California Polytechnic State University, San Luis Obispo

Department of Architectural Engineering

The Use and Explanation of Phase Angles in Forced Vibration Testing

Senior Research Thesis

Authors: Nicholas Slavin
Ryan Thornton

Advisor: Dr. Graham Archer

Date: July 07, 2020

Table of Contents

1.0 Introduction

1.1 Forced Vibration Testing

Forced vibration testing is a technique to identify and understand the dynamic behavior of an existing building. During forced vibration testing, a known mass is oscillated through a band of frequencies of interest for a particular structure. The oscillating mass is placed at a strategic location within the building, vibrated through the band of frequencies, then the response is recorded from various nodes throughout the structure. An example of a forced vibration test can be observed in Figure 1. From these tests, data that is corresponding to dynamic behavior can be gathered, including natural frequencies, mode shapes, damping ratios, and stress/strain distributions. Often, this information is collected before and after a seismic event or certain length of time to help determine how the structure has changed and if the building is damaged.

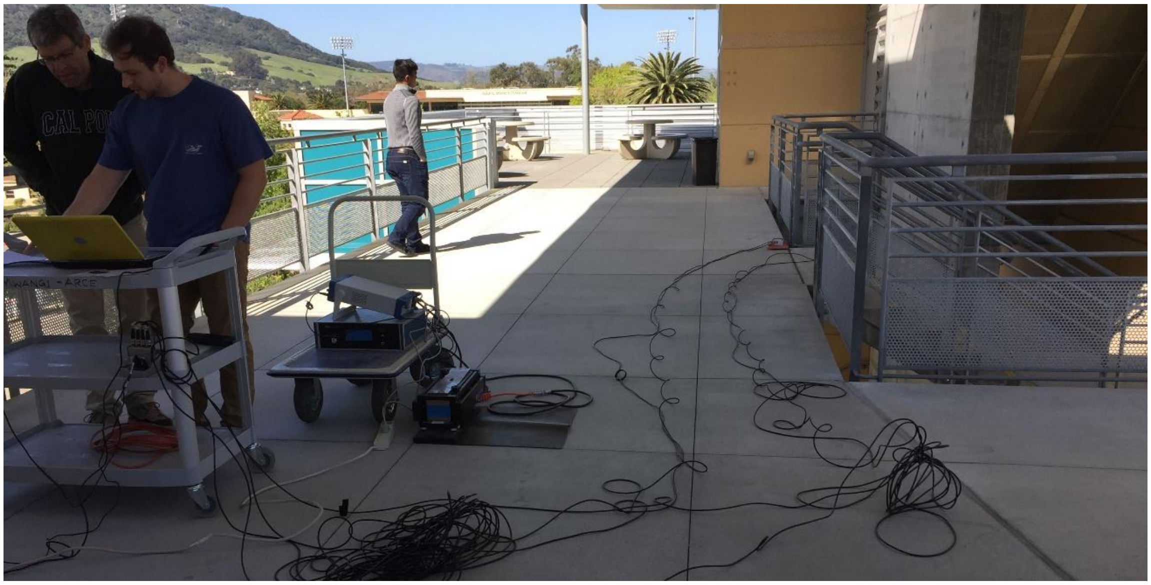


Figure 1. Forced Vibration Testing Equipment and Setup

1.2 Dynamic Response of a SDOF Oscillator

A single degree of freedom (SDOF) system that is subjected to a harmonic load will have two responses to the excitation: a transient response and a steady-state response [1]. The transient response is the free vibration due to the initial load. Internal properties such as damping will cause the transient response to decay quickly, leaving the steady-state response to continue. The steady-state response is defined as the "forced response" due to the harmonic load on the structure. Thus, the response will persist while the load persists. The steady-state displacement response is derived in the Appendix and has the general form:

$$u(t) = \frac{p_0}{k} R_d \sin(\omega t - \theta) \quad \text{(Eq. 1)}$$

The initial term in the equation p_0/k is the initial force over the stiffness of the system, defined as the *peak static displacement* (u_{st}) or the peak deformation of the structure due to the harmonic load. The second term, R_d, is the *response amplification factor*. The frequency ratio (ω/ω_n) dictates the behavior of the response amplification factor. The frequency ratio is the ratio of the frequency of the applied harmonic force (ω) to the structure's fundamental frequency (ω_n) and quantifies how close the structure is to resonance with the load being applied. If the ratio is close to zero, the response of the system will be unaffected; if the ratio is close to one, the system will find resonance and displacements will increase; and if the ratio is large ($\omega/\omega_n \gg 1$), the system will have a softened response. The last term is the *phase angle* (θ) or phase lag and defines the delay of the structure's response behind the harmonic force. The phase angle is affected by two parameters of the system: the frequency ratio (ω/ω_n) and damping ratio (ζ). The effect of these parameters on the phase angle can be observed in Figure 2 below:

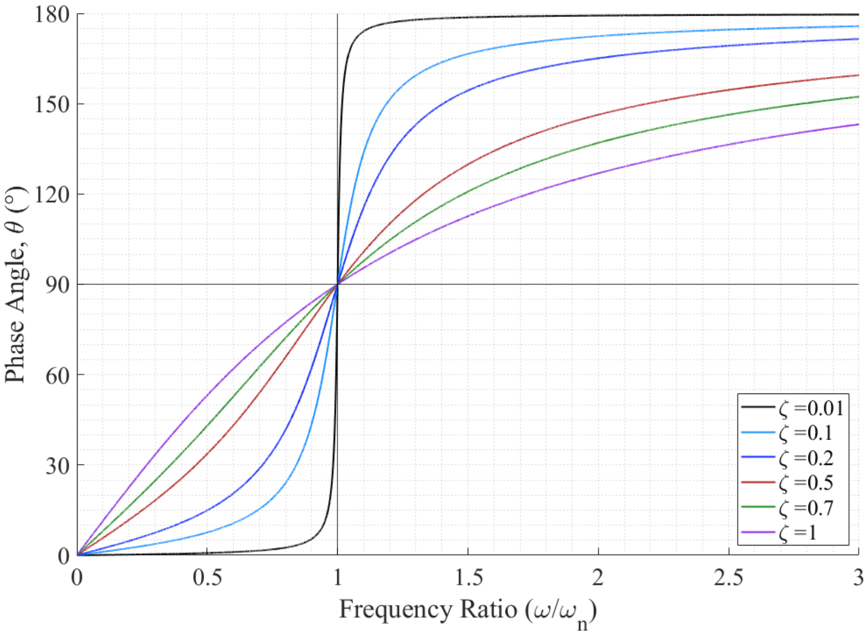


Figure 2. The Affect of Frequency Ratios and Damping on the Phase Angle

The graph above can be assessed in three distinct regions to examine the effect of the frequency ratio on the phase angle. When the frequency ratio is negligible ($\omega/\omega_n \ll 1$), the harmonic load is slowly varying, and the phase angle is nearly zero. This implies that the displacement response of the structure is in phase with the applied harmonic load. Secondly, when the frequency ratio is at resonance ($\omega/\omega_n = 1$), the phase angle is 90°, thus the displacement response reaches its maxima as the harmonic load passes through zero. Lastly, when the frequency ratio is large ($\omega/\omega_n \gg 1$), the force is rapidly varying, and the phase angle is approaching 180°. In this instance, the displacement response is opposite phase of the harmonic load, and the displacement peaks occur at the same time but in the opposite direction as the applied load peaks.

1.3 Purpose

When conducting forced vibration tests, the recorded data represents the global or "combined" response of the structure. It is difficult to isolate a single mode and capture its contribution to the global response when conducting forced vibration tests. Furthermore, it is uncertain if the response being recorded is truly a "pure" mode response, or if other modes are contributing significantly, contaminating the response. Often, when forced vibration tests are conducted, components such as the amplitude (R_d or u_{st}) of the response are analyzed in detail but the phase angle is overlooked. The purpose of this project is to investigate the phase angle and determine whether it contains information about how the overall dynamic response of the structure is affected by modal contributions. A comprehensive model was built in MATLAB to simulate a forced vibration test, and numerical and algebraic methods were used to capture the dynamic response of the structure and determine what patterns were associated with the phase angle.

2.0 Computational Methods to Determine Displacements of a MDOF System

2.1 Overview

The dynamic response for any multi-degree of freedom (MDOF) linear system with viscous damping can be reasonably captured through classical modal analysis. [1] These systems have characteristic natural frequencies and modes, and when the equations of motion defining the structure's response are transformed to modal coordinates, these equations become decoupled. Thus, the dynamic response of each individual mode can be analyzed independently to determine the characteristic mode shape, the pattern of deformation, the damping, and the natural frequency of that mode. The individual mode can be assessed as a SDOF system with the characteristic properties for that particular mode. A SDOF system can be solved with a closed form analytical solution or a numerical time stepping method when subjected to a harmonic dynamic excitation. Thus, each mode can be analyzed through these analytical or numerical methods. After the individual responses for each mode are captured, the modes can be combined to determine the total response for a given DOF.

For this investigation, instrumenting a building and conducting a real forced vibration test was not possible given the current circumstances. So, the numerical method was used to simulate forced vibration test data recorded in the field. The algebraic method was utilized to analyze the same structure and the results of the two methods were compared.

2.2 Numerical Method

The dynamic response for a classically damped linear MDOF system can be solved through modal analysis to determine the modal coordinates $q_n(t)$ due to an applied load [1]. Each modal equation has the same form as the equation of motion for a SDOF system, adapted to obtain the solutions for $q_n(t)$. Once the modal coordinates $q_n(t)$ are calculated, the modal contribution to the nodal (DOF) displacement can be determined by multiplying the mode shape ϕ_n (characteristic to the structure based on mass and stiffness) by the modal coordinate $q_n(t)$ (dependent on the applied load on the structure):

$$u_n(t) = \phi_n q_n(t) \qquad \text{(Eq. 2)}$$

Once the modal contribution $u_n(t)$ at the node is calculated, the summation of all the modal contributions can be determined to find the total displacement at that node:

$$u(t) = \sum_{n=1}^{N} u_n(t) = \sum_{n=1}^{N} \phi_n q_n(t) \qquad \text{(Eq. 3)}$$

The individual modal coordinates $q_n(t)$ and modal responses $u_n(t)$ are calculated through uncoupled modal equations, converged on through iterative solution methods, then superposed to determine the total response at the node of interest.

When modal analyses are conducted to capture the dynamic response of MDOF structures, numerical methods are most often used to combine the modal contributions at every DOF, at a moment in time t, to capture the structure's excitation to the applied load. When combining modal contributions at a given DOF, numerical methods are excellent tools in capturing the combined response. However, numerical methods are incapable of partitioning the combined response into its modal constituents. Due to this constraint, numerical methods are only applicable to modal analysis results with computational models that are attempting to characterize a structure into its mode shapes and peak responses.

2.3 Algebraic Method

Algebraic methods are mathematical principles that solve pairs or groups of linear functions. These methods can be utilized to combine individual modal sinusoidal responses due to each mode at a single DOF. Each modal response at a particular DOF in a MDOF system will have a sinusoidal response with its own characteristic amplitude and phase angle [1]:

$$u_i(t) = A_i \sin(\omega t - \theta_i) \qquad \text{(Eq. 1)}$$

Summed together, the combination of all the individual modal responses will create the global response of the system at the DOF. [2]:

$$u_{total} = \sum_{i=1}^{n} A_i \sin(\omega t - \theta_i) = A \sin(\omega t - \theta) \qquad \text{(Eq. 2)}$$

Where:

$$A^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} A_i A_j \sin(\theta_i - \theta_j) = \sum_{i=1}^{n} {A_i}^2 + 2\sum_{i=1}^{n}\sum_{j>1}^{n} A_i A_j \sin(\theta_i - \theta_j) \qquad \text{(Eq. 3)}$$

And…

$$\tan\theta = \frac{\sum_{i=1}^{n} A_i \sin\theta_i}{\sum_{i=1}^{n} A_i \cos\theta_i} \qquad \text{(Eq. 4)}$$

Unlike the numerical results which captures discrete points, the response of the algebraic method captures a linear sinusoidal harmonic response. The algebraic method is not an approximate method. It can determine the exact amplitude and phase angle that defines the harmonic response, compared to the numerical method which is only able to provide an approximate value through an iterative convergence method.

Similar to the numerical method, the algebraic method also combines the individual modal responses at every DOF. Because the algebraic method is defining the response as a harmonic equation rather than a series of discrete points, it requires a mathematical theorem to combine responses to capture the global response. The Harmonic Addition Theorem is used to combine numerous sine waves into a single harmonic wave. Utilizing the Harmonic Addition Theorem, the individual sinusoidal response for each mode can be combined to capture the global response at that DOF and compared to the recorded response from the forced vibration test.

2.4 Structure Parameters

To investigate the phase angle and modal contributions to displacement response, a MATLAB model was created that could be subjected to forced vibration testing. The MATLAB model is a three-story structure with nine degrees of freedom (DOFs) located at the center of mass of each floor plate and in the directions shown in Figure 3. DOFs 1, 4, and 7 are in the x translational direction, DOFs 2, 5, and 8 are in the y translational direction, and DOFs 3, 6, and 9 are rotational.

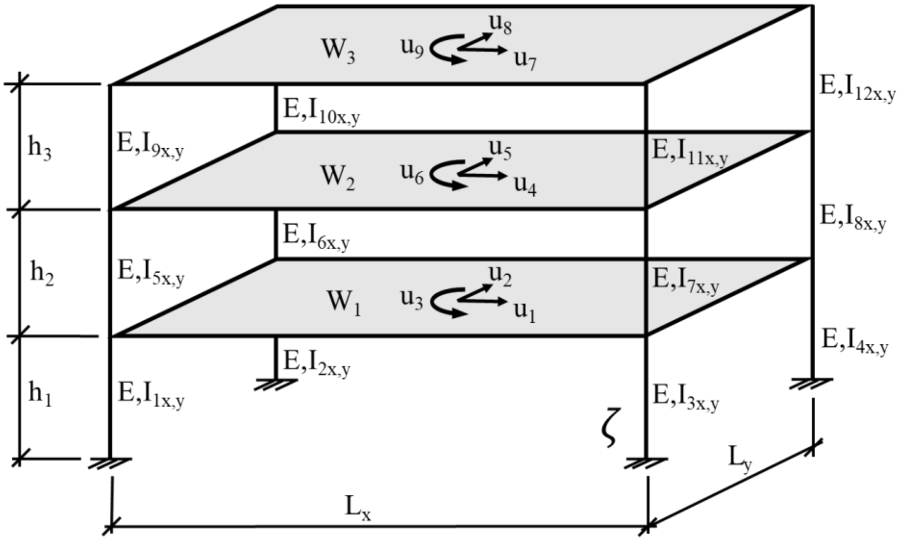


Figure 3. MATLAB model structure

The first step in the MATLAB code is to define the structural parameters. The user can specify and easily change the weight of each floor, the rectangular plan dimensions, the height of each floor, the damping ratio, the modulus of elasticity of the columns, and the moment of inertia of each column for each floor in each principle direction (Fig. 3). Each column is modeled as fixed-fixed and because the moments of inertia have so much variability within the structure, the stiffnesses can easily be made unsymmetrical. This forced irregularity in stiffness allows the user to create a configuration that more accurately represents a real building and real forced vibration testing. The specific model parameters used for the structure subjected to forced vibration testing and described in this report are provided in Table 1.

Table 1. Parameters for tested structure

Damping Ratio	**Column Modulus of Elasticity**	**Plan X Dimension**	**Plan Y Dimension**
3%	29000 ksi	200 ft	100 ft

	Floor 1	Floor 2	Floor 3
Weight (psf)	150	150	75
Height (ft)	15	12	12

	Column Moment of Inertia (in^4)					
	Floor 1		Floor 2		Floor 3	
	X Direction	Y Direction	X Direction	Y Direction	X Direction	Y Direction
SW Column	20800	960	800	800	800	480
NW Column	5600	7200	4400	5200	560	1360
NE Column	17600	19200	800	4000	960	640
SE Column	12000	4000	2800	6000	240	480

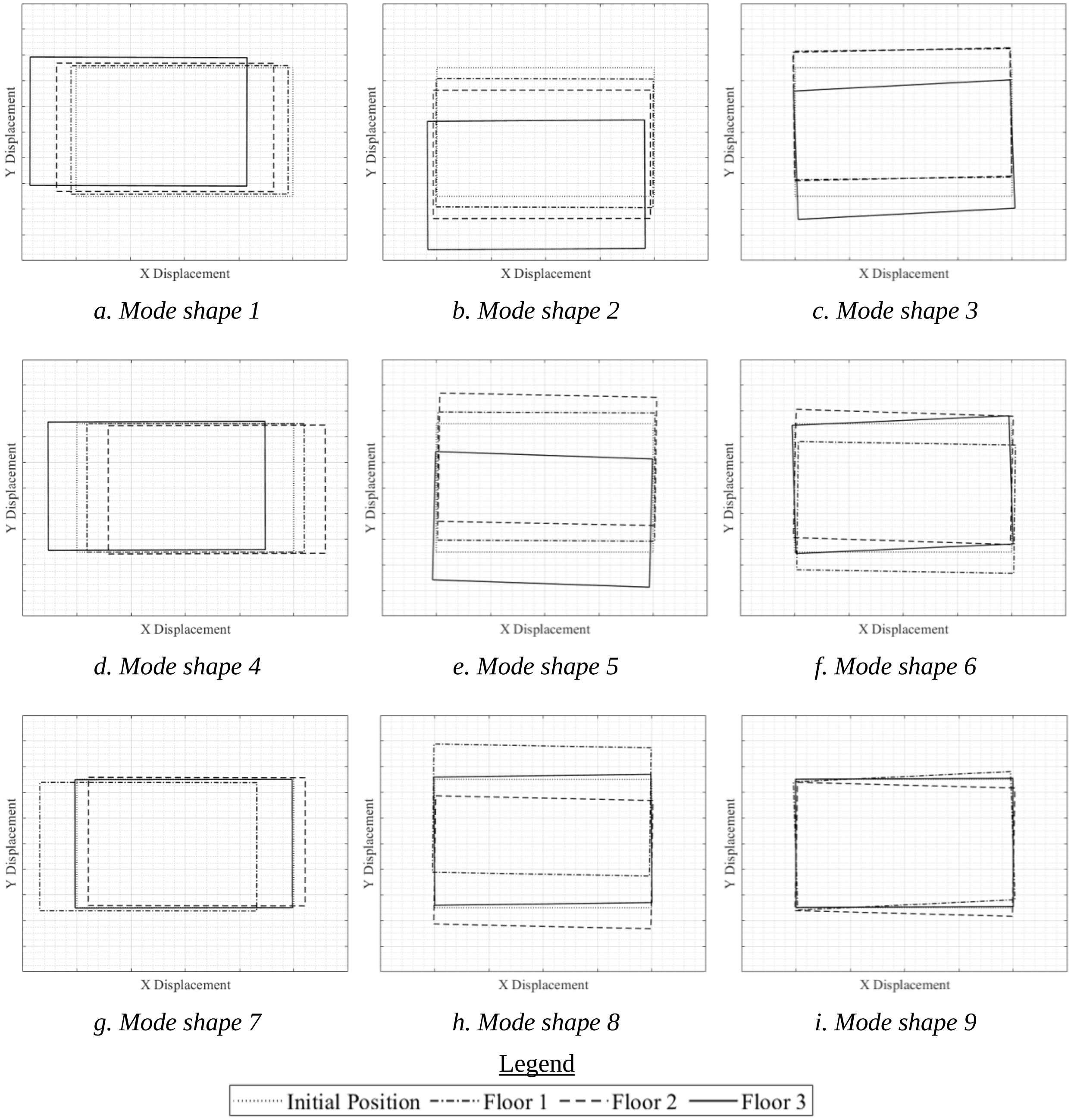
2.5 Mode Shapes

After inputting structural parameters, the MATLAB code calculates a stiffness matrix and mass matrix for the structure. The "eig" function is then used to determine the mode shape and frequency matrices. The natural frequencies and periods of the tested structure are provided in Table 2 and the mode shapes are provided in Figure 4. Within the figure, each plot illustrates a different mode shape, where each line type defines a floor and the horizontal motion of each floor is shown in plan view. The axes are labeled to show displacements in the x and y directions, but mode shape displacements are relative to each other and otherwise arbitrary, so the axes are not numbered.

The mode shapes appear to be influenced by the unsymmetrical stiffness built into the structure. A completely symmetrical structure might show pure translation in the x direction in the first mode, pure translation in the y direction in the second mode, and pure rotation in the third mode. However, Figure 4a shows that the floors translate along a diagonal for the first mode, Figure 4b shows diagonal motion with slight rotation for the second mode, and Figure 4c shows rotation and y translation for the third mode. Higher order modes similarly incorporate multiple components of motion. In real forced vibration testing, the experimenter would aim to isolate specific modes to obtain relative displacements and characteristics for the mode being tested. For the testing program described herein, modes 1, 3, 4, and 8 were selected for a total of four forced vibration tests run using the MATLAB code.

Table 2. Fundamental frequencies and periods for each mode of the structure

	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7	Mode 8	Mode 9
Period (s)	0.94	0.91	0.56	0.50	0.47	0.30	0.26	0.22	0.14
Frequency (rad/s)	6.7	6.9	11.3	12.7	13.5	21.0	24.6	28.0	45.1



a. Mode shape 1 *b. Mode shape 2* *c. Mode shape 3*

d. Mode shape 4 *e. Mode shape 5* *f. Mode shape 6*

g. Mode shape 7 *h. Mode shape 8* *i. Mode shape 9*

Figure 4. Plan view of each floor's motion for each mode shape of the MATLAB model structure

2.6 Harmonic Input Force

To simulate forced vibration testing, a virtual mass shaker characterized by the forcing function shown in Equation (8), was set to different frequencies and placed in different locations and orientations on the MATLAB model for each test.

$$p(t) = p_0 \sin(\omega t) \qquad \text{(Eq. 8)}$$

In this harmonic forcing function, p_0 is the amplitude of the sinusoidal force (kip), and ω is the frequency of the sinusoidal force (rad/s). Within the MATLAB code, a time vector is set where the duration and time step can be adjusted. For the testing program described in this report, a force amplitude of 100 kips was used. This value would be uncharacteristically large for a real forced vibration test but enabled larger displacements in the results.

To isolate and excite specific modes for each test, the forcing frequency and location of the mass shaker were varied. The forcing frequency was set to the natural frequency of the mode being tested. Because the structure is unsymmetrical, it was difficult to place the shaker in a way that would only excite a single mode. Therefore, results may show contributions from other modes affecting the response, which would likely be the case in a forced vibration test of a real building. Figure 5 shows the location, orientation, and forcing frequency of the shaker for each test. For each test, a single shaker was placed in the location indicated in the figure. For example, for the test in which mode 1 was isolated, the shaker was set to a frequency of 6.7 Hz, placed in the middle of the third floor, and rotated 161° from horizontal. The location and orientation of the shaker aim to excite the corresponding mode shapes shown in Figure 4.

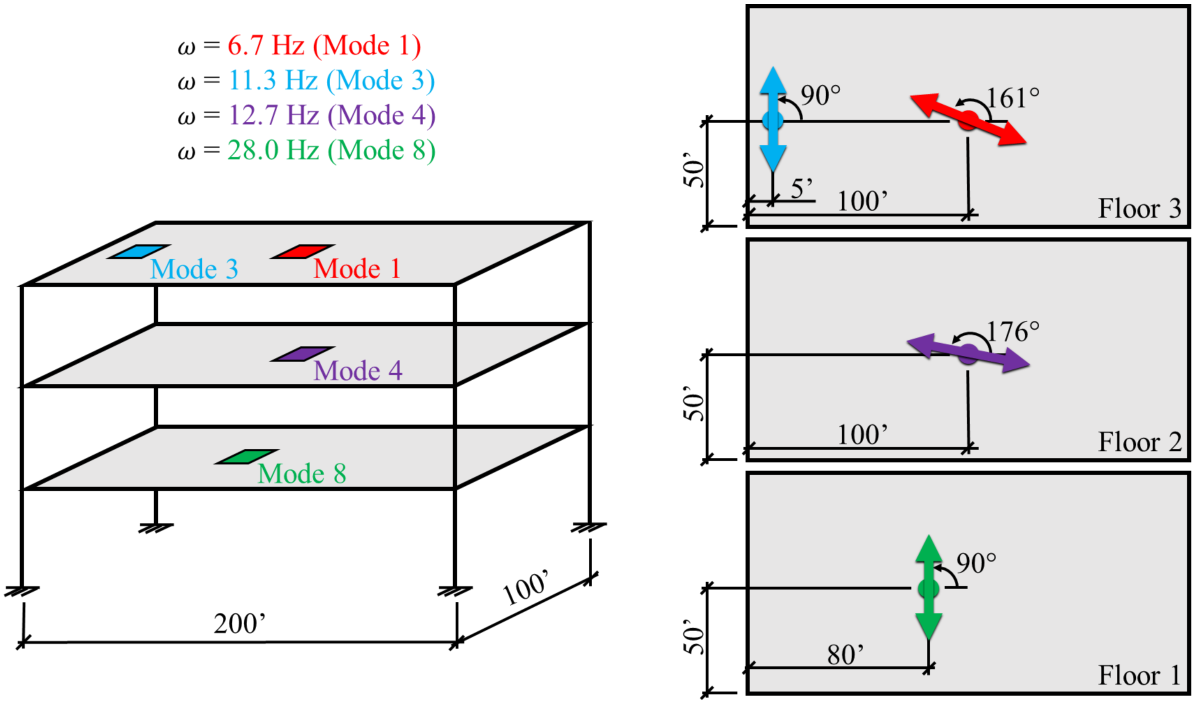


Figure 5. Location and frequency of mass shaker on MATLAB model for each test

2.7 MATLAB Process and Capabilities

After inputting the structure and force parameters, the MATLAB code is set to calculate the mode shapes, frequencies, and periods for each mode before running a forced vibration test using both the numerical and algebraic methods. To run a forced vibration test, the applied harmonic load is first translated into its components at each DOF. Then, the forces are converted from the DOFs into their modal coordinates and terms in the steady state solution are calculated for each mode (Eq. 1). The modal displacement responses are enumerated from the steady state equation and converted back to real displacements at each DOF. The numerical method is used to sum up the responses at each DOF to determine a global combined response, which represents data collected from a real forced vibration test. The algebraic method and Harmonic

Addition Theorem equations are used as an alternate way to determine the overall displacement response at each DOF. To do so, specific terms in the steady state solution are converted from modal coordinates to constituents at each DOF and the terms are combined using the algebraic method to mathematically solve for overall amplitudes and phase angles. After these parameters are found, the displacement response is calculated.

Within the MATLAB code, it is also possible for the user to specify the location on the structure for which they want to obtain displacement results. The displacements due to the harmonic force are automatically calculated at each DOF, but an additional recording location can be specified by the user. Because the forced vibration testing is done in MATLAB rather than real testing, the code can determine the responses contributed by each mode to displacement at each DOF and at the selected location. In real forced vibration testing, the experimenter would only be able to collect a string of displacement data representing the combined contributions from all the modes. The MATLAB code, however, can break down the overall displacement into its modal constituents, allowing the user to determine if secondary modes are being activated and contaminating the response when a single mode is trying to be isolated.

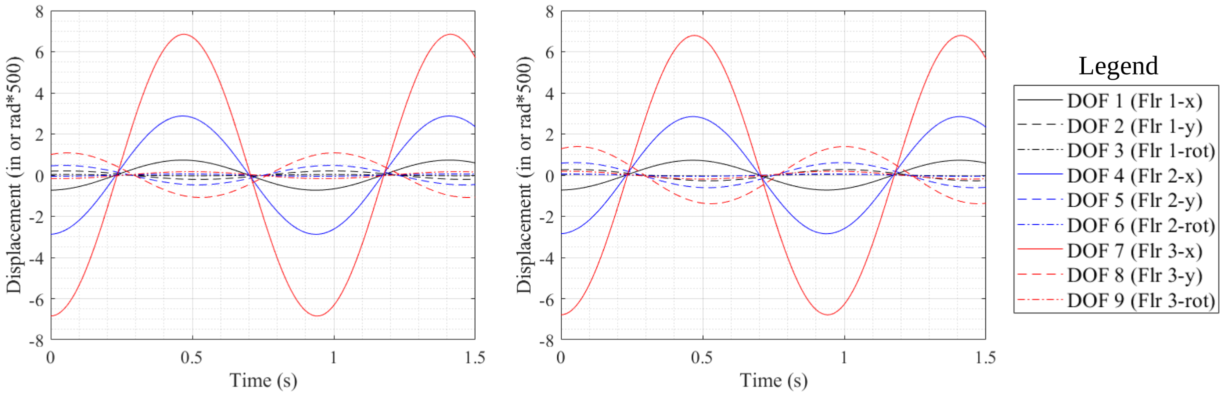
The entire MATLAB script used for forced vibration testing of a three-story structure is provided in the Appendix of this report.

3.0 Results

The MATLAB code was run four times to simulate different forced vibration tests. For each test, a specific mode was isolated by altering the frequency, location, and direction of the applied harmonic force. The goal of each test was to isolate a mode to obtain responses for that mode. The results compiled from each test describe how successful the test was at isolating the mode at several locations on the structure. Unanticipated contributions from other modes are also shown and analyzed for each test. Results are provided for four tests, in which modes 1, 3, 4, and 8 were isolated. Detailed test descriptions based on response observations are provided for each test, and an overall quantitative summary is provided at the end of this results section.

3.1 Mode 1 Excitation Test

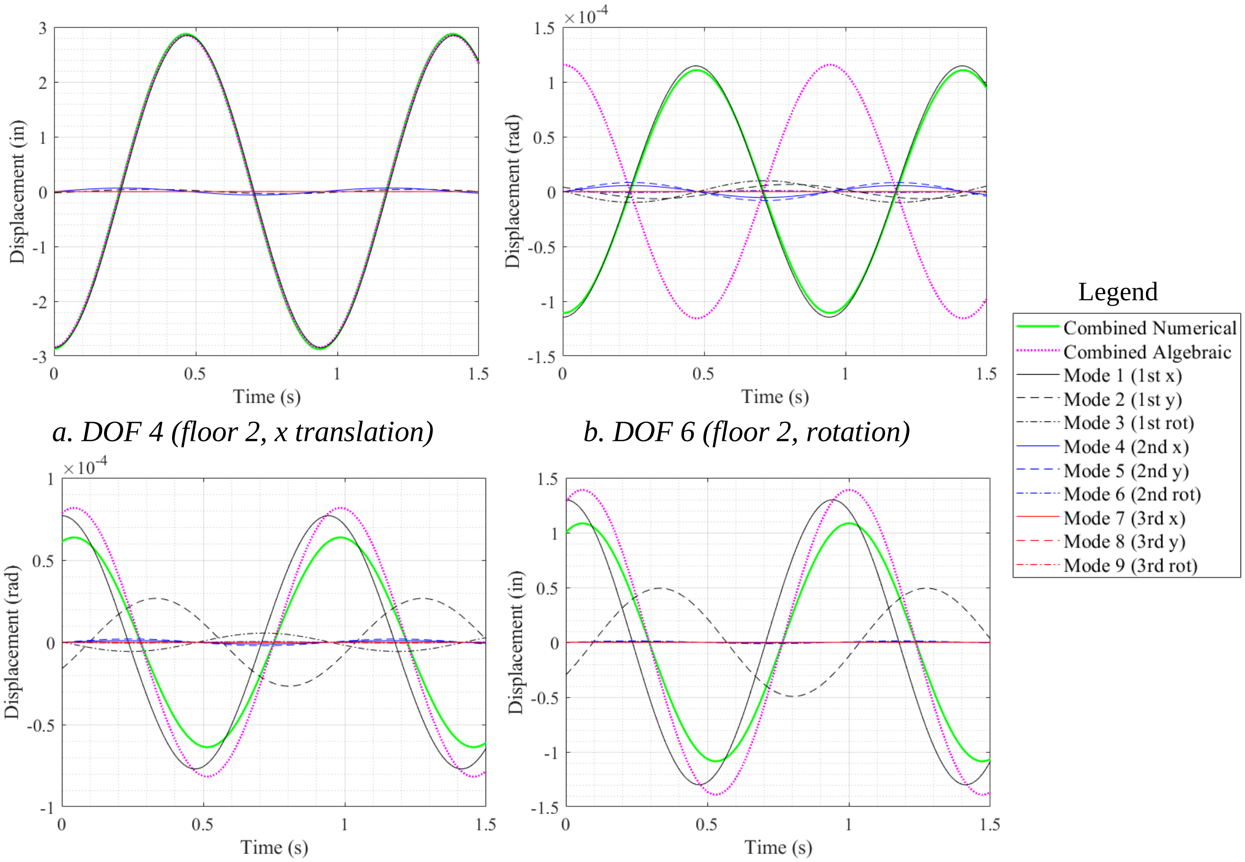
Mode 1 displacements due to the applied harmonic force were determined using the numerical method (Fig. 6a) and the algebraic method (Fig 6b). Each sine wave presents the response at a specific DOF in the structure. Because DOFs 1, 2, 4, 5, 7, and 8 are translational in the units of inches while DOFs 3, 6, and 9 are rotational in the units of radians, it is difficult to compare displacement magnitudes between the two sets of DOFs. The rotational DOF displacements are scaled by a factor of 500 and illustrated on the same plots as the translational DOFs. During the mode 1 test, rotation is minimized so DOF 3, 6, and 9 values remain close to zero. However, in subsequent tests, the rotational response is much larger and the scale factor of 500 captures the response while maintaining clarity with the translational DOFs.



a. Numerical method *b. Algebraic method*

Figure 6. Mode 1 excitation test – combined displacement response at each DOF

Figure 6 indicates that the mode 1 forced vibration test is exciting displacement primarily in the x translational direction (solid line type), with greater displacements at the third floor (red line color). Upon cursory inspection, it seems like the numerical and algebraic methods have produced the same displacement results. However, distinct differences become apparent when isolating specific DOF responses. For example, it is difficult to see in Figure 6 because the rotational responses are close to zero, but the DOF 6 and DOF 9 algebraic method responses are approximately 180° out of phase from the corresponding numerical method results. Additionally, displacement amplitudes at DOFs 2, 3, 5, and 8 are slightly greater for the algebraic method than for the numerical method. To investigate these discrepancies, the response was broken down at each DOF into its modal constituents. Because the harmonic force was applied in a way that aimed to isolate mode 1, the expected modal content is that most of the response would be contained in mode 1. Breaking up the response into its modal constituents, as shown in Figure 7 for DOFs 3, 4, 6, and 8, illustrates how successful the test was at isolating mode 1 and how other modes may have contaminated the response.



a. DOF 4 (floor 2, x translation)

b. DOF 6 (floor 2, rotation)

c. DOF 3 (floor 1, rotation)

d. DOF 8 (floor 3, y translation)

Figure 7. Mode 1 excitation test – modal contributions to displacement response at various DOFs

Different locations in the structure exhibited a variety of responses. Figure 7a shows a typical "good" result, where mode 1 is successfully isolated and the combined numerical and algebraic methods agree. The solid green combined numerical line, dotted pink combined algebraic line, and solid black mode 1 line all lie on top of each other in this figure. All the other modal constituents are approximately zero. The combined numerical line represents the summation of all the modal contributions and imitate the results that would be obtained from a real forced vibration test. The combined algebraic line is also a combination of all the modes to provide the real total displacements, but it uses the Harmonic Addition Theorem and theoretical equations to combine sine waves. The Figure 7a response shows successful isolation of mode 1 and is apparent at DOFs 1, 4, and 7.

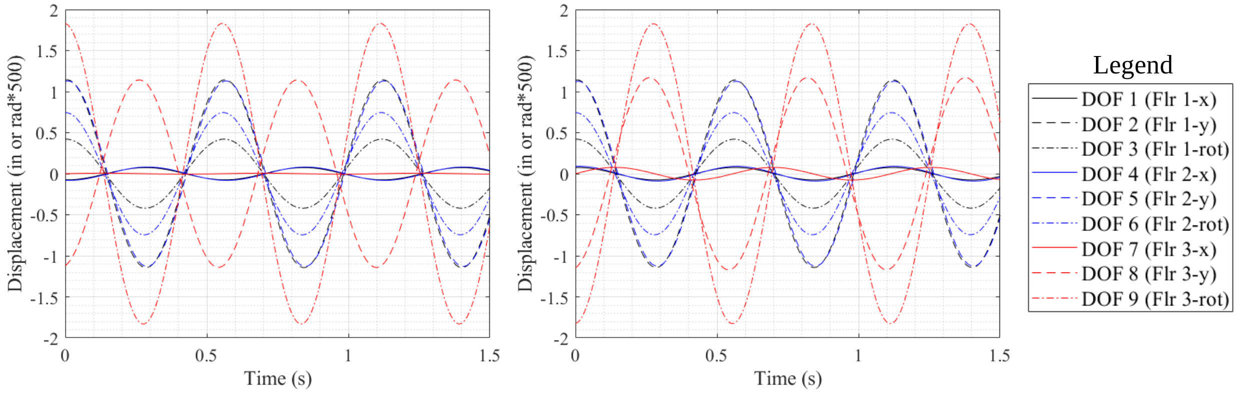
In Figure 7b, the algebraic response is flipped 180° from the numerical response. The numerical method is correctly summing up the modal contributions because it aligns closely to the mode 1 response but is lowered slightly by interference from other modes. This error in the algebraic response is apparent in DOFs 6 and 9 for the mode 1 excitation test. To investigate the "flipping" error apparent in Figure 7b, the research team looked for patterns in the algebraic method and the Harmonic Addition Theorem. It was found that whenever the summation of the cosine terms within Equation 7 was negative, the flipping error would occur. This error can be attributed to taking the arctangent of the overall term. The cause of the error is likely because the arctangent function is undefined at certain locations, resulting in the flipped phase angle. An attempted fix to the problem involved adjusting the theorem to force the cosine term to be positive.

This modification corrected the flipping error but resulted in a different phase angle shift. It was decided to note when the flipping error occurs but maintain the original algebraic method and Harmonic Addition Theorem rather than make alterations for all subsequent results.

To investigate the amplitude differences between the numerical and algebraic methods that were apparent in several Figure 6 responses, the modal constituents were broken down and are shown in Figure 7c and d for DOFs 3 and 8. Again, the numerical method appears to be accurately summing up the modal contributions while the algebraic method is in error. In the case of DOFs 2, 3, 5, and 8, mode 1 has not been successfully isolated; the mode 2 response is unintentionally activated and is contaminating mode 1 results. The combined numerical displacements accurately capture the slight destructive interference while the algebraic method fails to capture the cancellation effects. Both methods, however, show a phase shift that occurs due to mode 2 contamination. For all DOFs that have a pure mode 1 response, the phase angle for the combined numerical and algebraic responses is generally close to 90° or -90°. This is apparent in Figure 7a, where DOF 4 has a phase angle of 88.1° and Figure 7b, where DOF 6 has a phase angle of 88.7°. When modal contamination occurs, the phase angle is not close to 90° or -90°. For example, in Figure 7c, the DOF 3 response has a phase angle of -73.5°, and in Figure 7d, the DOF 8 response has a phase angle of -67.4°. This result is significant because phase angles substantially different from 90° or - 90° may indicate that the response is being contaminated and combined displacements may not be representative of the pure mode 1 response that the forced vibration test is trying to isolate.

3.2 Mode 3 Excitation Test

After conducting the mode 1 excitation test, the harmonic force was applied to isolate mode 3 in a separate test. The results were analyzed to determine if similar flipping and amplitude discrepancies occurred between the numerical and algebraic methods and if phase angle differences could predict modal contamination. The mode 3 test results are summarized in Figure 8, which illustrates the combined displacements at different locations on the structure. Movement primarily occurs rotationally (dash-dot line type) and in the y direction (dashed line type), with slightly higher displacements on the third floor. Like the mode 1 test, numerical and algebraic results are very similar. However, the flipping error appears to occur at DOFs 1, 4, and 9 and the amplitude discrepancy occurs at DOF 7.

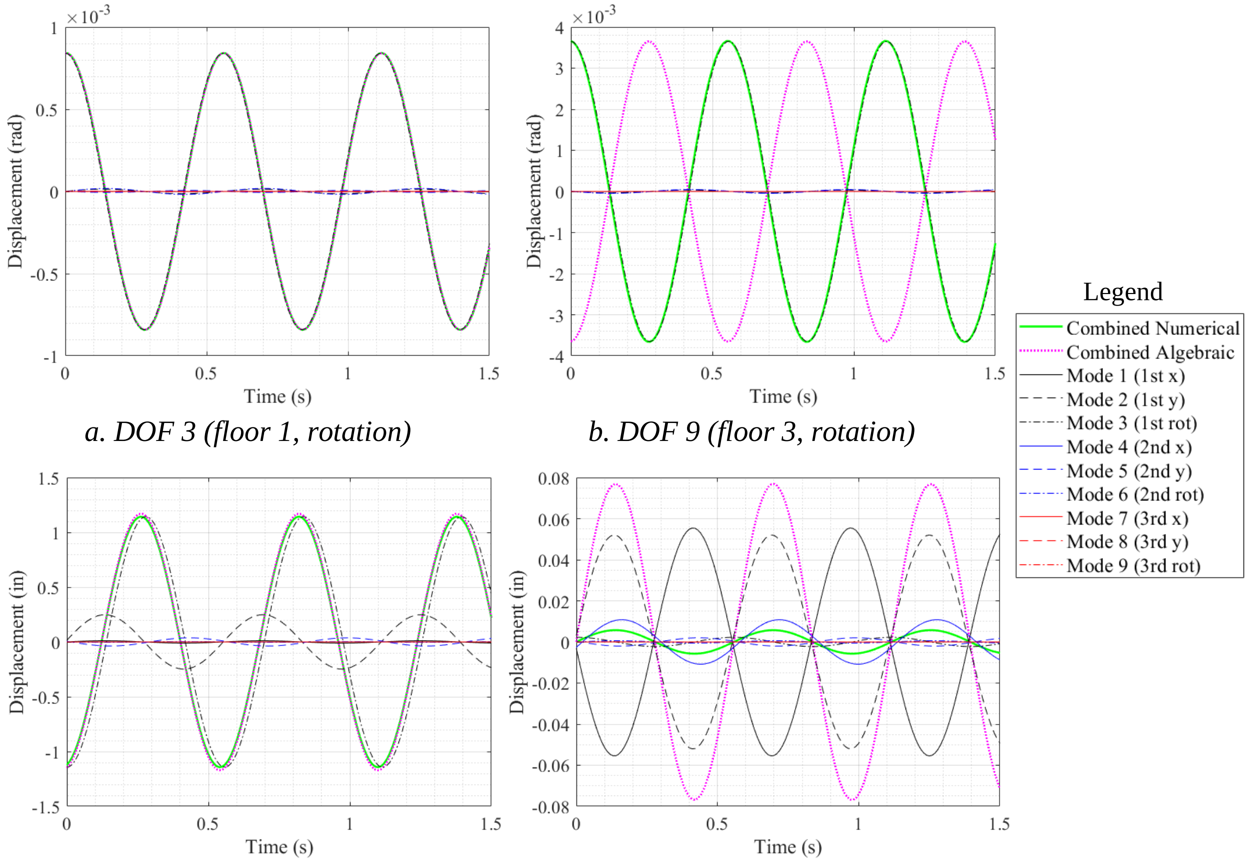


a. Numerical method *b. Algebraic method*

Figure 8. Mode 3 excitation test – combined displacement response at each DOF

To investigate the differences between the numerical and algebraic results and determine if a pure mode 3 response was achieved, the sine waves were broken down into their modal constituents for each DOF. At DOFs 2, 3, 5, and 6, a pure mode 3 response was achieved, all other modal contributions were minimized, and the algebraic method matched the numerical method. Figure 9a shows this response for DOF 3, where the combined numerical and algebraic lines lie on top of the mode 3 dash-dot black line and correctly sum up the modal contributions. Figure 9b shows the flipping error, which occurs at DOFs 1, 4, and 9. This error in the algebraic method is again attributed to the undefined arctangent step in the Harmonic Addition Theorem.

At DOFs 7 and 8, the mode 3 response is being slightly contaminated by other modes. Figure 9c shows that for DOF 8, the response is being only slightly altered by mode 2, which shifts the sine wave to a phase angle of 79.0°. Figure 9d illustrates a more dramatic alteration, where the overall DOF 7 response is nearly cancelled out to zero by modes 1 and 2. The combined numerical response captures this cancellation affect while the combined algebraic response fails to correctly sum up the modal contributions, resulting in an amplitude error. The combined numerical results for DOF 7 indicate an issue in forced vibration testing because the numerical values represent the real values that would be obtained when collecting physical test data. Although attempting to isolate the mode 3 response, the experiment would only show combined data, thus neglecting the significant contributions that are coming from modes 1 and 2 and inaccurately presenting mode 3 results. The DOF 7 results do, however, have a phase angle of -0.1°. This phase angle, which is significantly different from -90° or 90°, could indicate that modal contamination is affecting results and the pure mode 3 response is not being successfully isolated.



a. DOF 3 (floor 1, rotation)

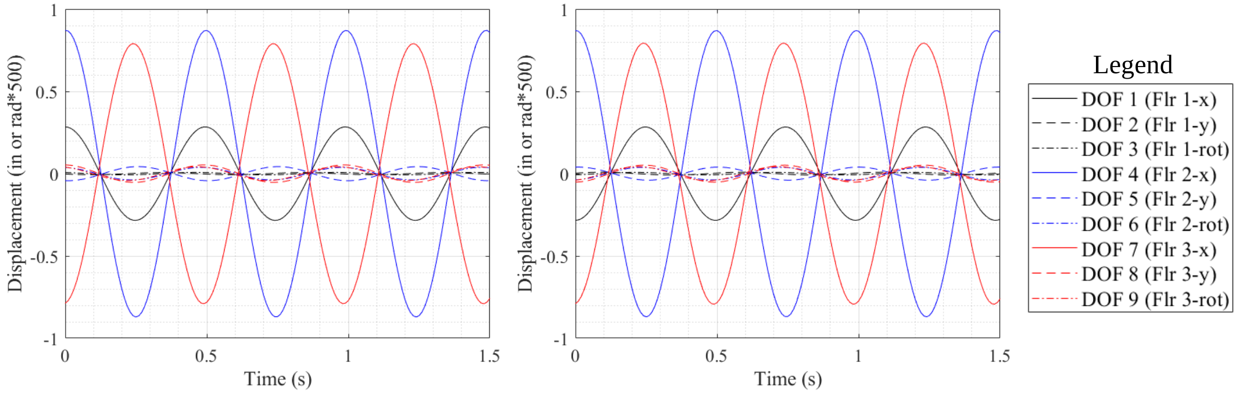
b. DOF 9 (floor 3, rotation)

c. DOF 8 (floor 3, y translation)

d. DOF 7 (floor 3, x translation)

Figure 9. Mode 3 excitation test – modal contributions to displacement response at various DOFs

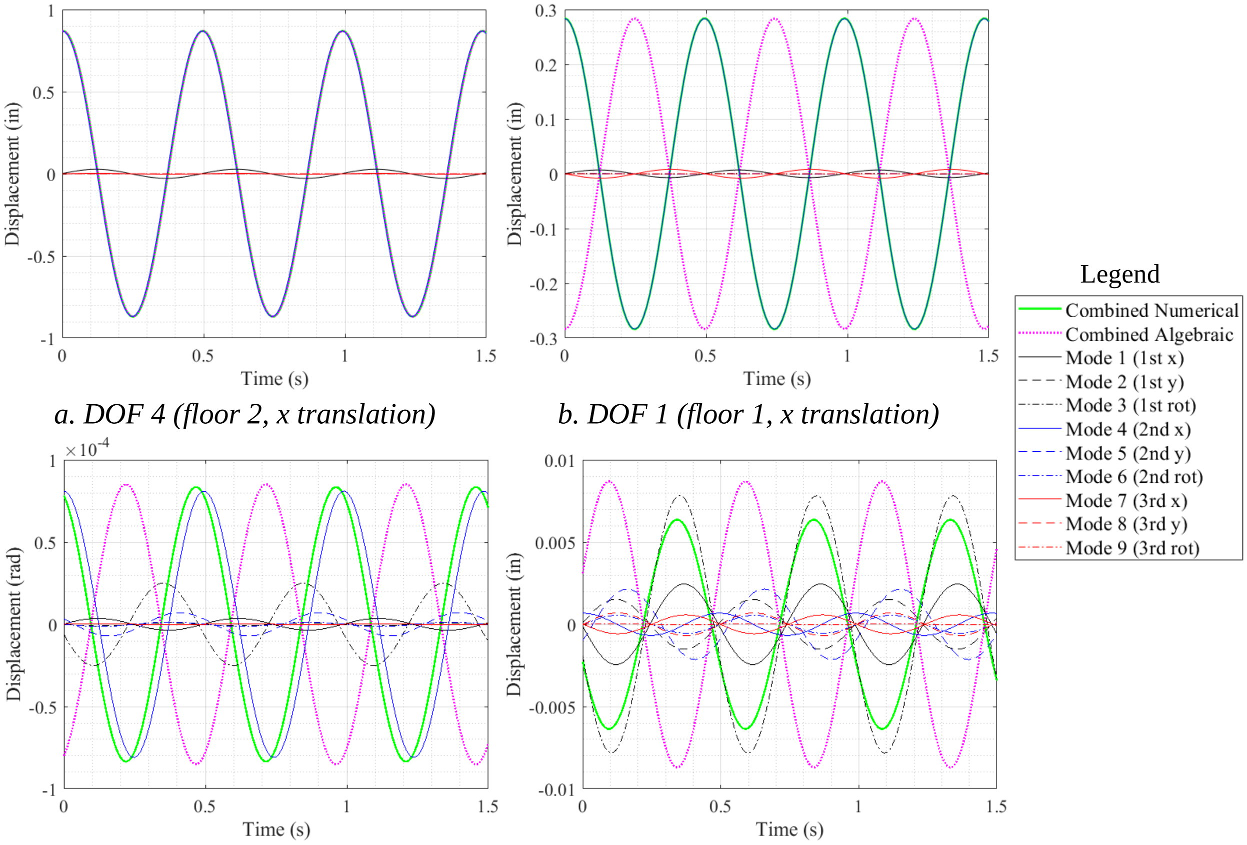
3.3 Mode 4 Excitation Test



a. Numerical method *b. Algebraic method*

Figure 10. Mode 1 excitation test – combined displacement response at each DOF

After the modes 1 and 3 excitation tests, the shaker was placed in a manner that aimed to isolate mode 4 and another test was conducted. Figure 10a shows the combined results using the numerical method and Figure 10b shows the combined results using the algebraic method. Displacement occurs primarily in the x direction (solid line type) and in floors two and three (blue and red line colors). Of the four tests conducted, the mode 4 test had the most occurrences of the flipping error in the algebraic method; it was found at DOFs 1, 2, 3, 5, 6, 8, and 9. The amplitude discrepancy only occurred at DOF 2.



a. DOF 4 (floor 2, x translation) *b. DOF 1 (floor 1, x translation)*

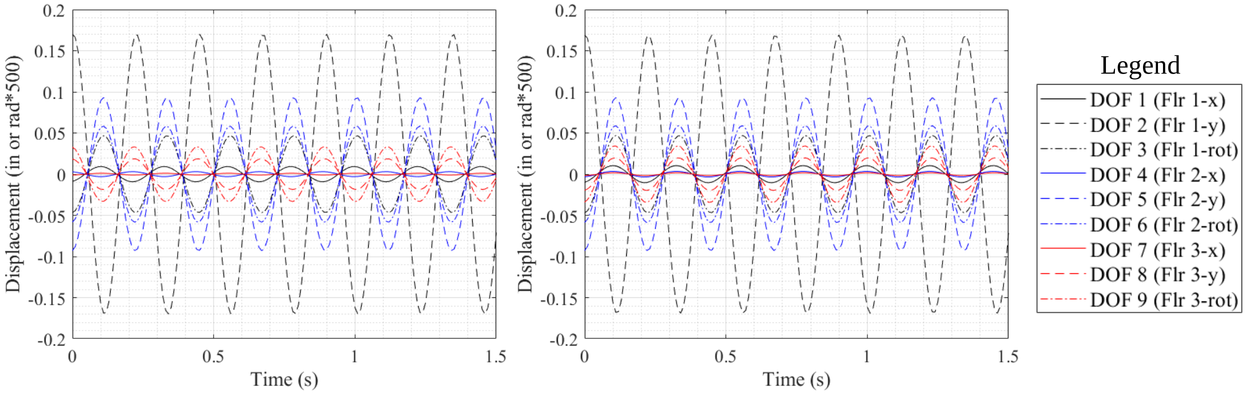
c. DOF 9 (floor 3, rotation) *d. DOF 2 (floor 1, y translation)*

Figure 11. Mode 4 excitation test – modal contributions to displacement response at various DOFs

The responses at each DOF were broken down into their modal constituents and are provided in Figure 11. Pure mode 4 responses where algebraic results matched numerical results occurred at DOFs 4 and 7, with an example provided in Figure 11a. DOFs 1, 5, 6, and 8 each have the algebraic flipping error but lack significant modal contamination. Figure 11c shows that isolation of mode 4 is nearly successful at DOF 9, but contribution from mode 3 draws the combined numerical response to the left, resulting in a phase angle of -69.9°. Similar minor mode 3 contamination occurs at DOF 3. Figure 11d illustrates that the test failed to isolate mode 4 at DOF 2, and the response was primarily characterized by mode 3 contribution. Many modes affect the displacements at DOF 2, and the phase angle of the combined response is 20.8°. Because locations with slight or no modal contamination have phase angles close to -90° or 90°, the 20.8° phase angle at DOF 2 reinforces the hypothesis that modal contamination can be indicated by phase angles that are not close to -90° or 90°.

3.4 Mode 8 Excitation Test

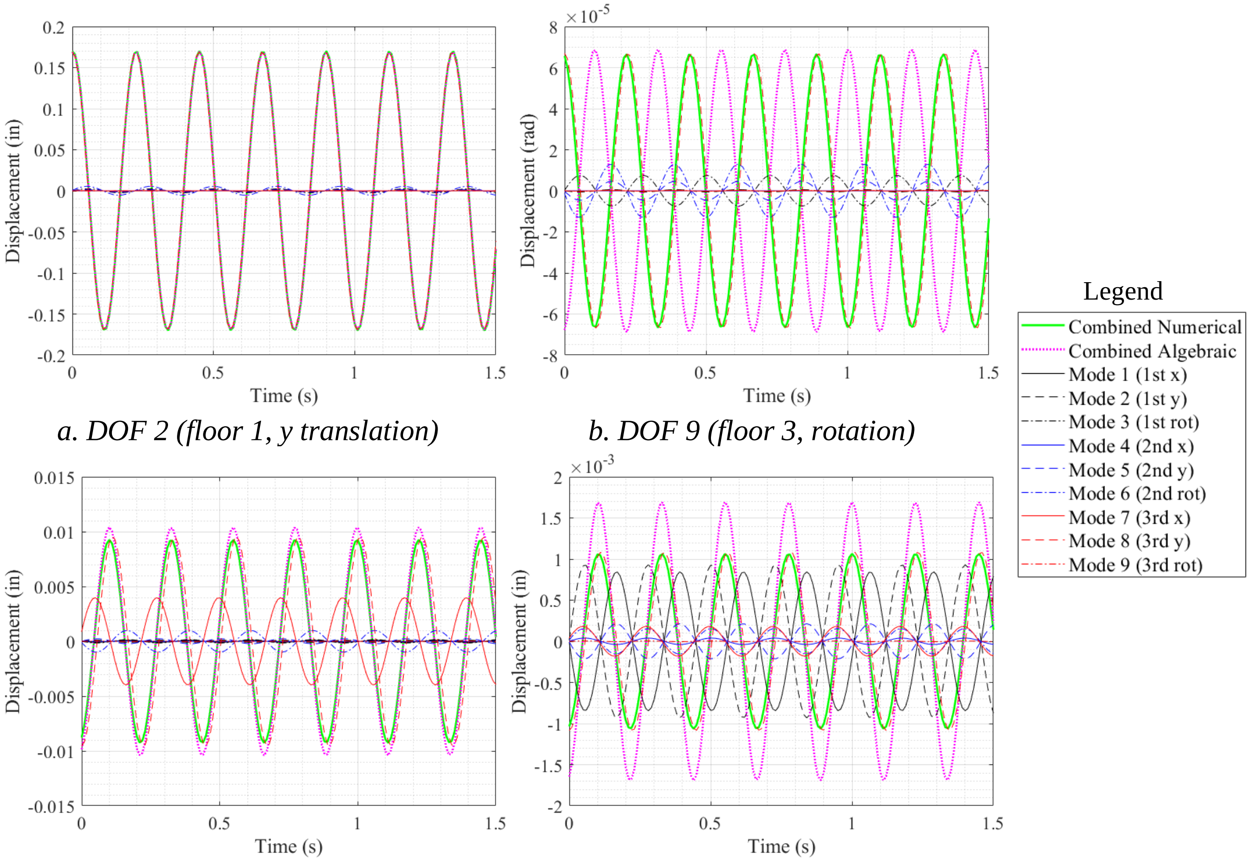
Mode 8 was isolated and tested to determine if results varied for higher modes. The overall results for each DOF are provided in Figure 12 for the numerical and algebraic methods. During this test, greater movement occurred in the y translational direction (dashed line type) and at the first floor (black line color). The arctangent flipping error occurs for DOFs 4, 8, and 9 and the largest discrepancy in amplitude between the two methods occurs at DOF 7.



a. Numerical method *b. Algebraic method*

Figure 12. Mode 1 excitation test – combined displacement response at each DOF

Figure 13 displays several DOF responses partitioned into their modal constituents. At DOFs 2, 3, 5, and 6, mode 8 was successfully isolated, other modes were minimized, and the combined numerical and algebraic responses matched, correctly summing up the modal contributions as shown in Figure 13a for DOF 2. The flipping error occurred with the algebraic method but mode 8 was successfully isolated at DOFs 8 and 9 (Fig. 13b). Figure 13c shows that at DOFs 1 and 4, the numerical response was contaminated by mode 7, resulting in slightly reduced amplitudes and phase angles of 71.2° at both DOFs. Modal contamination occurs most significantly at DOF 7 (Fig. 13d). Here, the combined numerical results end up being close to the mode 8 sine wave by chance because large contributions in modes 1 and 2 cancel each other out. The algebraic method amplitude fails to capture this cancellation, but the numerical method accurately captures it, resulting in an amplitude close to the mode 8 response. The phase angle of the combined response is 76.4°, which is only somewhat distanced from the 90° pure, isolated response. Many modes contaminate the response, but because the overall displacements are close to the mode 8 values, the phase angle remains only mildly separated from 90° or -90°.



a. DOF 2 (floor 1, y translation) *b. DOF 9 (floor 3, rotation)*

c. DOF 1 (floor 1, x translation) *d. DOF 7 (floor 3, x translation)*

Figure 13. Mode 8 excitation test – modal contributions to displacement response at various DOFs

3.5 Results Summary

A brief account of the results from all four tests are provided in Table 3. For each test, at each DOF, the purple squares represent when there is a significant difference between the amplitude of the mode being isolated and the amplitude of the combined numerical response, which represents the data collected from a real forced vibration test. The green squares represent when modes other than the one being isolated are unintentionally being excited and have a significant amplitude, contaminating the overall response. Green and purple squares therefore both describe when a pure isolated response is not being achieved for the mode being tested. Threshold values are described in the Table 3 legend.

Table 4 tabulates phase angle values of the combined numerical response at each DOF for each test. Phase angles shown in darker shades of blue are values further from -90° or 90°. It was found that at DOFs with phase angles further from -90° or 90°, modal contamination was generally greater. This result is significant because in a real test, only the combined numerical response would be obtained, and it wouldn't be possible to break the response down into its modal constituents. A difference in amplitude would not be apparent, but the phase angle would indicate that a pure response is not being achieved.

Modal contamination often occurred because of lower-order modes and modes of similar natural frequencies to the one being tested. This pattern likely occurs because lower natural frequencies will have lower stiffness values, thus increasing displacement in the steady state solution (Eq. 1), and a higher frequency ratio will increase the response amplification factor, thus increasing displacement (Eq. 1).

Additionally, modal contamination occurred more often at DOFs that have lower displacement values and are not being highly activated during the forced vibration test. The mode shape for the mode being tested does not show significant movement at these DOFs, so displacement becomes dominated by secondary modes during testing.

The flipping error in the combined algebraic response is also indicated in Table 3 with orange squares. The error occurs due to the arctangent function being undefined in the Harmonic Addition Theorem, and it appears to have no connection to specific locations in the structure or other model parameters.

Table 3. Characteristics of response at each DOF for each test

Test	DOF 1	DOF 2	DOF 3	DOF 4	DOF 5	DOF 6	DOF 7	DOF 8	DOF 9
Mode 1		🟪🟩	🟪🟩		🟪🟩	🟧		🩷🟩	🩷🟧
Mode 3	🟧			🟩🟧			🟪🟢		🟧
Mode 4	🟧	🟪🟩🟧	🟧		🩷🟧	🟧		🟧	🟧
Mode 8	🟩			🟩🟧			🟢	🟩🟧	🟧

Table X legend

🩷 **Discrepancy between tested mode and combined numerical response:**
The amplitude of the mode being tested has a >10% error from being equal to the amplitude of the combined numerical response.

🟪 **Large discrepancy between tested mode and combined numerical response:**
The amplitude of the mode being tested has a >20% error from being equal to the amplitude of the combined numerical response.

🟩 **Modal contamination:**
A mode other than the one being tested has an amplitude >40% of the amplitude of the combined numerical response; or two or more modes other than the one being tested each have an amplitude >20% of the amplitude of the combined numerical response.

🟢 **High modal contamination:**
A mode other than the one being tested has an amplitude >70% of the amplitude of the combined numerical response; or two or more modes other than the one being tested each have an amplitude >40% of the amplitude of the combined numerical response.

🟧 **Flipping error:**
The combined algebraic response is flipped approximately 180° from the combined numerical response.

Table 4. Phase angle of combined numerical response at each DOF for each test (degrees)

Test	DOF 1	DOF 2	DOF 3	DOF 4	DOF 5	DOF 6	DOF 7	DOF 8	DOF 9
Mode 1	87.8	-70.5	-73.5	88.1	-72.0	88.7	89.8	-67.4	80.7
Mode 3	86.5	-85.9	-88.2	82.3	-83.2	-89.7	-0.1	79.0	-88.0
Mode 4	90.0	-20.8	63.1	-87.9	-82.8	80.6	85.2	86.8	69.9
Mode 8	71.2	-86.5	87.8	-71.2	87.0	84.5	76.4	-82.1	-80.5

4.0 Conclusions

4.1 Summary

After conducting forced vibration testing of a three-story MATLAB model using both numerical and algebraic methods, several conclusions can be drawn about modal contamination and the testing process.

The algebraic method is generally successful in predicting displacements using theoretical equations but has a couple discrepancies with the numerical method. First, a flipping error may arbitrarily occur that puts the algebraic response 180° out of phase from the numerical data. Second, when substantial modal contamination occurs within the test, the algebraic method produces a faulty amplitude in its response and fails to capture cancellation due to interference from other modes. These two situations are somewhat infrequent, and the algebraic method is typically accurate in predicting displacement responses.

Within this project, a three-story MATLAB model with nine degrees of freedom was developed that can easily be subjected to forced vibration testing. The structural configuration can be altered, and the code can be used for future forced vibration tests. The MATLAB model is useful because the displacement response can easily be broken down into its modal constituents.

Phase angles in the overall displacement response that are significantly different from -90° or 90° may indicate that modal contamination is affecting the response and that forced vibration testing is not successfully capturing a pure response of the mode being tested.

4.2 Future Investigations

Further studies should be conducted to investigate the phase angle and modal contamination in forced vibration testing. The next step would be to adjust the structural parameters of the MATLAB model so that additional symmetrical and unsymmetrical configurations can be tested and compared to the current model. After conducting a thorough computer analysis program, the project would necessitate implementation of a testing procedure on a real structure so that analytical results can be compared to physical forced vibration data. Additionally, improvements can be made to the algebraic method by using advanced numerical theories to adjust the Harmonic Addition Theorem to account for the inherent flipping and amplitude errors. With additional physical and virtual tests and refinement of the algebraic method, a comprehensive understanding of the phase angle and modal contamination can be developed.

5.0 Acknowledgements

We would like to thank our advisor, Graham Archer, for his continuous help, support, and guidance throughout this project. Additionally, Peter Laursen provided valuable contributions and suggestions. Finally, this project was made possible with the support of Al Estes and the faculty and staff of the Cal Poly ARCE department.

6.0 References

[1] Chopra, A.K. 2012. Dynamics of Structures: Theory and Applications to Earthquake Engineering. Pearson Education, Inc., Prentice Hall.

[2] Weisstein, E.M. 1999. Harmonic Addition Theorem. Michigan State University.

7.0 Appendix

8.1 Derivation of the Harmonic Response of a SDOF Oscillator

The equation that describes the governing response to a SDOF system under a harmonic load including viscous damping is a second-order nonhomogeneous equation:

$$m\ddot{u} + c\dot{u} + ku = p(t) \qquad \text{(Eq. 5)}$$

There are two initial conditions defined:

$$u = u(0)$$

$$\dot{u} = \dot{u}(0)$$

Additionally, the SDOF system is subjected to the harmonic load:

$$p(t) = p_0 \sin(\overline{\omega} t)$$

The solution for a second-order nonhomogeneous differential equation is as follows:

$$ay'' + by' + cy = g(x)$$

$$y = y_c + y_p$$

where y_c the complementary solution, the transient response of the system, and y_p is the particular solution, the steady state response of the system. The transient response contains an exponential term which decays within a few periods when damping is considered, leaving only the steady state response after a short period of time. Thus, the response of the SDOF system under a harmonic load with viscous damping is only the particular solution. The equation describing a SDOF system subjected to a sinusoidal load considering damping is of the form:

$$m\ddot{u} + c\dot{u} + ku = p_0 \sin(\omega t) \qquad \text{(Eq. 6)}$$

To more easily determine the particular solution for Equation (2), the equation will be manipulated by dividing each side of the equation by the mass m:

$$\ddot{u} + \frac{c}{m}\dot{u} + \frac{k}{m}u = \frac{p_0}{m}\sin(\omega t) \qquad \text{(Eq. 7)}$$

The following two known relationships can be used to further simplify Equation (3):

$$\omega_n{}^2 = \frac{k}{m}$$

And…

$$\zeta = \frac{c}{2m\omega_n}$$

Substitute into Equation (3) and simplify:

$$\ddot{u} + 2\zeta\omega_n\dot{u} + \omega_n{}^2\, u = \frac{p_0}{m}\sin(\omega t) \qquad \text{(Eq. 4)}$$

The particular solution of the differential equation of the form $g(x) = C\sin(x)$ is as follows:

$$y_p = \ C_1 \cos(x) + \ C_2 \sin(x)$$

Equation (4) has a similar form to the equation $g(x) = C\sin(x)$, thus it has a particular solution in the form:

$$u_p(t) = C\sin(\omega t) + D\cos(\omega t) \qquad \text{(Eq. 5)}$$

Taking the first and second derivatives of the particular solution [Equation (5)] yields the velocity and acceleration terms for a SDOF system:

$$\dot{u}_p(t) = \omega C\cos(\omega t) - \omega D\sin(\omega t) \qquad \text{(Eq. 6)}$$

$$\ddot{u}_p(t) = -\omega^2 C\sin(\omega t) - \omega^2 D\cos(\omega t) \qquad \text{(Eq. 7)}$$

Substitute the velocity [Equation (6)] and acceleration [Equation (7)] into the equation of motion for the SDOF system [Equation (4)]:

$$\begin{aligned}&(-C\omega^2 - 2\zeta\omega_n D\omega + C{\omega_n}^2)\sin(\omega t)\\ &\qquad - \ (-D\omega^2 - 2\zeta\omega_n C\omega + D{\omega_n}^2)\cos(\omega t) = \frac{p_0}{m}\sin(\omega t)\end{aligned} \qquad \text{(Eq. 8)}$$

In Equation (8), the terms in front of the respective sine and cosine terms must equal:

$$-C\omega^2 - 2\zeta\omega_n D\omega + C{\omega_n}^2 = \frac{p_0}{m} \qquad \text{(Eq. 9)}$$

$$-D\omega^2 - 2\zeta\omega_n C\omega + D{\omega_n}^2 = 0 \qquad \text{(Eq. 10)}$$

There are two equations, and two unknown constants, C and D, so the system of equations can be solved to determine their value:

$$(-\omega^2 + {\omega_n}^2)C - (2\zeta\omega_n\omega)D \ = \frac{p_0}{m} \qquad \text{(Eq. 9b)}$$

Knowing…

$${\omega_n}^2 = \frac{k}{m}$$

Simplify Equation (9b) by dividing each term by ${\omega_n}^2$:

$$\left[1 - \left(\frac{\omega}{\omega_n}\right)^2\right]C + \left[-2\zeta\frac{\omega}{\omega_n}\right]D = \frac{p_0}{k} \qquad \text{(Eq. 9c)}$$

Similarly for Equation (10)…

$$\left[-2\zeta\frac{\omega}{\omega_n}\right]C + \left[1 - \left(\frac{\omega}{\omega_n}\right)^2\right]D = 0 \qquad \text{(Eq. 10c)}$$

Solving Equation (9c) and Equation (10c) to solve for the constants C and D yields…

$$C = \frac{p_0}{k}\frac{\left(1-\left({}^{\omega}/_{\omega_n}\right)^2\right)}{\left(1-\left({}^{\omega}/_{\omega_n}\right)^2\right)^2+\left(2\zeta\left({}^{\omega}/_{\omega_n}\right)\right)^2} \qquad \text{(Eq. 11)}$$

$$D = \frac{p_0}{k}\frac{2\zeta\left({}^{\omega}/_{\omega_n}\right)}{\left(1-\left({}^{\omega}/_{\omega_n}\right)^2\right)^2+\left(2\zeta\left({}^{\omega}/_{\omega_n}\right)\right)^2} \qquad \text{(Eq. 12)}$$

So, the particular solution for the given second-order non-homogenous differential equation as defined in Equation (1) is:

$$u_p(t) = \frac{p_0}{k}\frac{\left(1-\left({}^{\omega}/_{\omega_n}\right)^2\right)}{\left(1-\left({}^{\omega}/_{\omega_n}\right)^2\right)^2+\left(2\zeta\left({}^{\omega}/_{\omega_n}\right)\right)^2}\sin(\omega t) + \frac{p_0}{k}\frac{2\zeta\left({}^{\omega}/_{\omega_n}\right)}{\left(1-\left({}^{\omega}/_{\omega_n}\right)^2\right)^2+\left(2\zeta\left({}^{\omega}/_{\omega_n}\right)\right)^2}\cos(\omega t)$$

8.2 Derivation of the Harmonic Response of a SDOF Oscillator (Alternate Form)

Recall the particular solution to the second order nonhomogeneous differential equation:

$$u_p(t) = C\sin(\bar{\omega}t) + D\cos(\bar{\omega}t) \qquad \text{(Eq. 1)}$$

Where the constants C and D are defined as:

$$C = \frac{p_0}{k}\frac{\left(1-\left({}^{\omega}/_{\omega_n}\right)^2\right)}{\left(1-\left({}^{\omega}/_{\omega_n}\right)^2\right)^2+\left(2\zeta\left({}^{\omega}/_{\omega_n}\right)\right)^2}$$

$$D = \frac{p_0}{k}\frac{2\zeta\left({}^{\omega}/_{\omega_n}\right)}{\left(1-\left({}^{\omega}/_{\omega_n}\right)^2\right)^2+\left(2\zeta\left({}^{\omega}/_{\omega_n}\right)\right)^2}$$

The alternate form of the harmonic response of a SDOF oscillator as proposed by Anil K. Chopra has the following form [1]:

$$u(t) = \frac{p_0}{k}R_d\sin(\omega t-\theta) \qquad \text{(Eq. 2)}$$

The general form of Equation (2) can be decomposed utilizing a trigonometric addition formula to obtain an expanded form of the equation:

$$\sin(\alpha-\beta) = \sin\alpha\cos\beta - \sin\beta\cos\alpha$$

Hence…

$$u(t) = \frac{p_0}{k}R_d[\sin\omega t\cos\theta - \sin\theta\cos\omega t] \qquad \text{(Eq. 3)}$$

Which can be rewritten as…

$$u(t) = \left(\frac{p_0}{k}R_d\cos\theta\right)\sin\omega t + \left(-\frac{p_0}{k}R_d\sin\theta\right)\cos\omega t \qquad \text{(Eq. 4)}$$

Next, equate the coefficients from Equation (1) to the coefficients of Equation (4):

$$C = \frac{p_0}{k} R_d \cos\theta \qquad \text{(Eq. 5)}$$

$$D = -\frac{p_0}{k} R_d \sin\theta \qquad \text{(Eq. 6)}$$

Rearrange the equations to isolate the phase angle term (θ):

$$\cos\theta = {}^{C}\!\Big/\!\left(\frac{p_0}{k} R_d\right) \qquad \text{(Eq. 7)}$$

$$\sin\theta = {}^{-D}\!\Big/\!\left(\frac{p_0}{k} R_d\right) \qquad \text{(Eq. 8)}$$

Use a trigonometric identity to combine Equation (7) and Equation (8):

$$\tan\theta = \frac{\sin\theta}{\cos\theta} = \frac{-D/\left(\frac{p_0}{k} R_d\right)}{C/\left(\frac{p_0}{k} R_d\right)} \qquad \text{(Eq. 9)}$$

Thus…

$$\tan\theta = \frac{-D}{C} \qquad \text{(Eq. 10)}$$

Solving for the phase angle (θ): yields:

$$\theta = \tan^{-1}\left({}^{-D}\!/_{C}\right) \qquad \text{(Eq. 11)}$$

Now, solving for the amplitude of the equation of the alternate form, square Equation (5) and Equation (6):

$$C^2 = \left(\frac{p_0}{k} R_d\right)^2 \cos^2\theta \qquad \text{(Eq. 12)}$$

$$D^2 = \left(-\frac{p_0}{k} R_d\right)^2 \sin^2\theta \qquad \text{(Eq. 13)}$$

Adding Equation (12) and Equation (13) gives:

$$C^2 + D^2 = \left(\frac{p_0}{k} R_d\right)^2 \left(\sin^2\theta + \cos^2\theta\right) \qquad \text{(Eq. 14)}$$

Utilizing a trigonometric identity, Equation (14) can be simplified to:

$$C^2 + D^2 = \left(\frac{p_0}{k} R_d\right)^2 \qquad \text{(Eq. 15)}$$

Therefore…

$$\left(\frac{p_0}{k} R_d\right) = \sqrt{C^2 + D^2} \qquad \text{(Eq. 16)}$$

In summary:

$$u(t) = \sqrt{C^2 + D^2}\,\sin\left(\omega t - \tan^{-1}\left({}^{-D}\!/_{C}\right)\right)$$

$$u(t) = \frac{p_0}{k} R_d \sin(\omega t - \theta) \qquad \text{(Eq. 2)}$$

8.3 MATLAB Code

```
% Nicholas Slavin and Ryan Thornton
% Spring 2020
% Cal Poly ARCE Senior Project
% 3-Story Forced Vibration Model

clear all; close all; clc;

% This MATLAB script runs a forced vibration test on a three-story
% structure.  The floors are rectangular with columns at each corner and
% degrees of freedom (DOFs) are at the center of each floor.  The user
% should adjust parameters in the first three sections of code in order to
% define the structure, the applied harmonic load, and the displacement
% recording location.
```

Define Model

```
% The user can define the following parameters to change the structural
% configuration.

% Damping Ratio
zeta = .03;
```

```
% Weight of Each Floor
w.f1 = 150;               % 1st Floor Weight (psf)
w.f2 = 150;               % 2nd Floor Weight (psf)
w.f3 = 75;                % 3rd Floor Weight (psf)

% Structure Dimensions
h.f1 = 15;                % 1st Floor Height (ft)
h.f2 = 12;                % 2nd Floor Height (ft)
h.f3 = 12;                % 3rd Floor Height (ft)
x = 200*12;               % Structure Length (in)
y = 100*12;               % Structure Width (in)

% Column Properties
E = 29000;                                     % Column Modulus of Elasticity (ksi)
I.f1 = 8000*[2.6 .7 2.2 1.5; 1.2 .9 2.4 .5];   % Moment of Inertia (in^4) 1st floor matrix where
cols = each corner column, row 1 = x principle direction of each column, row 2 = y principle
direction
I.f2 = 4000*[.2 1.1 .2 .7; .2 1.3 1.0 1.5];    % Moment of Inertia (in^4) 2nd floor matrix
I.f3 = 800*[1 .7 1.2 .3; .6 1.7 .8 .6];        % Moment of Inertia (in^4) 3rd floor matrix

% Column Stiffnesses (assuming all are fix-fix)
kcol.f1 = 12*E*I.f1/(h.f1*12)^3;     % 1st Floor Column Stiffnesses (k/in)
kcol.f2 = 12*E*I.f2/(h.f2*12)^3;     % 2nd Floor Column Stiffnesses (k/in)
kcol.f3 = 12*E*I.f3/(h.f3*12)^3;     % 3rd Floor Column Stiffnesses (k/in)
```

Define Loading

```
% In this section, the user can define the applied harmonic force
% amplitude, frequency, time vector, and location and direction within the
% structure.

dt = .01;                   % Time Step (s)
t = 0:dt:1.5;               % Time Interval (s)
F.p0 = 100;                 % Force Amplitude (k)

% Forcing Frequency (rad/s): change to the natural frequency of the mode
% being tested
% wbar = 6.6719;              % Mode 1
% wbar = 11.262;              % Mode 3
wbar = 12.7035;             % Mode 4
% wbar = 28.0279;             % Mode 8

% Location of Load: change to activate the mode shape of the mode being
% tested

% % Mode 1 Excitation
% F.x = 100*12;               % x-coordinate from bottom left corner (in)
% F.y = 50*12;                % y-coordinate from bottom left corner (in)
% F.floor = 3;                % floor (1, 2, or 3)
% F.rot = 160.62;             % counterclockwise rotation of load from positive x-direction
(degrees)

% % Mode 3 Excitation
% F.x = 5*12;                  % x-coordinate from bottom left corner (in)
% F.y = 50*12;                 % y-coordinate from bottom left corner (in)
% F.floor = 3;                 % floor (1, 2, or 3)
% F.rot = 90;                  % counterclockwise rotation of load from positive x-direction
(degrees)

% Mode 4 Excitation
F.x = 100*12;                % x-coordinate from bottom left corner (in)
F.y = 50*12;                 % y-coordinate from bottom left corner (in)
F.floor = 2;                 % floor (1, 2, or 3)
F.rot = -4.4938;             % counterclockwise rotation of load from positive x-direction
(degrees)

% % Mode 8 Excitation
% F.x = 80*12;                 % x-coordinate from bottom left corner (in)
% F.y = 50*12;                 % y-coordinate from bottom left corner (in)
% F.floor = 1;                 % floor (1, 2, or 3)
% F.rot = 90;                  % counterclockwise rotation of load from positive x-direction
(degrees)
```

Define Displacement Recording Location

```
% In addition to displacement being recorded at each DOF, here the user can
% specify where they want to record the displacement at any location in the
% structure.

uchosen.floor = 1;                 % floor (1, 2, or 3)
uchosen.xcoord = 100*12;           % x-coordinate from bottom left corner (in)
uchosen.ycoord = 50*12;            % y-coordinate from bottom left corner (in)
```

Stiffness Matrix

```
% This section develops a stiffness matrix for the 3-story structure.

% Initiate K Matrix
K = zeros(9,9);

% K Matrix Column 1
K(1,1) = sum(kcol.f1(1,:))+sum(kcol.f2(1,:));
K(3,1) = y/2*(-kcol.f1(1,1)-kcol.f1(1,2)+kcol.f1(1,3)+kcol.f1(1,4)-kcol.f2(1,1)-
kcol.f2(1,2)+kcol.f2(1,3)+kcol.f2(1,4));
K(4,1) = -sum(kcol.f2(1,:));
K(6,1) = y/2*(kcol.f2(1,1)+kcol.f2(1,2)-kcol.f2(1,3)-kcol.f2(1,4));

% K Matrix Column 2
K(2,2) = sum(kcol.f1(2,:))+sum(kcol.f2(2,:));
K(3,2) = x/2*(-kcol.f1(2,1)+kcol.f1(2,2)-kcol.f1(2,3)+kcol.f1(2,4)-kcol.f2(2,1)+kcol.f2(2,2)-
kcol.f2(2,3)+kcol.f2(2,4));
K(5,2) = -sum(kcol.f2(2,:));
K(6,2) = x/2*(kcol.f2(2,1)-kcol.f2(2,2)+kcol.f2(2,3)-kcol.f2(2,4));

% K Matrix Column 3
```

```
K(1,3) = y/2*(-kcol.f1(1,1)-kcol.f1(1,2)+kcol.f1(1,3)+kcol.f1(1,4)-kcol.f2(1,1)-
kcol.f2(1,2)+kcol.f2(1,3)+kcol.f2(1,4));
K(2,3) = x/2*(-kcol.f1(2,1)+kcol.f1(2,2)-kcol.f1(2,3)+kcol.f1(2,4)-kcol.f2(2,1)+kcol.f2(2,2)-
kcol.f2(2,3)+kcol.f2(2,4));
K(3,3) =
(y/2)^2*(sum(kcol.f1(1,:))+sum(kcol.f2(1,:)))+(x/2)^2*(sum(kcol.f1(2,:))+sum(kcol.f2(2,:)));
K(4,3) = y/2*(kcol.f2(1,1)+kcol.f2(1,2)-kcol.f2(1,3)-kcol.f2(1,4));
K(5,3) = x/2*(kcol.f2(2,1)-kcol.f2(2,2)+kcol.f2(2,3)-kcol.f2(2,4));
K(6,3) = (y/2)^2*-sum(kcol.f2(1,:))+(x/2)^2*-sum(kcol.f2(2,:));

% K Matrix Column 4
K(1,4) = -sum(kcol.f2(1,:));
K(3,4) = y/2*(kcol.f2(1,1)+kcol.f2(1,2)-kcol.f2(1,3)-kcol.f2(1,4));
K(4,4) = sum(kcol.f2(1,:))+sum(kcol.f3(1,:));
K(6,4) = y/2*(-kcol.f2(1,1)-kcol.f2(1,2)+kcol.f2(1,3)+kcol.f2(1,4)-kcol.f3(1,1)-
kcol.f3(1,2)+kcol.f3(1,3)+kcol.f3(1,4));
K(7,4) = -sum(kcol.f3(1,:));
K(9,4) = y/2*(kcol.f3(1,1)+kcol.f3(1,2)-kcol.f3(1,3)-kcol.f3(1,4));

% K Matrix Column 5
K(2,5) = -sum(kcol.f2(2,:));
K(3,5) = x/2*(kcol.f2(2,1)-kcol.f2(2,2)+kcol.f2(2,3)-kcol.f2(2,4));
K(5,5) = sum(kcol.f2(2,:))+sum(kcol.f3(2,:));
K(6,5) = x/2*(-kcol.f2(2,1)+kcol.f2(2,2)-kcol.f2(2,3)+kcol.f2(2,4)-kcol.f3(2,1)+kcol.f3(2,2)-
kcol.f3(2,3)+kcol.f3(2,4));
K(8,5) = -sum(kcol.f3(2,:));
K(9,5) = x/2*(kcol.f3(2,1)-kcol.f3(2,2)+kcol.f3(2,3)-kcol.f3(2,4));

% K Matrix Column 6
K(1,6) = y/2*(kcol.f2(1,1)+kcol.f2(1,2)-kcol.f2(1,3)-kcol.f2(1,4));
K(2,6) = x/2*(kcol.f2(2,1)-kcol.f2(2,2)+kcol.f2(2,3)-kcol.f2(2,4));
K(3,6) = (y/2)^2*-sum(kcol.f2(1,:))+(x/2)^2*-sum(kcol.f2(2,:));
K(4,6) = y/2*(-kcol.f2(1,1)-kcol.f2(1,2)+kcol.f2(1,3)+kcol.f2(1,4)-kcol.f3(1,1)-
kcol.f3(1,2)+kcol.f3(1,3)+kcol.f3(1,4));
K(5,6) = x/2*(-kcol.f2(2,1)+kcol.f2(2,2)-kcol.f2(2,3)+kcol.f2(2,4)-kcol.f3(2,1)+kcol.f3(2,2)-
kcol.f3(2,3)+kcol.f3(2,4));
K(6,6) =
(y/2)^2*(sum(kcol.f2(1,:))+sum(kcol.f3(1,:)))+(x/2)^2*(sum(kcol.f2(2,:))+sum(kcol.f3(2,:)));
K(7,6) = y/2*(kcol.f3(1,1)+kcol.f3(1,2)-kcol.f3(1,3)-kcol.f3(1,4));
K(8,6) = x/2*(kcol.f3(2,1)-kcol.f3(2,2)+kcol.f3(2,3)-kcol.f3(2,4));
K(9,6) = (y/2)^2*-sum(kcol.f3(1,:))+(x/2)^2*-sum(kcol.f3(2,:));

% K Matrix Column 7
K(4,7) = -sum(kcol.f3(1,:));
K(6,7) = y/2*(kcol.f3(1,1)+kcol.f3(1,2)-kcol.f3(1,3)-kcol.f3(1,4));
K(7,7) = sum(kcol.f3(1,:));
K(9,7) = y/2*(-kcol.f3(1,1)-kcol.f3(1,2)+kcol.f3(1,3)+kcol.f3(1,4));

% K Matrix Column 8
K(5,8) = -sum(kcol.f3(2,:));
K(6,8) = x/2*(kcol.f3(2,1)-kcol.f3(2,2)+kcol.f3(2,3)-kcol.f3(2,4));
K(8,8) = sum(kcol.f3(2,:));
K(9,8) = x/2*(-kcol.f3(2,1)+kcol.f3(2,2)-kcol.f3(2,3)+kcol.f3(2,4));
```

```
% K Matrix Column 9
K(4,9) = y/2*(kcol.f3(1,1)+kcol.f3(1,2)-kcol.f3(1,3)-kcol.f3(1,4));
K(5,9) = x/2*(kcol.f3(2,1)-kcol.f3(2,2)+kcol.f3(2,3)-kcol.f3(2,4));
K(6,9) = (y/2)^2*-sum(kcol.f3(1,:))+(x/2)^2*-sum(kcol.f3(2,:));
K(7,9) = y/2*(-kcol.f3(1,1)-kcol.f3(1,2)+kcol.f3(1,3)+kcol.f3(1,4));
K(8,9) = x/2*(-kcol.f3(2,1)+kcol.f3(2,2)-kcol.f3(2,3)+kcol.f3(2,4));
K(9,9) = (y/2)^2*sum(kcol.f3(1,:))+(x/2)^2*sum(kcol.f3(2,:));
```

Mass Matrix

```
% Here, a mass matrix is developed for the 3-story structure.

% Point Mass of Each Floor
m.f1 = w.f1*x*y/1000/386.4/144;     % Floor 1 mass (k*s^2/in)
m.f2 = w.f2*x*y/1000/386.4/144;     % Floor 2 mass (k*s^2/in)
m.f3 = w.f3*x*y/1000/386.4/144;     % Floor 3 mass (k*s^2/in)

% Mass Matrix
M = zeros(9,9);
M(1,1) = m.f1;
M(2,2) = m.f1;
M(3,3) = m.f1/12*(x^2+y^2);
M(4,4) = m.f2;
M(5,5) = m.f2;
M(6,6) = m.f2/12*(x^2+y^2);
M(7,7) = m.f3;
M(8,8) = m.f3;
M(9,9) = m.f3/12*(x^2+y^2);
```

Mode Shapes and Frequencies

```
% In this section, the eig function is used to calculate the mode shape
% matrix and frequency matrix for the structure

[phi omega2] = eig(K,M);

% Check if mode shape matrix is mass-normalized
Identity = phi'*M*phi;
```

Mode Shape Plots

```
% The mode shapes are then plotted so that the user can easily visualize
% the inherent movements of the structure.  In this section, nine plots are
% developed that show plan views of how the three floors move for each mode
% shape.

% Initial Position of Floors
a = 1/12*[-x/2,x/2,x/2,-x/2,-x/2];          % X-coordinates of Diaphragm
b = 1/12*[-y/2,-y/2,y/2,y/2,-y/2];          % Y-coordinates of Diaphragm

% Displacement Scale Factor
```

```
Sms = 100;                                   % Scale displacements to make them visible

% Loop through Each Mode Shape Plot
for ii = 1:9
    figure

    % Initial Position
    plot(a,b,':k','Linewidth',1);
    hold on;

    % Floor 1
    ums.dof1 = a+phi(1,ii)*Sms;                           % Floor 1 Displacement in the X dir.
    ums.dof2 = b+phi(2,ii)*Sms;                           % Floor 1 Displacement in the Y dir.
    R.f1 = plot(ums.dof1,ums.dof2,'-.k','Linewidth',1);   % Translations Only
    rotate(R.f1,[0,0,1],rad2deg(phi(3,ii))*Sms);         % Displacement about Z axis
    hold on

    % Floor 2
    ums.dof4 = a+phi(4,ii)*Sms;                           % Floor 2 Displacement in the X dir.
    ums.dof5 = b+phi(5,ii)*Sms;                           % Floor 2 Displacement in the Y dir.
    R.f2 = plot(ums.dof4,ums.dof5,'--k','Linewidth',1);   % Translations Only
    rotate(R.f2,[0,0,1],rad2deg(phi(6,ii))*Sms);         % Displacement about Z axis
    hold on

    % Floor 3
    ums.dof7 = a+phi(7,ii)*Sms;                           % Floor 3 Displacement in the X dir.
    ums.dof8 = b+phi(8,ii)*Sms;                           % Floor 3 Displacement in the Y dir.
    R.f3 = plot(ums.dof7,ums.dof8,'-k','Linewidth',1);    % Translations Only
    rotate(R.f3,[0,0,1],rad2deg(phi(9,ii))*Sms);         % Displacement about Z axis

    % Plot Labels/Characteristics
    legend('Initial Position','Floor 1','Floor 2','Floor
3','location','southoutside','numcolumns',4,'fontsize',14,'fontname','times')
    title(['Mode Shape ',num2str(ii),': \Phi_',num2str(ii)],'fontsize',14);
    xlabel('X Displacement','fontsize',14,'fontname','times');
    ylabel('Y Displacement','fontsize',14,'fontname','times');
    set(gca,'XTickLabel',[],'YTickLabel',[])
    xlim(1/12*[-x,x])
    ylim(1/12*[-y,y])
    grid on
    grid minor
end
```

Natural Periods

```
% Here, the fundamental frequencies and periods of each mode are
% calculated.  This is the last section of code that deals only with the
% structure before the harmonic force is applied.

for ii = 1:size(omega2,1);
    wn(ii) = sqrt(omega2(ii,ii));       % Frequency (rad/s)
    T(ii) = 2*pi/wn(ii);                % Period (s)
end
```

Apply Force

```
% In this section, the applied harmonic force is translated from its
% designated location to its component at each DOF.

F.dof = zeros(9,1);
F.p0x = F.p0*cosd(F.rot);                               % x-component of applied force (k)
F.p0y = F.p0*sind(F.rot);                               % y-component of applied force (k)
F.p0rot = F.p0x*(y/2-F.y)-F.p0y*(x/2-F.x);              % rotational component of applied force
(k*in)
F.dof(F.floor*3-2:F.floor*3) = [F.p0x F.p0y F.p0rot];   % Force vector for each DOF (k or k*in)
```

Steady State Response for Each Mode

```
% After determining the modes and applied forces, the displacement response
% of the structure is determined in a modal context.  This section
% calculates the terms in the steady state solution for each mode.

% Loop through Each Mode
for ii = 1:9
    p0(ii) = phi(:,ii)'*F.dof;                              % Force amplitude in modal context
    k(ii) = wn(ii)^2;                                       % Stiffness used in steady state
equation
    beta(ii) = wbar/wn(ii);                                 % Frequency ratio between forcing
frequency and natural frequency
    Rd(ii) = 1/sqrt((1-beta(ii)^2)^2+(2*zeta*beta(ii))^2);  % Dynamic amplification factor
    rho(ii,1) = p0(ii)/k(ii)*Rd(ii);                        % Amplitude of steady state
displacement response in modal context
    thetaq(ii,1) = atand(2*zeta*beta(ii)/(1-beta(ii)^2));   % Phase angle of steady state
displacement response in modal context (degrees)
    q(ii,:) = rho(ii)*sin(wbar*t-thetaq(ii)*pi/180);        % Modal displacement response of
structure
end
```

Convert to Real Displacements at Each DOF

```
% After calculating displacements in the modal world, the displacement are
% converted to the real world and summed up using a numerical method in
% this section.  Contribution of each mode to displacement at each DOF is
% tracked in addition to the real combined displacements.

% The DOFVector is used as labels to cycle through when creating data
% structures and tracking displacements for each DOF
DOFVector = ["DOF1","DOF2","DOF3","DOF4","DOF5","DOF6","DOF7","DOF8","DOF9"];

% Real Displacements (in or rad)
% Cycle through each DOF
for ii = 1:9
```

```
    % Cycle through each mode
    for jj = 1:9
        u.(char(DOFVector(ii)))(jj,:) = phi(ii,jj)*q(jj,:);     % Modal contributions to
displacement at each DOF
    end
    % Cycle through each time step
    for jj = 1:size(q,2)
        u.numerical(ii,jj) = sum(u.(char(DOFVector(ii)))(:,jj)); % Combined displacement due to
summation of modal contributions at each DOF
    end
end

% Phase Angle for Each DOF (degrees)
% Note: these may generate imaginary numbers or be undefined because values
% are being backed out of the numerical response using the arcsine
% function.  It is recommended to retrieve phase angle using the algebraic
% method to avoid this problem.
% Cycle through each DOF
for ii = 1:9
    % Cycle through each mode
    for jj = 1:9
        theta.(char(DOFVector(ii)))(jj,1) = -
asind(u.(char(DOFVector(ii)))(jj,1)/max(u.(char(DOFVector(ii)))(jj,:))); % Phase angle of each
modal contribution
    end
    theta.numerical(ii,1) = -asind(u.numerical(ii,1)/max(u.numerical(ii,:)));  % Phase angle of
combined displacement response
end

% Amplitude for each DOF (in or rad)
% Cycle through each DOF
for ii = 1:9
    % Cycle through each mode
    for jj = 1:9
        amp.(char(DOFVector(ii)))(jj,1) = max(u.(char(DOFVector(ii)))(jj,:)); % Amplitude of each
modal contribution
    end
    amp.numerical(ii,1) = max(u.numerical(ii,:));     % Amplitude of combined displacement
response
    % Express amplitudes for each modal contribution as a percentage of the
    % combined response amplitude for each DOF
    for jj = 1:9
        amp.percent.(char(DOFVector(ii)))(jj,1) =
amp.(char(DOFVector(ii)))(jj,1)/amp.numerical(ii,1);
    end
end
```

Real Displacements at Selected Location

```
% In addition to the response at each DOF, here, the code calculates the
% response at the location on the structure that was selected by the user.

% Contribution of displacements for each mode
```

```
uchosen.d1 = sqrt((uchosen.ycoord-y/2)^2+(uchosen.xcoord-x/2)^2);   % Distance of selected
location from the center of mass (in)
uchosen.alpha = atan((uchosen.ycoord-y/2)/(uchosen.xcoord-x/2));    % Angle of selected location
from positive x direction about the center of mass (rad)
% Cycle through each mode
for ii = 1:9
    uchosen.d2(ii,:) = uchosen.d1*tan(u.(char(DOFVector(uchosen.floor*3)))(ii,:));  % Movement at
selected location due to rotation at the selected floor's center of mass (in)
    if uchosen.xcoord == x/2 && uchosen.ycoord == y/2                               % If the
selected location is at the center of mass, all the motion is due to the translational DOFs
        uchosen.x(ii,:) = u.(char(DOFVector(uchosen.floor*3-2)))(ii,:);             % Movement in
x direction at selected location due to applied force (in)
        uchosen.y(ii,:) = u.(char(DOFVector(uchosen.floor*3-1)))(ii,:);             % Movement in
y direction at selected location due to applied force (in)
    elseif uchosen.xcoord >= x/2                                                    % If the
selected location is not at the center of mass, the rotational movement needs to be incorporated
        uchosen.x(ii,:) = -uchosen.d2(ii,:)*sin(uchosen.alpha)+u.(char(DOFVector(uchosen.floor*3-
2)))(ii,:);    % X direction at selected location (in)
        uchosen.y(ii,:) = uchosen.d2(ii,:)*cos(uchosen.alpha)+u.(char(DOFVector(uchosen.floor*3-
1)))(ii,:);     % Y direction at selected location (in)
    else
        uchosen.x(ii,:) = uchosen.d2(ii,:)*sin(uchosen.alpha)+u.(char(DOFVector(uchosen.floor*3-
2)))(ii,:);     % X direction at selected location (in)
        uchosen.y(ii,:) = -uchosen.d2(ii,:)*cos(uchosen.alpha)+u.(char(DOFVector(uchosen.floor*3-
1)))(ii,:);    % Y direction at selected location (in)
    end
end

% Combined displacement response at the selected location
uchosen.d2combined = uchosen.d1*tan(u.numerical(uchosen.floor*3,:));
if uchosen.xcoord == x/2 && uchosen.ycoord == y/2
    uchosen.xcombined = u.numerical(uchosen.floor*3-2,:);
    uchosen.ycombined = u.numerical(uchosen.floor*3-1,:);
elseif uchosen.xcoord >= x/2
    uchosen.xcombined = -uchosen.d2combined*sin(uchosen.alpha)+u.numerical(uchosen.floor*3-2,:);
    uchosen.ycombined = uchosen.d2combined*cos(uchosen.alpha)+u.numerical(uchosen.floor*3-1,:);
else
    uchosen.xcombined = uchosen.d2combined*sin(uchosen.alpha)+u.numerical(uchosen.floor*3-2,:);
    uchosen.ycombined = -uchosen.d2combined*cos(uchosen.alpha)+u.numerical(uchosen.floor*3-1,:);
end
```

Algebraic Method

```
% This section conducts the same analysis as above but uses the algebraic
% method to combine sine waves and solve for displacement responses.

% Cycle through DOFs and modes
for ii = 1:9
    for jj = 1:9
        Alg.A(ii,jj) = phi(ii,jj)*rho(jj);  % Convert amplitudes from modal to real context
    end
end
```

```
Alg.thetaq_rad = deg2rad(thetaq);           % Convert phase angle degrees to radians (phase
angles don't change from modal to real context before combining modes)

% Cycle through DOFs and modes
for ii = 1:9
    for jj = 1:9
        Alg.delta_c(ii,jj) = Alg.A(ii,jj)*cos(Alg.thetaq_rad(jj));  % Term in harmonic addition
theorem for combining phase angles
        Alg.delta_s(ii,jj) = Alg.A(ii,jj)*sin(Alg.thetaq_rad(jj));  % Term in harmonic addition
theorem for combining phase angles
    end
end

% Cycle through DOFs
for ii = 1:9
    Alg.delta_cos(ii,1) = sum(Alg.delta_c(ii,:));   % Term in harmonic addition theorem for
combining phase angles (Note: whenever this term is negative, the flipping error occurs in the
combined algebraic response)
    Alg.delta_sin(ii,1) = sum(Alg.delta_s(ii,:));   % Term in harmonic addition theorem for
combining phase angles
end

% Cycle through DOFs
for jj = 1:9
    theta.algebraic(jj) = atand(Alg.delta_sin(jj)/Alg.delta_cos(jj)); % Phase angle for the
combined displacement response using the algebraic method (degrees)
end

% Cycle through DOFs and modes
for ii = 1:9
    for jj = 1:9
        Alg.A1(ii,jj) = Alg.A(ii,jj)^2;             % Term in harmonic addition theorem for
combining amplitudes
        for kk = jj+1:9
            Alg.A2(ii,jj) = Alg.A(ii,jj)*Alg.A(ii,kk)*cos(Alg.thetaq_rad(jj)-Alg.thetaq_rad(kk));
% Term in harmonic addition theorem for combining amplitudes
        end
    end
end

% Cycle through DOFs
for ii = 1:9
    Alg.A_sq(ii) = sum(Alg.A1(ii,:))+2*sum(Alg.A2(ii,:));   % Term in harmonic addition theorem
for combining amplitudes
    amp.algebraic(ii) = sqrt(Alg.A_sq(ii));               % Amplitude for the combined
displacement response using the algebraic method (in)
    u.algebraic(ii,:) = amp.algebraic(ii)*sin(wbar*t-deg2rad(theta.algebraic(ii))); %
Displacement response at each DOF
end
```

Plot of Modal Displacements

```
% This plot shows the modal "q" displacements for each mode before being
% converted back into a real context

% Line styles cycled through in many of the plots
linestylevector = ["k-","k--","k-.","b-","b--","b-.","r-","r--","r-."];

% Plot of modal displacements
figure
for ii = 1:9
    plot(t,q(ii,:),(char(linestylevector(ii))))
    hold on
end
title('Steady State Response within Modal "q" Context')
xlabel('Time (s)')
ylabel('Displacement/Modal Contribution')
legend('Mode 1','Mode 2','Mode 3','Mode 4','Mode 5','Mode 6','Mode 7','Mode 8','Mode 9')
grid on
grid minor
```

Plots of Modal Contributions to Displacement at Each DOF

```
% This series of nine plots (one for each DOF) shows the displacement that
% occurs due to each mode at each DOF.  These displacements are after the
% responses have been converted back into a real context, before the
% different modal contributions have been summed up.  The summed up
% responses are also shown on the plots, including both the numerical and
% algebraic combination methods.

% Plots of modal contributions to displacement at each DOF
% Cycle through each DOF
for ii = 1:9
    figure
    plot(t,u.numerical(ii,:),'g','Linewidth',1.5)    % plot numerical combined response
    hold on
    plot(t,u.algebraic(ii,:),':m','Linewidth',1.5)          % plot algebraic combined response
    hold on
    for jj = 1:9                                    % cycle through each mode
        plot(t,u.(char(DOFVector(ii)))(jj,:),(char(linestylevector(jj))))   % plot modal
contribution to displacement at each DOF
        hold on
    end
    title(['Mode 3 Excitation Response: DOF ',num2str(ii)],'fontsize',14)
    xlabel('Time (s)','fontsize',14,'fontname','times')
    if ii == 3 || ii == 6 || ii == 9
        ylabel('Displacement (rad)','fontsize',14,'fontname','times')
    else
        ylabel('Displacement (in)','fontsize',14,'fontname','times')
    end
    legend('Combined Numerical','Combined Algebraic','Mode 1 (1st x)','Mode 2 (1st y)','Mode 3
(1st rot)','Mode 4 (2nd x)','Mode 5 (2nd y)','Mode 6 (2nd rot)','Mode 7 (3rd x) ','Mode 8 (3rd
y)','Mode 9 (3rd rot)','fontsize',14,'fontname','times')
    set(gca,'fontsize',14,'fontname','times')
    grid on
    grid minor
end
```

Plots of Combined Displacements at Each DOF

```
% These plots show the combined (real) displacements at each DOF all on the
% same graph.  There are two figures: one for the responses calculated with
% the numerical method and one for the responses found using the algebraic
% method.

% Scale factor so that responses in radians can be visible when plotted on
% the same graph as responses in inches
scalerad = 500;

% Plot of Numerical Method Responses
figure
% Cycle through DOFs
for ii = 1:9
    if ii == 3 || ii == 6 || ii == 9
        plot(t,scalerad*(u.numerical(ii,:)),(char(linestylevector(ii)))) % Rotational DOFs
    else
        plot(t,u.numerical(ii,:),(char(linestylevector(ii))))    % Translational DOFs
    end
    hold on
end
title('Combined DOFs: Numerical Method','fontsize',14,'fontname','times')
xlabel('Time (s)','fontsize',14,'fontname','times')
ylabel('Displacement (in or rad*500)','fontsize',14,'fontname','times')
legend('DOF 1 (Flr 1-x)','DOF 2 (Flr 1-y)','DOF 3 (Flr 1-rot)','DOF 4 (Flr 2-x)','DOF 5 (Flr 2-
y)','DOF 6 (Flr 2-rot)','DOF 7 (Flr 3-x)','DOF 8 (Flr 3-y)','DOF 9 (Flr 3-
rot)','fontsize',14,'fontname','times')
set(gca,'fontsize',14,'fontname','times')
grid on
grid minor

% Plot of Algebraic Method Responses
figure
% Cycle through DOFs
for ii = 1:9
    if ii == 3 || ii == 6 || ii == 9
        plot(t,scalerad*u.algebraic(ii,:),(char(linestylevector(ii))))  % Rotational DOFs
    else
        plot(t,u.algebraic(ii,:),(char(linestylevector(ii))))           % Translational DOFs
    end
    hold on
end
title('Combined DOFs: Algebraic Method','fontsize',14,'fontname','times')
xlabel('Time (s)','fontsize',14,'fontname','times')
ylabel('Displacement (in or rad*500)','fontsize',14,'fontname','times')
legend('u1 (flr 1-x)','u2 (flr 1-y)','u3 (flr 1-rot)','u4 (flr 2-x)','u5 (flr 2-y)','u6 (flr 2-
rot)','u7 (flr 3-x)','u8 (flr 3-y)','u9 (flr 3-rot)','fontsize',14,'fontname','times')
set(gca,'fontsize',14,'fontname','times')
grid on
grid minor
```

Plots of Displacement at Selected Location

```
% Here, two plots are generated.  The first one shows the harmonic
% displacements in the x direction at the location on the structure
% previously specified by the user, and the second figure shows the
% displacements in the y direction.  Displacements are given for each modal
% contribution as well as the combined numerical response.

% Plot of Displacements in X Direction
figure
plot(t,uchosen.xcombined,'g','Linewidth',1.5)           % Combined numerical response (in)
hold on
% Cycle through contributions from each mode
for ii = 1:9
    plot(t,uchosen.x(ii,:),(char(linestylevector(ii)))) % Response from each modal contribution
(in)
    hold on
end
title('Steady State Response: Selected Location, x-direction')
xlabel('Time (s)')
ylabel('Displacement (in)')
legend('Combined Numerical','Mode 1','Mode 2','Mode 3','Mode 4','Mode 5','Mode 6','Mode 7','Mode
8','Mode 9')
grid on
grid minor

% Plot of Displacements in Y Direction
figure
plot(t,uchosen.ycombined,'g','Linewidth',1.5)           % Combined numerical respons (in)
hold on
% Cycle through contributions from each mode
for ii = 1:9
    plot(t,uchosen.y(ii,:),(char(linestylevector(ii)))) % Response from each modal contribution
(in)
    hold on
end
title('Steady State Response: Selected Location, y-direction')
xlabel('Time (s)')
ylabel('Displacement (in)')
legend('Mode 1','Mode 2','Mode 3','Mode 4','Mode 5','Mode 6','Mode 7','Mode 8','Mode
9','Combined')
grid on
grid minor
```

Illustration of Displaced Structure

```
% This section aims to create an approximate illustration of the displaced
% structure but may need some work in order to be an accurate
% representation.

% Initial Position of Floors
a = 1/12*[-x/2,x/2,x/2,-x/2,-x/2];              % X-coordinates of Diaphragm
b = 1/12*[-y/2,-y/2,y/2,y/2,-y/2];              % Y-coordinates of Diaphragm

% Displacement Scale Factor
Sf = 15/max(max(u.numerical));                 % Scale displacements to make them visible

% Labels/Styles to Loop through When Plotting
linestyle2 = ["-.k","--k","-k"];
linewidth2 = [.4,.7,1];
RotFlrs = ["f1","f2","f3"];

% Plot Plan View of Displaced Structure
figure
plot(a,b,':k')                                  % Initial Position
hold on
% Loop through each floor
for ii = [1,4,7]
    if u.numerical(ii,round(pi/(2*wbar*dt))) >= 0         % Displacement in x direction
        uf(ii,:) = a+max(u.numerical(ii,:))*Sf;
    else
        uf(ii,:) = a+min(u.numerical(ii,:))*Sf;
    end
    if u.numerical(ii+1,round(pi/(2*wbar*dt))) >= 0       % Displacement in y direction
        uf(ii+1,:) = b+max(u.numerical(ii+1,:))*Sf;
    else
        uf(ii+1,:) = b+min(u.numerical(ii+1,:))*Sf;
    end
    Rf.(char(RotFlrs((ii+2)/3))) =
plot(uf(ii,:),uf(ii+1,:),(char(linestyle2((ii+2)/3))),'Linewidth',linewidth2((ii+2)/3));
    if u.numerical(ii+2,round(pi/(2*wbar*dt))) >= 0       % Rotational displacement
        rotate(Rf.(char(RotFlrs((ii+2)/3))),[0,0,1],rad2deg(max(u.numerical(ii+2,:)))*Sf);
    else
        rotate(Rf.(char(RotFlrs((ii+2)/3))),[0,0,1],rad2deg(min(u.numerical(ii+2,:)))*Sf);
    end
    hold on
end
legend('Initial Position','Floor 1','Floor 2','Floor 3')
title('Illustration of Displaced Structure Shape')
xlabel('Displacement')
ylabel('Displacement')
xlim(1/12*[-x,x])
ylim(1/12*[-y,y])
```