from matplotlib import pyplot as plt
import scipy.linalg as sp

m=1.83 #parameters of model
L=0.2
N=3
b=0.08
E=210*(10**9)
d=0.001
I=(b*d*d*d)/12
k=(24*E*I)/(L**3)
g=8.77
mab=0.3 #5% of the building's mass is the extra mass for the absorber
#parameters of the damper
while True:
    n_dampers=int(input("How many dampers do you want to add: "))
    mass_on_floor=int(input("What floor should the masses be added to: "))
    max_amp_on_floor=[]
    mab=1/n_dampers
    M= m*np.identity(n_dampers+3)
    for i in range(3,n_dampers+3):
        M[i][i]=1
    #creating the mass matrix
    K=np.zeros((n_dampers+3,n_dampers+3))
    K[0][0:3]=[2,-1,0]
    K[1][0:3]=[-1,2,-1]
    K[2][0:3]=[0,-1,1]
    K[3][0:3]=[0,0,0]
    K[mass_on_floor-1][mass_on_floor-1]+=n_dampers

    for i in range(3,n_dampers+3):
        K[i][mass_on_floor-1]-=1
    K=k*K
    for i in range(3,n_dampers+3):
        kspring = (i**2)*mab*(100/n_dampers)

        K[mass_on_floor-1][i]=-kspring
        K[i][i]=kspring

    #creates stiffness matrix
    G=np.zeros((n_dampers+3,n_dampers+3),dtype=np.complex)
    G[0][0:3]=[2j,-1j,0]
    G[1][0:3]=[-1j,2j,-1j]
    G[2][0:3]=[0,-1j,1j]
    G[3][0:3]=[0,0,0]
    G[mass_on_floor-1][mass_on_floor-1]+=1j
    G[mass_on_floor-1][mass_on_floor-1]*=n_dampers
    for i in range(3,n_dampers+3):
        G[i][mass_on_floor-1]-=1j
        G[mass_on_floor-1][i]=-1j
        G[i][i]=1j
    G=g*G
    #creates damping matrix
    vect=np.zeros((n_dampers+3,1))
    vect[0]=1
    #this is equivalent to a forcing function which provides a unit harmonic input on floor 1
    V,D=(sp.eig(K,M))
    V=np.real(np.sqrt(V)) ##
    print("Natural angular frequencies are: ",V) #natural frequencies in radians (without damping)
    for i in range(3): #iterate for each floor
        amplitude=[]
        frequencies=np.linspace(0,130,1001)
        for f in frequencies:
            B=K +(f*G)-((f**2)*M)
            B_inv=(np.linalg.inv(B))
            amplitude.append((np.absolute(B_inv.dot(vect)[i]))) #this provides the y vector for each floor
        if (abs(max(amplitude)))>(abs(min(amplitude))):   #prints the largest magnitude of amplitude for each floor
            max_amp_on_floor.append(abs(max(amplitude)))

        else:
            max_amp_on_floor.append(abs(min(amplitude)))
        plt.plot(frequencies,amplitude)
        n=str(i+1)
        plt.title("Floor number: " + n)
        plt.show()

    print("Maximum amplitude is ", max(max_amp_on_floor))
    print("This occurs on floor: ", max_amp_on_floor.index(max(max_amp_on_floor))+1)
```