Homework 4

Louis Lin (PID: A59006427)

Kyoungyeon Lee (PID: A59000512)

March 21, 2021

University of California, San Diego

Prof. Joel Conte

SE 201B Nonlinear Structural Analysis

Table of Contents

List of Figure

List of Tables

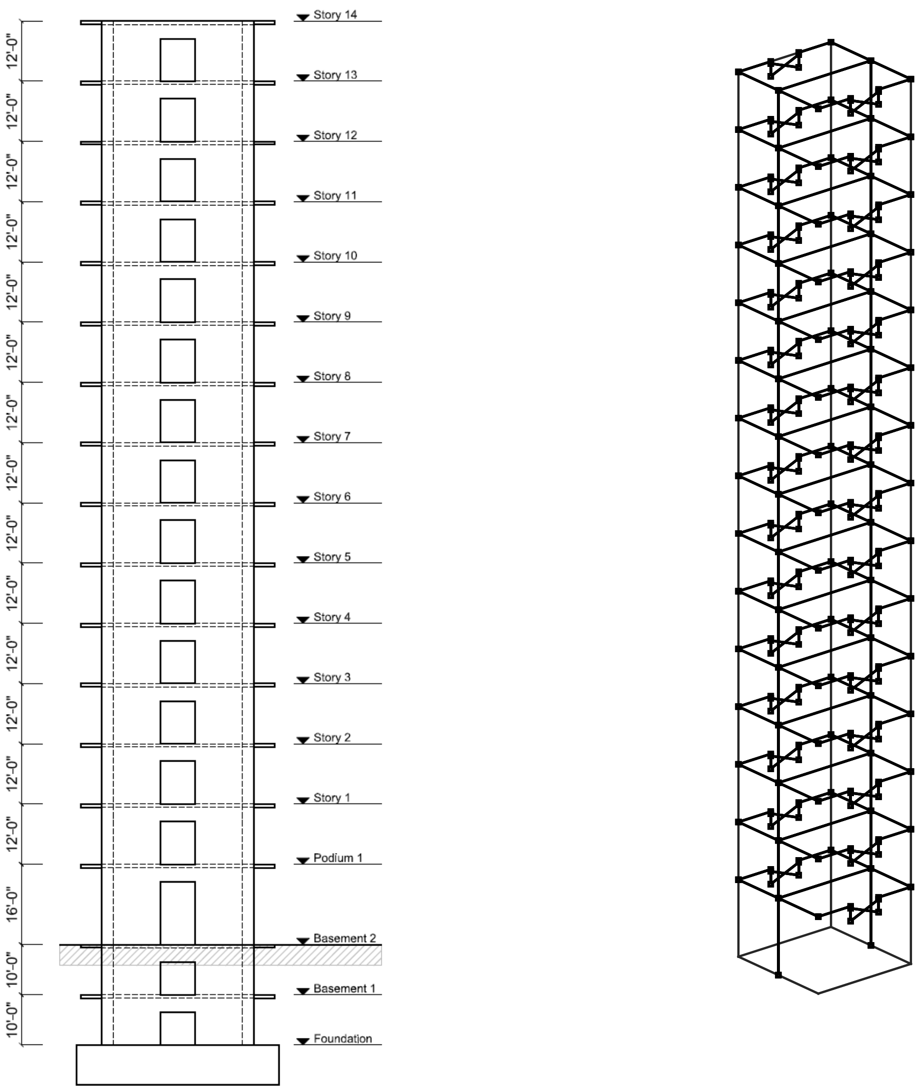
Acknowledgement

The authors would like the thank Prof. Joel Conte for the wealth of knowledge that he provided in nonlinear analysis.

The authors would like to recognize that Angshuman Deb and Maitreya Manoj Kurumbhati for their generous contribution in establishing the base model files for the analyses.

Project Description

The objective of this report was to perform a nonlinear time history analysis of a coupled core wall system to be used in a residential building in downtown Los Angeles, California. The building is comprised of 17 stories: two basement levels, one podium level, and 14 typical tower floors see Figure 1. The core wall is the primary lateral force resisting system of the structure and extended from the top story to the foundation. The core wall was considered fixed at the ground floor and the basement levels was not modeled. Opensees was used as the primary analysis software while MATLAB was used to post-process the results. A cyclic quasi-static pushover analysis was first performed in the North-South and then the East-West directions. A dynamic time history analysis was performed on the building by applying the Northridge earthquake motions in the EW direction as well as a bi-directional dynamic time history analysis by simultaneously applying the Northridge NS and EW earthquake records. Comparisons were made between different material modeling assumptions and types of elements for the quasi-static analyses and comparison were made between different material models, assumed damping algorithms, and earthquake record's time steps for the dynamic analyses.



(a) Elevation View *(b) 3D View*

Figure 1 Elevation and Plan View of Core Wall System

Model Descriptions

The real coupled core wall system is comprised of two c-shaped walls connected with a diagonally reinforced coupling beam. The preliminary design of the core wall is shown in Figure 2 and the preliminary design for the coupling beam is shown in Figure 4. The core wall was modeled using one force-based beam-column element per level, each with 5 typical section discretization as shown in Figure 3. Dimensions for section discretization of the core wall and the fiber counts are shown in Figure 3 and Table 1 respectively. The coupling beam was modeled as a truss element with cross sectional areas of the concrete and the steel found through the equivalent truss method. The concrete fiber of the coupling beam had an area of $0.362 \cdot b \cdot d$ where b and d are the width and height of the coupling beam, and the steel fiber had the combined area of 12 #9 rebars. One fictious beam column element was used to model the behavior of the diaphragm which connected the two core wall elements. This was to prevent spurious disassociations between the two walls such as twisting or displacing separately. Rigid beams were used to connect the core wall beam-column elements to the coupling beam elements in order to account for the geometric effect of the system.

Table 1 Section Discretization for Core Wall

Core Wall Discretization	Number of Layers Y Direction	No. of Layers X Direction	Material
Main Boundary Element (typ.)	6	10	Confined Concrete
Horizontal Cover of Boundary Element (typ.)	5	30	Unconfined Concrete
Vertical Cover of Boundary Element (typ.)	60	5	Unconfined Concrete
Core Wall NS Web	2	5	Unconfined Concrete
Core Wall EW Webs (typ.)	5	2	Unconfined Concrete
Steel Reinforcement in EW Webs (typ.)	4	18	Reinforcing Steel
Steel Reinforcement in NS Web	4	32	Reinforcing Steel

Table 2 Section Discretization for Coupling Beam

Coupling Beam Discretization	Number of Layers Y Direction	No. of Layers X Direction	Material
Concrete	1	1	Coupling Beam Concrete
Steel Reinforcement	1	1	Reinforcing Steel

In the model, different material constitutive models were used for each element section and section fiber. The material used for each section fiber were listed above with the properties of concrete are shown in Table 3 and material properties of the steel are shown in Table 4. In Opensees, the steel rebar used the STEEL02 constitutive model while the all the concrete used Concrete02, both of which

Table 3 Opensees' Concrete Model Parameters for Core Wall and Coupling Beam Concrete

Opensees Parameters	Units	Unconfined Concrete		Confined Concrete		Coupling Beam	
$fpc	[ksi]	f_c'	-9.1	f_{cc}'	-12.1	f_c'	$10 * f_{cr}'$
$E	[ksi]	E_c	$57\sqrt{f_c'}$	E_c	$57\sqrt{f_{cc}'}$	E_c	$57\sqrt{f_c'}$
$epsc0	[-]	ϵ'_c	$2f_c'/E_c$	ϵ'_{cc}	$2f_{cc}'/E_{cc}$	ϵ'_c	$2f_c'/E_c$
$ft	[ksi]	f_{cr}'	0.236	f_{cr}'	0.236	f_{cr}'	0.236
$lambda	[-]	λ	0.25	λ	0.25	λ	0.25
$Ets	[ksi]	Et	$0.1E_c$	Et	$0.1E_c$	Et	$0.1E_c$
$fpcU	[ksi]	f_{cu}'	$0.2f_c'$	f_{ccu}'	$0.85f_{cc}'$	f_{cu}'	$0.2f_c'$
$epsU	[-]	ϵ_{cu}'	-0.006	ϵ_{ccu}'	-0.015	ϵ_{cu}'	-0.006

Table 4 Opensees' Steel Models Parameters for Core Wall and Coupling Beam Steel Rebar

Opensees Parameters	Units	Reinforcing Steel	
$Fy	[ksi]	F_y	455
$E	[ksi]	E_s	215,000
$b	[-]	b	0.01
$R0	[-]	R_0	20
$cR1	[-]	cR_1	0.925
$cR2	[-]	cR_2	0.15
$a1	[-]	a_1	0
$a2	[-]	a_2	1
$a3	[-]	a_3	0
$a4	[-]	a_4	1
$sigInit	[ksi]	-	0

For each floor, it was assumed that the superimposed dead load was 20 psf and the weight of the façade was 10 psf. The self-weight of reinforced concrete was considered 150 pcf and was included in the model. The inertia properties of the system are derived as 87,727.8 lbf-s^2/ft for the first floor at the podium level, 41,734.8 lbf-s^2/ft at stories 1 through 13, and 33,734.8 lbf-s^2/ft at story 14. Although not previously shown, the building is symmetrical about the NS axis, but asymmetrical about the EW axis. Thus, the mass of the system was redistributed into two nodes accordingly, 65% in a center node and 35% in a south node, located at the nodes at the base of every floor.

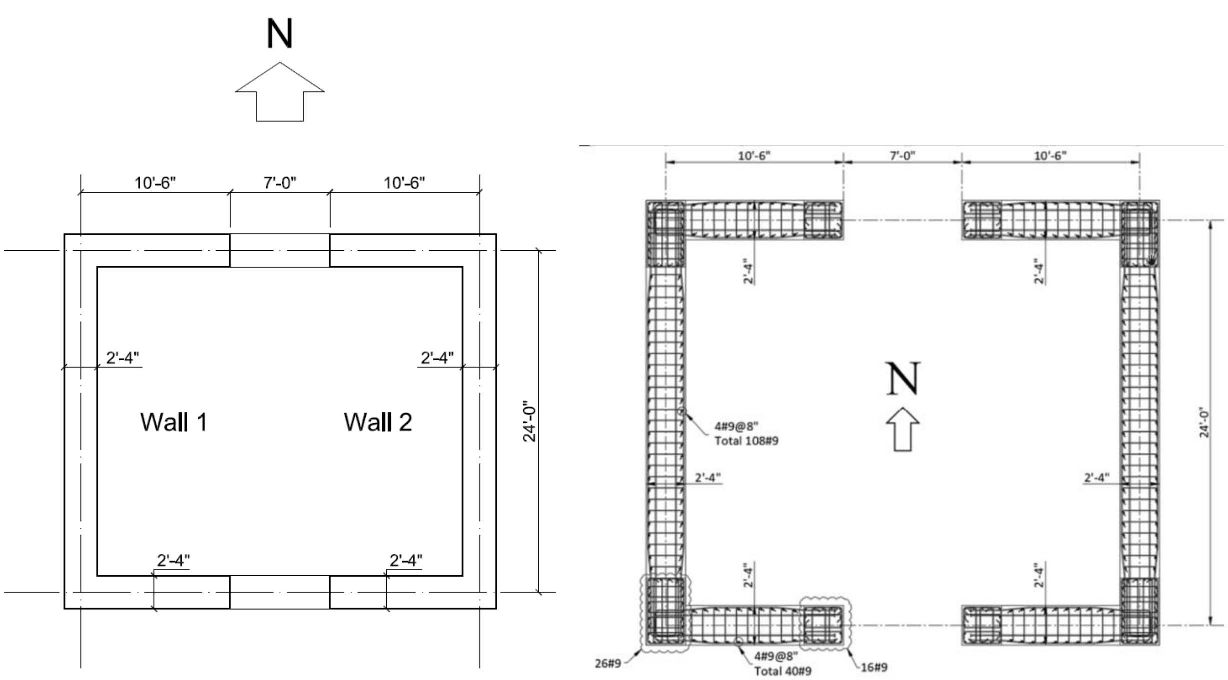
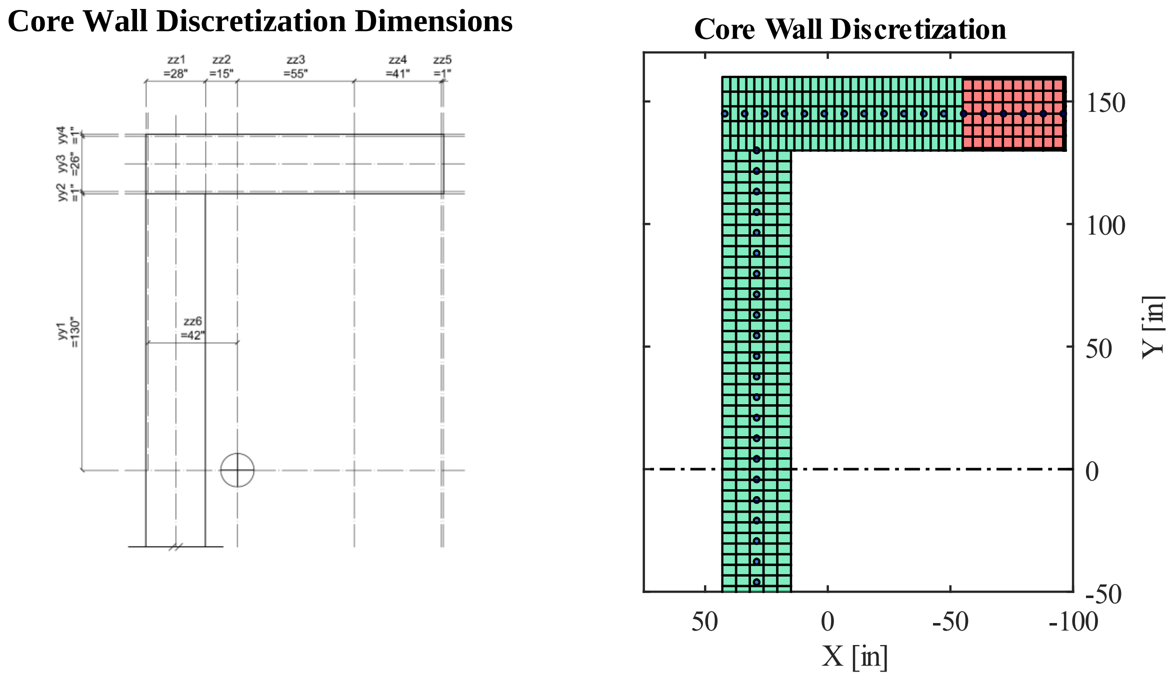


Figure 2 Preliminary Reinforcement Design of Core Walls



(a) Dimensions *(b) Fiber Section*

Figure 3 Discretization of Core Wall Concrete and Steel Rebar

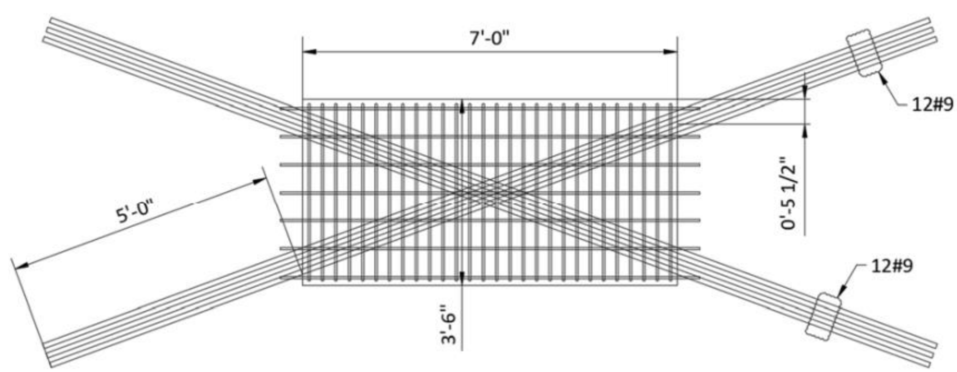
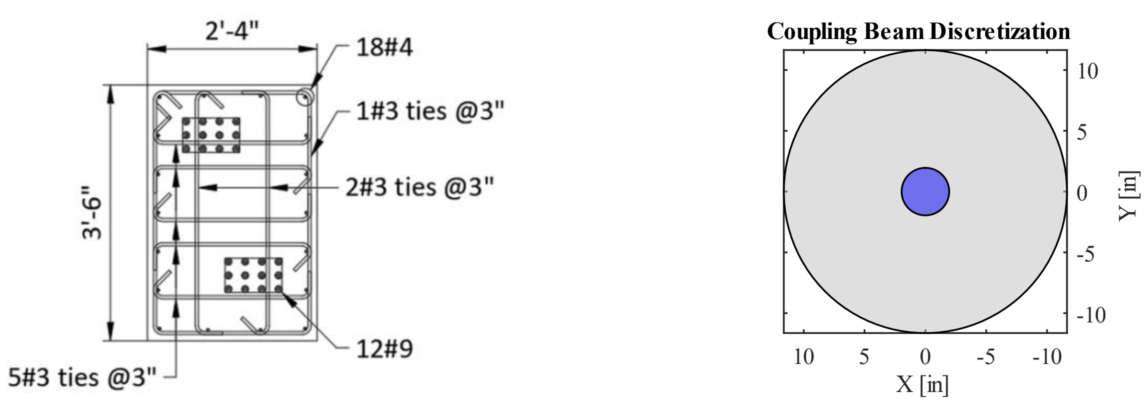


Figure 4 Preliminary Design of Coupling Beam



(a) Dimensions *(b) Fiber Section*

Figure 5 Coupling Beam Discretization

Nonlinear Cyclic Pushover Analysis

In this portion two quasi-static cyclic pushover analysis were performed, one in the EW direction and the other in the NS direction. The beam-column elements used to model the core wall used the force-based Bernoulli-Euler beam elements as described previously. Plots of the base shear and the moment curvature at the base of the core wall is presented as well as analysis of the axial strain versus curvature. The shear deformation and shear force developed by the coupling beam is also shown for EW direction analysis where the beams are actually engaged. A profile of the horizontal displacement along the height of the building is shown along with the axial and moment diagrams developed at the ends of the beam-column element as well as the Gauss Lobatto integration points. These results were compared with displacement-based Bernoulli-Euler beam elements as well as the same force-based elements but using linear elastic materials properties.

In the analysis, the dead loads mentioned previously are applied first (the yellow arrows shown in Figure 6) through a load-controlled solver. This method solved the nonlinear system by assuming that at every load step, the next converged load is always set and remains constant while the solver iterates to find the equilibrium of the fiber, section, and element resisting forces. Next, a cyclic quasi-static pushover analysis is performed using a displacement-based solver. This algorithm uses a preselected node as a control node and iterates to find the equilibrium of fiber section and element resisting forces in order to match the prescribed displacement of the control node. The control node was chosen as the top node of wall 2 for analysis in both directions. For the displacement controlled analysis, the displacements were set as drift ratios, given as $[0.0, 0.5, 0.5, 1.0, 1.0, 2.0, 2.0, 2.5, 2, 0.0](\%)$ in the direction of interest.

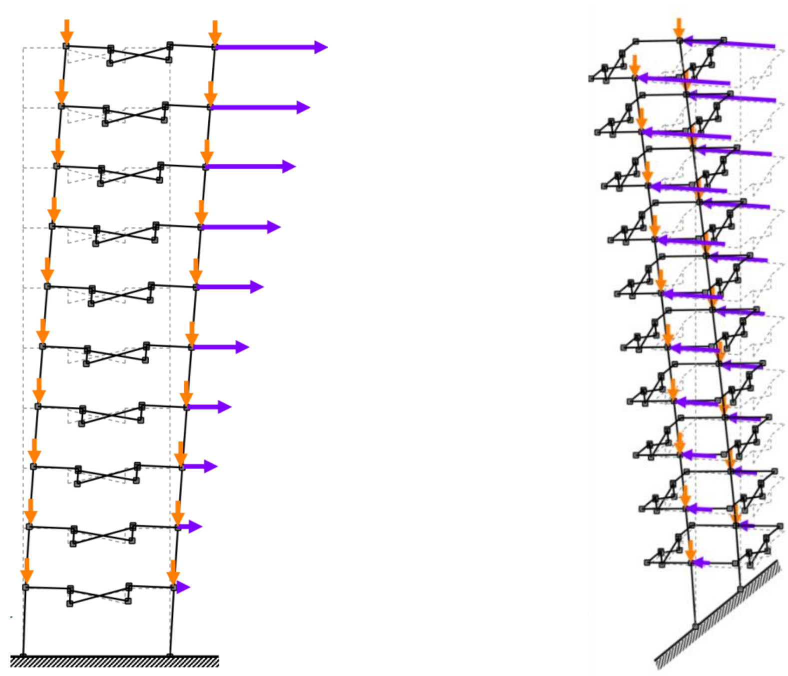


Figure 6 Sketch of the Application of the Cyclic Quasi-Static Pushover Analysis

Quasi-Static Cyclic Pushover Analysis in the EW Direction

Base Shear

Figure 7 shows the cyclic pushover curve as given by the base shear vs. roof drift ratio for both wall 1 and wall 2 as well as the total base shear of the system. The was measured as the horizontal reaction at the base of both walls in the EW direction. Positive displacement as well as positive base shear is defined as towards the east direction. In the case of positive roof drift, wall 2 is the leading wall and wall 1 is the trailing wall. The leading wall will always have a larger base shear while the trailing wall has a smaller base shear. This phenomenon is apparent even when the load is reversed as shown. When the two walls are displaced towards the east, the leading wall (in blue), will take on more of the shear force since it is having the control node where the displacement is enforced. Since there is greater displacement, the forces at the base of the wall will be larger to resist the external load. The trailing wall will receive less of the force since the coupling beams do not transfer the displacement completely, while the beams are also deforming themselves in the process of transferring the load. Since the coupling of the two walls are not perfect, the response of both walls will differ slightly as the transfer of displacement and force is equally shared between the two walls.

Since the leading wall is also experiencing higher forces, the constitutive models that were chosen for the section fibers will also dictate the behavior of the wall. It is apparent in the overall shape of the base shear curve that yielding and thus strain hardening has occurred in large parts of the leading wall. Small bumps show strength degradation caused by concrete crushing as well steel yielding to carry the new load. Pinching in the pushover curve also denote the presence of cracks in the wall opening and closing, which would affect the shape of the curve during displacement reversals events. The base shear developed by the trailing wall as the displacement reverses is less than that developed by the leading in the opposite direction due to the presence of internal strains in the wall due to the first loading. When the displacement reverses, the strains in the walls need to be unloaded before strain in the opposite direction can be applied. The maximum total base shear of developed by the walls was 4992.3 kips in the positive drift ratio of 2.5% and a negative maximum of 4831.0 kips with the negative maximum drift ratio of -2.5%.

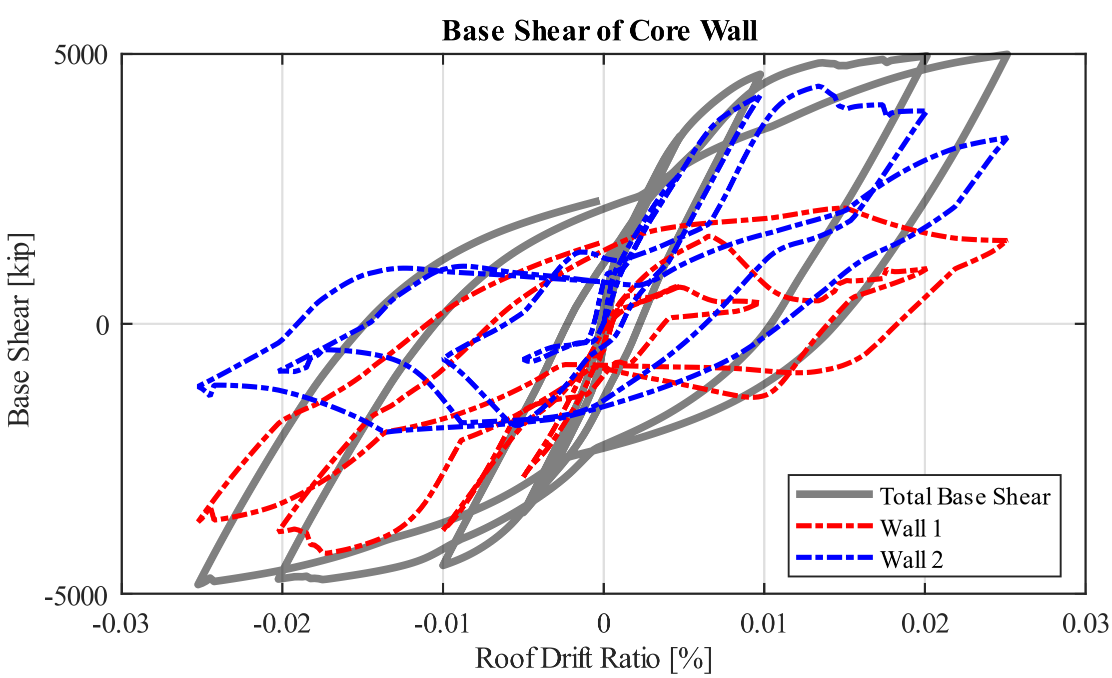


Figure 7 Base Shear vs Roof Drift Ratio in the EW Direction Under Quasi-Static Loading

Moment Curvature

Figure 9 shows the moment vs. curvature response of the first core wall section of wall 1 and wall 2 in the EW quasi-static loading case. The section moment curvature is found by taking the moment of the forces developed by the individual fibers of the section about the centroid of the section. The curve indicates that there the section is strongly dictated by the ductile behavior of steel as the Bauschinger effect is readily apparent. The slight drops in moment developed by the section indicate where the concrete has crushed but there is steel rebar that will take on the strain and the strength will increase slightly. Pinching is also evident in the moment curvature which indicates that the bottom section has cracked. The point where the structure reaches maximum drift is also shown which indicates that when the maximum curvature is reached by the leading wall, the trailing wall experiences only a fraction of the curvature.

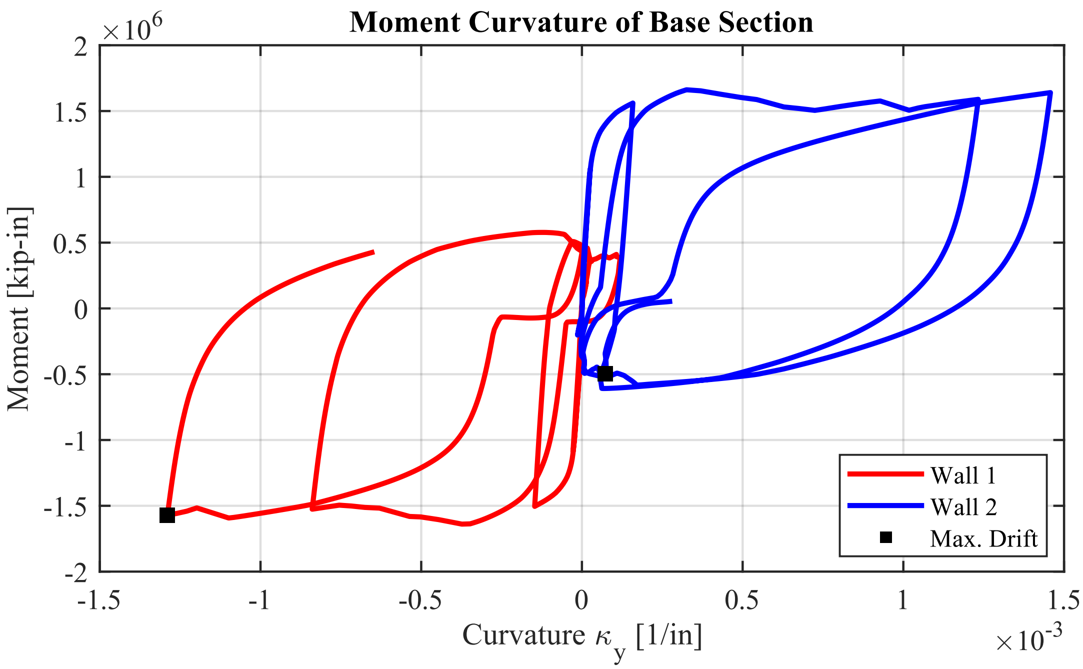


Figure 8 Moment Curvature Response of Wall 1 and Wall 2 in the EW Direction

Axial Strain versus Curvature Response

Figure 9 shows the axial stain at the section centroid versus the curvature response of the base section of wall 1 and wall 2 when it undergoes cyclic quasi-static loading in the EW direction.

The centroidal strain of both walls are seen to be increasing due to residual axial strain after displacement reversals. When cyclic curvature is applied, plastic hinge elongation is observed. This phenomenon is characterized by the increase of permanent residual axial strain and sectional plastic deformation.

The curvature-axial curve of each wall moves further and further away from the origin over the course of the pushover analysis. This is because the walls have developed permanent plastic deformations (rotations), where wall 1 rotates more and more towards the west and wall 2 rotates towards the east. The rotation between the two walls is further increase by the elongation of the coupling beams with each cycle, which also undergoes plastic deformations.

The shape of the axial-curvature curve is slightly different between the two walls due to the difference in applied displacements. While wall 2 is sometimes the trailing wall, it always has the control node during the displacement controlled iteration, meaning the forces developed by it will always be higher.

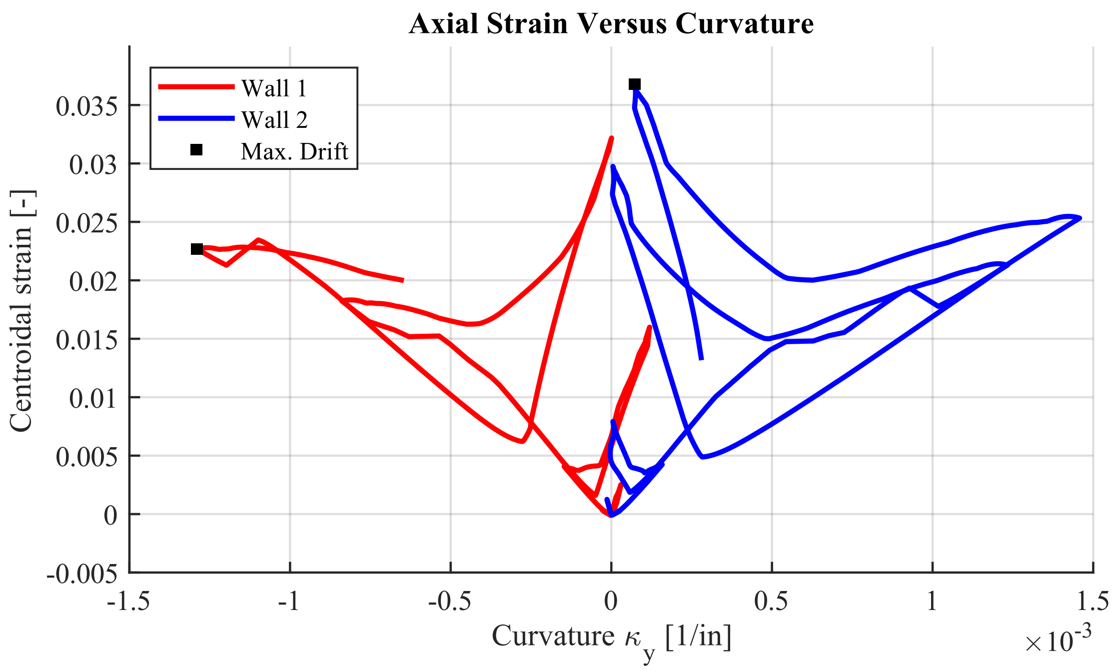


Figure 9 Axial Strain vs Curvature Curve of Wall 1 and Wall 2 in the EW Direction

Figure 10 shows the strain profile of wall 1 and wall 2 at the maximum roof drift of +2.5%. The load is applied in the east direction, making wall 2 the leading wall and wall 1 the trailing wall. For wall, the neutral axis is 25.6in away from the extreme compression fiber, located at east side of the wall. For Wall 1, the entire section is seen to be in tension. It is observed that the strain from wall 2 continues to wall 1, transferred by the coupling beams, although not perfectly since the magnitudes are greatly different. This shows the shift in neutral axis while the wall undergoes positive drift is significant especially when considering the trailing wall. This demonstrates that walls can undergo tension during extreme lateral loading and how earthquake motions might induce such a scenario.

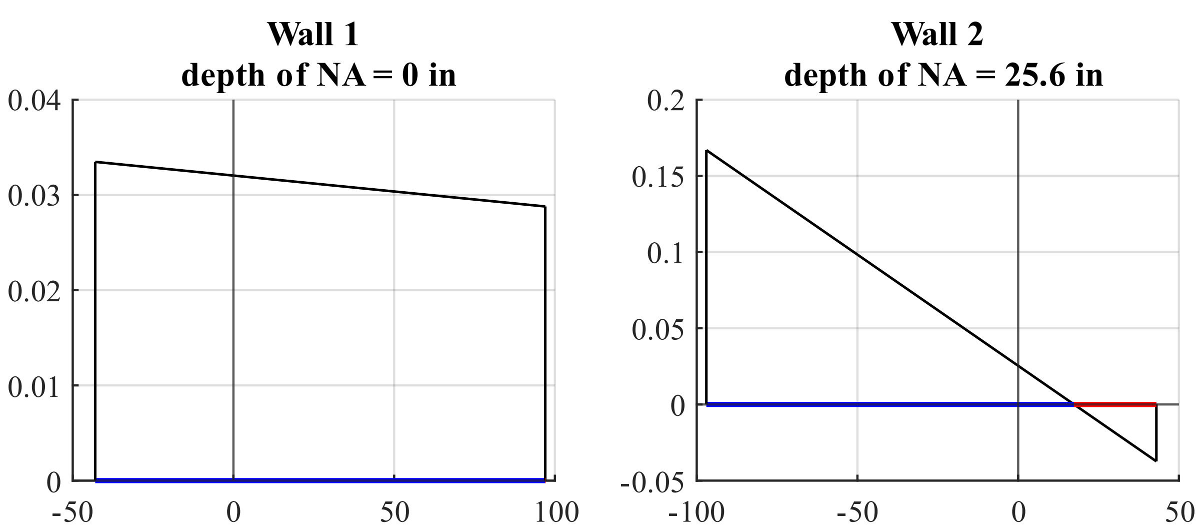


Figure 10 Location of the Neutral Axis

Shear Force and Shear Deformation of Coupling Beams

Figure 12 shows the shear force vs. shear deformation response of the coupling beams at the top of the 1st, 8th, and 15th story.

The coupling beam at the first story develops the most shear capacity. This is because the lateral constraint of the pier walls is the strongest at the first floor and weakest at the top floor (i.e., the stiffness coefficient of the wall is larger at the bottom). When the walls are pushed to the side the steel experiences tension and yields, creating permanent deformation. However, after a certain amount deformation, the walls prevent further elongation and compressive axial force is developed in the coupling beams due to the pier wall's confinement. The coupling beam at the bottom floor with the most constraint develops the most compressive force and the coupling beam at the top floor, which is freer to elongate, develops the least compressive force. Also, lower floor, having more compression has less crack, therefore there is less degradation of concrete strength.

The shear deformation is calculated as $\gamma = \frac{\delta_e - \delta_s}{2\, l_t}\left(tan\theta + \frac{1}{tan\theta}\right)$, where l_t is the undeformed length of each truss element, and δ_e and δ_s are the elongation and shortening of the equivalent truss elements as shown in Figure 11. Note that δ_e and δ_s represents algebraic values, therefore $\delta_e > 0$ and $\delta_s < 0$.

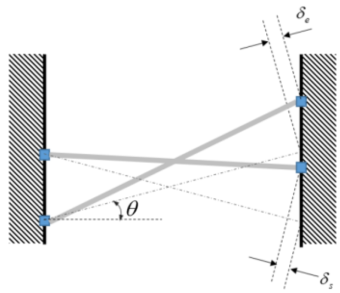


Figure 11 Shear deformation in coupling beam from equivalent truss system

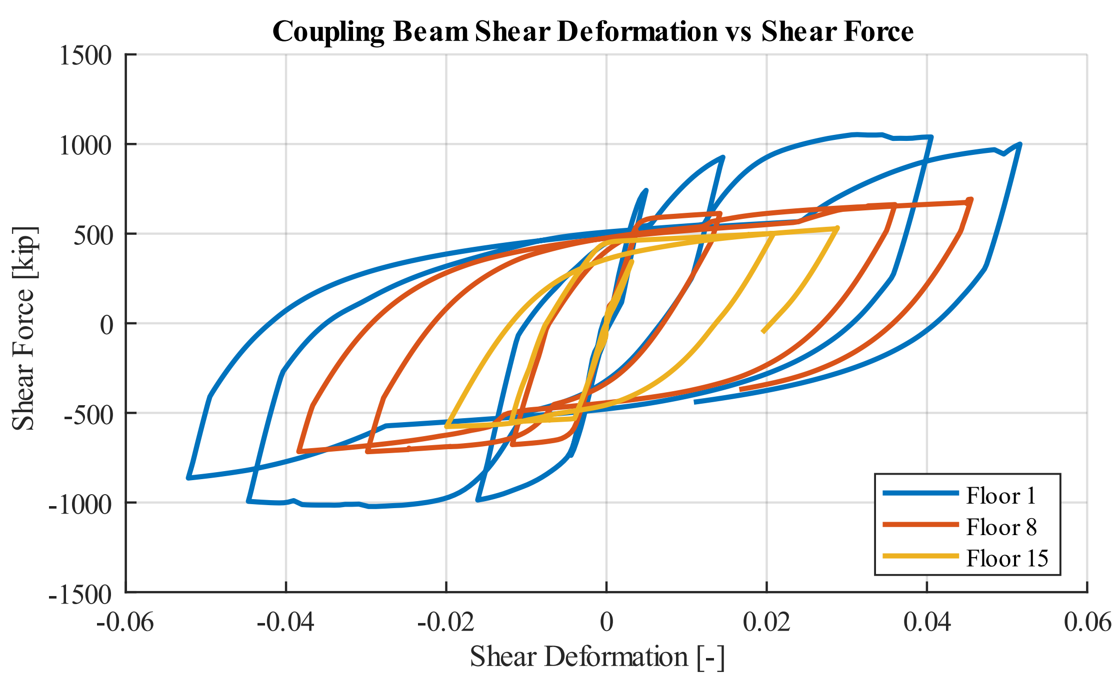


Figure 12 Shear force vs Shear Deformation Response of Coupling Beam Under Quasi-Static Loading

Horizontal Displacement Along Height of Building

Figure 13 shows the horizontal displacemnet in the EW dirextion of wall 1 and wall 2 along the height of building at the maximum positive drift (+2.5%). The maximum displacement of the roof at wall 2 is 54.37in while for wall 1 it is 53.73in. This is reasonable since the displacement is enforced at wall 2 while wall 1 is the trailing wall. The shape of the two walls are similar but not exact because the coupling beams do not enforce the exact shape from one wall to the other, especially considering how they are defomring themselves. The sharp interstory drift is not as appartent in wall 1 as in wall 2 because the control node is on wall 2.

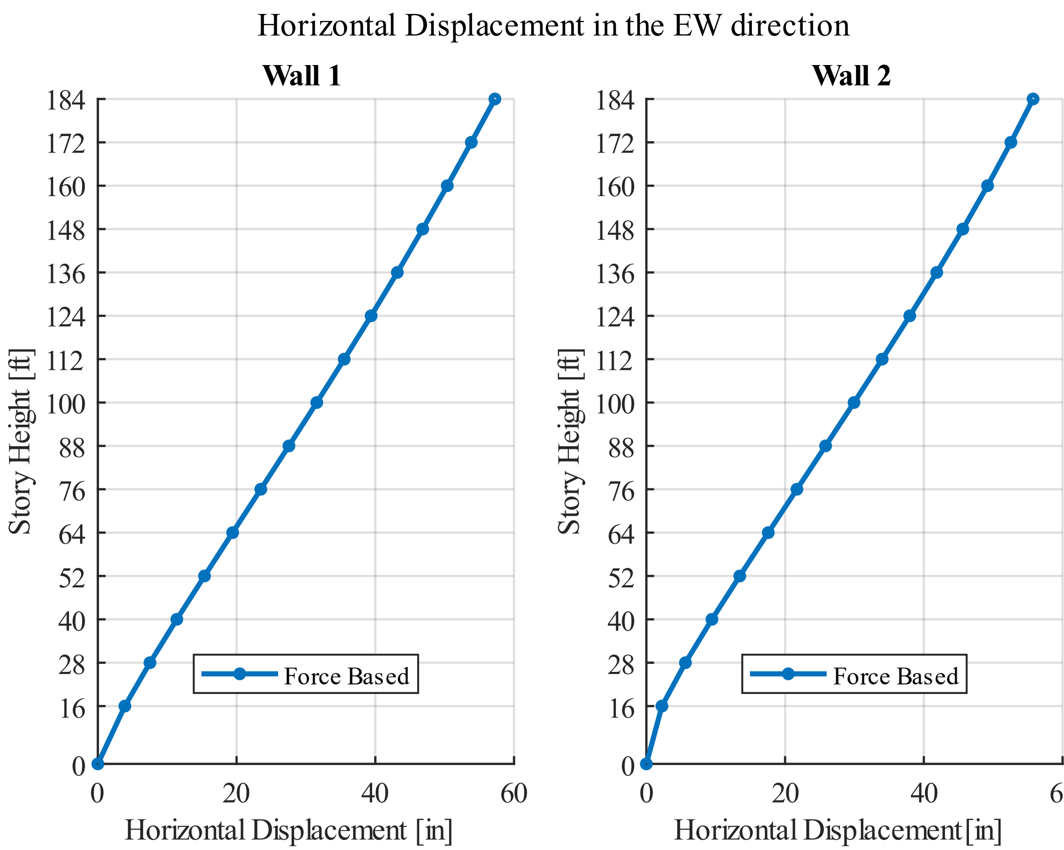


Figure 13 Horizontal Drift Along Centroid in the EW Direction

Axial and Moment Diagram Along Height of Building

Figure 14 shows the axial force and bending moment diagram of Wall 2 over the full height of the building during the maximum enforce drift (+2.5%). The blue lines represent the axial forces and bending moment developed by the element forces while the orange lines represent the section forces at all Gauss-Lobatto integration points. The element forces are the summation of the section forces using the Gauss-Lobatto approach, taking a weighted sum of each cross section. For 5 sections, these weights are given as $\left[\frac{1}{10}, \frac{49}{90}, \frac{32}{45}, \frac{49}{90}, \frac{1}{10}\right]$.

Using force-based Bernoulli-Euler beam column elements, the element end forces and section forces match well. This is because force-based formulation approximates the distribution of forces within the element which is more exact than displacement-based formulation which assumes a distribution of deformation within an element. The force interpolation functions (force shape function), $N_Q(x)$ imposes a constant field for the axial force and a linear field for the bending moment. These assumptions are always exact in beam theory and it is independent of the section material behavior. There is only secondary approximation which is the numerical integration along the element, and it can create some numerical errors, but it is not significant. Because the formulation is exact, it is sufficient to use one element per story.

The axial force accumulates as it goes down the building which is largely caused by the floor weight from each level but can also be caused by the coupling beam transferring lateral forces into axial forces. The increase of moment at every floor is caused by the additional moment produced by the coupling beam. This structure has zero moment at 64 ft, which is at the bottom of 6th floor. There is a zero moment point which indicates the point of inflection which is caused by the presence of the coupling beams. If the walls were a perfect cantilever system, the roof would deflect perfectly without any inflection point. But since the coupling of the beam do not perfectly transfer the forces between the two walls, the walls will act as a hybrid cantilever and frame system. The point of inflection is created due to coupling effect from the coupling beams and the point of inflection moves downward as the stiffness of the coupling beams increase. High coupling between the walls will lead to larger frame effect, which lowers the inflection point. Higher coupling would also reduce inter-story drift as the introduction of the inflection point reduces the displacement of each floor.

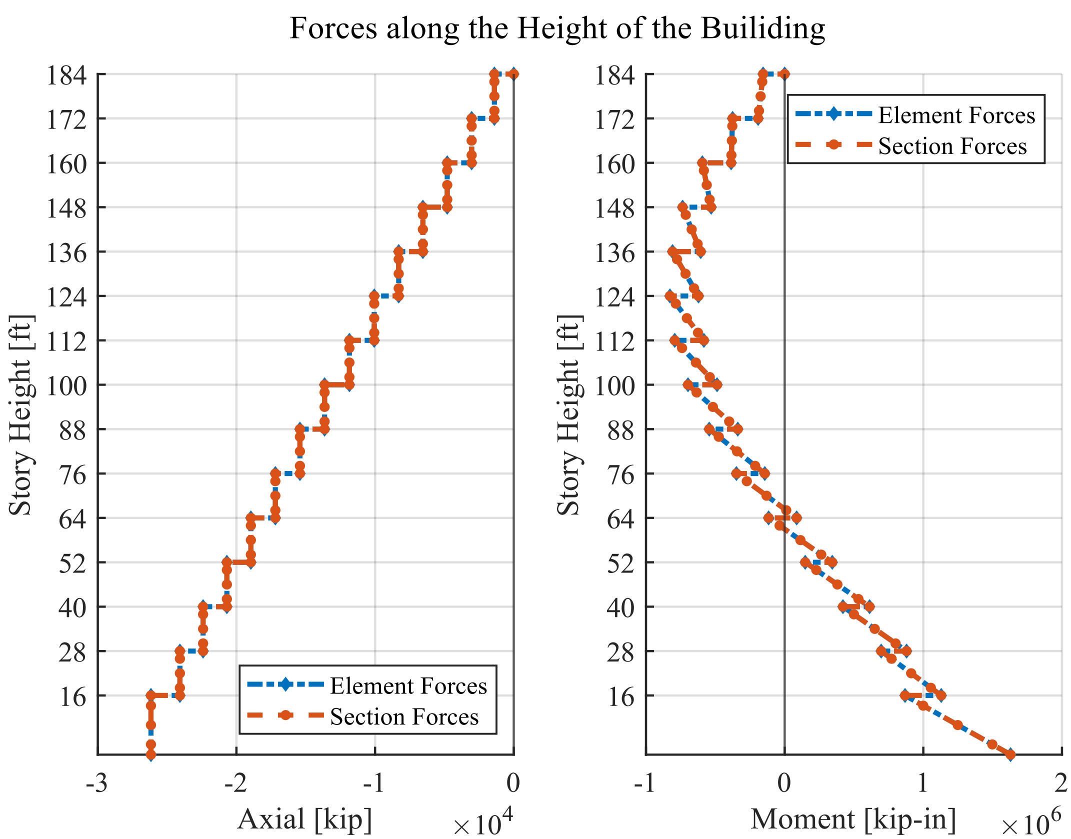


Figure 14 Axial and Moment Diagram Along Height of Building in the EW Direction

<u>Comparison with Displacement-Based Bernoulli-Euler Beam-Column Elements</u>

This section compares the use of the force-based Bernoulli-Euler beam-column element and the displacement-based Bernoulli-Euler beam-column element. Whereas the force-based element used one element per story, the displacement-based element uses 2 elements per story with 5 Gauss-Lobatto integration points each since the approximation by the displacement beam-columns lead to large discrepancies in the results.

Compare Horizontal Displacement in the EW direction

Figure 15 shows the horizontal displacemnet in the EW direction of wall 1 and wall 2 along the height of the building at the maximum positive drift imposed during the cyclic pushover analysis (+2.5%).

The maximum displacement at the roof is same for the displacement-based element and the force-based element because this was enforeced. At lower levels, the displacement-based elements deflect slightly less. The approximation of the deformation fields would lead to smaller dispalcements than those elements which have assumed force functions since displacement of the elments are enforeced rather than forces. Additonally, two displacement-basd bernoulli beam-column elements would result in a a smoother deflected shape leading to less deflection along the height of the building.

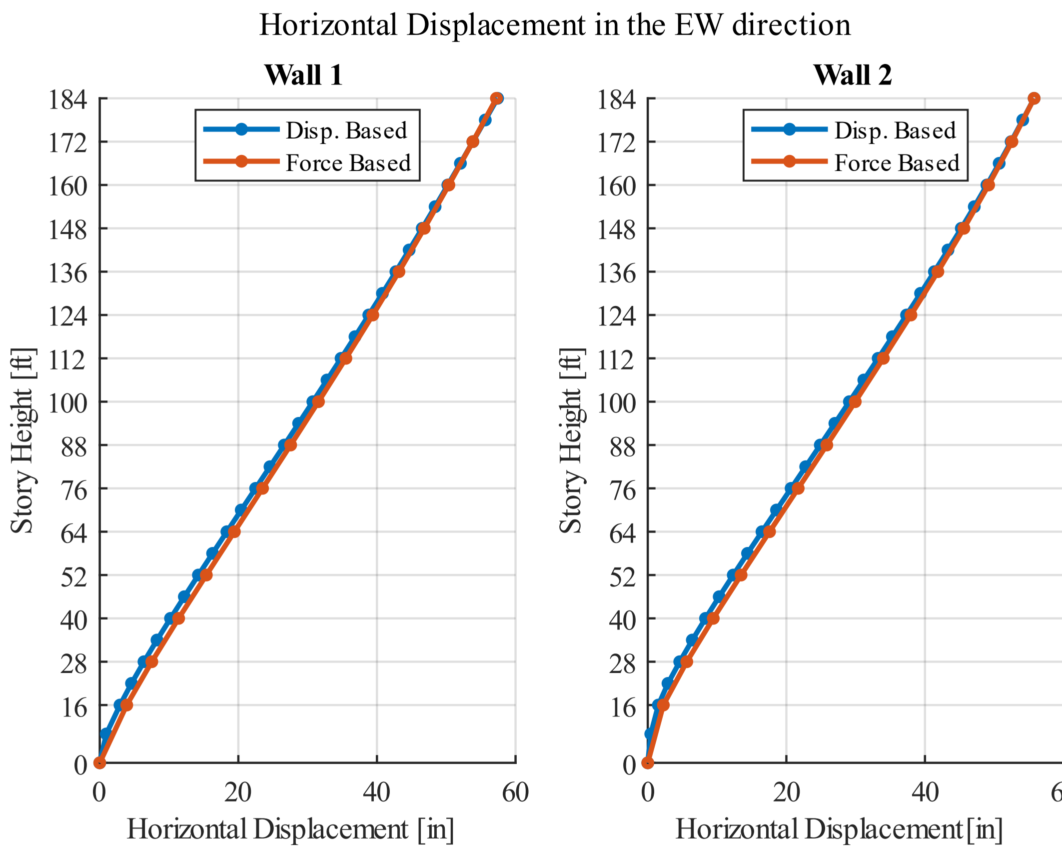


Figure 15 Comparing Horizontal Drift of Force-Based Element and Disp.-Based Elements

Compare Axial and moment diagrams

Figure 16 shows the axial force and bending moment diagram of Wall 2 over the full height of the building at maximum positive enforced drift (+2.5%). The blue lines represent the axial forces and bending moment using the element end forces, and the orange lines represent the section forces at all Gauss-Lobatto integration points only. The element end forces are the summation of the Gauss-Lobatto points given specific weights for each point.

The section forces and the element end forces calculated using the displacement-based element do not match well. The is because the displacement-based element assumes a strain-displacement transformation matrix, $B(x)$. $B(x)$ is a shape function for the section deformations, and it assumes that ε_0 is constant and κ is linear which is the primary source of approximation. This approximation between element level and section level is only true when the material is linear-elastic, and the section is prismatic. Since in the couple walls, there is nonlinear material, the section forces show very different result from the element forces.

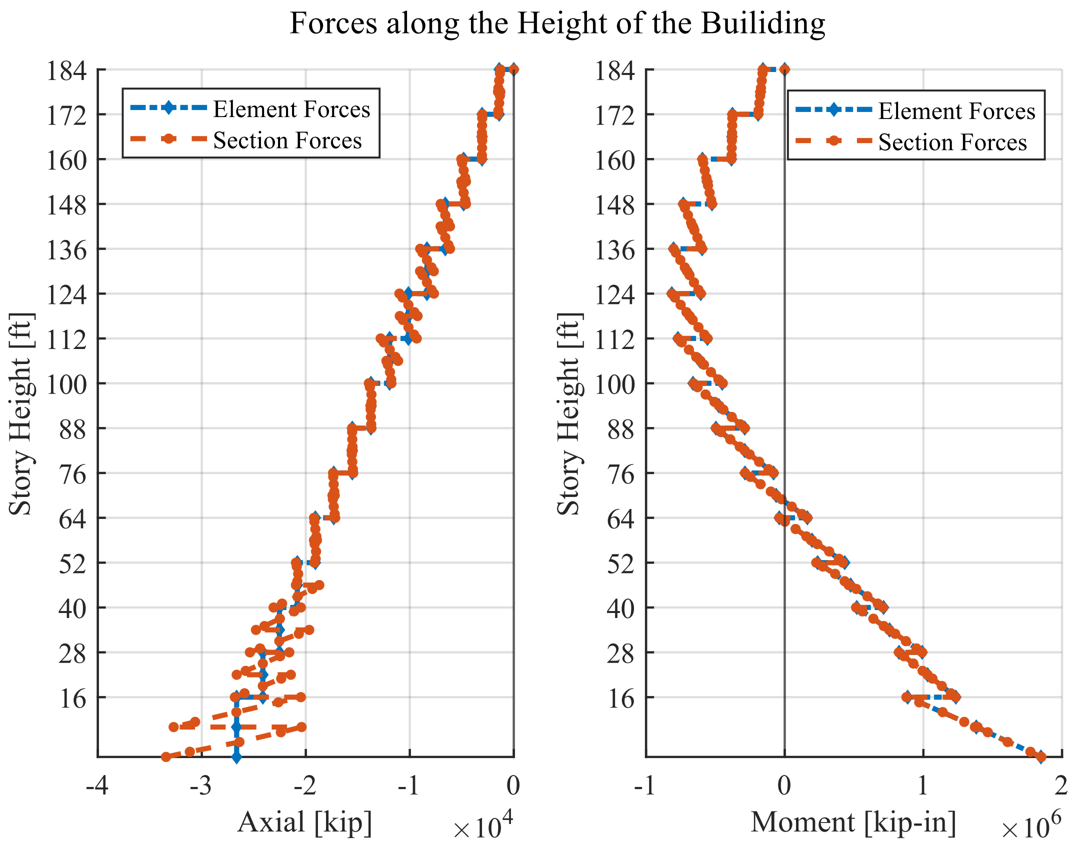


Figure 16 Comparing Axial Moment Diagram of Force-Based Element and Disp.-Based Elements

Comparison with Linear Elastic Material Models

This section compares the use of nonlinear and linear constitutive models.

Base shear of core wall

Figure 17 shows the cyclic pushover curve through the base shear vs. roof drift ratio curve. The figure presents the base shear for wall 1 and wall 2 as well as the total base shear.

The base shear calculated using linear material constitutive models gives a line instead of loops. However, we can notice that the lines for wall 1 and wall 2 is not truly linear; the base shear of wall 1 is higher at both negative and positive drift ratio. This is because of the P-Delta effect we have applied to the model. The line is curved due to geometric nonlinearity of the structure.

It can also be observed that the behavior of the linear material is similar but opposite to that of nonlinear material. It is similar in a way that one wall has greater magnitude of base shear to one side and the other wall has greater base shear on the other side.

And it is opposite in a way that, whereas the leading wall had a larger base shear than the trailing wall in nonlinear material model, the trailing wall has greater base shear than the leading wall in linear material model.

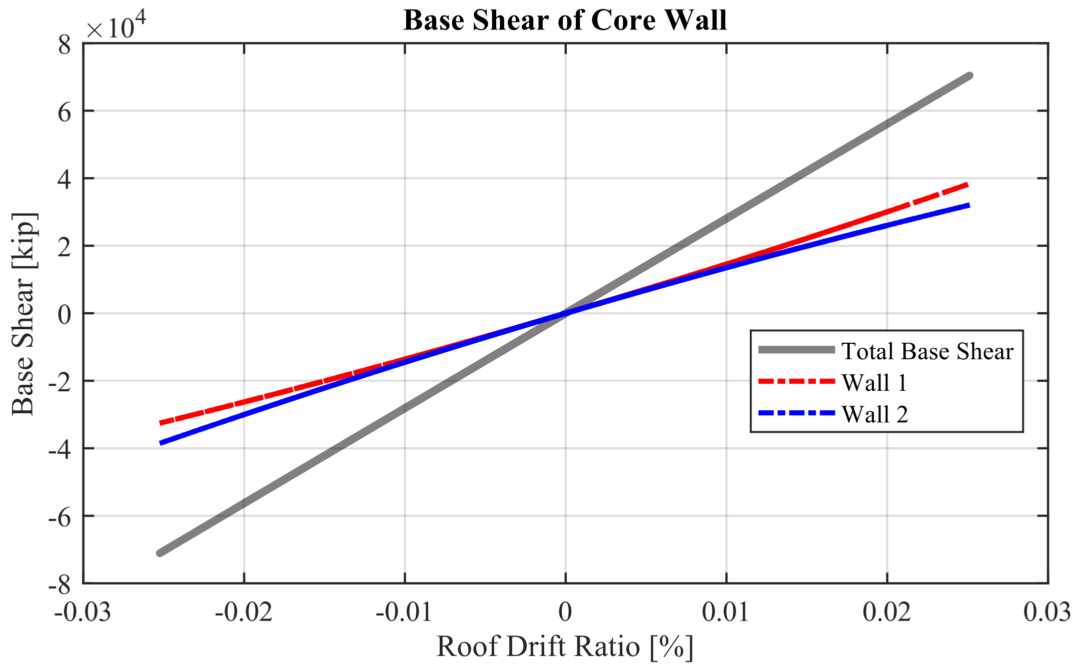


Figure 17 Comparison of Base Shear vs Roof Drift Ratio for Different Material Models

<u>Axial Strain</u>

Figure 18 shows the axial stain (at the section centroid) (on the vertical axis) versus the curvature response (along the horizontal axis) of the base section of wall 1 and wall 2. The axial curvature plot is a straight line since the curvature is always related to the strain. The intersection is not at (0,0) but at little below due to the superimposed dead load and self-weight of the structure. There is some axial strain present. There is no permanent plastic section deformation since the material is always elastic and thus, the axial curvature curve will not migrate.

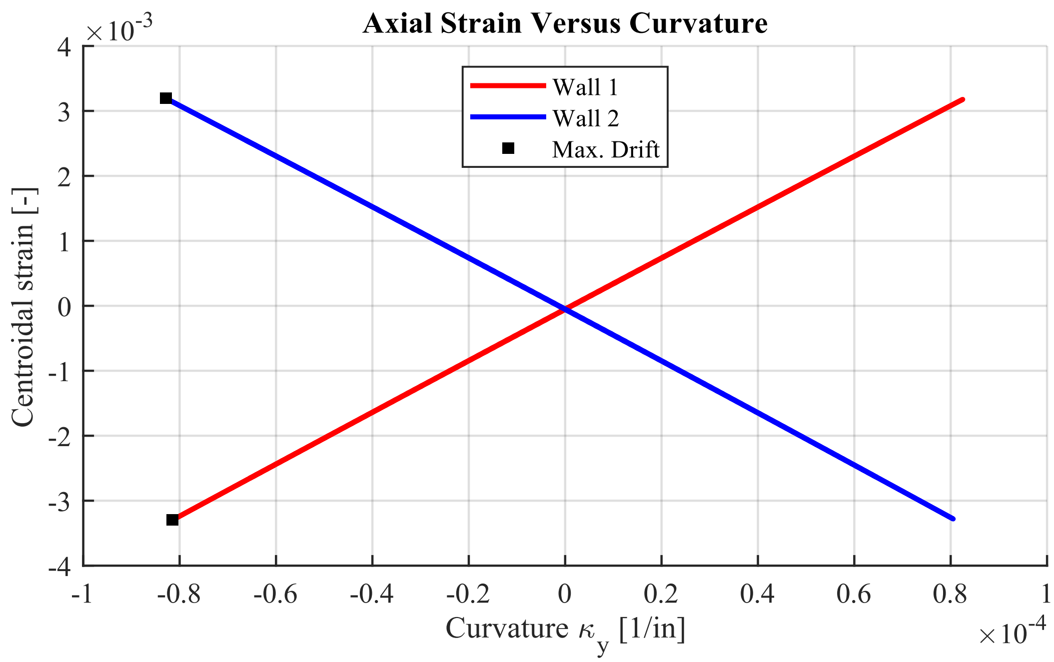


Figure 18 Comparison of Axial Strain vs Curvature for Different Material Models

<u>Horizontal displacement</u>

Figure 19 shows the horizontal displacemnet in the EW dirextion of wall 1 and wall 2 along the height of at the maximum positive drift imposed during the cyclic pushover analysis (roof drift ratio = 2.5%).

In both plots, the beam-columns elements for the walls are force-based elements. The deflected shape of the structure using nonlinear material has a point of inflection as well as the linear material model. The elastic material model deflects less because the materials does not undergo hardening or softening, but remain perfectly elastic.

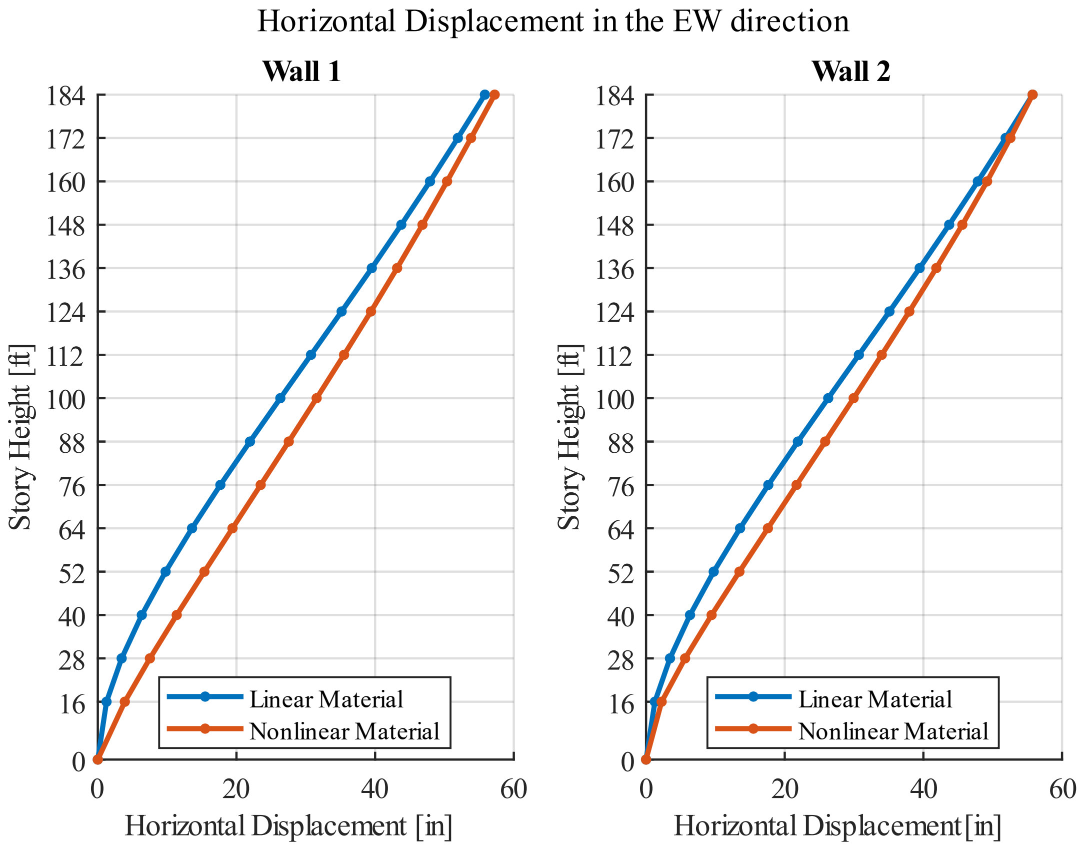


Figure 19 Comparison of Horizontal Drift for Different Material Models

<u>Element End forces for Axial and Moment</u>

The forces developed by the linear model is far greater than that of the nonlinear material. This is because the material does not yield and thus continues to develop strain as the system undergoes enforced displacement. Even though the structure deflects the same amount, the choice of constitutive model, dictates the force generated by the section. Because there is no plastic deformation that is able to take away some of the enforced displacement, the moment and axial force in the linear model is far greater.

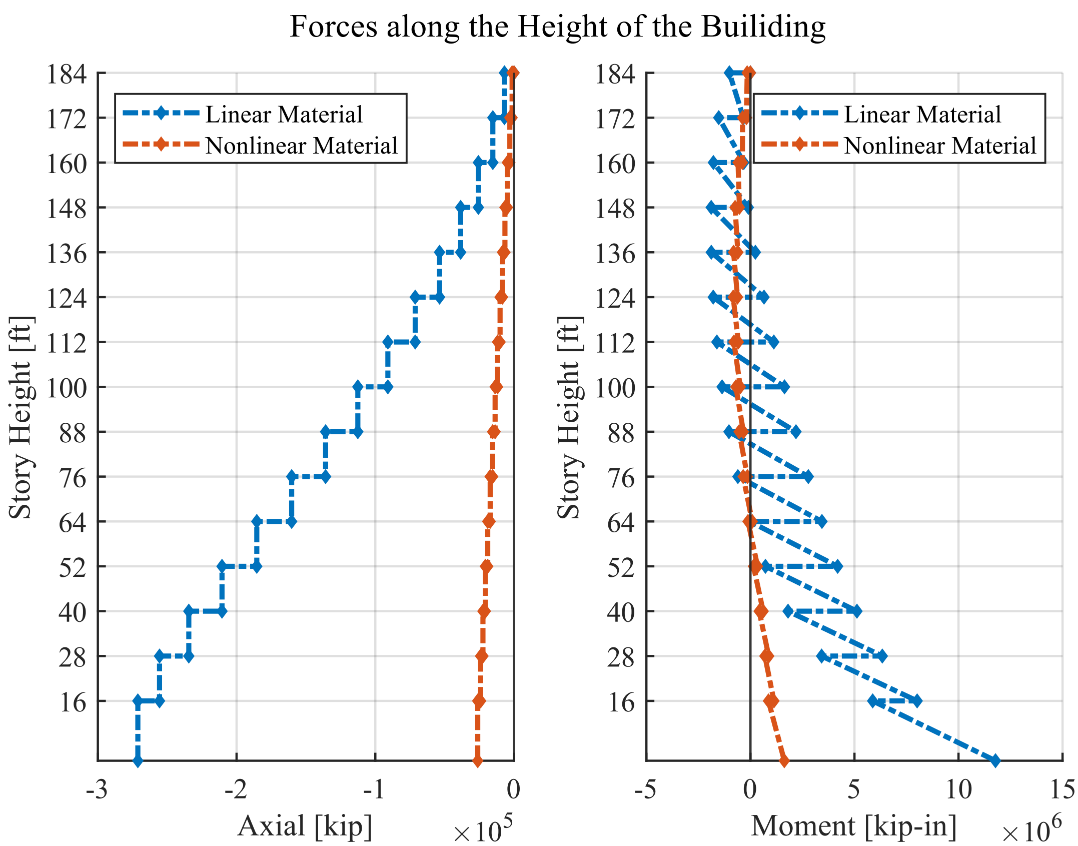


Figure 20 Comparison of Axial Moment Diagram Element End Forces

Part II Quasi-Static Cyclic Pushover Analysis in the NS Direction

<u>Base Shear</u>

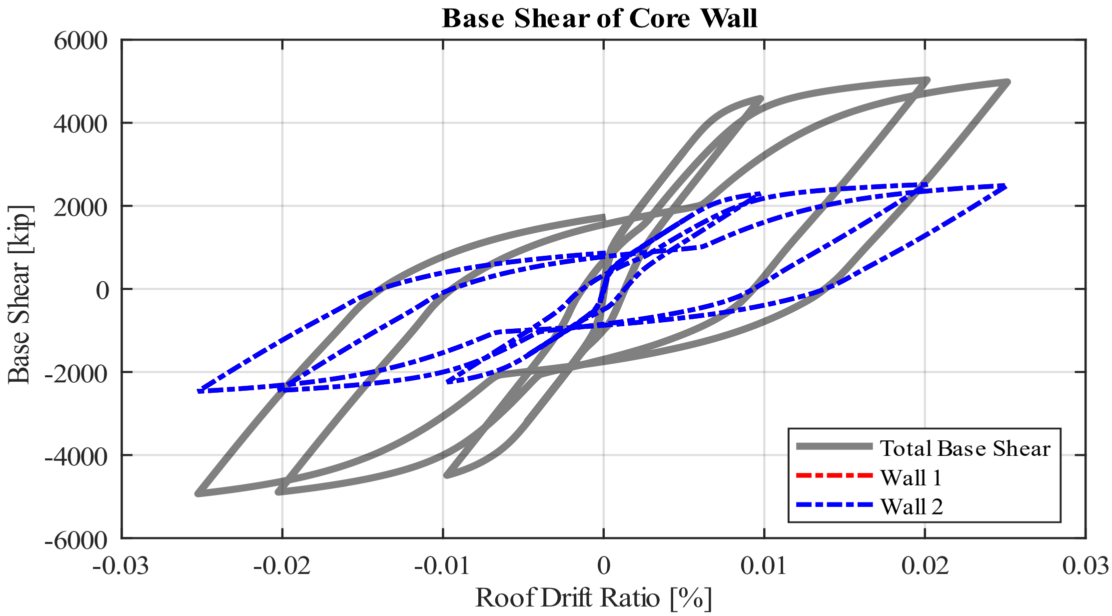


Figure 21 Base Shear vs Roof Drift Ratio in the NS Direction Under Quasi-Static Loading

The base shear in the NS direction for wall 1 and wall 2 are similar due to their similar geometries. This is the stronger axis for the couple core wall as is easily seen from the geometry but also by the pushover curve given in Figure 21. The ductile behavior of the wall system shows that less crushing occurs in the NS direction when loaded than in the EW direction.

<u>Moment Curvature</u>

The moment curvature shows again the similar behavior of wall 1 and wall 2 in the NS direction. Pinching is apparent meaning that cracks form, but the over shape of the curve demonstrates that the building is adequately designed well in the NS direction.

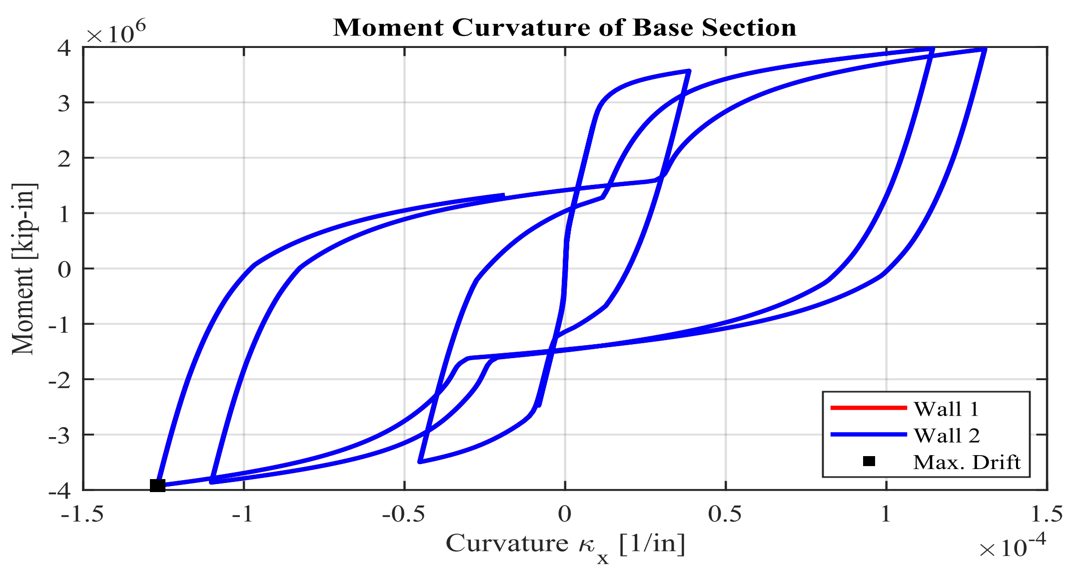


Figure 22 Moment Curvature Response of Wall 1 and Wall 2 in the NS Direction

Axial Strain versus Curvature Response

The axial strain versus curvature in the NS direction shows that the wall does not undergoes as much plastic deformation as the wall in the EW direction. Figure 23 shows that the curvature has a linear relationship with the axial strain. The neutral axis is shown in Figure 24.

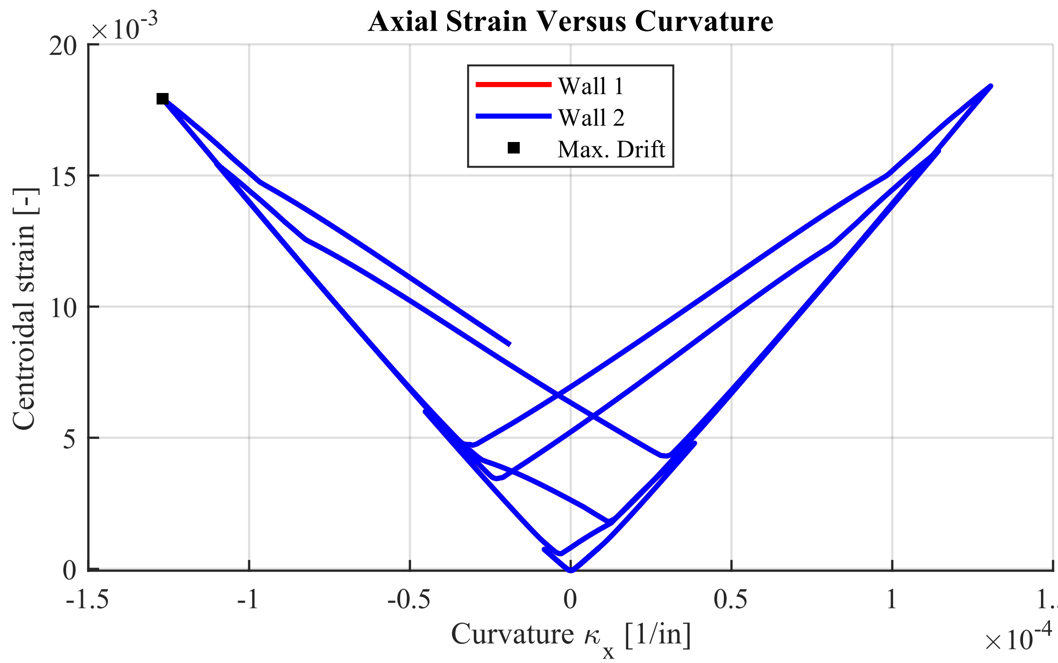


Figure 23 Axial Strain vs Curvature Curve of Wall 1 and Wall 2 in the NS Direction

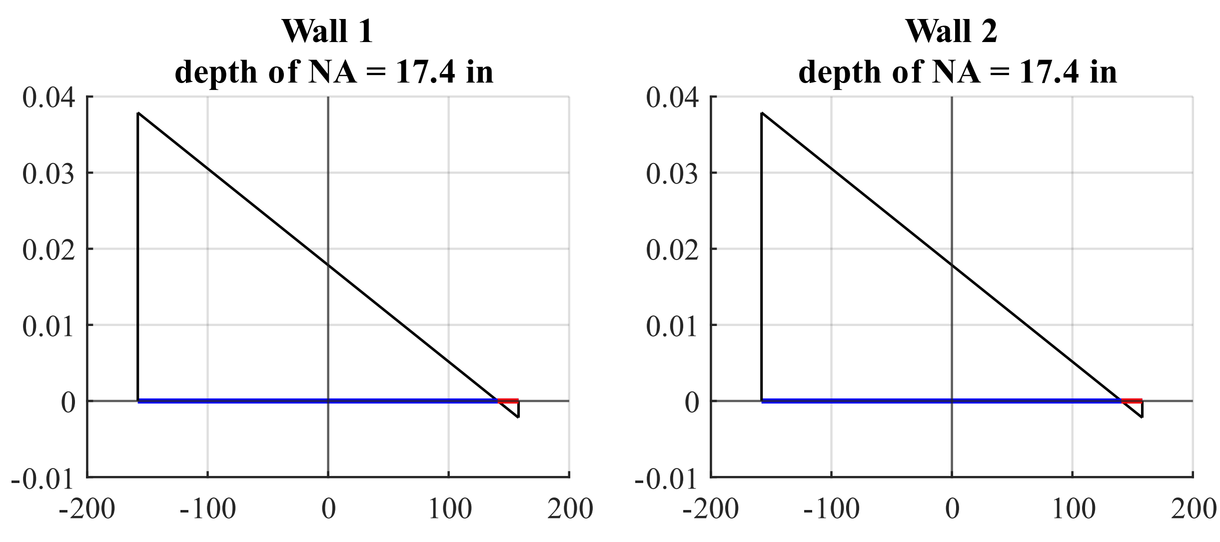


Figure 24 Location of the Neutral Axis

<u>Horizontal Displacement Along Height of Building</u>

The deflected shape between wall 1 and wall 2 are more similar as both walls are engaged more equally in the NS direction. This coupling beam is able to transfer the displacement between the two walls better with deforming. The structure is stiff in the NS direction and doesn't show any sign of inflection points. The coupling between the walls seem to indicate more a of cantilever deflection.

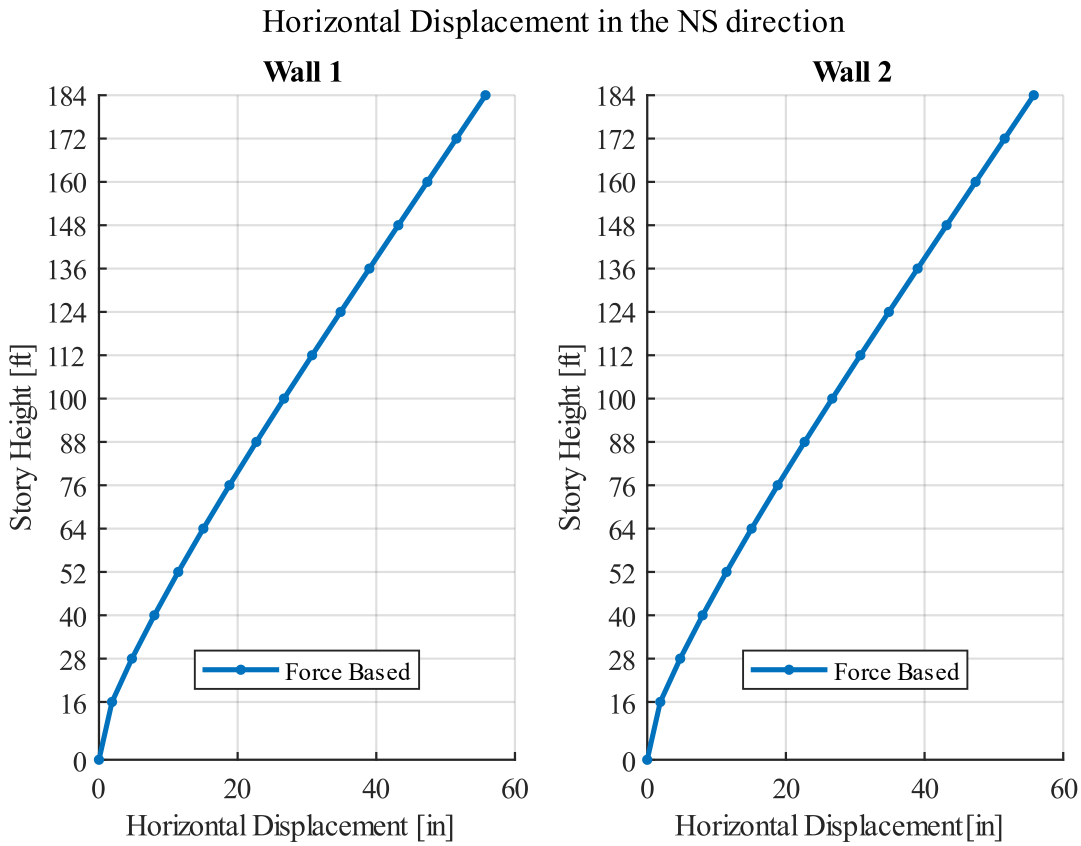


Figure 25 Horizontal Drift Along Centroid in the NS Direction

Axial and Moment Diagram Along Height of Building

The axial diagram along the height of the building is similar to that of the EW direction. Every floor contributes its self-weight, which culminates towards the bottom. The moment does not change signs which means that the wall does not have an inflection point. The moment diagram looks like a cantilever beam with small disturbances between floors due to the coupling beam transferring tiny moments between the walls.

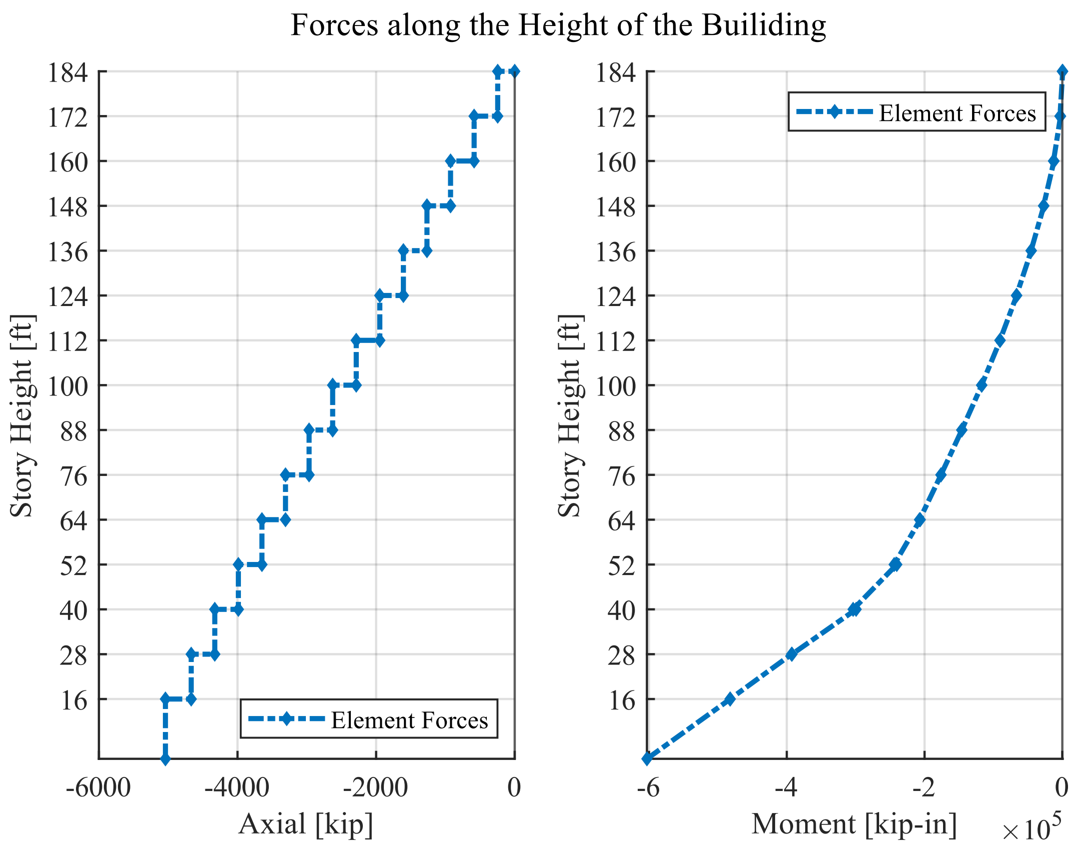


Figure 26 Axial and Moment Diagram Along Height of Building in the NS Direction

Part B: Nonlinear Time History Analysis

The dynamic analysis of the structure was performed by applying the Northridge earthquake motion in the NS and EW directions. The overturning moment at the base of the were taken and plotted against the roof drift ratios. The time history of displacement and acceleration is given for selected stories along the height of the building. The normalized base shear, to the weight of the building, is also provided. Moment curvature in the EW direction is provided for the section at the base of wall 1 and wall 2. The axial strain versus curvature was also shown. Shear deformation and shear force developed by the coupling beam is given for the podium level, story 7, and story 14. Comparisons were made for linear elastic material models as well as with earthquake records with smaller time steps. Bi-directional earthquake motion was imposed on the building with the NS and EW earthquake motions applied simultaneously.

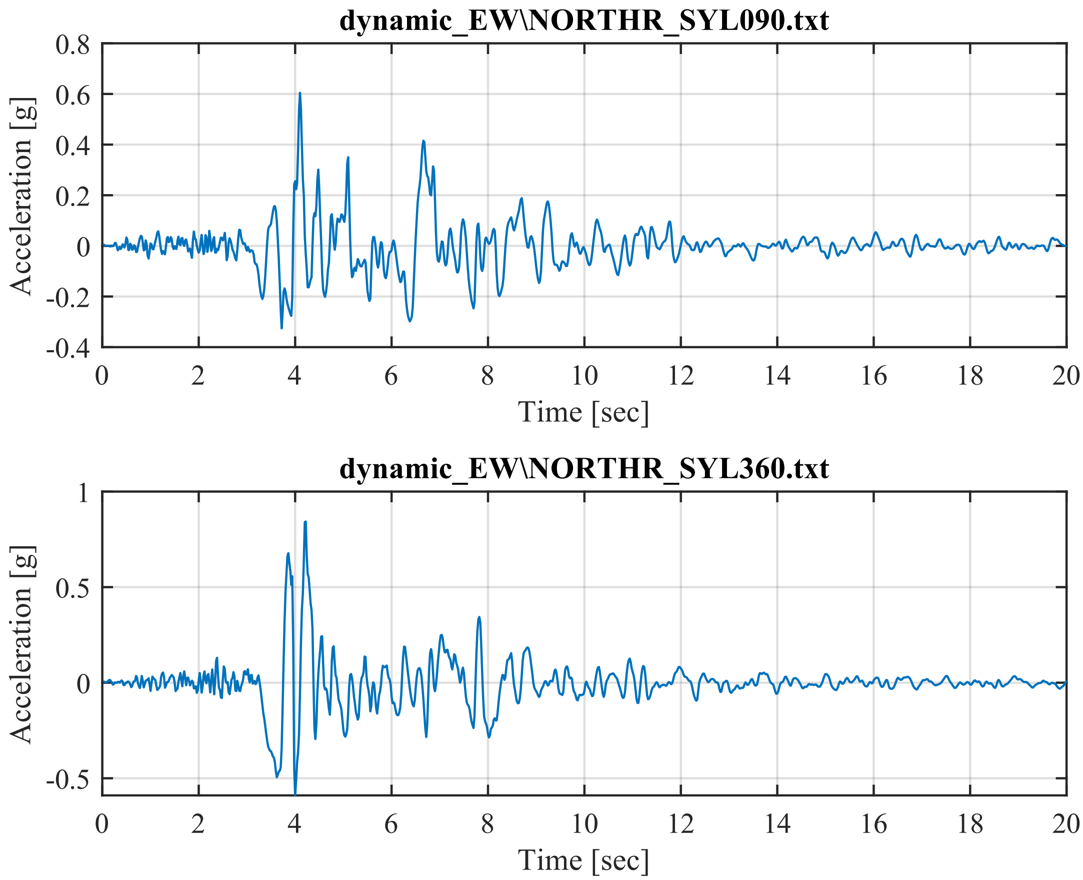


Figure 27 Earthquake Motions Imposed on the Building for Dynamic Loading

Analyzing Damping Ratio of the Core Wall

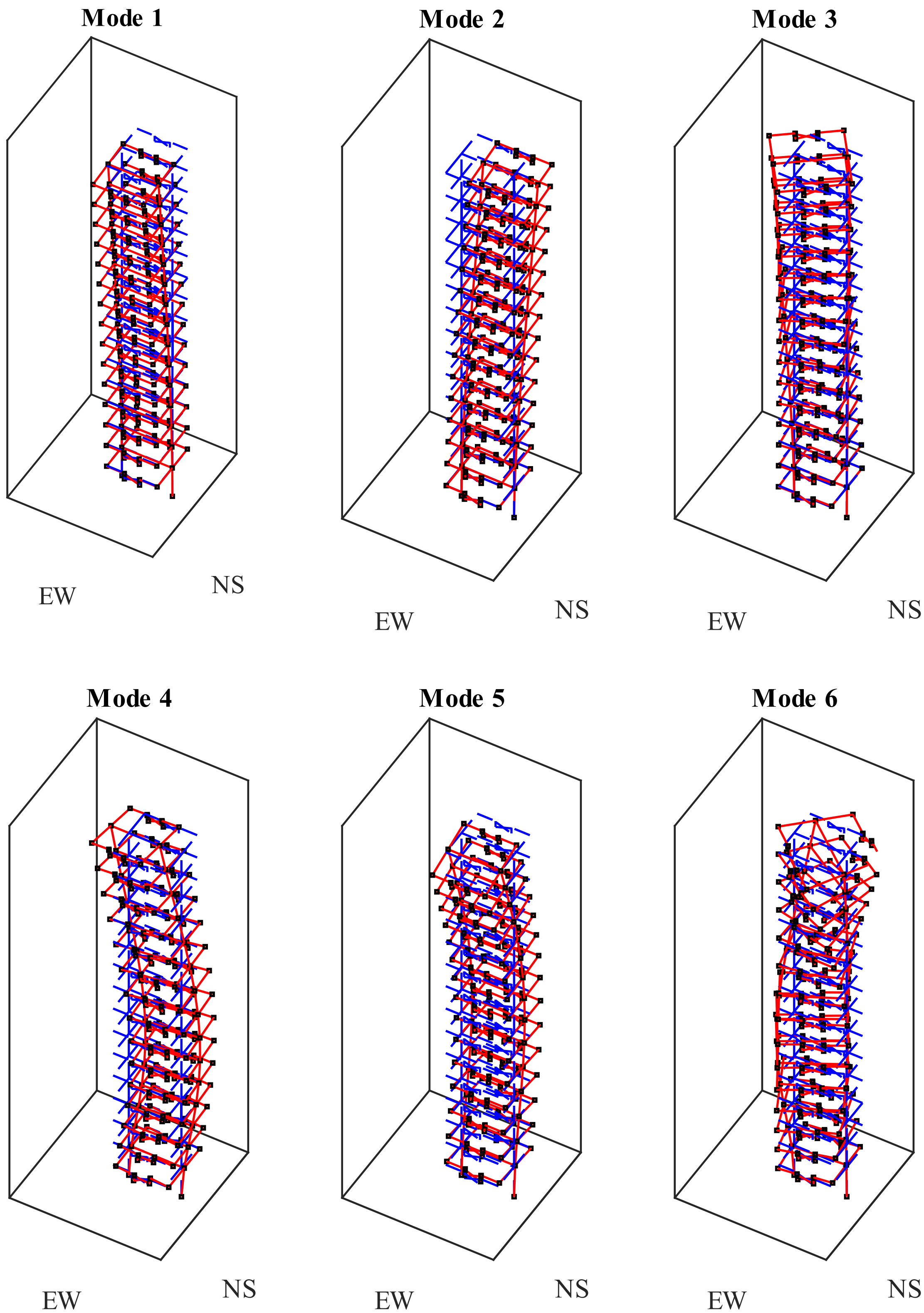


Figure 28 3D Views of Mode Shapes 1 through 6

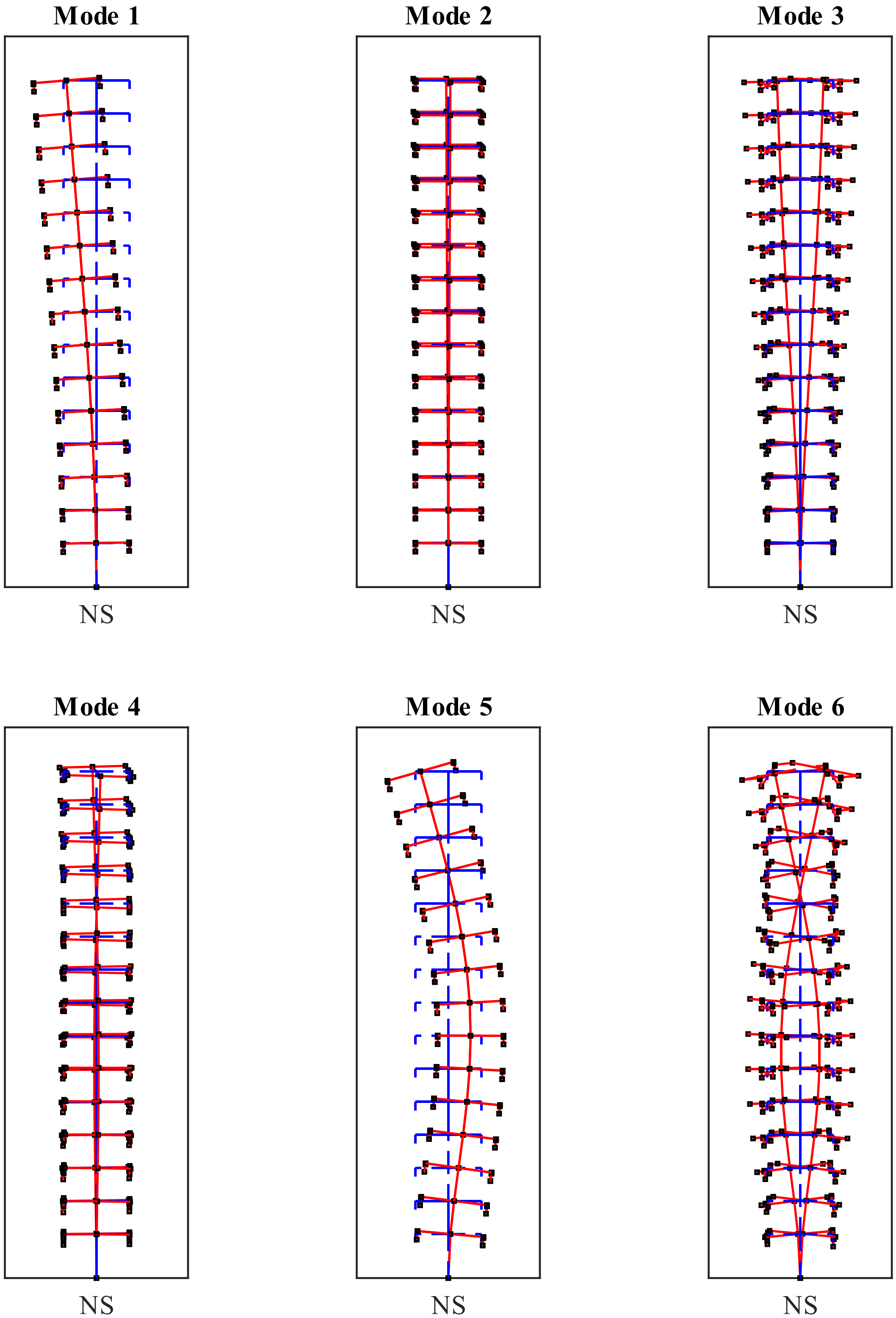


Figure 29 NS Elevation View of Mode Shapes 1 through 6

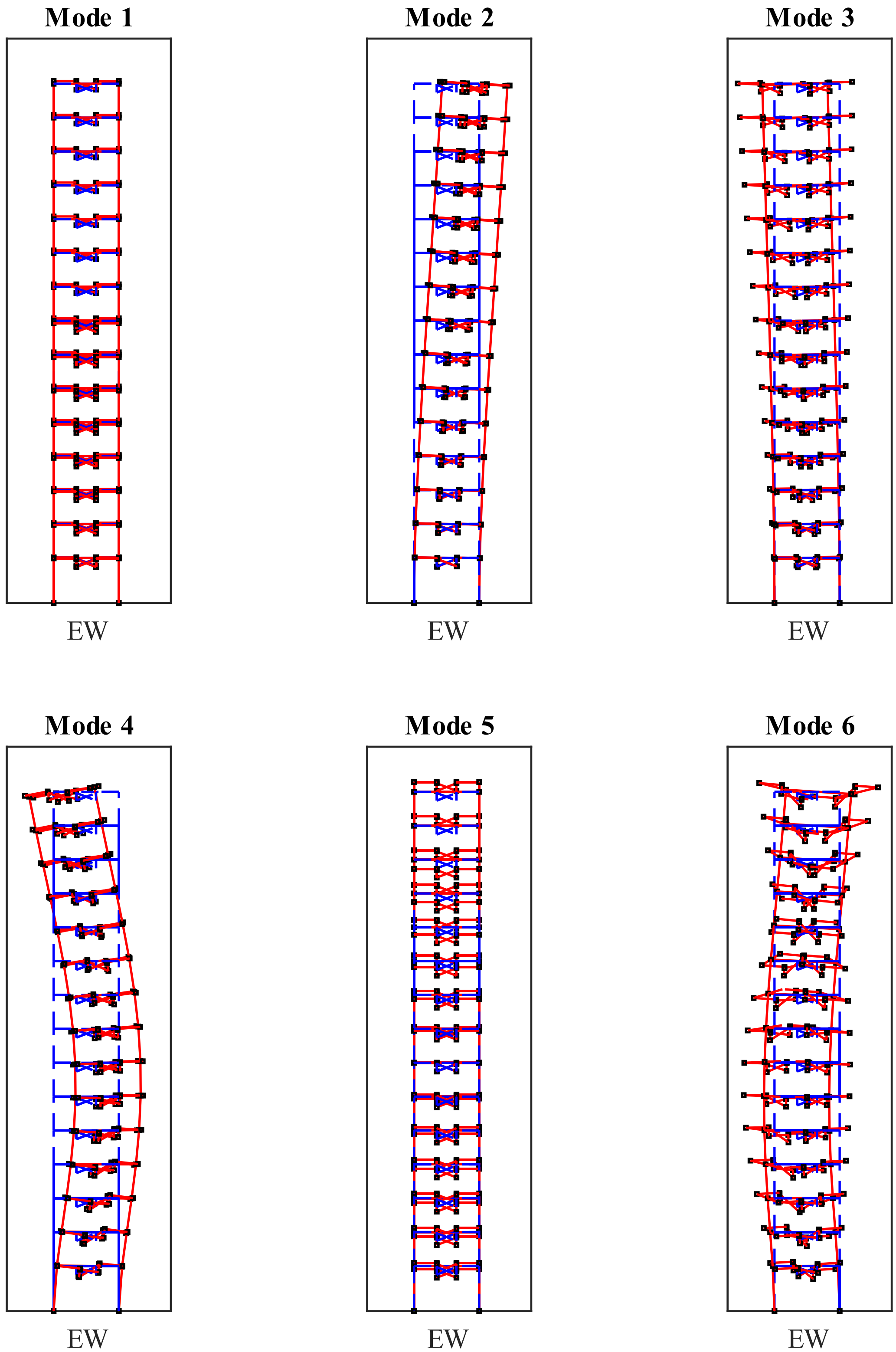


Figure 30 EW Elevation View of Mode Shapes 1 through 6

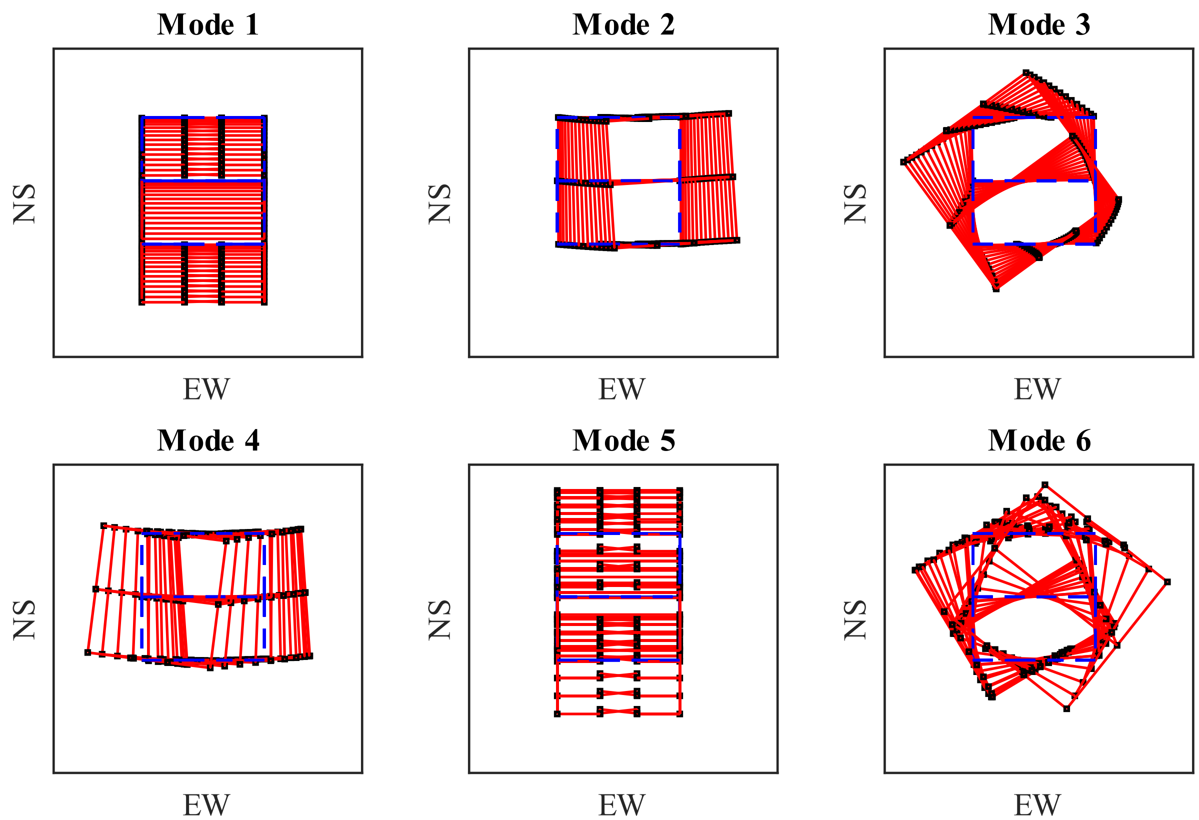


Figure 31 Top View of Mode Shapes 1 through 6

Different views of the mode shapes of the structure are given in Figure 28 through Figure 31. Mode shapes are fundamental shapes that are orthogonal with one another and can be used as the basis for any combination of deformations the structure can take on. The first mode and fifth mode are predominantly in the NS direction while mode 2 and mode 4 are primarily in the EW direction. Mode 3 and 6 exhibit uni-directional and bi-directional torsion. Because the building is asymmetrical, the mode shapes in mode 1, 2, 4, and 5 exhibit torsion as well as their predominant displacement directions. The natural periods of the first 6 modes are given in Table 5 Natural Periods of the Structure for Modes 1-6

Table 5 Natural Periods of the Structure for Modes 1-6

Modes	Undamped Natural Periods (Uncracked)	Undamped natural periods (Cracked)
1	0.862	0.869
2	0.842	0.847
3	0.384	0.385
4	0.198	0.199
5	0.138	0.139
6	0.104	0.104

<u>Determine Parameters for Raleigh Damping Model</u>

Figure 32 shows the damping ratio as a function of frequency after assigning two percent damping ratio to the first two translational modes of the structure in the EW direction. The two predominant modes in the EW direction are mode 1 and mode 4.

Natural frequencies of the two predominant EW modes are found by $\omega = 2\pi/T$.

T_1 = 0.862, T_4 = 0.198

ω_1 = 7.4196, ω_4 = 31.646

The parameters a_0 and a_1 for the Rayleigh damping model is determined as

$$\begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = 2\begin{bmatrix} \frac{1}{\omega_m} & \omega_m \\ \frac{1}{\omega_n} & \omega_n \end{bmatrix}^{-1} \begin{bmatrix} \xi_m \\ \xi_n \end{bmatrix} = \begin{bmatrix} 0.23551 \\ 0.0010288 \end{bmatrix}$$

where $\xi_m = \xi_n = 0.02$. The damping ratio is calculated below, where $\frac{a_0}{2}\frac{1}{\omega}$ is the mass proportional damping and $\frac{a_1}{2}\omega$ is the stiffness proportional damping

$$\xi(\omega) = \frac{a_0}{2}\frac{1}{\omega} + \frac{a_1}{2}\omega = \frac{0.23551}{2\omega} + \frac{0.0010288}{2}\omega$$

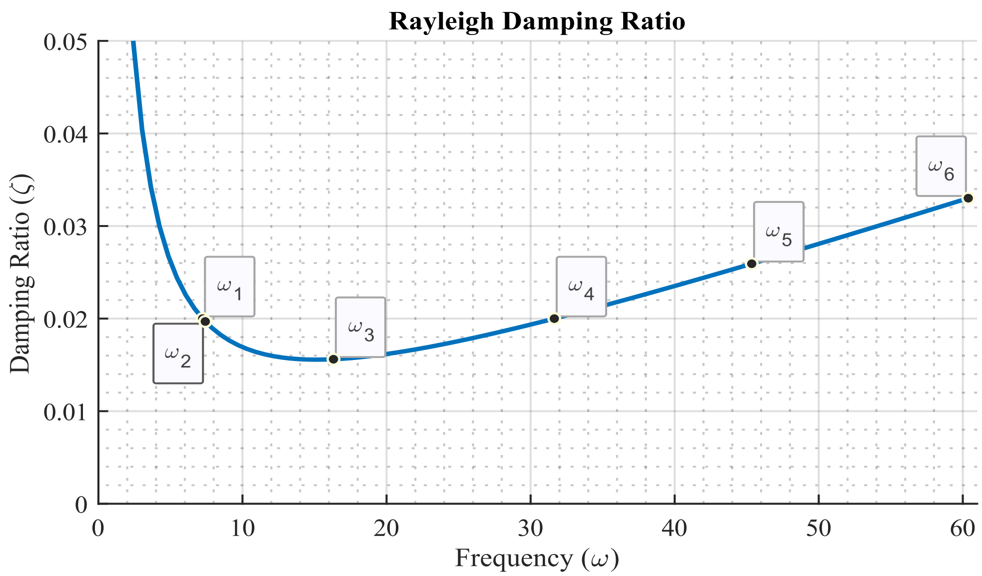


Figure 32 Damping Ratio vs Frequency

Compute Damping Ratio of third vibration mode

The first mode and the fourth were set to a damping ratio of 0.02 with the other damping ratios for the other modes are solved for. The damping ratios of the rest vibration modes of the structure are found in Table 6.

Table 6 Natural Frequencies and Damping Ratios for Uni-directional Earthquake Motion

Modes	Natural Frequencies (1/s)	Damping Ratios
1	7.2335	0.02
2	7.4196	0.019687
3	16.317	0.01561
4	31.646	0.02
5	45.35	0.03301
6	60.38	0.025925

Part I Nonlinear Time History Analysis in the Uniaxial Direction

Overturning Moment at Base of Core Wall

Figure 33 show the moment arms needed to be considered when calculating the overturning moment of the building.

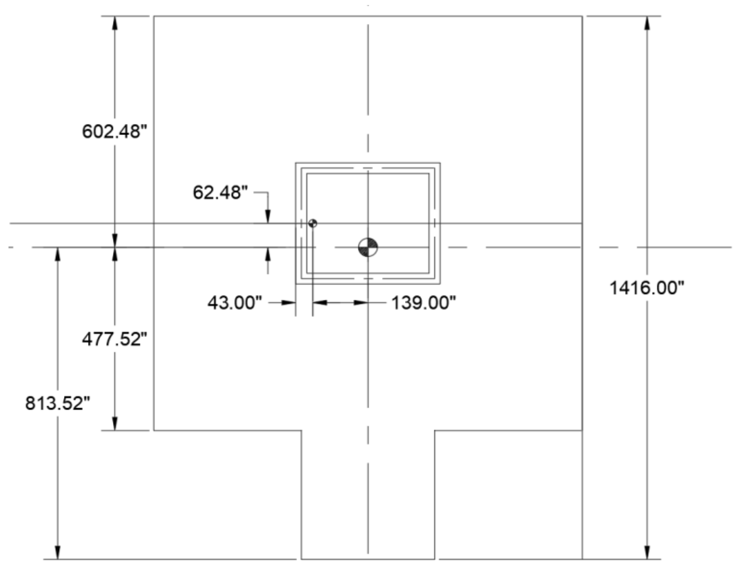


Figure 33 Locating the Centroid of the Building and the Core Wall

Figure 34 shows individual overturning moment vs. the roof drift ratio of wall 1 and wall 2, and the total overturning moment vs. the roof drift ratio. The curve of the overturning moment vs. drift ratio from wall 1 and wall 2 have the same shape. The behavior at the positive drift ratio and the negative drift ratio is not similar, whereas that of nonlinear static pushover analysis had a similar response. Also, it is noted that the overturning moment at negative drift ratio is larger than that at positive drift ratio.

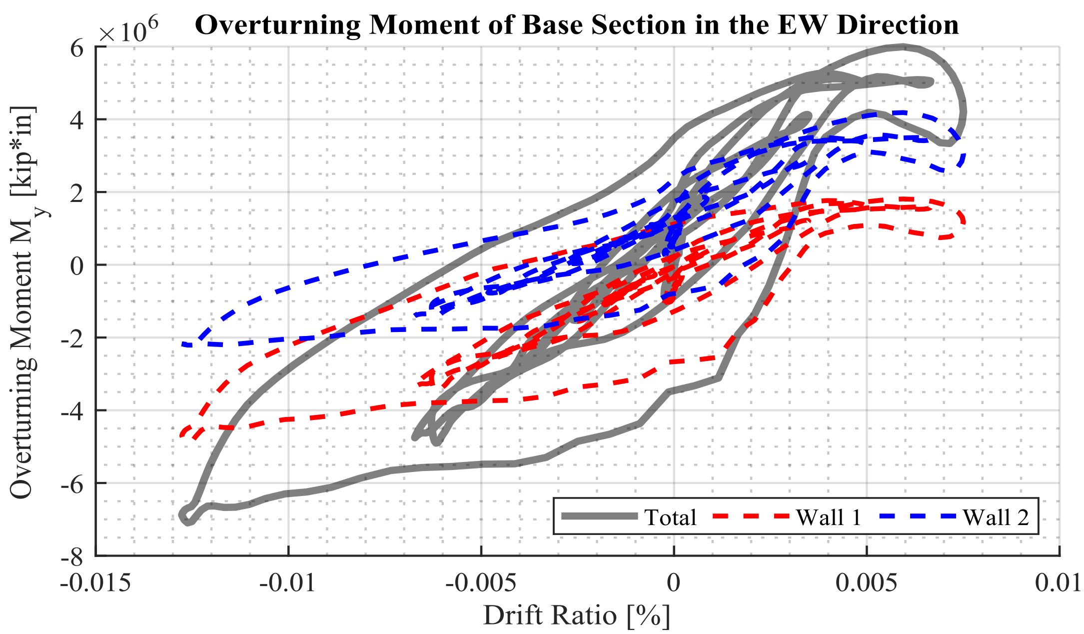


Figure 34 Total Base Overturning Moment vs Roof Drift Ratio in the EW Direction Under Dynamic Loading

Absolute Acceleration Response

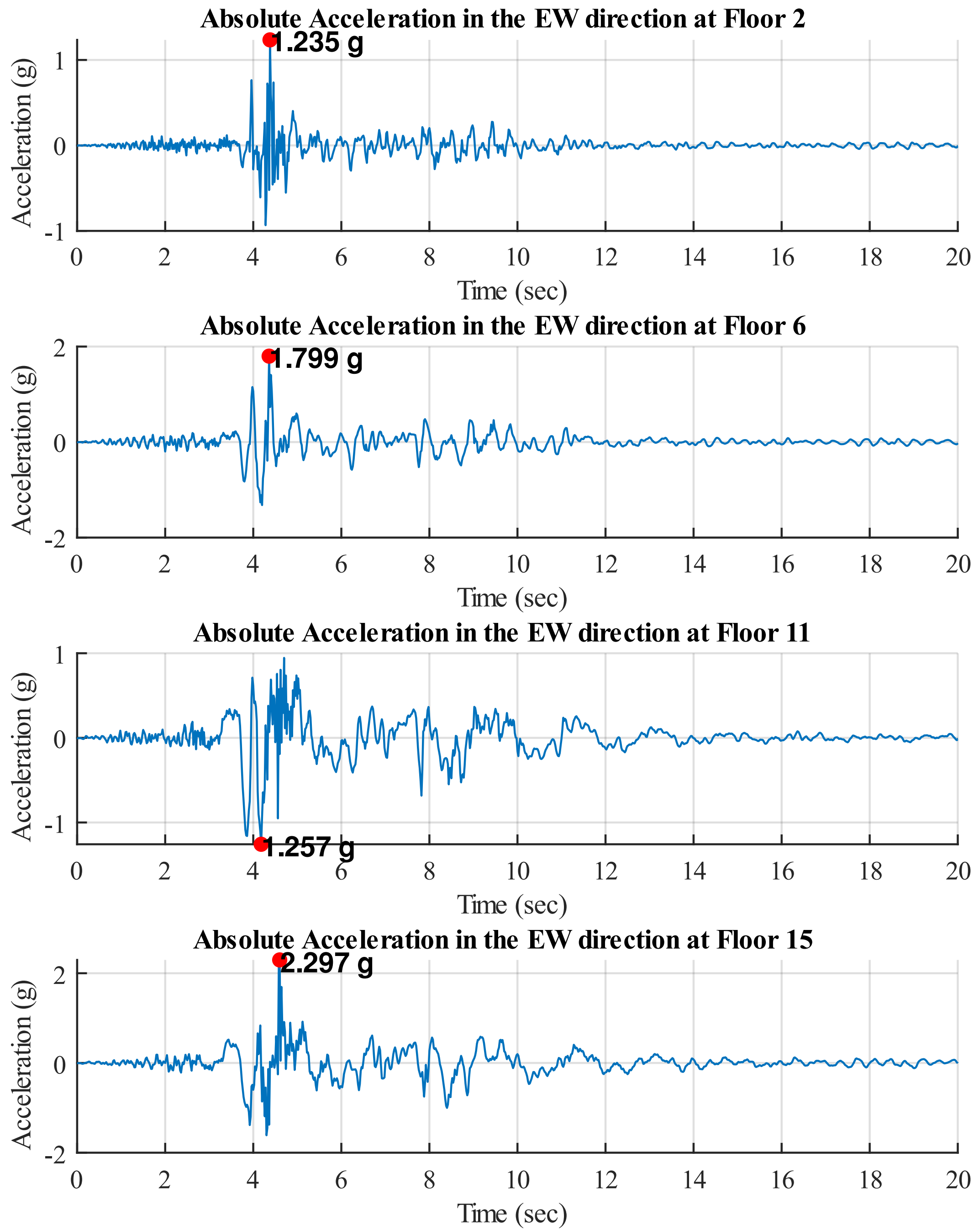


Figure 35 Absolute Acceleration Response

Relative Displacement Response

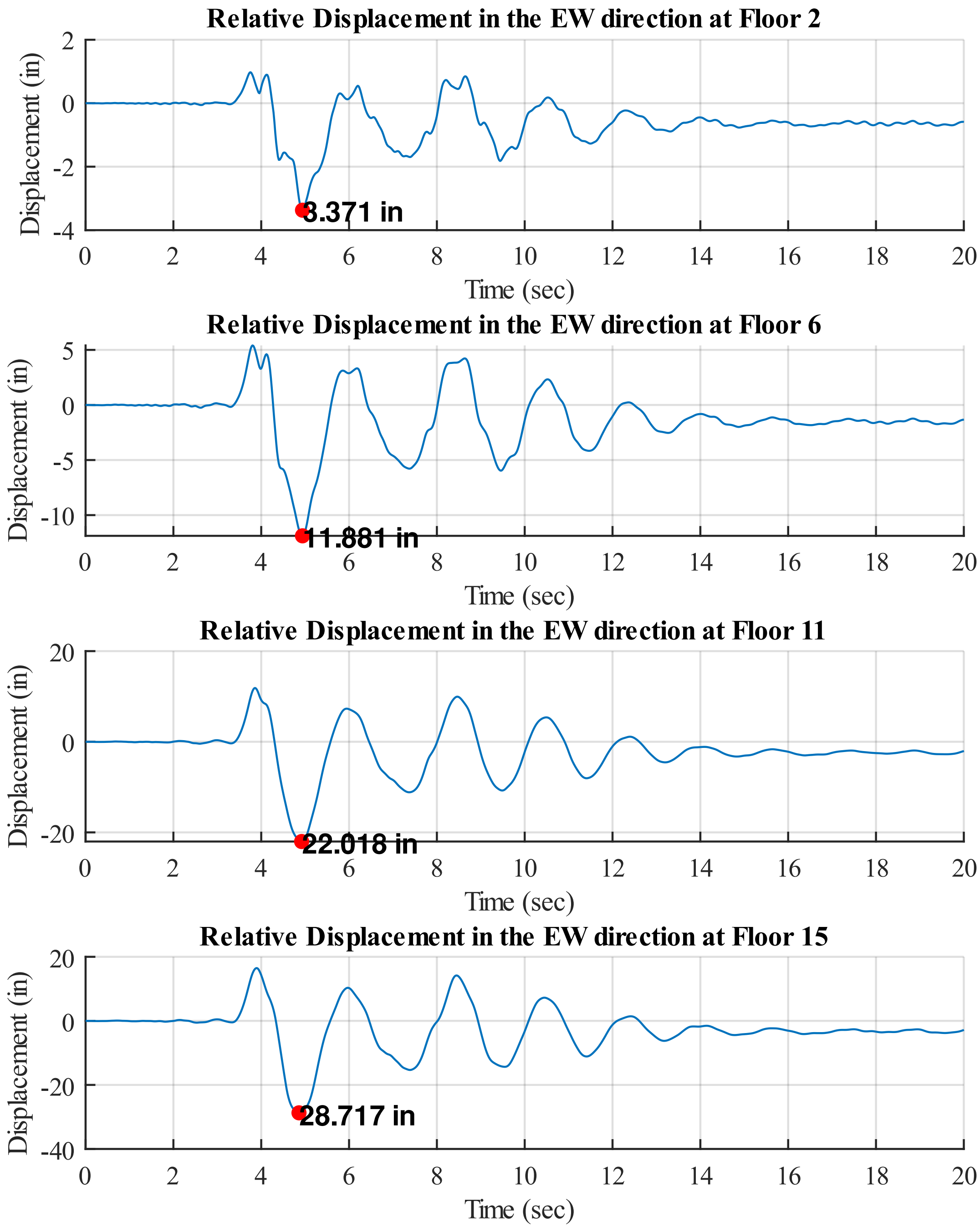


Figure 36 Relative Displacement Response Under Dynamic

Normalized Base Shear

Figure 37 shows the total normalized base shear response of the coupled core-wall unit in the EW direction.

The maximum normalized base shear is approximately 30%, which means that 30% of the weight is participating in the dynamic effect. So, having 100% of normalized base shear would means that the entire weight of the structure is participating in the motion, but it does not happen in the real world.

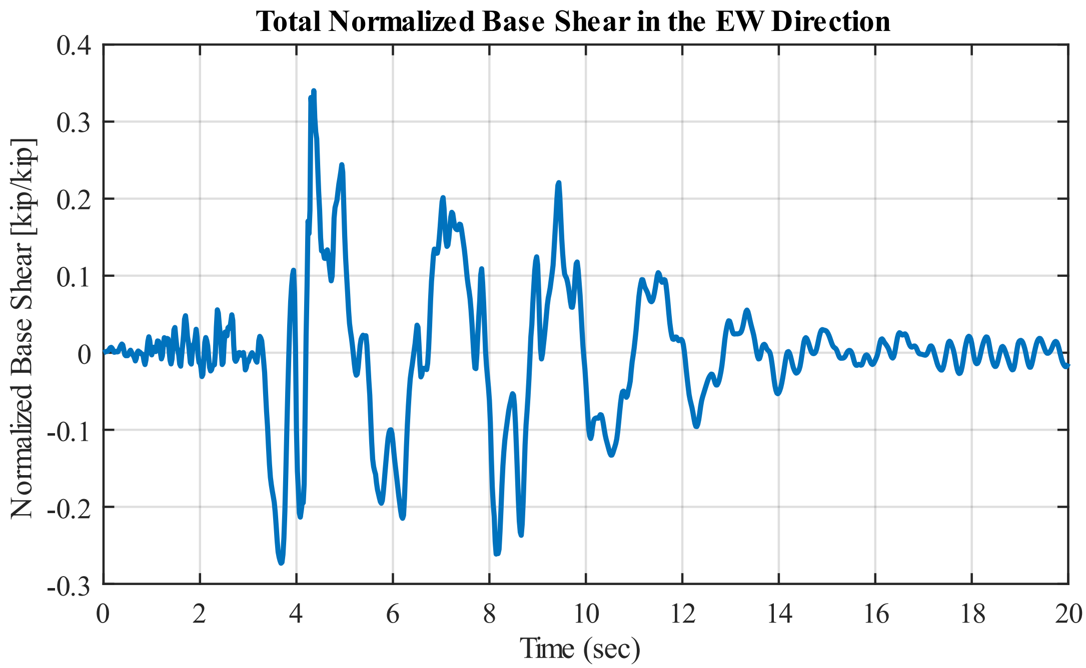


Figure 38 Normalized Base Shear in the EW Direction

Moment Curvature

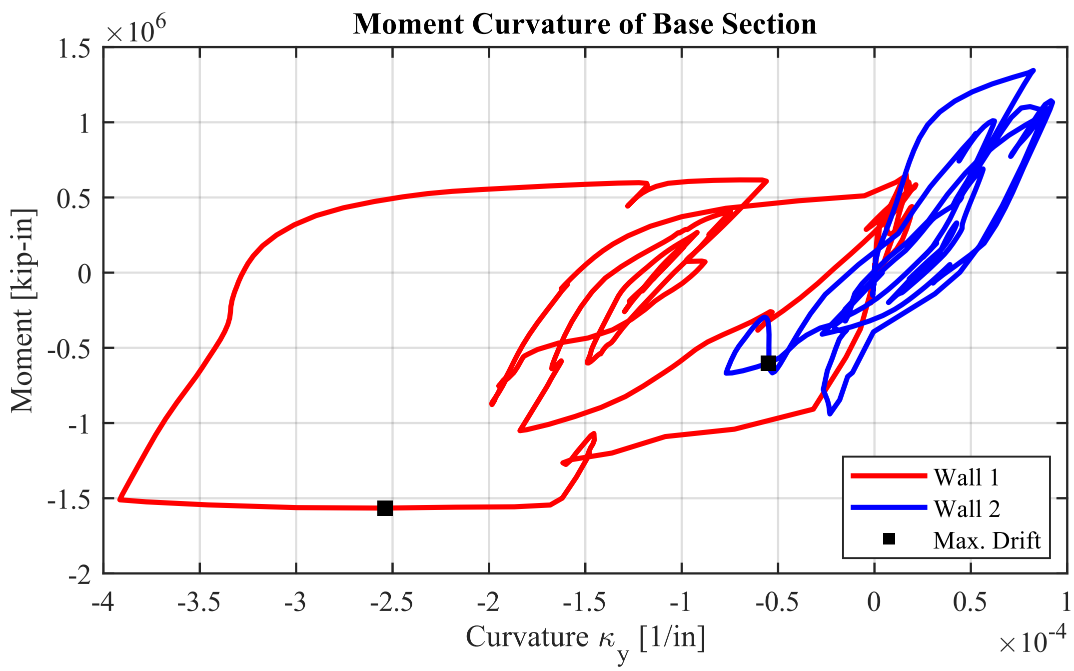


Figure 39 Moment Curvature Response of Wall 1 and Wall 2 in the EW Direction Under Dynamic Loading

Axial Strain versus Curvature Response

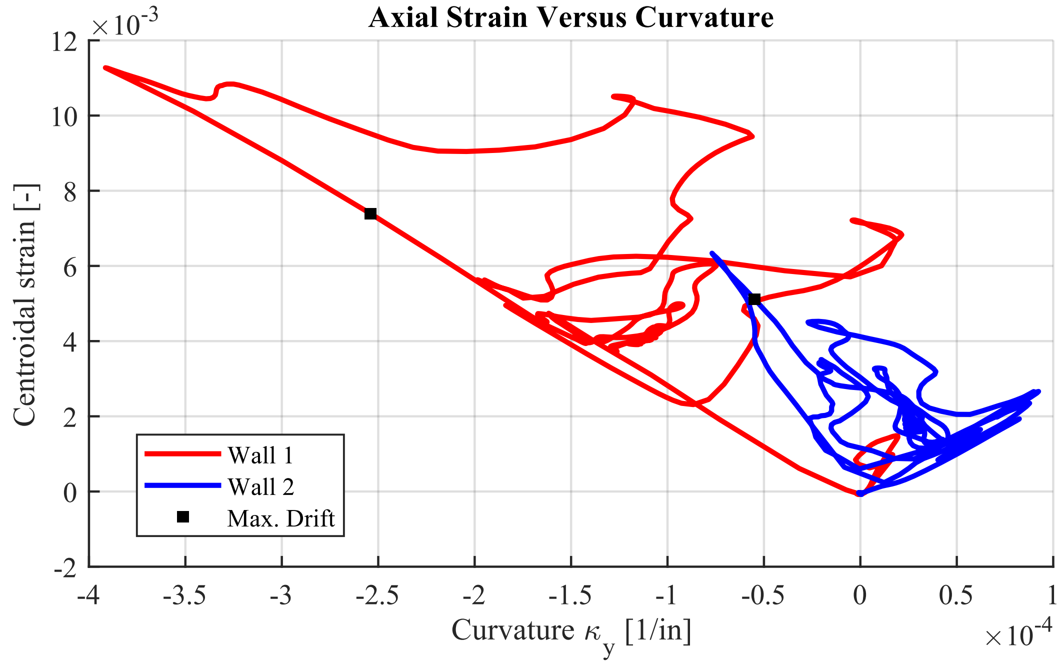


Figure 40 Axial Strain vs Curvature of Wall 1 and Wall 2 under Dynamic Loading

Shear Force and Shear Deformation of Coupling Beams at Selected Floors

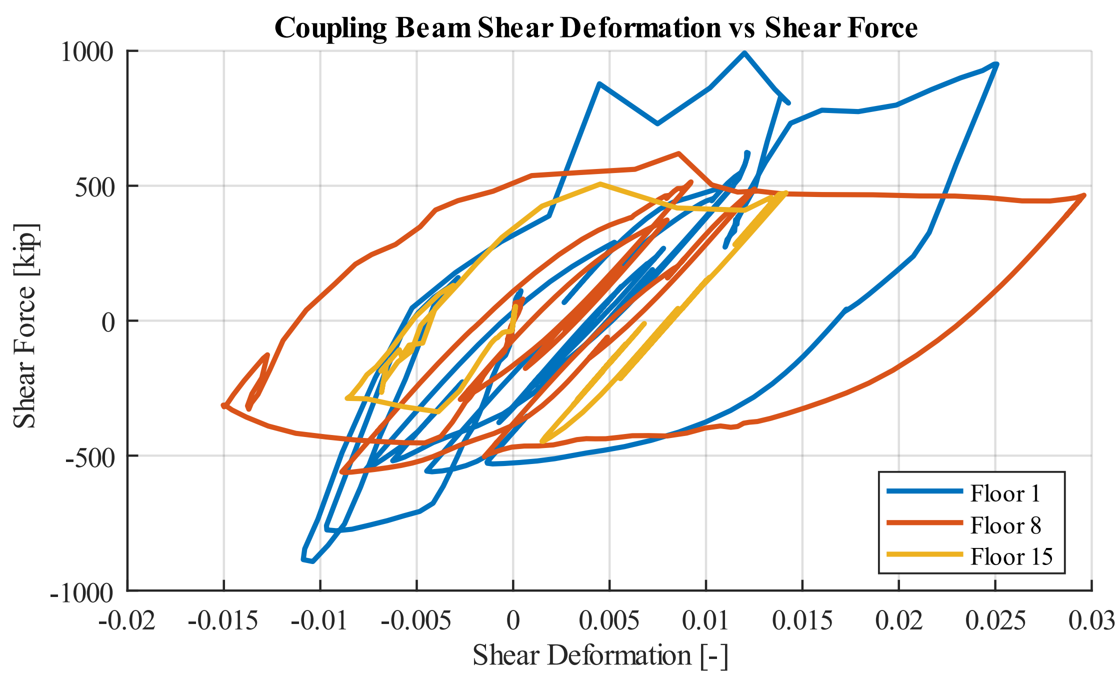


Figure 41 Shear Force vs Shear Deformation Response of Coupling Beams Under Dynamic Loading

Comparison with Linear Elastic Material Models

A comparison of the uni-directional earthquake ground motion is made for nonlinear and linear material constitutive models.

Overturning Moment at Base of Core Wall

The overturning moment at the base of the core wall has a more linear and similar shape than that of the nonlinear material. The irregularities come from taking into account the non-geometric deformities of the structure through the P-delta approximation.

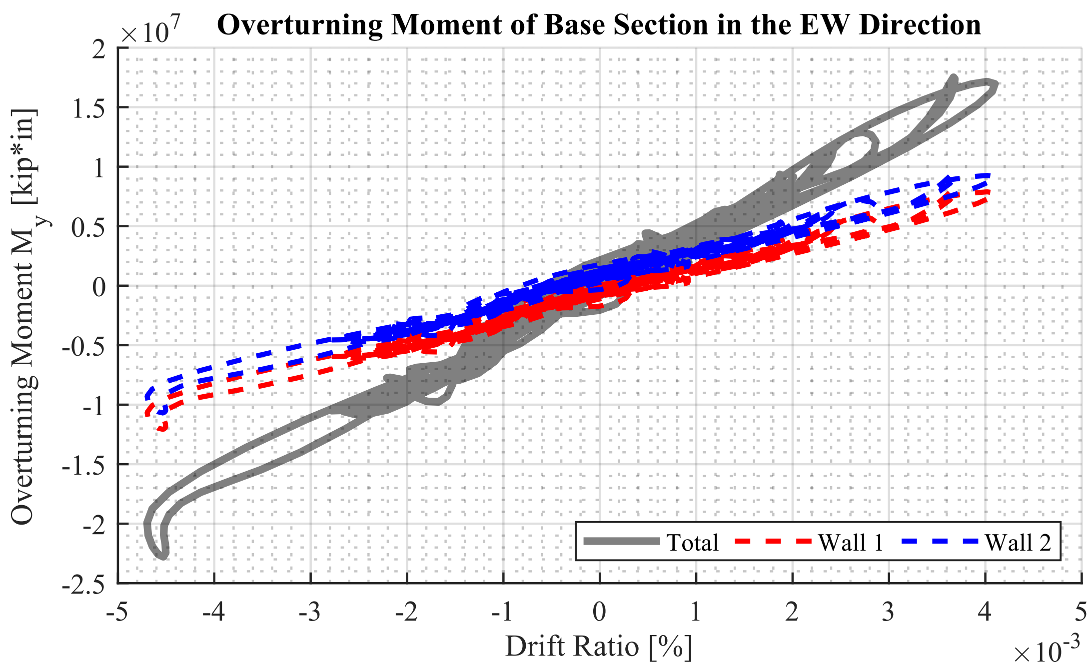


Figure 42 Comparison of Total Base Overturning Moment vs Roof Drift Ratio Under Dynamic Loading for Different Material Models

Absolute Acceleration Response

The average period of vibration of the linear elastic material model is shorter than that of nonlinear material model.

And the magnitudes of the pulses reduce at a slower rate when the material is linear elastic than when the material is nonlinear. In other words, the vibration lasts longer.

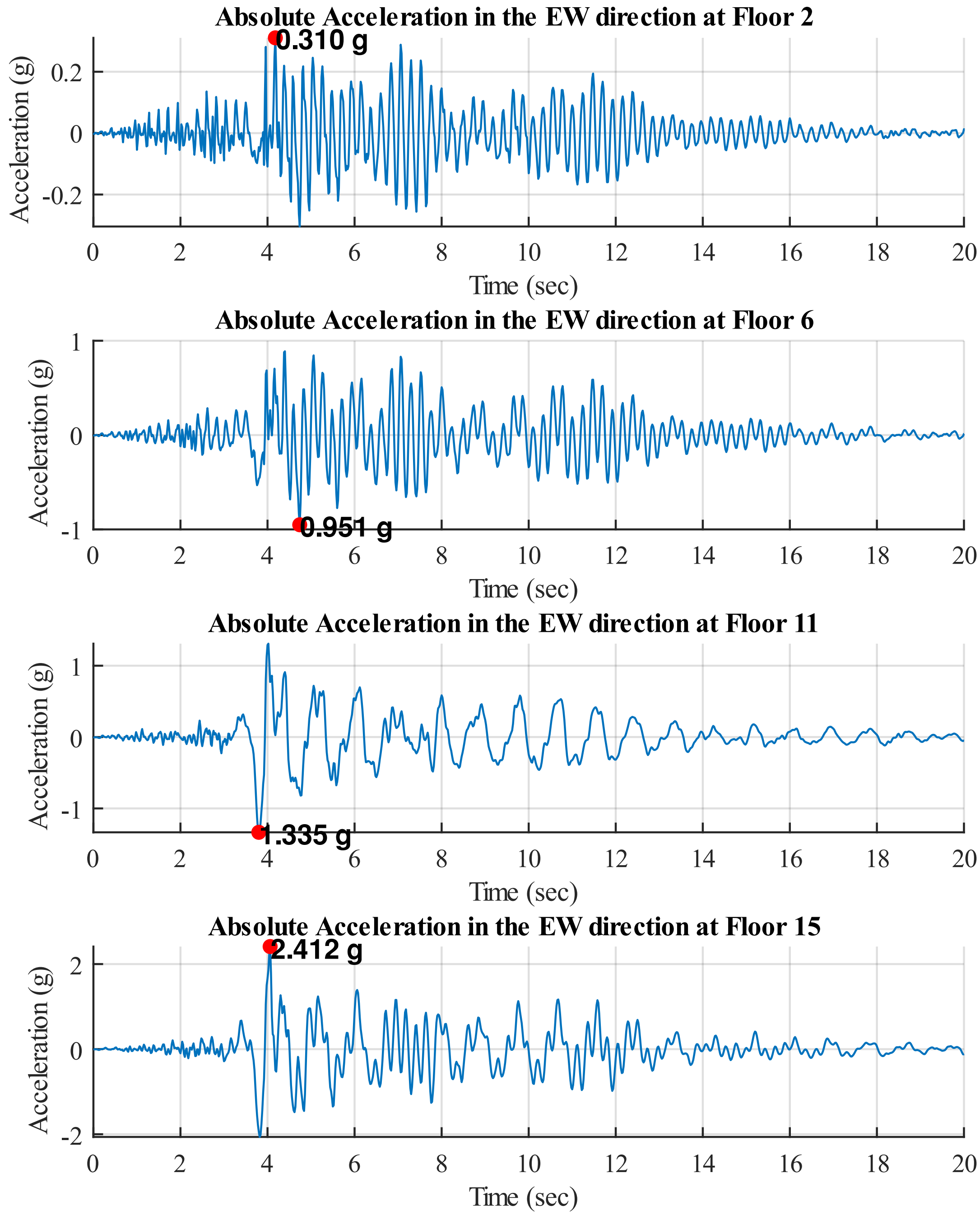


Figure 43 Comparison of Absolute Acceleration Response Under Dynamic Loading for Different Material Models

Relative Displacement Response

The magnitude of the peak displacement and frequency of the relative displacement response is smaller when linear elastic material is used than when the material is nonlinear; the vibration lasts longer.

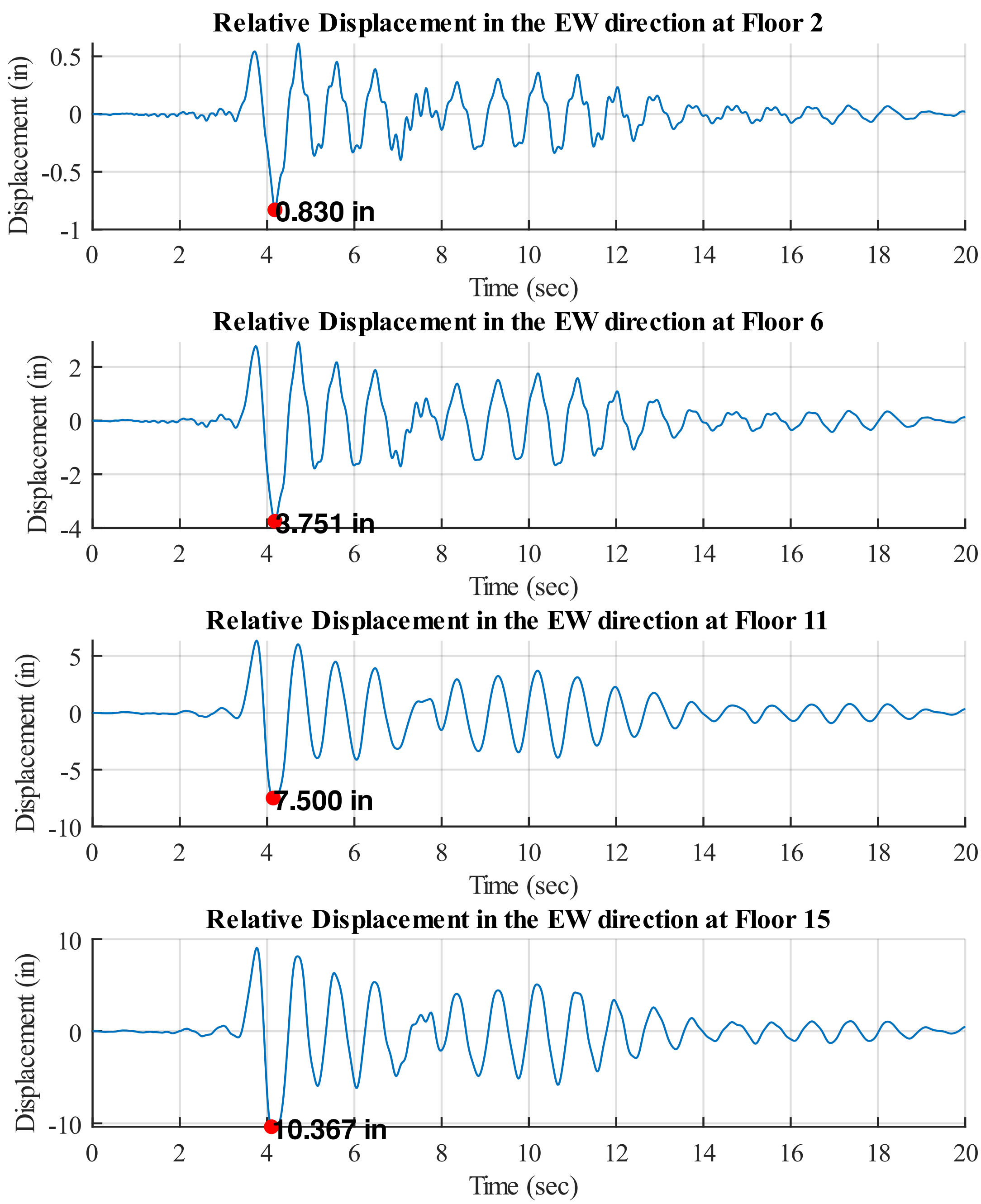


Figure 44 Comparison of Relative Displacement Response Under Dynamic Loading for Different Material Models

Normalized Base Shear

Figure 45 shows the total normalized base shear using linear elastic material. Total normalized base shear using linear elastic material is significantly larger than that of using nonlinear material model. The approximate maximum normalized base shear is 100% which is unrealistic for a structure.

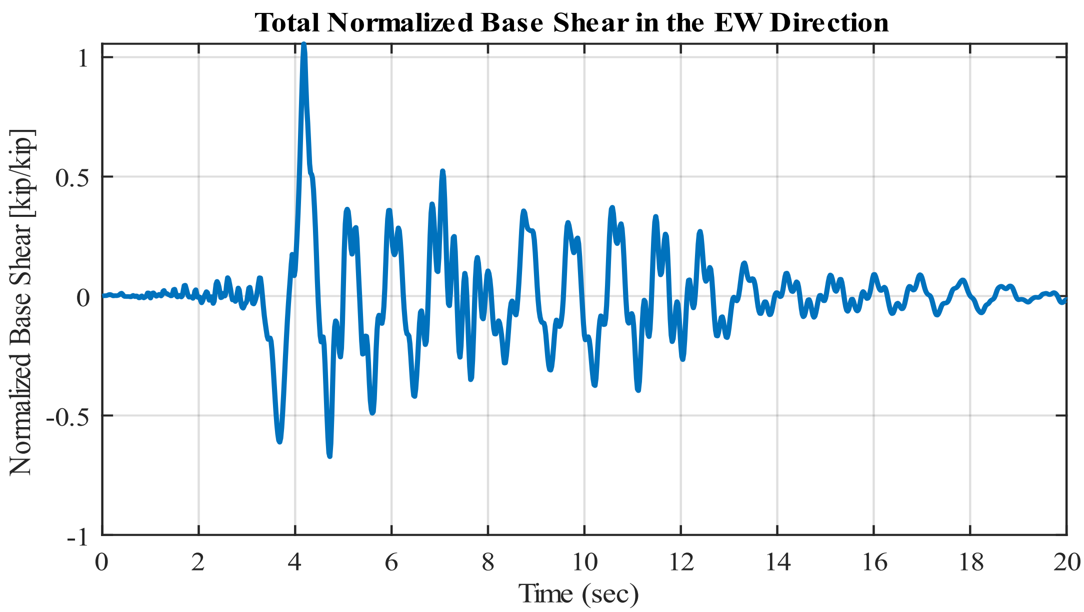


Figure 45 Comparison of Normalized Base Shear in the EW Direction for Different Material Models

Moment Curvature

In nonlinear materials, the walls are weaker in tension because of asymmetric loading. For linear material properties, the response is the same for both walls.

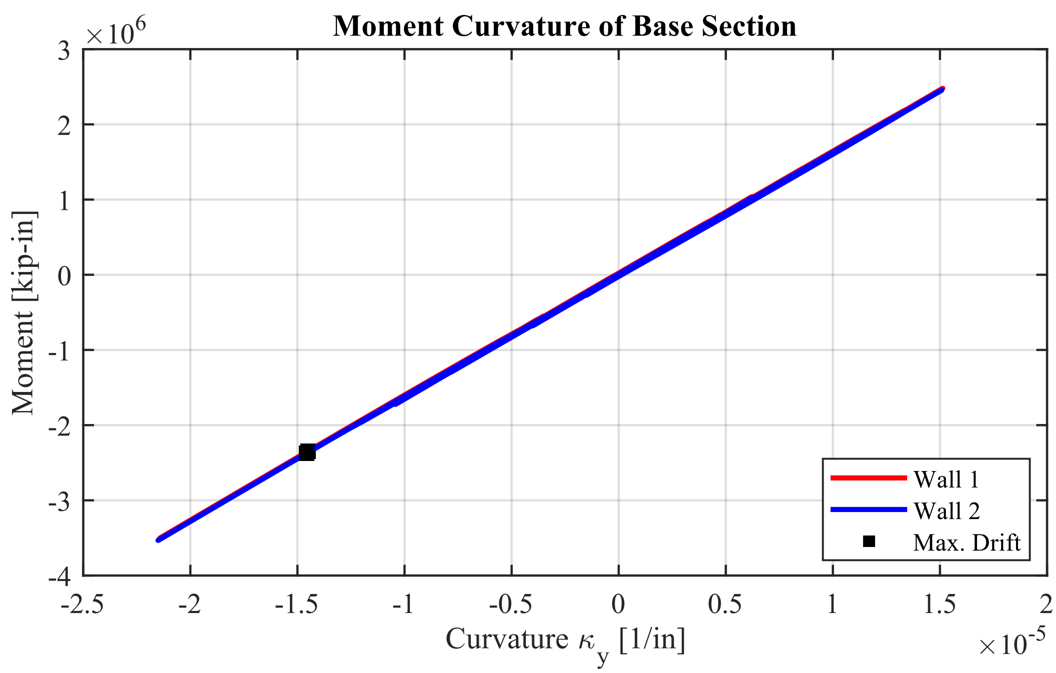


Figure 46 Comparison of Moment Curvature Under Dynamic Loading for Different Material Models

Axial Strain versus Curvature Response

In the linear material quasi-static case, the axial curvature plot was straight however in the linear material dynamic loading case there are hysteresis loops due to higher mode of structure.

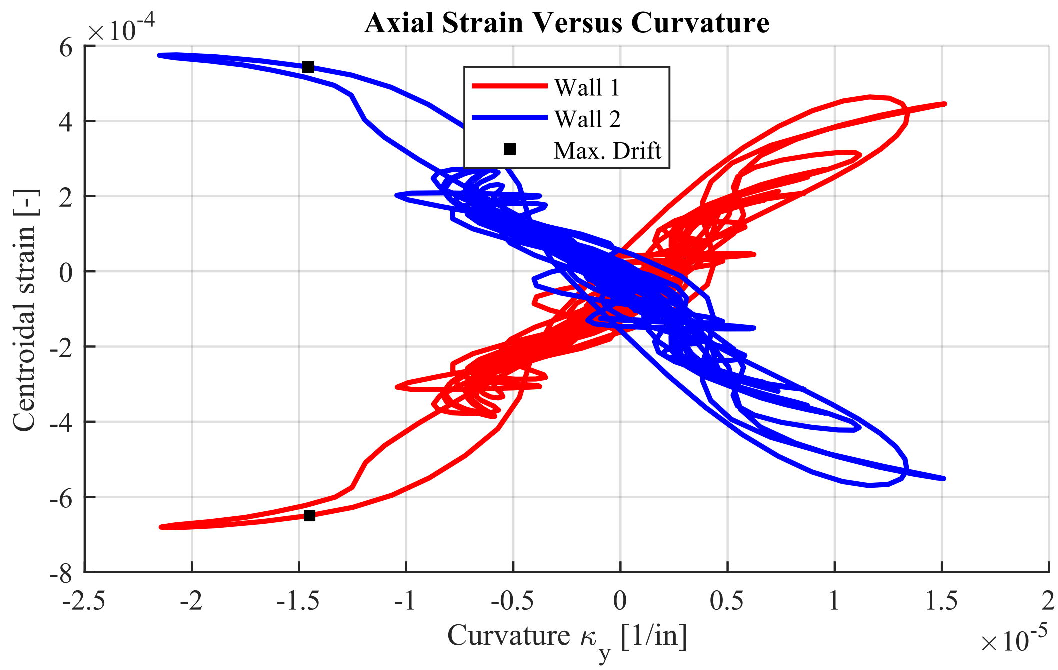


Figure 47 Comparison of Axial Strain Under Dynamic Loading for Different Material Models

Shear Force and Shear Deformation of Coupling Beams at Selected Floors

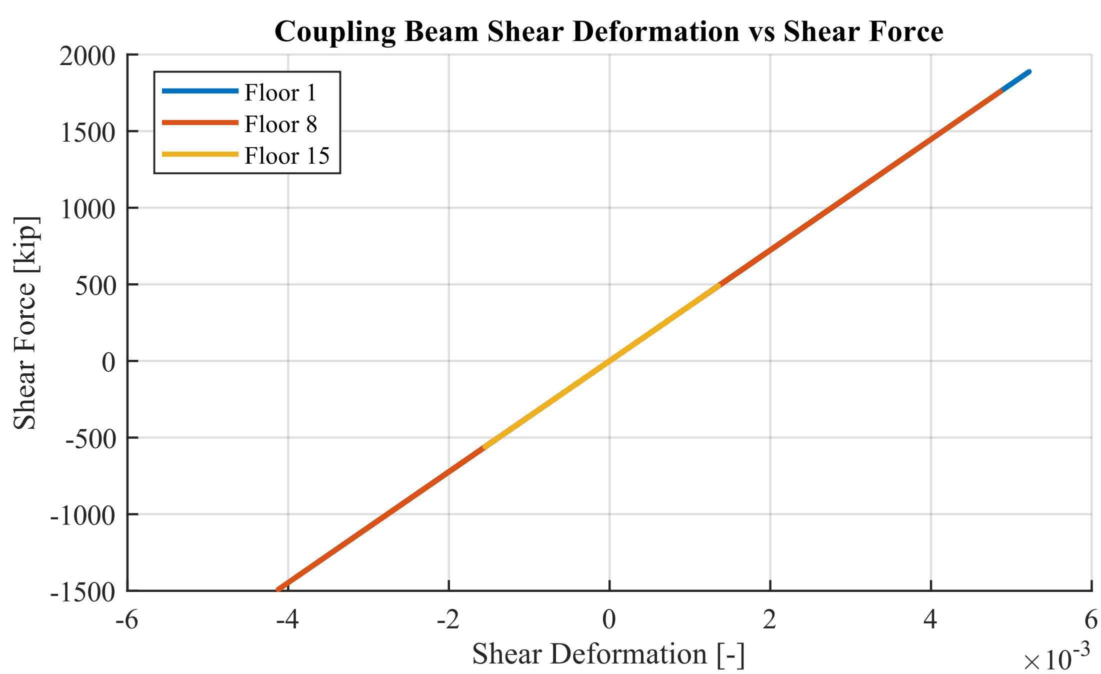


Figure 48 Comparison of Coupling Beam Responses Under Dynamic Loading for Different Material Models

Comparison with Results of Finer Time Steps in Earthquake Records

This section compares the effect of the time step used in nonlinear time history analysis.

Delta t, Δt is the time step to integrate the equations of motions numerically. And delta t record, Δtrec is the time step of the earthquake record. Typical time step of the earthquake record is 0.02 second and sometimes it is 0.01 second and 0.005 second (with digital seismometers). So, Δt and Δtrec are defined independently.

The response will not be accurate if Δt is too large, and if Δt is too small it is uneconomical because it takes longer time to compute. How small Δt can be figured out by dividing the current Δt by 2 and comparing the result with the original. If the results are very different, it means the response is not very accurate. So, we need to divide Δt until the result response are converged in terms of Δt and use that Δt for analyzing the response.

Figure 51 shows the absolute acceleration response, using the two different time step Δt = 0.02 and Δt = 0.01. And the two responses are almost identical.

Figure 52 shows the relative displacement response with Δt = 0.01 sec and it is overlapped with the same analysis using Δt = 0.02 sec. And we can observe that the curves are almost identical.

Therefore, we could conclude that the Δt = 0.02 sec is a reasonably small time-step, and good to use for the analysis.

Overturning Moment at Base of Core Wall

Figure 49 show the overturning moment using half the time step. We can observe that the curves are smoother when a finer step is used.

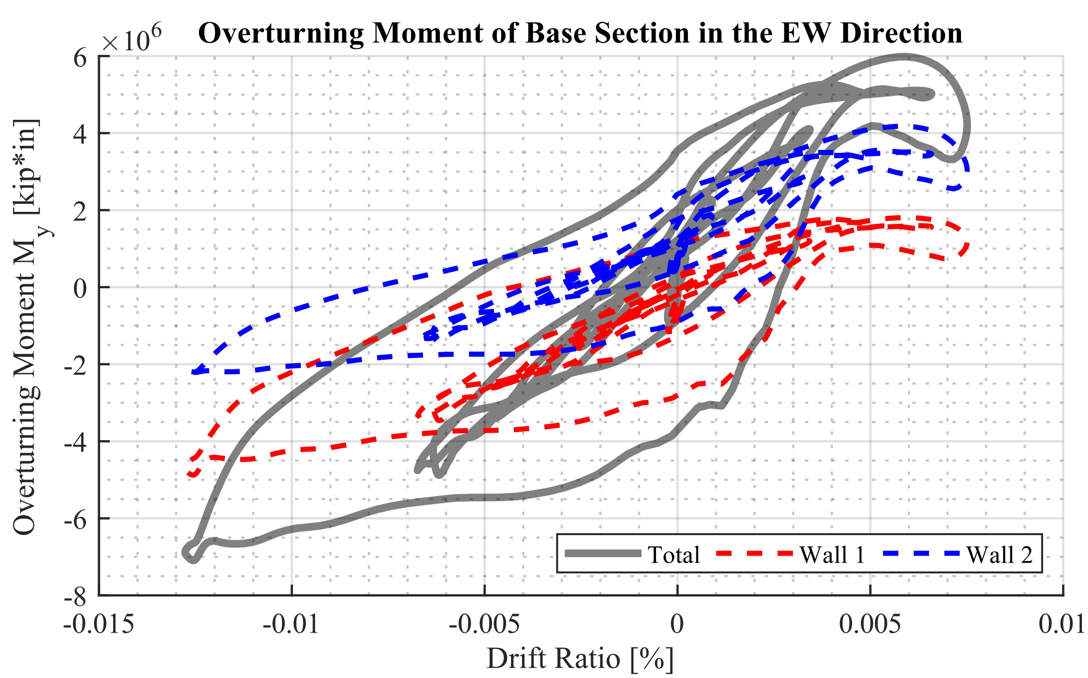


Figure 49 Comparison of Total Base Overturning Moment vs Roof Drift Ratio Under Dynamic Loading for Different Time Steps

Normalized Base Shear

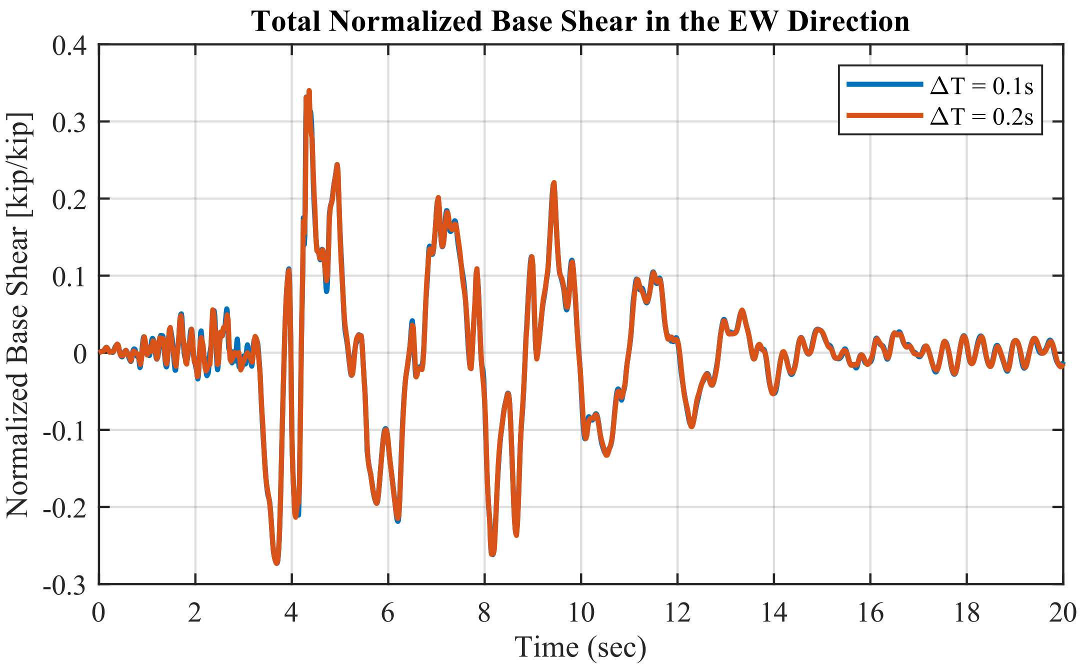


Figure 50 Comparison of Normalized Base Shear in the EW Direction Under Dynamic Loading for Different Time Steps

Absolute Acceleration Response

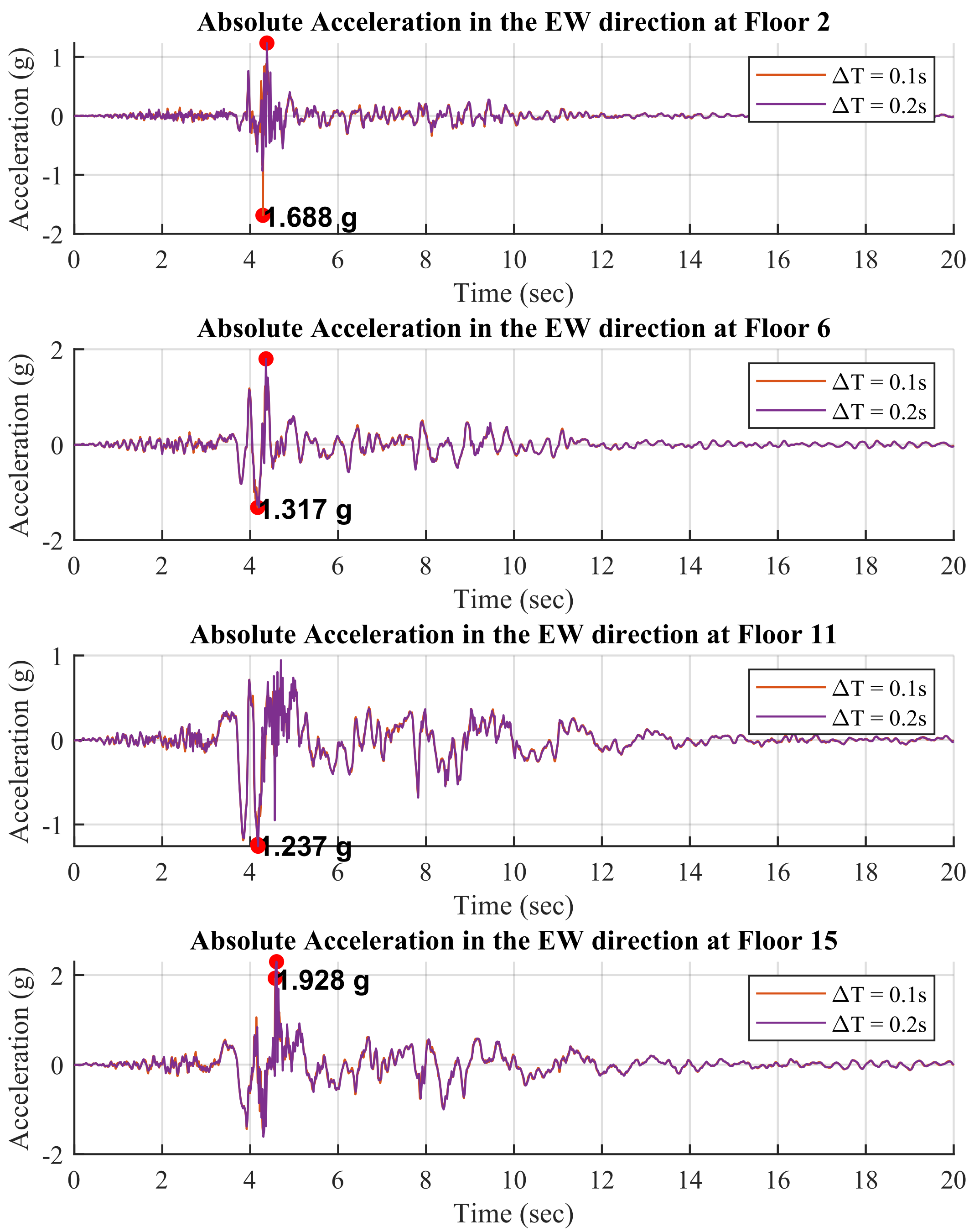


Figure 51 Comparison of Absolute Acceleration Response Under Dynamic for Different Time Steps

<u>Relative Displacement Response</u>

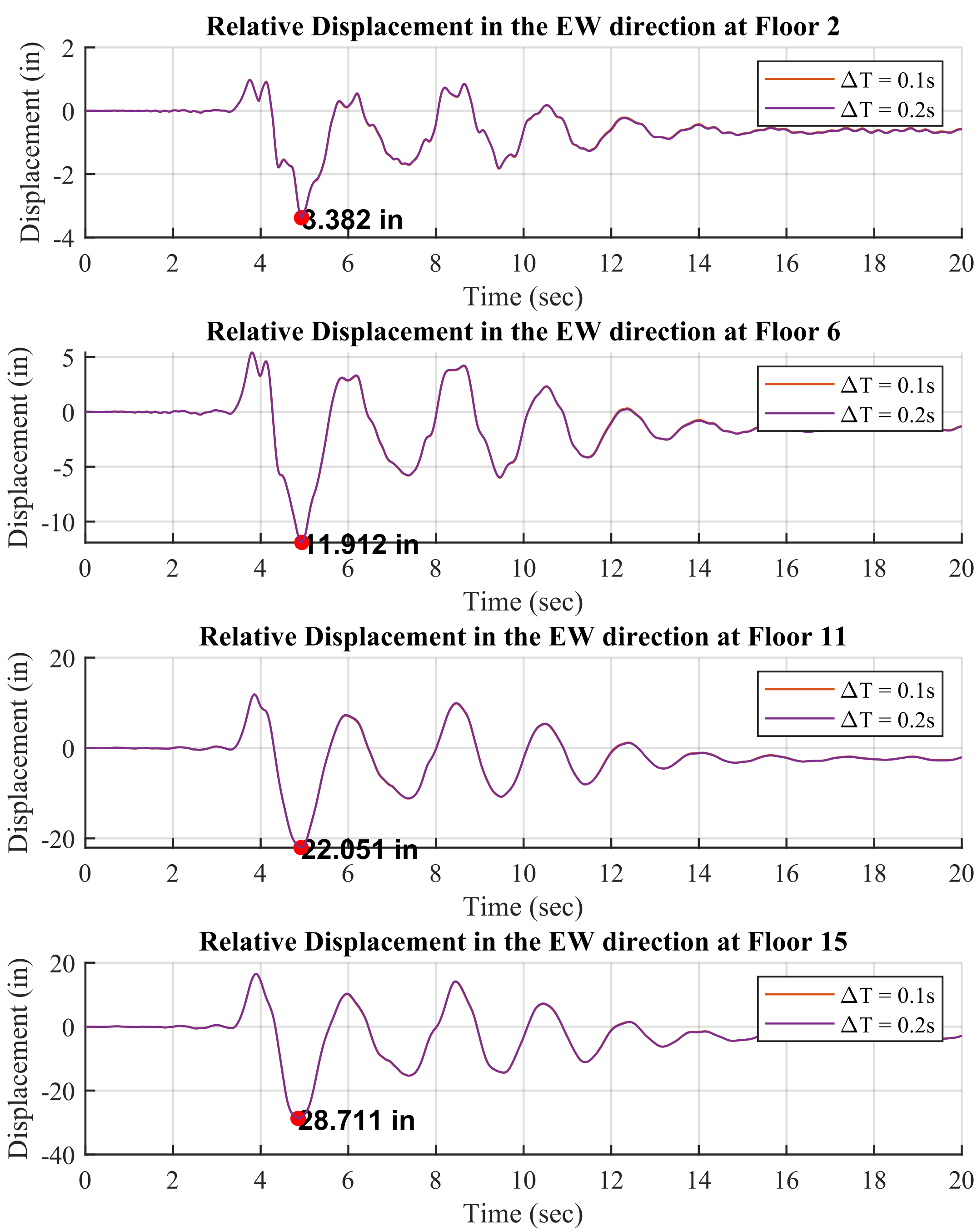


Figure 52 Comparison of Relative Displacement Response Under Dynamic for Different Time Steps

Comparison with Results with Different Tangent Stiffness for Rayleigh Damping model.

This section compares the simulated structural responses obtained by using a model with a Rayleigh damping model $C = a_0 M + a_1 K_{cracked,\ initial}$ with that of using a model with a Rayleigh damping model $C = a_0 M + a_1 C_{ommitted}$.

$K_{cracked,\ initial}$ is the tangent stiffness matrix after applying gravity loads. And $K_{committed}$ the committed tangent stiffness matrix which is the tangent stiffness matrix of the last converged state.

Committed tangent stiffness is smaller than the initial tangent stiffness because the stiffness of the structure will soften more and more as load is applied.

Overturning Moment at Base of Core Wall

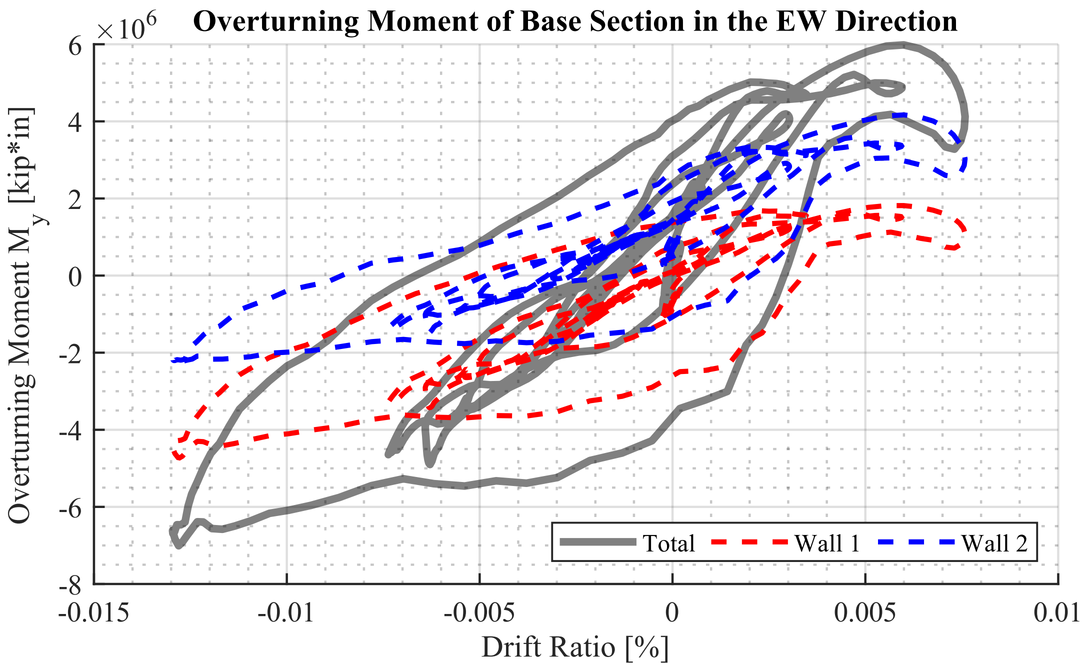


Figure 53 Comparison of Total Base Overturning Moment vs Roof Drift Ratio Under Dynamic Loading for Different Damping Models

Normalized Base Shear

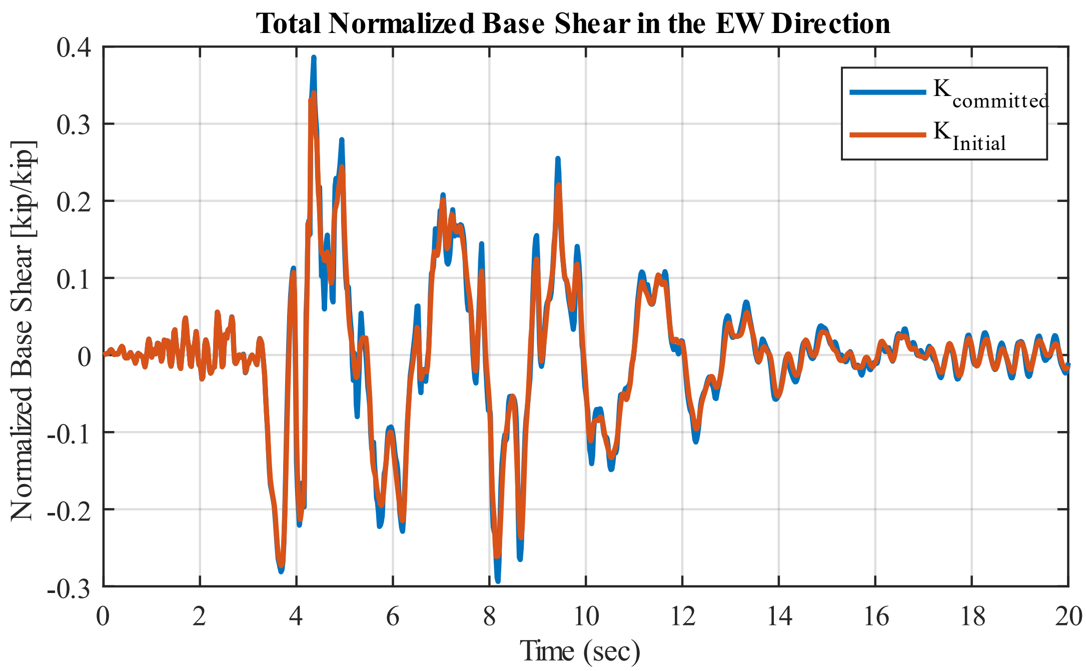


Figure 54 Comparison of Normalized Base Shear in the EW Direction Under Dynamic Loading for Different Damping Models

Absolute Acceleration Response

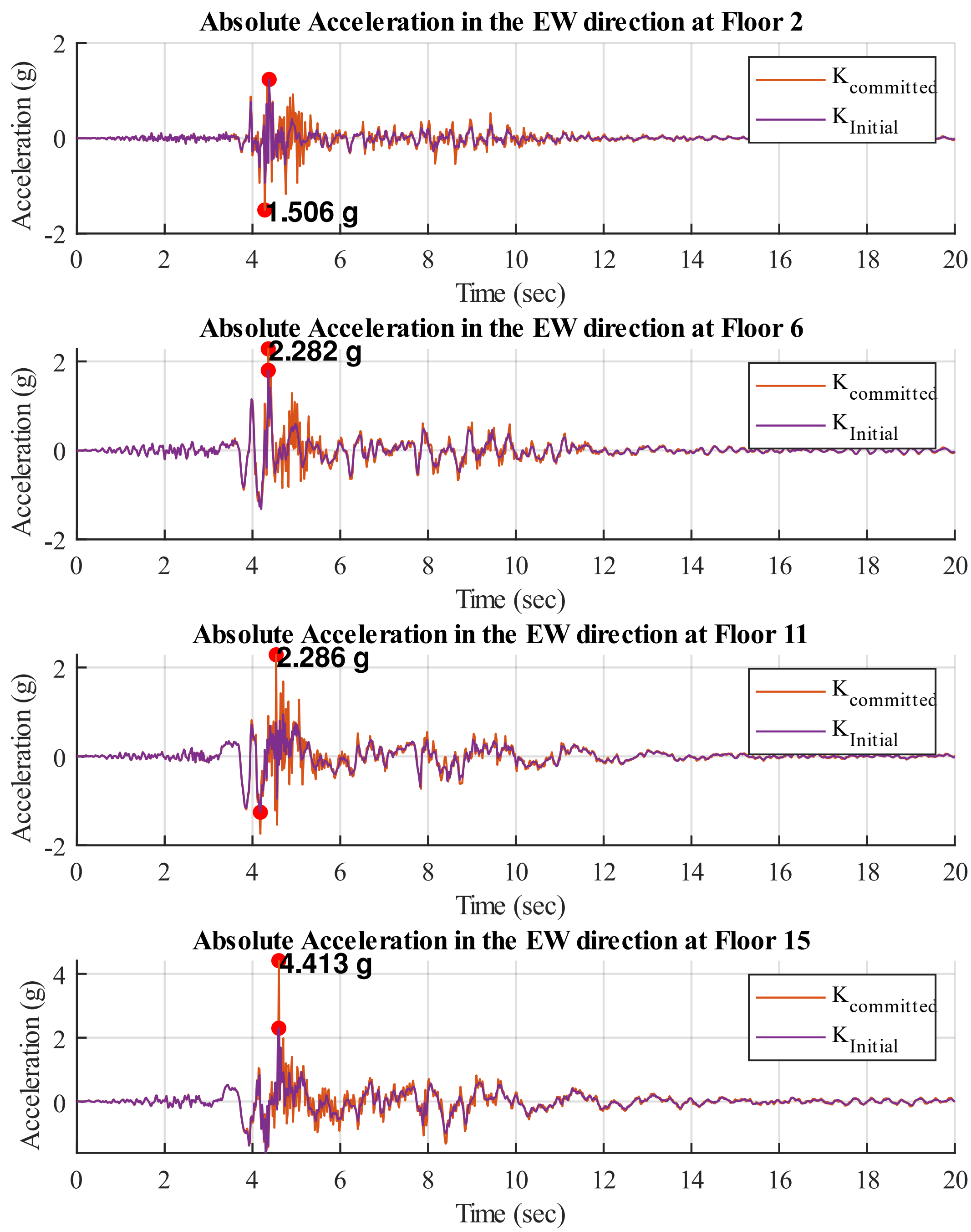


Figure 55 Comparison of Absolute Acceleration Response Under Dynamic for Different Damping Models

Relative Displacement Response

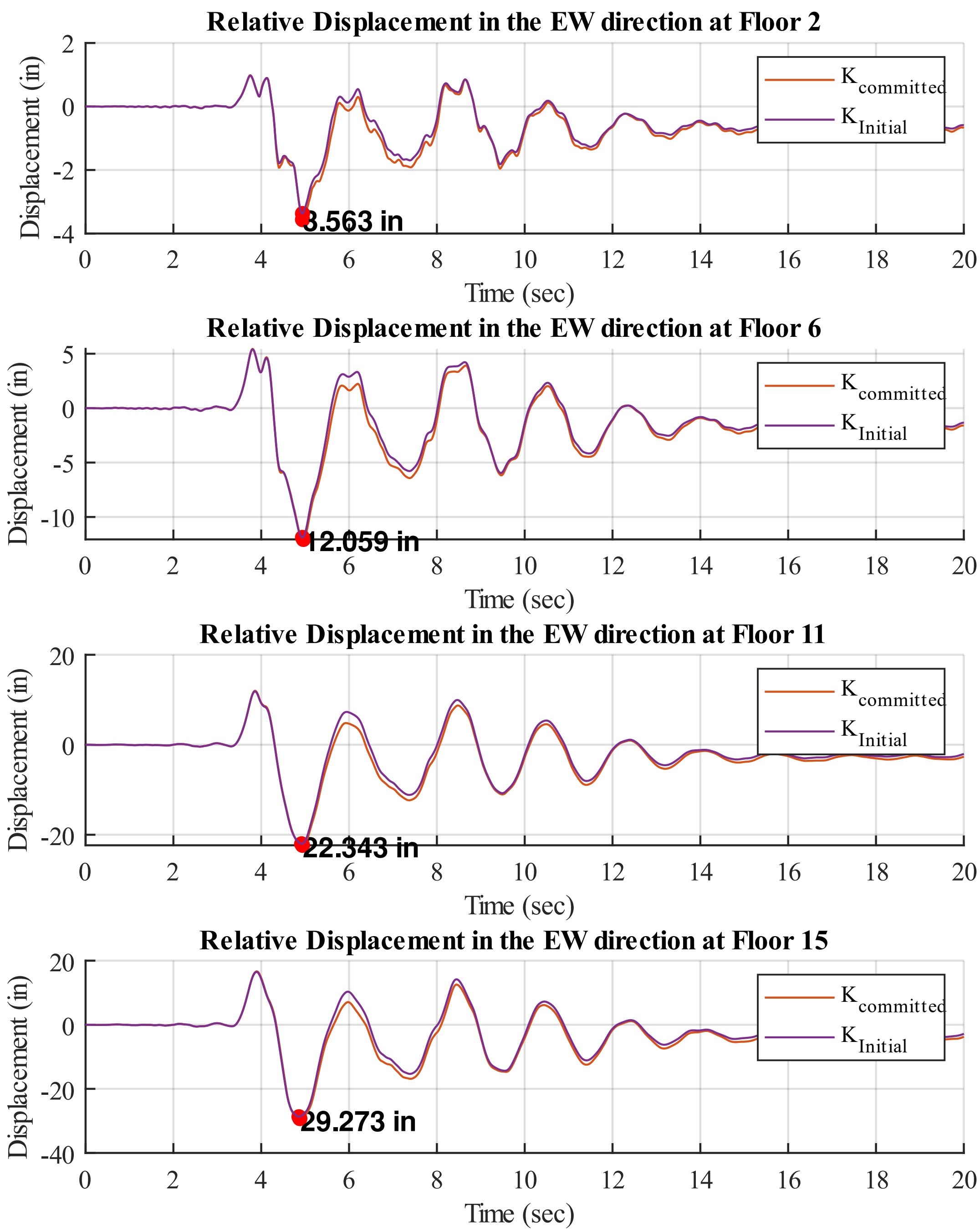


Figure 56 Comparison of Relative Displacement Response Under Dynamic for Different Damping Models

Part II Nonlinear Time History Analysis in the Bi-Axial Direction

Determine parameters for Raleigh Damping Model using Biaxial Modes

Figure 57 shows the damping ratio as a function of frequency after assigning two percent damping ratio to the first two modes of the structure. The two predominant modes are mode 1 and mode 2.

Natural frequencies of the two predominant modes are found by: $\omega = 2\pi/T$.

T_1 = 0.869, T_2 = 0.847

ω_1 = 7.2335, ω_2 = 7.4196

And the parameters a_0 and a_1 for the Rayleigh damping model is determined below equation, where $\xi_m = \xi_n = 0.02$.

$$\begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = 2\begin{bmatrix} \frac{1}{\omega_1} & \omega_1 \\ \frac{1}{\omega_2} & \omega_2 \end{bmatrix}^{-1} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} = \begin{bmatrix} 0.14651 \\ 0.0027298 \end{bmatrix}$$

Table 7 Natural Frequencies and Damping Ratios for Bi-directional Earthquake Motion

Modes	Natural Frequencies (1/s)	Damping Ratios
1	7.2335	0.02
2	7.4196	0.02
3	16.317	0.026761
4	31.646	0.045509
5	45.35	0.063514
6	60.38	0.083627

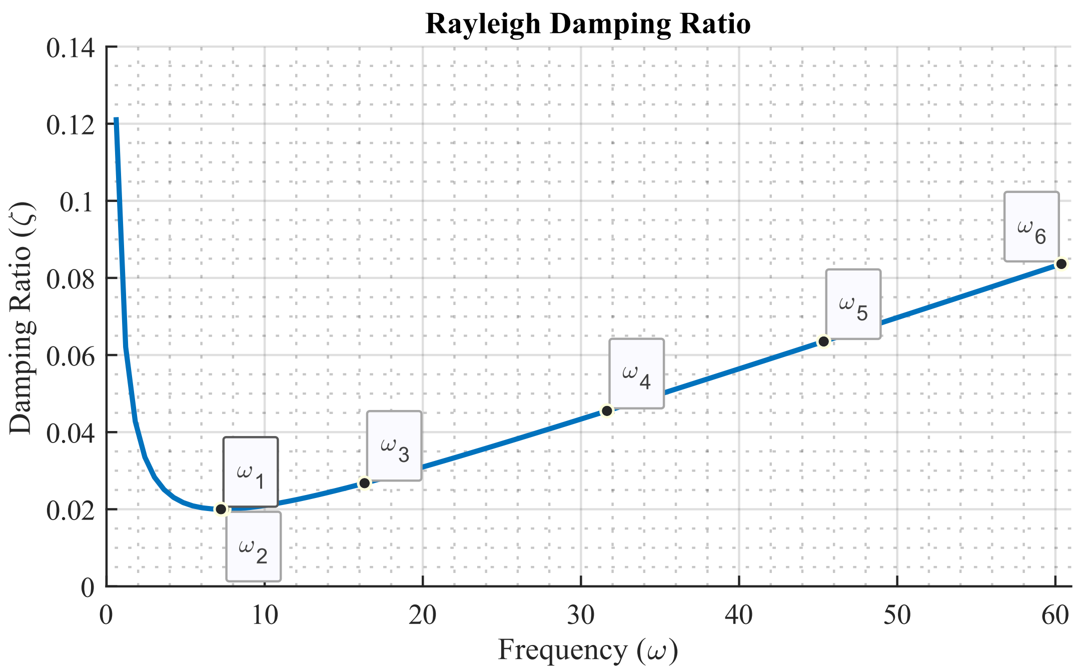


Figure 57 Damping ratio of Bi-directional Earthquake Motion

Overturning Moment

The overturning moment of NS direction is greater than that of EW direction. Also, the overturning moment in NS direction show more linear behavior than that in the EW direction. This is because the coupling beam activates in the EW direction influencing the structure to have more frame action. So, the structure movement in NS direction, which do not have the coupling effect will behave more like a cantilever.

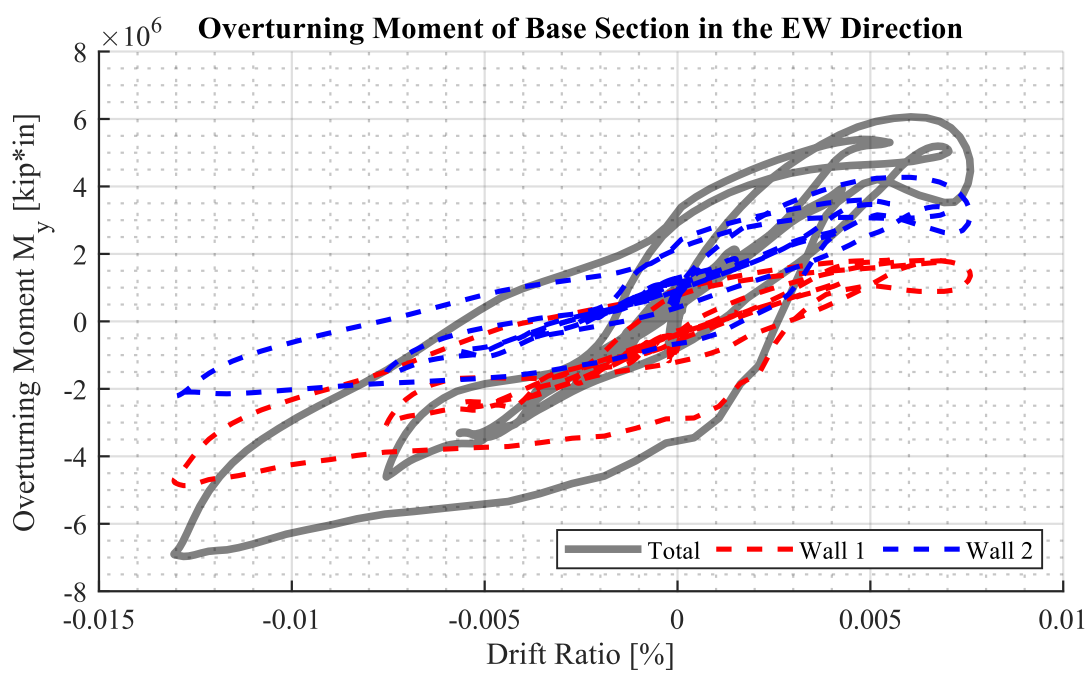


Figure 58 Total Base Overturning Moment in the EW Direction

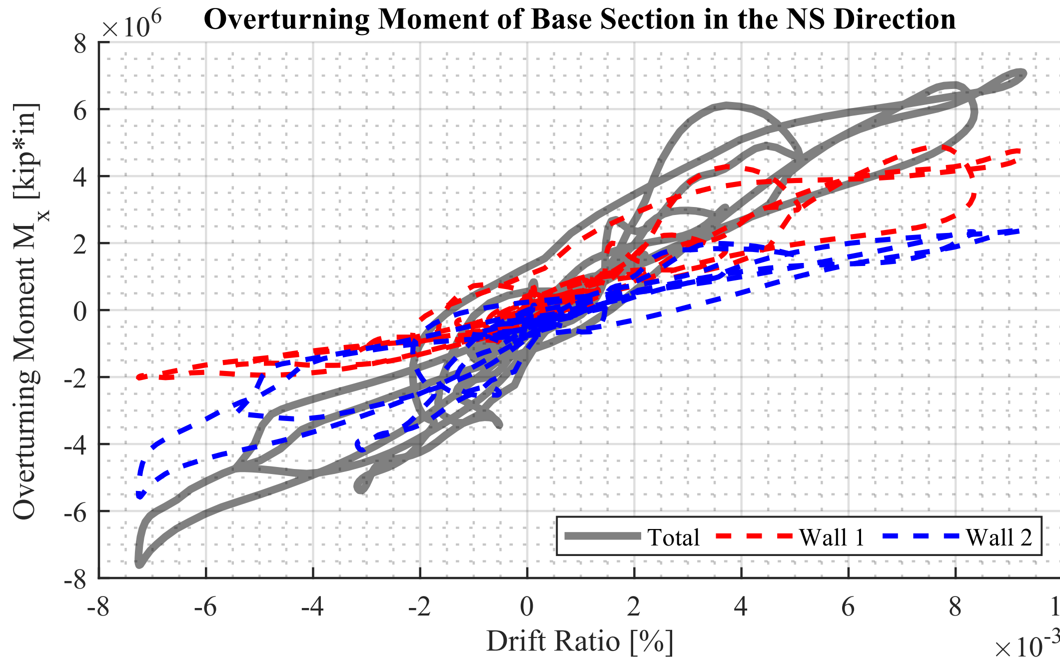


Figure 59 Total Base Overturning Moment in the NS Direction

<u>Absolute Acceleration Response for Selected Floors</u>

Absolute Acceleration in the NS direction have smeller average period than that in the EW direction because the structure is stiffer in the NS direction.

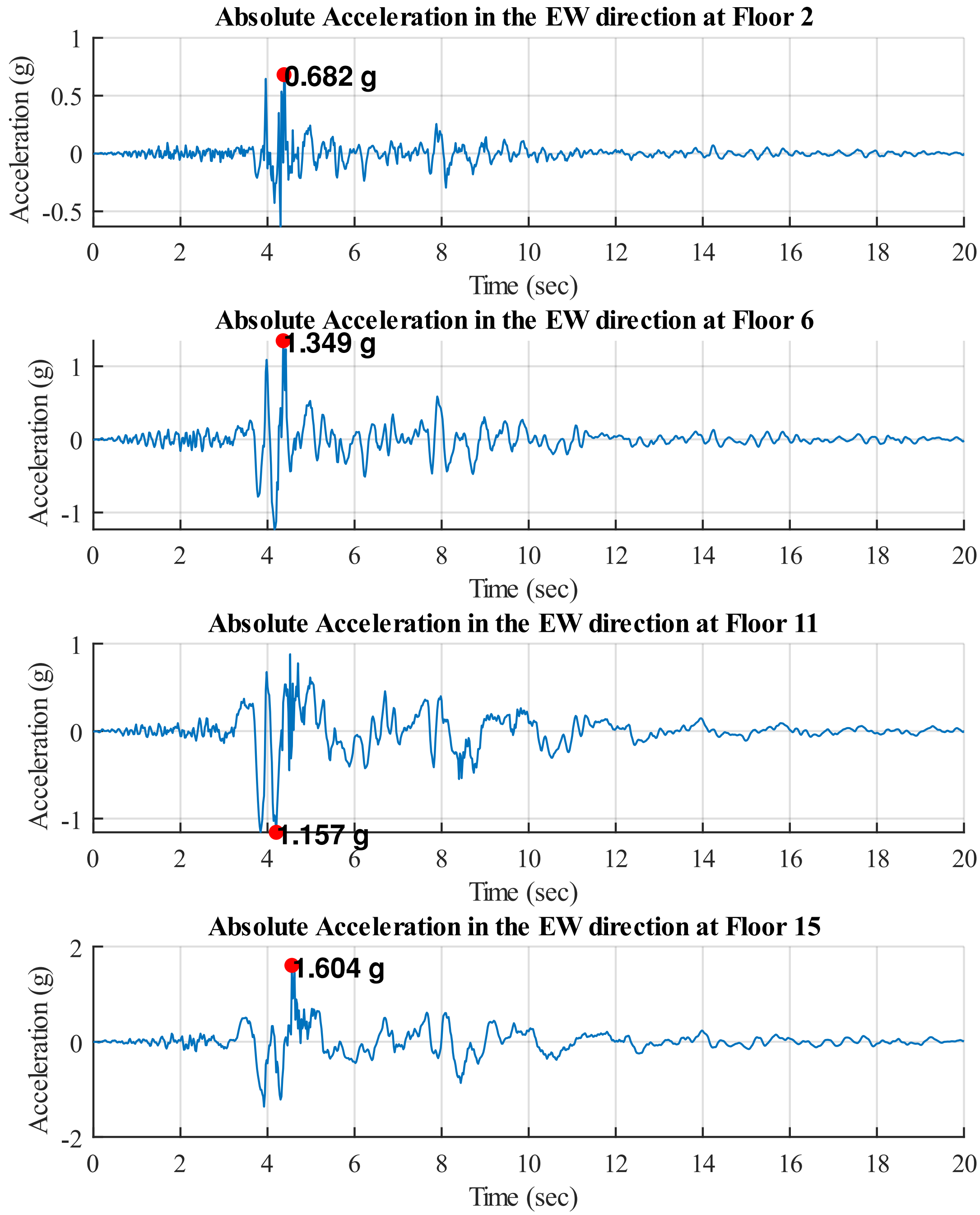


Figure 60 Time History of the Absolute Acceleration of the Wall 1 and Wall 2 in Selected Floors in the EW Direction

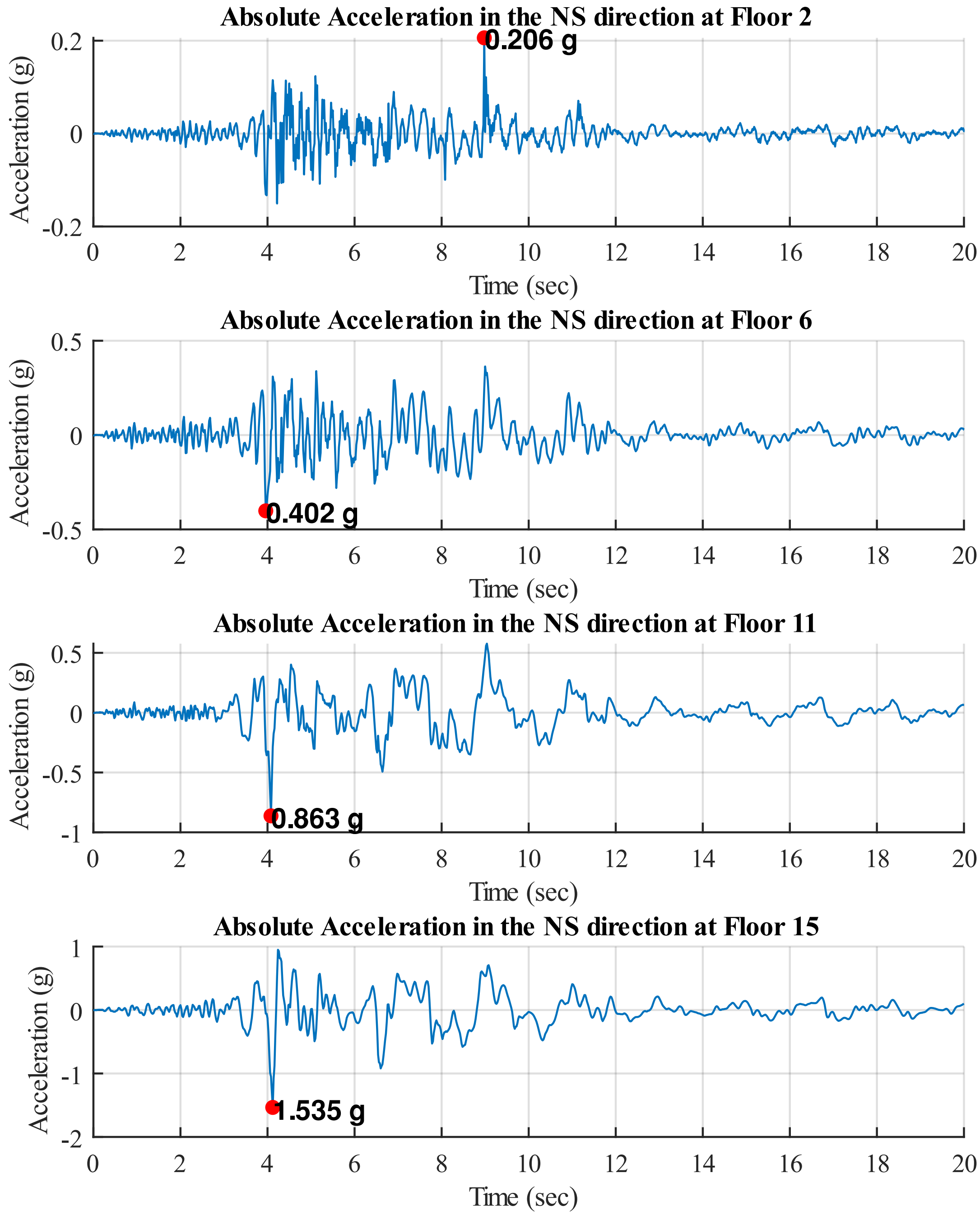


Figure 61 Time History of the Absolute Acceleration of the Wall 1 and Wall 2 in Selected Floors in the NS Direction

Displacements Response for Selected Floors

The displacement response is larger in the EW direction because the structure is less stiff in the EW direction where the walls are coupled by the coupling beams. And the structure is stiffer in the NS direction.

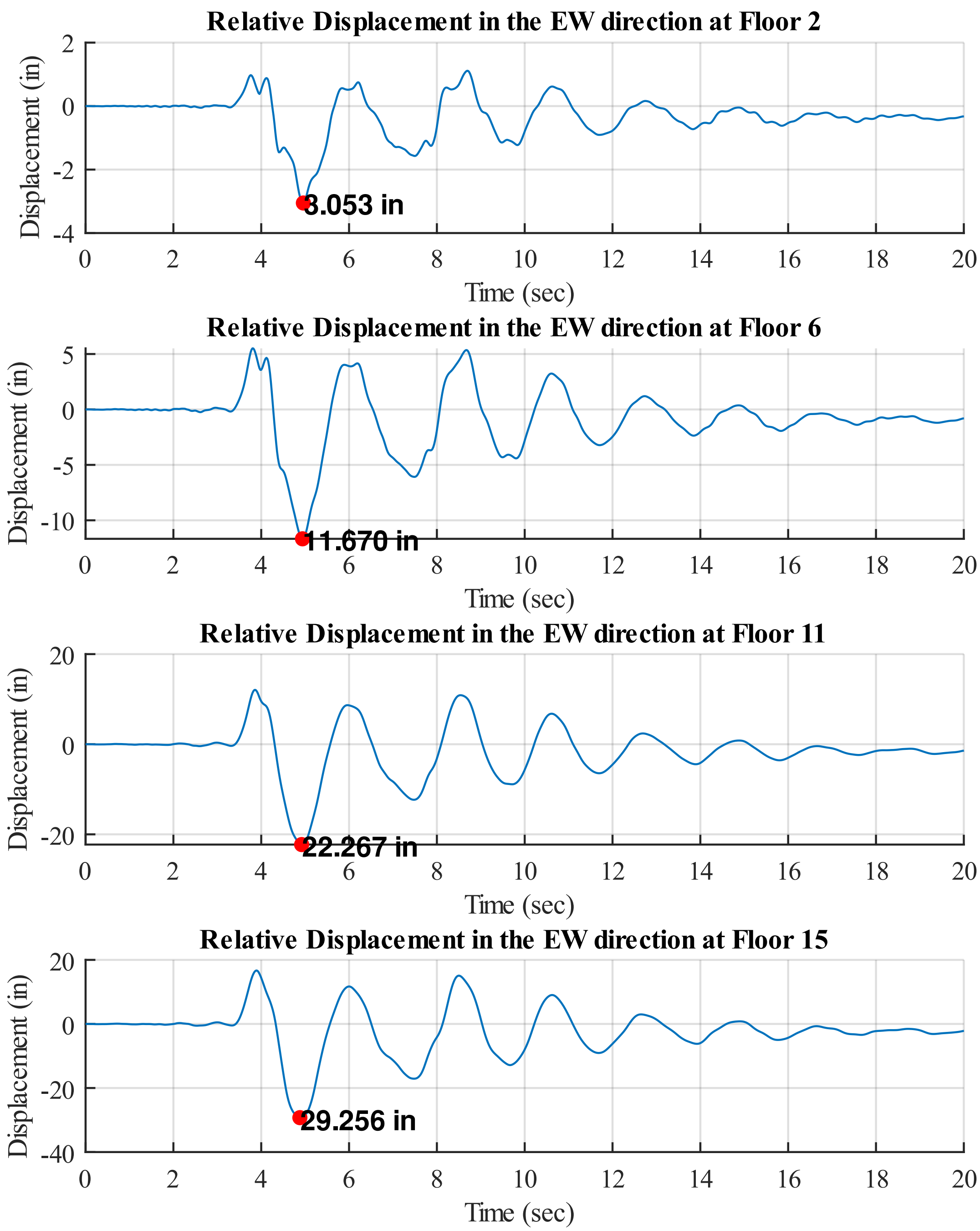


Figure 62 Time History of the Relative Displacements of the Wall 1 and Wall 2 in Selected Floors in the EW Direction

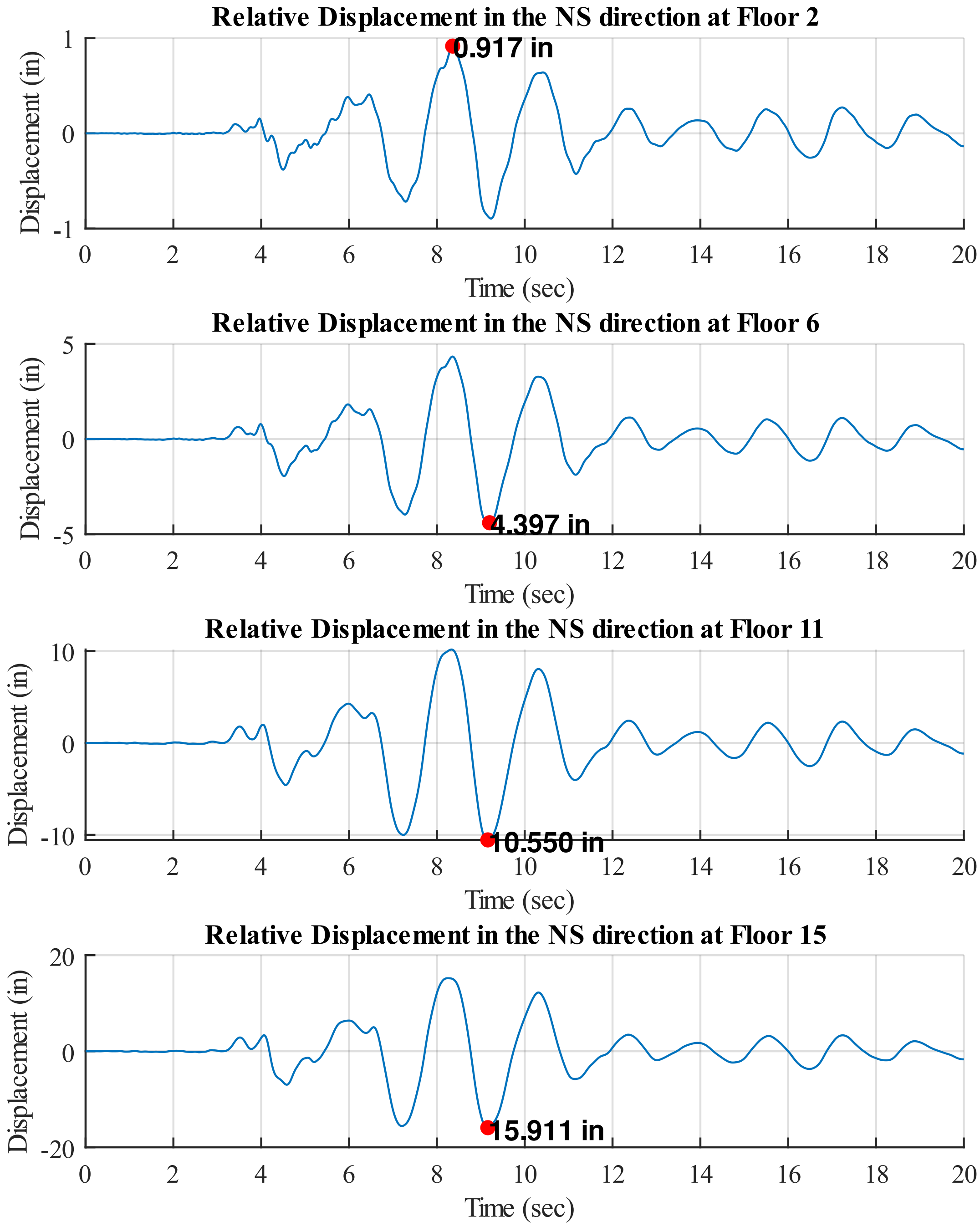


Figure 63 Time History of the Relative Displacements of the Wall 1 and Wall 2 in Selected Floors in the NS Direction

Figure 64 shows the NS and EW displacements. The displacement increases as the number of floor increases. And we can observe that the structure is stiffer in the NS direction because the displacement is larger to the EW direction.

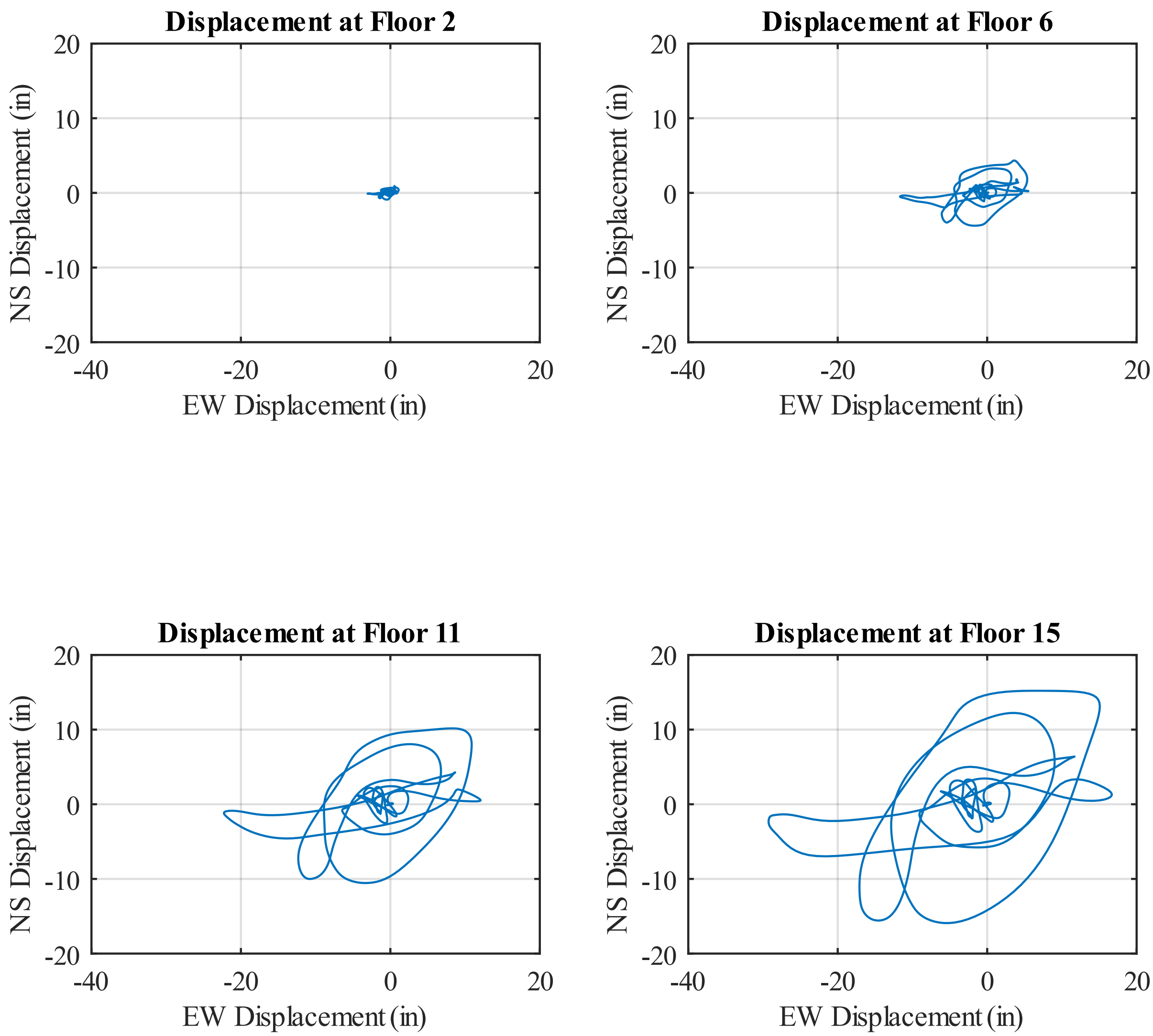


Figure 64 Relative Displacement in the NS vs EW Direction Under Dynamic Loading

The displacement increases along the height of the building. Similar shapes of displacements can be seen along the height of the building. The smallest displacement is at the base of the building.

Time History Normalized Base Shear

The normalized base shear is greater in the EW direction, and it means more mass is contributing to the dynamic effect in the EW direction. This is because the structure is less stiff in EW direction.

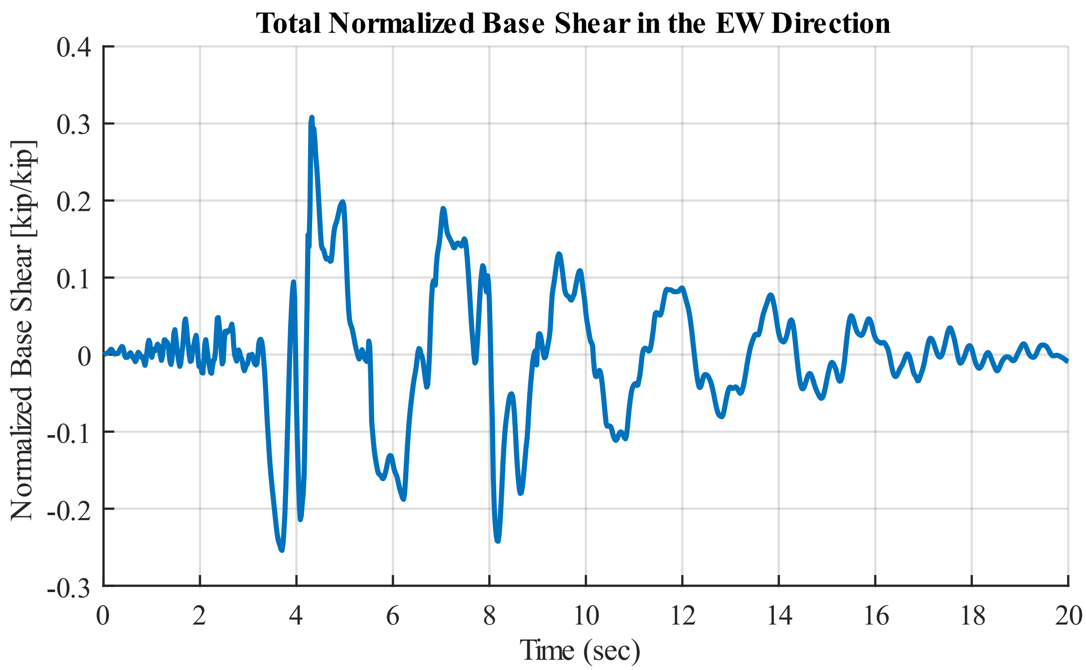


Figure 65 Total Normalized Base in the NS Direction Under Dynamic Loading

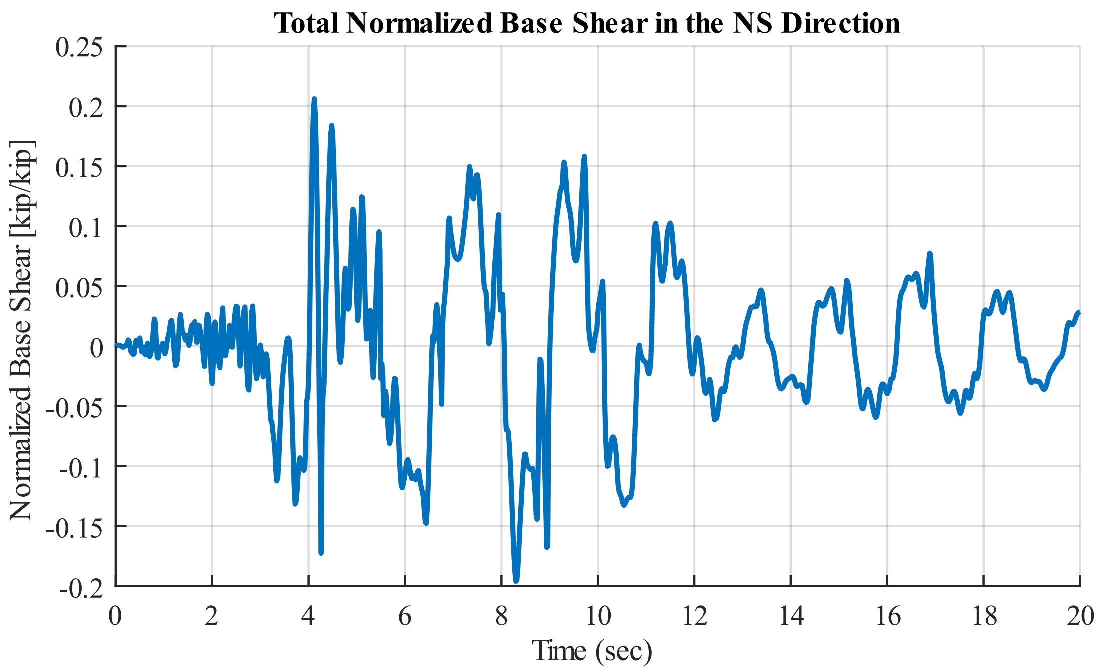


Figure 66 Total Normalized Base in the EW Direction Under Dynamic Loading

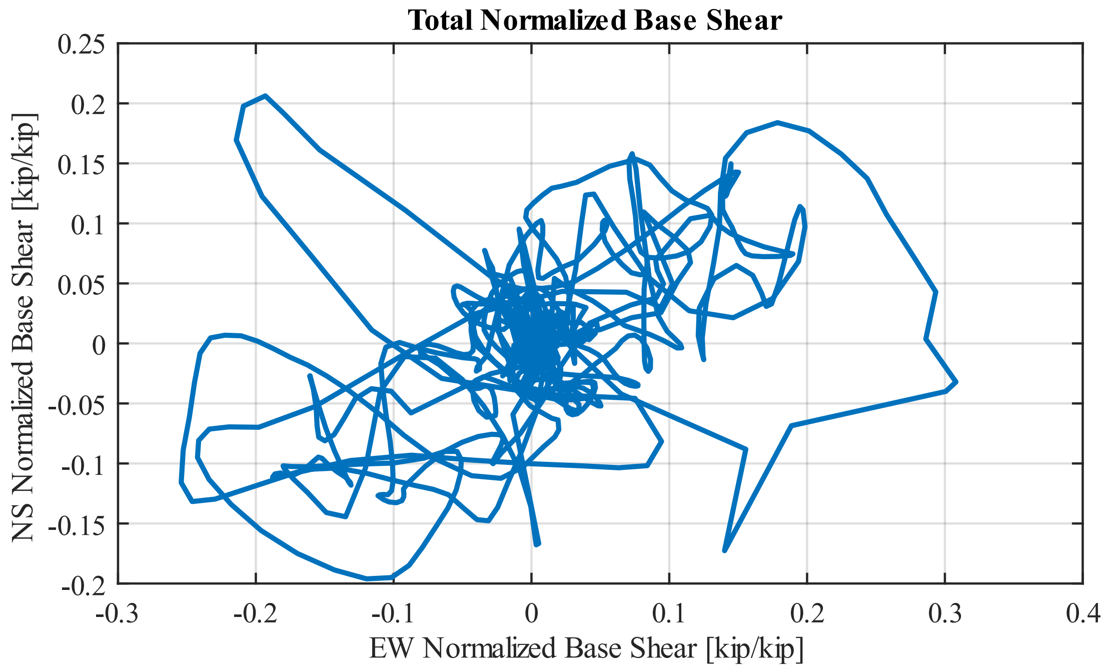


Figure 67 Total Normalized Base in the NS vs EW Direction Under Dynamic Loading

Moment Curvature

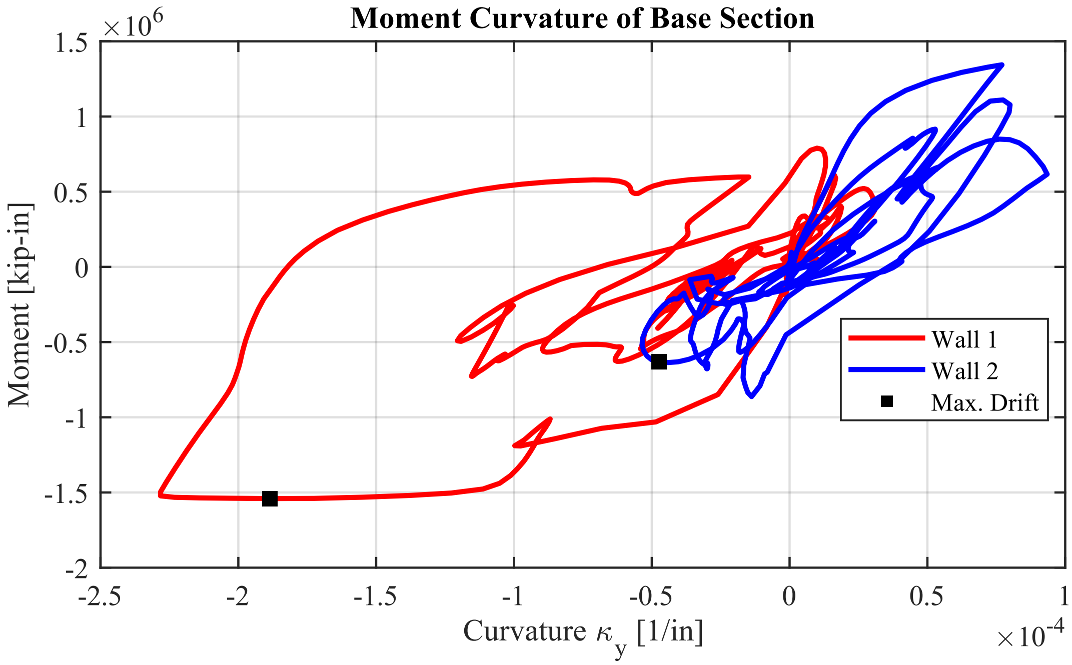


Figure 68 Moment Curvature Response of Wall 1 and Wall 2 in the EW Direction Under Bi-Directional Dynamic Loading

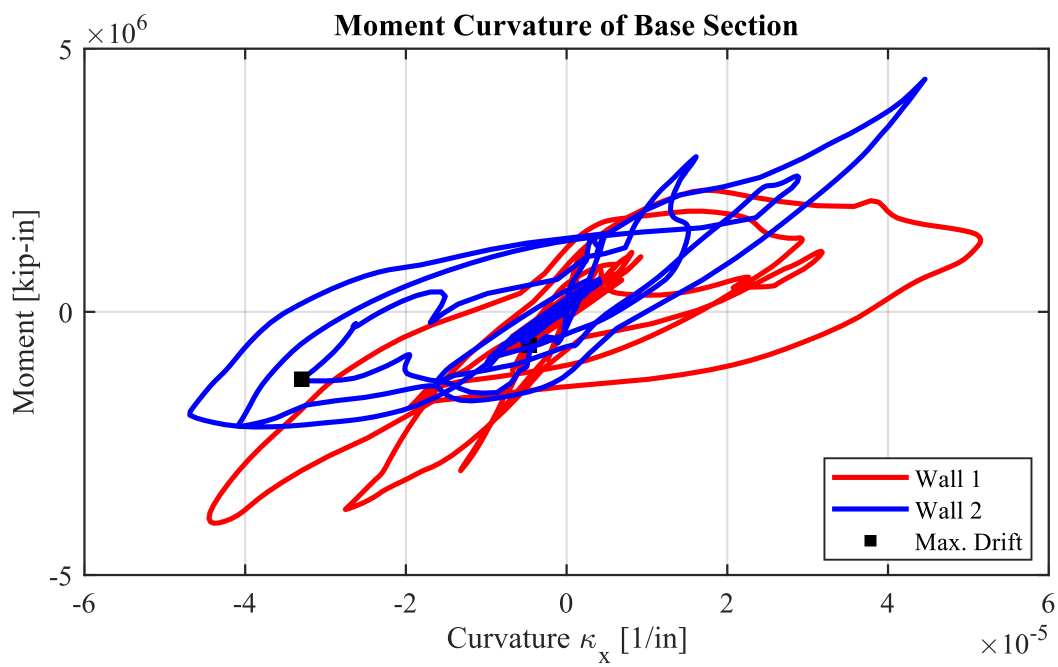


Figure 69 Moment Curvature Response of Wall 1 and Wall 2 in the NS Direction Under Bi-Directional Dynamic Loading

Appendix A MATLAB Files

```
function plotDamping(figName, folder, mode)
  arguments;
    figName = "temp";
    folder = "static_EW_force";
    mode = [1, 4];
  end

%    T_pre = load(".\"+folder+"\ModalAnalysis\Pre-gravity\periods.txt");
  T_post = load(".\"+folder+"\ModalAnalysis\Post-gravity\periods.txt");

   wn_post = 2* pi()./T_post;
   alpha_matrix_post = [1/wn_post(mode(1)) wn_post(mode(1))
                   1/wn_post(mode(2)) wn_post(mode(2))];
   alpha_damping_ratio = [0.02; 0.02];

   alphas_post = 2*inv(alpha_matrix_post)*alpha_damping_ratio;
   disp(alphas_post)
   damp_post = @(w) alphas_post(1)/2./w + alphas_post(2)/2*w;
   damping_ratios = damp_post(wn_post);

   w_post = linspace(0,60,100);
   d_post = damp_post(w_post);

   figure(1)
   hold on
   plot(w_post,d_post)

   for i = 1:6
%           wn_post(i)
           damping_ratios(i)
           plt = plot(wn_post(i),damping_ratios(i),"o");
           datatip(plt, 'DataIndex', i);
%           plt.DataTipTemplate.DataTipRows(1).Label = "\omega_" +i;
%           plt.DataTipTemplate.DataTipRows(1).Value = "";
%           plt.DataTipTemplate.DataTipRows(2) = [];
   end

%    ylim([0,0.05]);
   xlim([0,61]);
   h = findobj('Type','line'); set(h,'LineWidth',2,'MarkerSize',2,'MarkerFaceColor','none');
   grid on; grid minor;
   xlabel("Frequency (\omega)")
   ylabel("Damping Ratio (\zeta)")
   title("Rayleigh Damping Ratio")
end
```

```matlab
function plotNormBaseShear(figName, folder, direction, legendName)
  arguments;
    figName = "temp";
    folder = "dynamic_EW";
    direction = "EW"
    legendName = ""
  end


filename = ".\" +folder+"\Results_dynamic\Reaction.txt";
dataLines = [1, Inf];
opts = delimitedTextImportOptions("NumVariables", 13);
opts.DataLines = dataLines;
opts.Delimiter = " ";
opts.VariableNames = ["time", "Fx1", "Fy1", "Fz1", "Mx1", "My1", "Mz1", "Fx2", "Fy2", "Fz2", "Mx2", "My2", "Mz2"];
opts.VariableTypes = ["double", "double", "double", "double", "double", "double", "double", "double", "double", "double", "double", "double", "double"];
Reaction = readtable(filename, opts);

W = 21365;

figure(1); hold on;
if direction =="EW"
    plt = plot(Reaction.time,(Reaction.Fx1+ Reaction.Fx2)/W);
elseif direction =="NS"
    plt = plot(Reaction.time,(Reaction.Fy1+ Reaction.Fy2)/W);
end

if legendName ~= ""; plt.DisplayName = legendName; legend(); end;

title("Total Normalized Base Shear in the " + direction +" Direction");
grid on;
xlabel("Time (sec)");
ylabel("Normalized Base Shear [kip/kip]");
h = findobj('Type','line'); set(h,'LineWidth',2);
xlim([0,20]);
print_figure(figName)
end
```

```
function plotDamping(figName, folder, mode)
  arguments;
    figName = "temp";
    folder = "static_EW_force";
    mode = [1, 4];
  end

%     T_pre = load(".\"+folder+"\ModalAnalysis\Pre-gravity\periods.txt");
  T_post = load(".\"+folder+"\ModalAnalysis\Post-gravity\periods.txt");

   wn_post = 2* pi()./T_post;
   alpha_matrix_post = [1/wn_post(mode(1)) wn_post(mode(1))
                1/wn_post(mode(2)) wn_post(mode(2))];
   alpha_damping_ratio = [0.02; 0.02];

   alphas_post = 2*inv(alpha_matrix_post)*alpha_damping_ratio;
   disp(alphas_post)
   damp_post = @(w) alphas_post(1)/2./w + alphas_post(2)/2*w;
   damping_ratios = damp_post(wn_post);

   w_post = linspace(0,60,100);
   d_post = damp_post(w_post);

   figure(1)
   hold on
   plot(w_post,d_post)

   for i = 1:6
%          wn_post(i)
        damping_ratios(i)
        plt = plot(wn_post(i),damping_ratios(i),"o");
        datatip(plt, 'DataIndex', i);
%          plt.DataTipTemplate.DataTipRows(1).Label = "\omega_" +i;
%          plt.DataTipTemplate.DataTipRows(1).Value = "";
%          plt.DataTipTemplate.DataTipRows(2) = [];
   end

%     ylim([0,0.05]);
   xlim([0,61]);
   h = findobj('Type','line'); set(h,'LineWidth',2,'MarkerSize',2,'MarkerFaceColor','none');
   grid on; grid minor;
   xlabel("Frequency (\omega)")
   ylabel("Damping Ratio (\zeta)")
   title("Rayleigh Damping Ratio")
end
```

```
function plotAxialStrain(figName, folder, direction)
  arguments;
    figName = "temp";
    folder = "static_EW_force";
    direction = "EW";
  end

  try
    [eo1, Kx1, Ky1, ~] = readvars(".\" +folder+ "\Results_static\SecD_W1_base.txt");
     [eo2, Kx2, Ky2, ~] = readvars(".\" +folder+ "\Results_static\SecD_W2_base.txt");
   catch
     [eo1, Kx1, Ky1, ~] = readvars(".\" +folder+ "\Results_dynamic\SecD_W1_base.txt");
     [eo2, Kx2, Ky2, ~] = readvars(".\" +folder+ "\Results_dynamic\SecD_W2_base.txt");
   end

   figure(1); hold on
   set(gca,'DefaultLineLineWidth',2)
   if direction == "EW"
     K1 = Ky1; K2 = Ky2;
   elseif direction == "NS"
     K1 = Kx1; K2 = Kx2;
   end

   plot(K1,eo1,'r-','DisplayName','Wall 1') % Wall 1
   plot(-K2,eo2,'b-','DisplayName','Wall 2') % Wall 2

   grid on; legend('Location','northwest');
   xlabel('Curvature \kappa_y [1/in]');
   ylabel('Centroidal strain \epsilon_o [-]');
   title("Axial Strain Versus Curvature");
   grid on; legend('Location','northwest');

   if direction == "EW"
     xlabel('Curvature \kappa_y [1/in]');
   elseif direction == "NS"
     xlabel('Curvature \kappa_x [1/in]');
   end

   ylabel('Centroidal strain [-]');

   idx = getMaxDispIdx(folder);
   scatter(K1(idx),eo1(idx),'ks','filled','HandleVisibility','off') % Wall 1
   scatter(-K2(idx),eo2(idx),'ks','filled','DisplayName','Max. Drift') % Wall 2
   legend('Location','north')
   print_figure(figName)

end
```

```
function [ ] = plot_fiberSection(secDefFilePath, secTag, figNum, fibColor)
% SE 201B
% Fiber section plotter
% Angshuman Deb
%% INPUT
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% secDefFilePath     : full path to file containing section definiton(s)
%                    (see note)
% secTag             : secTag of section to plot
% figNum             : Matlab figure number
% fibColor (optional) : Matrix of matTags along first column and
%                    r, g, b, alpha(optional) values along
%                    2nd, 3rd, 4th, 5th(optional) columns
%----------------------------------------------------------------------

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Note about secDefFilePath
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% For using this plotter, generate a text file with the
% full section definition as you would write in a tcl input file.
% Provide the full path of this text file to this plotter.
%----------------------------------------------------------------------
%% READ SECTION DATA
secDefFilePath = convertStringsToChars(secDefFilePath);
fid = fopen(fullfile(secDefFilePath), 'r');
RectPatch = [];
QuadPatch = [];
CircPatch = [];
Fiber = [];
StraightLayer = [];
CircLayer = [];

% RectPatch
% $matTag $numSubdivY $numSubdivZ $yI $zI $yJ $zJ

% QuadPatch
% $matTag $numSubdivIJ $numSubdivJK $yI $zI $yJ $zJ $yK $zK $yL $zL

% CircPatch
% $matTag $numSubdivCirc $numSubdivRad $yCenter $zCenter $intRad $extRad $startAng $endAng

% Fiber
% $matTag $yLoc $zLoc $A

% StraightLayer
% $matTag $numFiber $areaFiber $yStart $zStart $yEnd $zEnd

% CircLayer
% $matTag $numFiber $areaFiber $yCenter $zCenter $radius <$startAng $endAng>

while feof(fid) ~= 1
    currLine = fgetl(fid);
    currLine = strsplit(strtrim(strtok(currLine,';')));
    currLine = currLine(~cellfun('isempty',currLine));
    if ~isempty(currLine)
        if strcmp(currLine{1}, 'section') && strcmp(currLine{2},'Fiber')
            secTagCurr = str2double(currLine{3});
```

```
            if secTag == secTagCurr
                secDefEnd = false;
                while ~secDefEnd
                    currSecLine = fgetl(fid);
                    currSecLine = strsplit(strtrim(strtok(currSecLine,';')));
                    currSecLine = currSecLine(~cellfun('isempty',currSecLine));
                    if ~isempty(currSecLine)
                        if strcmp(currSecLine{1},'patch')
                            if strcmp(currSecLine{2},'rect')
                                RectPatch = [RectPatch;str2double(currSecLine{3}) str2double(currSecLine{4}) str2double(currSecLine{5}) str2double(currSecLine{6}) str2double(currSecLine{7}) str2double(currSecLine{8}) str2double(currSecLine{9})];
                            elseif strcmp(currSecLine{2},'quad')
                                QuadPatch = [QuadPatch;str2double(currSecLine{3}) str2double(currSecLine{4}) str2double(currSecLine{5}) str2double(currSecLine{6}) str2double(currSecLine{7}) str2double(currSecLine{8}) str2double(currSecLine{9}) str2double(currSecLine{10}) str2double(currSecLine{11}) str2double(currSecLine{12}) str2double(currSecLine{13})];
                            elseif strcmp(currSecLine{2},'circ')
                                CircPatch = [CircPatch;str2double(currSecLine{3}) str2double(currSecLine{4}) str2double(currSecLine{5}) str2double(currSecLine{6}) str2double(currSecLine{7}) str2double(currSecLine{8}) str2double(currSecLine{9}) str2double(currSecLine{10}) str2double(currSecLine{11})];
                            end
                        elseif strcmp(currSecLine{1},'fiber')
                            Fiber = [Fiber;str2double(currSecLine{5}) str2double(currSecLine{2}) str2double(currSecLine{3}) str2double(currSecLine{4})];
                        elseif strcmp(currSecLine{1},'layer')
                            if strcmp(currSecLine{2},'straight')
                                StraightLayer = [StraightLayer;str2double(currSecLine{3}) str2double(currSecLine{4}) str2double(currSecLine{5}) str2double(currSecLine{6}) str2double(currSecLine{7}) str2double(currSecLine{8}) str2double(currSecLine{9})];
                            elseif strcmp(currSecLine{2},'circ')
                                CircLayer = [CircLayer;str2double(currSecLine{3}) str2double(currSecLine{4}) str2double(currSecLine{5}) str2double(currSecLine{6}) str2double(currSecLine{7}) str2double(currSecLine{8}) str2double(currSecLine{9}) str2double(currSecLine{10})];
                            end
                        elseif strcmp(currSecLine{1},'}')
                            secDefEnd = true;
                        end
                    end
                end
            end
        end
    end
end
fclose(fid);

if ~exist('fibColor', 'var')
    fibColor = NaN;
end
%% View Section
figure(figNum);hold on

for i1 = 1:size(RectPatch,1)
    matTag = RectPatch(i1,1);
    numSubdivY = RectPatch(i1,2);
    numSubdivZ = RectPatch(i1,3);
    yI = RectPatch(i1,4);
    zI = RectPatch(i1,5);
    yJ = RectPatch(i1,6);
    zJ = RectPatch(i1,7);

    yVec = linspace(yI,yJ,numSubdivY+1);
```

```
    zVec = linspace(zI,zJ,numSubdivZ+1);

    for i2 = 1:length(yVec)-1
        for i3 = 1:length(zVec)-1
            xPatch = [0,0,0,0];
            yPatch = [yVec(i2) yVec(i2+1) yVec(i2+1) yVec(i2)];
            zPatch = [zVec(i3) zVec(i3) zVec(i3+1) zVec(i3+1)];
            if ismember(matTag, fibColor(:,1))
                if size(fibColor,2) == 5
                    p = patch(xPatch,yPatch,zPatch, fibColor(fibColor(:,1) == matTag, 2:size(fibColor,2)-1));
                    p.FaceAlpha = fibColor(fibColor(:,1) == matTag,5);
                else
                    p = patch(xPatch,yPatch,zPatch, fibColor(fibColor(:,1) == matTag, 2:size(fibColor,2)));
                    p.FaceAlpha = 1;
                end
            else
                p = patch(xPatch,yPatch,zPatch,'k');
                p.FaceAlpha = 0.2;
            end
        end
    end

end

for i1 = 1:size(QuadPatch,1)
    matTag = QuadPatch(i1,1);
    numSubdivIJ = QuadPatch(i1,2);
    numSubdivJK = QuadPatch(i1,3);

    yI = QuadPatch(i1,4);
    zI = QuadPatch(i1,5);

    yJ = QuadPatch(i1,6);
    zJ = QuadPatch(i1,7);

    yK = QuadPatch(i1,8);
    zK = QuadPatch(i1,9);

    yL = QuadPatch(i1,10);
    zL = QuadPatch(i1,11);

    yIJ_vec = linspace(yI,yJ,numSubdivIJ+1);
    zIJ_vec = linspace(zI,zJ,numSubdivIJ+1);

    yJK_vec = linspace(yJ,yK,numSubdivJK+1);
    zJK_vec = linspace(zJ,zK,numSubdivJK+1);

    yLK_vec = linspace(yL,yK,numSubdivIJ+1);
    zLK_vec = linspace(zL,zK,numSubdivIJ+1);

    yIL_vec = linspace(yI,yL,numSubdivJK+1);
    zIL_vec = linspace(zI,zL,numSubdivJK+1);

    numLines_IJ = numSubdivIJ + 1;
    numLines_JK = numSubdivJK + 1;

    for i2 = 1:(numLines_IJ-1)
```

```
    for i3 = 1:(numLines_JK-1)

        if i2 == 1 && i3 == 1
            y1 = yIL_vec(i3);
            z1 = zIL_vec(i3);

            y2 = yIJ_vec(i2+1);
            z2 = zIJ_vec(i2+1);

            [y3,z3] = polyxpoly(...
                [yIJ_vec(i2+1),yLK_vec(i2+1)],...
                [zIJ_vec(i2+1),zLK_vec(i2+1)],...
                [yIL_vec(i3+1),yJK_vec(i3+1)],...
                [zIL_vec(i3+1),zJK_vec(i3+1)]);

            y4 = yIL_vec(i3+1);
            z4 = zIL_vec(i3+1);
        elseif i2 == 1 && i3 == (numLines_JK-1)
            y1 = yIL_vec(i3);
            z1 = zIL_vec(i3);

            [y2,z2] = polyxpoly(...
                [yIJ_vec(i2+1),yLK_vec(i2+1)],...
                [zIJ_vec(i2+1),zLK_vec(i2+1)],...
                [yIL_vec(i3),yJK_vec(i3)],...
                [zIL_vec(i3),zJK_vec(i3)]);

            y3 = yLK_vec(i2+1);
            z3 = zLK_vec(i2+1);

            y4 = yIL_vec(i3+1);
            z4 = zIL_vec(i3+1);
        elseif i2 == (numLines_IJ-1) && i3 == 1
            y1 = yIJ_vec(i2);
            z1 = zIJ_vec(i2);

            y2 = yIJ_vec(i2+1);
            z2 = zIJ_vec(i2+1);

            y3 = yJK_vec(i3+1);
            z3 = zJK_vec(i3+1);

            [y4,z4] = polyxpoly(...
                [yIJ_vec(i2),yLK_vec(i2)],...
                [zIJ_vec(i2),zLK_vec(i2)],...
                [yIL_vec(i3+1),yJK_vec(i3+1)],...
                [zIL_vec(i3+1),zJK_vec(i3+1)]);
        elseif i2 == (numLines_IJ-1) && i3 == (numLines_JK - 1)
            [y1,z1] = polyxpoly(...
                [yIJ_vec(i2),yLK_vec(i2)],...
                [zIJ_vec(i2),zLK_vec(i2)],...
                [yIL_vec(i3),yJK_vec(i3)],...
                [zIL_vec(i3),zJK_vec(i3)]);

            y2 = yJK_vec(i3);
            z2 = zJK_vec(i3);

```

```
        y3 = yJK_vec(i3+1);
        z3 = zJK_vec(i3+1);

        y4 = yLK_vec(i2);
        z4 = zLK_vec(i2);
    elseif i2 == 1 && ( 1 < i3 < (numLines_JK - 1))
        y1 = yIL_vec(i3);
        z1 = zIL_vec(i3);

        [y2,z2] = polyxpoly(...
            [yIJ_vec(i2+1),yLK_vec(i2+1)],...
            [zIJ_vec(i2+1),zLK_vec(i2+1)],...
            [yIL_vec(i3),yJK_vec(i3)],...
            [zIL_vec(i3),zJK_vec(i3)]);

        [y3,z3] = polyxpoly(...
            [yIJ_vec(i2+1),yLK_vec(i2+1)],...
            [zIJ_vec(i2+1),zLK_vec(i2+1)],...
            [yIL_vec(i3+1),yJK_vec(i3+1)],...
            [zIL_vec(i3+1),zJK_vec(i3+1)]);

        y4 = yIL_vec(i3+1);
        z4 = zIL_vec(i3+1);
    elseif i2 == (numLines_IJ-1) && ( 1 < i3 < (numLines_JK - 1))
        [y1,z1] = polyxpoly(...
            [yIJ_vec(i2),yLK_vec(i2)],...
            [zIJ_vec(i2),zLK_vec(i2)],...
            [yIL_vec(i3),yJK_vec(i3)],...
            [zIL_vec(i3),zJK_vec(i3)]);

        y2 = yJK_vec(i3);
        z2 = zJK_vec(i3);

        y3 = yJK_vec(i3+1);
        z3 = zJK_vec(i3+1);

        [y4,z4] = polyxpoly(...
            [yIJ_vec(i2),yLK_vec(i2)],...
            [zIJ_vec(i2),zLK_vec(i2)],...
            [yIL_vec(i3+1),yJK_vec(i3+1)],...
            [zIL_vec(i3+1),zJK_vec(i3+1)]);
    elseif (1 < i2 < (numLines_IJ-1)) && i3 == 1
        y1 = yIJ_vec(i2);
        z1 = zIJ_vec(i2);

        y2 = yIJ_vec(i2+1);
        z2 = zIJ_vec(i2+1);

        [y3,z3] = polyxpoly(...
            [yIJ_vec(i2+1),yLK_vec(i2+1)],...
            [zIJ_vec(i2+1),zLK_vec(i2+1)],...
            [yIL_vec(i3+1),yJK_vec(i3+1)],...
            [zIL_vec(i3+1),zJK_vec(i3+1)]);
        [y4,z4] = polyxpoly(...
            [yIJ_vec(i2),yLK_vec(i2)],...
            [zIJ_vec(i2),zLK_vec(i2)],...
            [yIL_vec(i3+1),yJK_vec(i3+1)],...
```

```
            [zIL_vec(i3+1),zJK_vec(i3+1)]);
        elseif (1 < i2 < (numLines_IJ-1)) && i3 == (numLines_JK-1)
          [y1,z1] = polyxpoly(...
            [yIJ_vec(i2),yLK_vec(i2)],...
            [zIJ_vec(i2),zLK_vec(i2)],...
            [yIL_vec(i3),yJK_vec(i3)],...
            [zIL_vec(i3),zJK_vec(i3)]);

          [y2,z2] = polyxpoly(...
            [yIJ_vec(i2+1),yLK_vec(i2+1)],...
            [zIJ_vec(i2+1),zLK_vec(i2+1)],...
            [yIL_vec(i3),yJK_vec(i3)],...
            [zIL_vec(i3),zJK_vec(i3)]);

          y3 = yLK_vec(i2+1);
          z3 = zLK_vec(i2+1);

          y4 = yLK_vec(i2);
          z4 = zLK_vec(i2);
        else
          [y1,z1] = polyxpoly(...
            [yIJ_vec(i2),yLK_vec(i2)],...
            [zIJ_vec(i2),zLK_vec(i2)],...
            [yIL_vec(i3),yJK_vec(i3)],...
            [zIL_vec(i3),zJK_vec(i3)]);
          [y2,z2] = polyxpoly(...
            [yIJ_vec(i2+1),yLK_vec(i2+1)],...
            [zIJ_vec(i2+1),zLK_vec(i2+1)],...
            [yIL_vec(i3),yJK_vec(i3)],...
            [zIL_vec(i3),zJK_vec(i3)]);
          [y3,z3] = polyxpoly(...
            [yIJ_vec(i2+1),yLK_vec(i2+1)],...
            [zIJ_vec(i2+1),zLK_vec(i2+1)],...
            [yIL_vec(i3+1),yJK_vec(i3+1)],...
            [zIL_vec(i3+1),zJK_vec(i3+1)]);
          [y4,z4] = polyxpoly(...
            [yIJ_vec(i2),yLK_vec(i2)],...
            [zIJ_vec(i2),zLK_vec(i2)],...
            [yIL_vec(i3+1),yJK_vec(i3+1)],...
            [zIL_vec(i3+1),zJK_vec(i3+1)]);
        end

        yPatch = [y1,y2,y3,y4];
        zPatch = [z1,z2,z3,z4];
        xPatch = zeros(size(yPatch));

        if ismember(matTag, fibColor(:,1))
          if size(fibColor,2) == 5
            p = patch(xPatch,yPatch,zPatch, fibColor(fibColor(:,1) == matTag, 2:size(fibColor,2)-1));
            p.FaceAlpha = fibColor(fibColor(:,1) == matTag,5);
          else
            p = patch(xPatch,yPatch,zPatch, fibColor(fibColor(:,1) == matTag, 2:size(fibColor,2)));
            p.FaceAlpha = 1;
          end
        else
          p = patch(xPatch,yPatch,zPatch,'k');
          p.FaceAlpha = 0.2;
```

```
            end
        end
    end
end

for i1 = 1:size(CircPatch,1)
    matTag = CircPatch(i1,1);
    numSubdivCirc = CircPatch(i1,2);
    numSubdivRad = CircPatch(i1,3);
    yCenter = CircPatch(i1,4);
    zCenter = CircPatch(i1,5);
    intRad = CircPatch(i1,6);
    extRad = CircPatch(i1,7);
    startAng = CircPatch(i1,8)*pi/180;
    endAng = CircPatch(i1,9)*pi/180;

    rVec = linspace(intRad,extRad,numSubdivRad+1);
    thetaVec = linspace(startAng,endAng,numSubdivCirc+1);

    for i2 = 1:length(rVec)-1
        for i3 = 1:length(thetaVec)-1
            xPatch = [0,0,0,0];
            yPatch = yCenter + [rVec(i2)*cos(thetaVec(i3)) rVec(i2+1)*cos(thetaVec(i3)) rVec(i2+1)*cos(thetaVec(i3+1)) rVec(i2)*cos(thetaVec(i3+1))];
            zPatch = zCenter + [rVec(i2)*sin(thetaVec(i3)) rVec(i2+1)*sin(thetaVec(i3)) rVec(i2+1)*sin(thetaVec(i3+1)) rVec(i2)*sin(thetaVec(i3+1))];
            if ismember(matTag, fibColor(:,1))
                if size(fibColor,2) == 5
                    p = patch(xPatch,yPatch,zPatch, fibColor(fibColor(:,1) == matTag, 2:size(fibColor,2)-1));
                    p.FaceAlpha = fibColor(fibColor(:,1) == matTag,5);
                else
                    p = patch(xPatch,yPatch,zPatch, fibColor(fibColor(:,1) == matTag, 2:size(fibColor,2)));
                    p.FaceAlpha = 1;
                end
            else
                p = patch(xPatch,yPatch,zPatch,'k');
                p.FaceAlpha = 0.2;
            end
        end
    end
end

for i1 = 1:size(CircLayer,1)
    matTag = CircLayer(i1,1);
    numFiber = CircLayer(i1,2);
    areaFiber = CircLayer(i1,3);
    yCenter = CircLayer(i1,4);
    zCenter = CircLayer(i1,5);
    radius = CircLayer(i1,6);

    if length(CircLayer(i1,:)) > 6
        startAng = CircLayer(i1,7)*pi/180;
        endAng = CircLayer(i1,8)*pi/180;
    else
        startAng = 0.0;
        endAng = (360 - 360/numFiber)*pi/180;
    end
```

```

    rVec = linspace(radius,radius,numFiber);
    thetaVec = linspace(startAng,endAng,numFiber);
    yVec = yCenter + rVec.*cos(thetaVec);
    zVec = zCenter + rVec.*sin(thetaVec);

    for i2 = 1:length(yVec)
        if ismember(matTag, fibColor(:,1))
            filledCircle([0,yVec(i2),zVec(i2)],sqrt(areaFiber/pi),100, fibColor(fibColor(:,1) == matTag,2:size(fibColor,2)));
        else
            filledCircle([0,yVec(i2),zVec(i2)],sqrt(areaFiber/pi),100,'k');
        end
    end
end

for i1 = 1:size(StraightLayer,1)
    matTag = StraightLayer(i1,1);
    numFiber = StraightLayer(i1,2);
    areaFiber = StraightLayer(i1,3);
    yStart = StraightLayer(i1,4);
    zStart = StraightLayer(i1,5);
    yEnd = StraightLayer(i1,6);
    zEnd = StraightLayer(i1,7);

    yVec = linspace(yStart,yEnd,numFiber);
    zVec = linspace(zStart,zEnd,numFiber);

    for i2 = 1:length(yVec)
        if ismember(matTag, fibColor(:,1))
            filledCircle([0,yVec(i2),zVec(i2)],sqrt(areaFiber/pi),100, fibColor(fibColor(:,1) == matTag,2:size(fibColor,2)));
        else
            filledCircle([0,yVec(i2),zVec(i2)],sqrt(areaFiber/pi),100,'k');
        end
    end
end

for i1 = 1:size(Fiber,1)
    matTag = Fiber(i1,1);
    yVec = Fiber(i1,2);
    zVec = Fiber(i1,3);
    areaFiber = Fiber(i1,4);
    if ismember(matTag, fibColor(:,1))
        filledCircle([0,yVec,zVec],sqrt(areaFiber/pi),100, fibColor(fibColor(:,1) == matTag,2:size(fibColor,2)));
    else
        filledCircle([0,yVec,zVec],sqrt(areaFiber/pi),100,'k');
    end
end

ylabel('y')
zlabel('z')
view([1,0,0])
box on;
axis square;
axis equal;

end

```

```
% Helper function to plot filled circle
function h = filledCircle(center,r,N,color)
if isnumeric(color) && length(color) == 4
    alpha = color(4);
    color = color(1:3);
else
    alpha = 1;
end
THETA=linspace(0,2*pi,N);
RHO=ones(1,N)*r;
[Y,Z] = pol2cart(THETA,RHO);
Y=Y+center(2);
Z=Z+center(3);
X = zeros(size(Y));
h=patch(X,Y,Z,color);
h.FaceAlpha = alpha;
end
```

```
function plt = plotHorDisp(figName, folder, nNodeColumn, direction, legendName)
  arguments;
     figName = "temp";
     folder = "static_EW_force";
     nNodeColumn = 1;
     direction = "EW";
     legendName = "";
  end

   if direction == "EW"; DOF = 1; elseif direction == "NS"; DOF =2; end

   idx = getMaxDispIdx(folder);
   idx = 657;

   Nstories = 15;

   for wall = [1,2]
      displace = [];
      subplot(1,2,wall); hold on;
      for i = 1:Nstories
         for n = 1:nNodeColumn
            dispNode = load(".\" + folder + "\DeflectedShape\dispNode_" + num2str(1000*i +wall*100 + n) +".txt");
            displace = [displace, dispNode(idx,DOF)];
         end
      end
      height = 16:12/nNodeColumn:184;

      if nNodeColumn == 2
         height = [0, 8 , height];
      else
         height = [0, height];
      end

      plt = plot(abs([0, displace]),height,'-o');
      yticks([0, 16:12:184]);
      ylim([0,184])
      grid on;
      xlabel('Horizontal Displacement [in]');
      ylabel('Story Height [ft]');

      title("Wall "+wall);
      if legendName ~= ""; plt.DisplayName = legendName; legend('location','best'); end

   end
   sgtitle("Horizontal Displacement in the " +direction+" direction",'FontName','Times');
   h = findobj('Type','line'); set(h,'LineWidth',2,'MarkerSize',2,'MarkerFaceColor','none');
   print_figure(figName,[6.5,4.5]);
end
```

```
function plotMomentCurvature(figName, folder, direction)
   arguments;
      figName = "temp";
      folder = "static_EW_force";
      direction = "EW";
   end

   idx = getMaxDispIdx(folder);

    try
       [~, kx, ky, ~] = readvars(".\" +folder+ "\Results_static\SecD_W1_base.txt");
       [~, Mx, My, ~] = readvars(".\" +folder+ "\Results_static\SecF_W1_base.txt");
    catch
       [~, kx, ky, ~] = readvars(".\" +folder+ "\Results_dynamic\SecD_W1_base.txt");
       [~, Mx, My, ~] = readvars(".\" +folder+ "\Results_dynamic\SecF_W1_base.txt");
    end

    figure(1); hold on;
    set(gca,'DefaultLineLineWidth',2)
    title('Moment Curvature of Base Section');
    if direction == "EW"
       plot(ky,My,'r','DisplayName','Wall 1');
       scatter(ky(idx),My(idx),50,'ks','filled','HandleVisibility','off') % Wall 1
    elseif direction == "NS"
       plot(kx,Mx,'r','DisplayName','Wall 1');
       scatter(kx(idx),Mx(idx),50,'ks','filled','HandleVisibility','off') % Wall 1
    end

    try
       [~, kx, ky, ~] = readvars(".\" +folder+ "\Results_static\SecD_W2_base.txt");
       [~, Mx, My, ~] = readvars(".\" +folder+ "\Results_static\SecF_W2_base.txt");
    catch
       [~, kx, ky, ~] = readvars(".\" +folder+ "\Results_dynamic\SecD_W2_base.txt");
       [~, Mx, My, ~] = readvars(".\" +folder+ "\Results_dynamic\SecF_W2_base.txt");
    end

    if direction == "EW"
       plot(-ky,-My,'b','DisplayName','Wall 2');
       scatter(-ky(idx),-My(idx),50,'ks','filled','DisplayName','Max. Drift') % Wall 2
    elseif direction == "NS"
       plot(-kx,-Mx,'b','DisplayName','Wall 2');
       scatter(-kx(idx),-Mx(idx),50,'ks','filled','DisplayName','Max. Drift') % Wall 2
    end

    legend('Location','Best');
    grid on;
    box on;
    if direction == "EW"
       xlabel('Curvature \kappa_y [1/in]');
    elseif direction == "NS"
       xlabel('Curvature \kappa_x [1/in]');
    end

    ylabel('Moment [kip-in]');

    print_figure(figName)
end
```

```matlab
function [ fig, undefPlot, defPlot ] = plot_modeShape(modelDefFilePath, modeShapeDirPath, mode, scale, nDim, crdTransfMatrix, undefColor, undefLineWidth, defColor, defLineWidth, figNum)
%% DESCRIPTION

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% INPUT
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% modelDefFilePath         : full path to file containing model definition
%                          (see note)
% modeShapeDirPath         : full path to directory containing
%                          modeShape_$mode.txt file (see note)
% mode                     : mode number to plot
% scale                    : scale factor for mode shape
% nDim                     : number of dimensions (2 or 3)
% crdTransfMatrix          : coordinate transformation matrix (see note)
% undefColor               : color for undeformed shape
% undefLineWidth           : line width for undeformed shape
% defColor                 : color for deformed shape
% defLineWidth             : line width for deformed shape
% figNum                   : Matlab figure number
%-------------------------------------------------------------------------

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Note about modelDefFilePath
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% For using this plotter, generate a text output of your model as you
% write the .tcl input file. This text output should at least have all
% the nodal and element information. As you go on adding nodes and
% elements in the .tcl input file, it is required to write the command
% lines for adding nodes and elements to this text file. Provide the
% full path of this text file.
%-------------------------------------------------------------------------

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Note about modeShape_$mode.txt
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Save modeShape_$mode.txt for mode number = $mode. This file should have
% 1+6 columns for nDim = 3.
% 1+3 columns for nDim = 2.
% Column 1 should have tags of all nodes in the model
% Columns 2:end should have the node eigenvectors of all nodes in the model
% for mode number = $mode.
%-------------------------------------------------------------------------

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Note about crdTransfMatrix (= R):
% m: MATLAB
% o: OpenSees
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% [oX;oY;oZ] = R*[mX;mY;mZ]
% [mX;mY;mZ] = R'*[oX;oY;oZ]
% Assume (mX, mY, mZ) as basis
% 1st row of R = oX in (mX, mY, mZ)
% 2nd row of R = oY in (mX, mY, mZ)
% 3rd row of R = oZ in (mX, mY, mZ)
%-------------------------------------------------------------------------
%% READ MODEL DATA
```

```

[modelDefFilePath, modeShapeDirPath] = convertStringsToChars(modelDefFilePath, modeShapeDirPath);

if ~ismember(nDim,[2,3])
    error('Incorrect dimension! Should be 2 or 3.')
end

modelDataFile_fid = fopen(fullfile(modelDefFilePath,'modelData.txt'),'r');
str = textscan(modelDataFile_fid,'%s');
nodeCount = sum(ismember(str{:},'node'));
eleCount = sum(ismember(str{:},'element')) + sum(ismember(str{:},'rigidLink'));
nodeData = zeros(nodeCount, nDim + 1);
eleData = zeros(eleCount, 3);
eleTypes = cell(eleCount, 1);
nodeCtr = 0;
eleCtr = 0;
frewind(modelDataFile_fid);

while ~feof(modelDataFile_fid)
    currLine = fgetl(modelDataFile_fid);
    currLine = strtrim(strtok(currLine,';'));
    currLine = strsplit(currLine);

    if size(currLine,2) > 1 && strcmp(currLine{1},'node') == 1
        nodeCtr = nodeCtr + 1;
        nodeNum = str2double(currLine{2});
        coordinates = arrayfun(@(x) str2double(currLine{x}), 3:length(currLine), 'UniformOutput', 1);
        nodeData(nodeCtr,:) = [nodeNum coordinates];
    end

    if (size(currLine,2) > 1 && strcmp(currLine{1},'element') == 1) || (size(currLine,2) > 1 && strcmp(currLine{1},'rigidLink') == 1)
        eleCtr = eleCtr + 1;
        if strcmp(currLine{1},'rigidLink') == 1
            eleNum = 0;
            connectingNodes = [str2double(currLine{4}),str2double(currLine{3})];
            eleType = currLine{1};
        else
            eleNum = str2double(currLine{3});
            connectingNodes = [str2double(currLine{4}),str2double(currLine{5})];
            eleType = currLine{2};
        end
        eleData(eleCtr,:) = [eleNum connectingNodes];
        eleTypes{eleCtr} = eleType;
    end
end
fclose(modelDataFile_fid);

nodeData(:,2:size(nodeData,2)) = (crdTransfMatrix'*nodeData(:,2:size(nodeData,2))')';

%% PLOT UNDEFORMED SHAPE
fig = figure(figNum);
axis equal
hold on
grid on
box on
[~,element_iNode] = ismember(eleData(:,2),nodeData(:,1));
[~,element_jNode] = ismember(eleData(:,3),nodeData(:,1));
```

```
if nDim == 3
    for i = 1:size(eleData,1)
        undefPlot = ...
            plot3([nodeData(element_iNode(i),2) nodeData(element_jNode(i),2)],...
            [nodeData(element_iNode(i),3) nodeData(element_jNode(i),3)],...
            [nodeData(element_iNode(i),4) nodeData(element_jNode(i),4)],...
            '--','Color',undefColor,'LineWidth',undefLineWidth);hold on
    end
elseif nDim == 2
    for i = 1:size(eleData,1)
        undefPlot = ...
            plot([nodeData(element_iNode(i),2) nodeData(element_jNode(i),2)],...
            [nodeData(element_iNode(i),3) nodeData(element_jNode(i),3)],...
            '--','Color',undefColor,'LineWidth',undefLineWidth);hold on
    end
end

%% PLOT MODE SHAPE
%% READ MODE SHAPE INFORMATION
nodeDataDeformed = zeros(size(nodeData));
nodeEigenVector = load(fullfile(modeShapeDirPath,['modeShape_' num2str(mode) '.txt']));

for i = 1:size(nodeData,1)
    [~, ind] = ismember(nodeData(i,1), nodeEigenVector(:,1));
    if nDim == 2
        nodeDataDeformed(i,:) = [nodeData(i,1) nodeEigenVector(ind,2:size(nodeEigenVector,2)-1)];
    elseif nDim == 3
        nodeDataDeformed(i,:) = [nodeData(i,1) nodeEigenVector(ind,2:size(nodeEigenVector,2)-3)];
    end
end

nodeDataDeformed(:,2:size(nodeDataDeformed,2)) = (crdTransfMatrix'*nodeDataDeformed(:,2:size(nodeDataDeformed,2))')';
nodeDataDeformed(:,2:size(nodeDataDeformed,2)) = scale*nodeDataDeformed(:,2:size(nodeDataDeformed,2)) + nodeData(:,2:size(nodeData,2));
%% PLOT MODE SHAPE
figure(figNum)
if nDim == 3
    for i = 1:size(eleData,1)
        defPlot = ...
            plot3([nodeDataDeformed(element_iNode(i),2) nodeDataDeformed(element_jNode(i),2)],...
            [nodeDataDeformed(element_iNode(i),3) nodeDataDeformed(element_jNode(i),3)],...
            [nodeDataDeformed(element_iNode(i),4) nodeDataDeformed(element_jNode(i),4)],...
            'LineStyle','-','Color',defColor,'LineWidth',defLineWidth);
    end
    plot3(nodeDataDeformed(:,2),nodeDataDeformed(:,3),nodeDataDeformed(:,4),'ks','LineWidth',1,'MarkerFaceColor',[0.5 0.5 0.5])
elseif nDim == 2
    for i = 1:size(eleData,1)
        defPlot = ...
            plot([nodeDataDeformed(element_iNode(i),2) nodeDataDeformed(element_jNode(i),2)],...
            [nodeDataDeformed(element_iNode(i),3) nodeDataDeformed(element_jNode(i),3)],...
            'LineStyle','-','Color',defColor,'LineWidth',defLineWidth);
    end
    plot(nodeDataDeformed(:,2),nodeDataDeformed(:,3),'ks','LineWidth',1,'MarkerFaceColor',[0.5 0.5 0.5])
end


end
```

```
function plotModeShapeAll(folder)
    modelDefFilePath = ".\"+folder+"\Model";
    modeShapeDirPath = ".\"+folder+"\ModalAnalysis\Pre-gravity\";
for i = 1:6
    mode = i;
    scale = 500;
    nDim = 3;
    loadStep = 2;
    defScale = 10;
    crdTransfMatrix = eye(3);
    undefColor = 'b';
    undefLineWidth = 1.5;
    defColor = 'r';
    defLineWidth = 1.5;
    figNum = i;
    plotModeShape(modelDefFilePath, modeShapeDirPath, mode, scale, nDim, crdTransfMatrix, undefColor, undefLineWidth, defColor, defLineWidth, figNum);
    grid off;
    h = findobj('Type','line'); set(h,'LineWidth',1,'MarkerSize',1,'MarkerFaceColor','none');

    xlabel("EW");
    ylabel("NS");
    title("Mode " +i);

    set(gca,'xtick',[],'ytick',[],'ztick',[])
    set(gca,'xticklabel',[],'yticklabel',[],'zticklabel',[])

%     view(30,45)
%     print_figure("Mode "+i + " 3D View",[2,4]);

%     axis([-200 500 -200 500 0 2400])
%     view(0,0)
%     print_figure("Mode "+i+ " EW View",[2,4]);
    axis([-200 500 -400 300 0 2400])
    print_figure("Mode "+i+ " Top View",[2,2]);
end
```

```
%% Axial Moment
function plotElementAxialMoment(figName, folder, direction, beamType, nNodeColumn, flip, legendName,  GL, Nstories)
  arguments
    figName = "temp";
    folder = "static_EW_force";
    direction = 'EW';
    beamType = 'forceBeamColumn';
    nNodeColumn = 1;
    flip = -1;
     legendName ="Element Forces";
     GL = 5;
     Nstories = 15;
   end

   idx = getMaxDispIdx(folder);
   idx = 657;
   totalheights = [];
   culumheight = zeros(1,GL);
   A = []; My = [];
   wall = 2;

   for i = 1:Nstories
      for n = 1:nNodeColumn
         global_ele_force = load(".\"+folder+"\Results_Static\GlobEleF_" + beamType +num2str(1000*i +wall*100 + n)+ ".txt");
         local_heights = load(".\"+folder+"\Results_Static\IntPts_" + beamType + num2str(1000*i +wall*100 + n) +".txt");
         culumheight = culumheight(5) + local_heights(idx,:);
         totalheights = [totalheights, culumheight(1), culumheight(5)];
         A = [A, global_ele_force(idx, 3),-global_ele_force(idx, 9)];
         My= [My, global_ele_force(idx, 5),-global_ele_force(idx, 11)];
      end
   end

   figure(1);
   subplot(1,2,1); hold on;
   plot(flip*[A,0],[totalheights,totalheights(end)]/12,'-.d','DisplayName',legendName);
   grid on; xlabel('Axial [kip]');
   ylabel('Story Height [ft]');
   legend('Location','northwest');
   xline(0,'HandleVisibility','off');
   height = 16:12:184;
   yticks(height);
   ylim([0,184])

   subplot(1,2,2);  hold on;
   plot(flip*[My,0],[totalheights,totalheights(end)]/12,'-.d','DisplayName',legendName);
   grid on; xlabel('Moment [kip-in]');
   ylabel('Story Height [ft]');
   legend('Location','Northeast')
   xline(0,'HandleVisibility','off');
   yticks(height);
   ylim([0,184])

   h = findobj('Type','line'); set(h,'LineWidth',2,'MarkerSize',2,'MarkerFaceColor','none');
   sgtitle("Forces along the Builiding",'FontName','Times');
   print_figure(figName,[6.5,4.5])
end
```

```
function plotOTM(figName, folder, direction)
  arguments;
    figName = "temp";
    folder = "dynamic_EW";
    direction = "EW"
  end

Reaction = load(".\" +folder +"\Results_dynamic\Reaction.txt");
Disp = load(".\" +folder +"\Results_dynamic\Disp.txt");

figure(1); hold on;
title("Overturning Moment of Base Section in the " +direction+" Direction");

x_arm = 139;
y_arm = 62.48;
builidngHeight = 2208;

if direction =="EW"
   OTM1 = Reaction(:,6) + Reaction(:,4)*x_arm;
   OTM2 = Reaction(:,12) - Reaction(:,10)*x_arm;
   drift = Disp(:,2)/builidngHeight;
elseif direction =="NS"
   OTM1 = Reaction(:,5) - Reaction(:,4)*y_arm;
   OTM2 = Reaction(:,11) + Reaction(:,10)*y_arm;
   drift = -Disp(:,3)/builidngHeight;
end

OTM_total = OTM1 + OTM2;


plt = plot(drift ,-OTM_total ,'k','DisplayName','Total','LineWidth',3);  plt.Color(4) = 0.5;
plot(drift ,-OTM1 ,'r--','DisplayName','Wall 1','LineWidth',2);
plot(drift ,-OTM2,'b--','DisplayName','Wall 2','LineWidth',2);
xlabel("Drift Ratio [%]");

if direction =="EW"
   ylabel("Overturning Moment M_y [kip*in]");
elseif direction =="NS"
   ylabel("Overturning Moment M_x [kip*in]");
end
legend('Location','southeast','Orientation','horizontal');
grid on
grid minor

print_figure(figName)
end

```

```
function plotRecord(record)

data = load(record);  % Loads in time and acceleration
iter = data(1);
du = data(2);
time = (du:du:iter*du);
acc= data(3:iter+2);
A = acc;
[~, idx] = min(max(abs(A))-abs(A));
plot(time(idx),A(idx),'ro','MarkerFaceColor','r','HandleVisibility','off')
text(time(idx),A(idx),sprintf('  A_{abs, max} = %1.3fg',abs(A(idx))),'VerticalAlignment','middle','HorizontalAlignment','left','FontSize',12,'FontWeight','Bold');

plot(time,acc); grid on;
title(record); xlabel("Time [sec]");ylabel("Acceleration [g]");
print_figure("temp",[6.5, 2.25])
end
```

```matlab
function plotSection(figName, folder, secTag, size)
arguments
    figName = "temp"
    folder = "static_EW_force"
    secTag = 1;
    size = [3.25,2.5];
end

secDefFilePath = folder +"/Model/modelData.txt";
grid off;
figNum = 1;

fibColor = [1 0 0.85 0.5 0.5;
            2 1 0 0 0.5;
            3 .75 .75 .75 0.5
            4 0 0 1 0.5
            5 .0 0 1 0.5];
plotFiberSection(secDefFilePath, secTag, figNum, fibColor)
camroll(90)
title(figName);
ylabel('Y [in]');
zlabel('X [in]');
print_figure(figName,size)
end
```

```
addpath('.\function');
% run_opensees(".\static_EW_force\run.tcl");
% run_opensees(".\static_EW_disp\run.tcl");
% run_opensees(".\static_EW_lin\run.tcl");
% run_opensees(".\static_NS_force\run.tcl");
% run_opensees(".\dynamic_EW\run.tcl");
% run_opensees(".\dynamic_EW_kcommit\run.tcl");
% run_opensees(".\dynamic_EWNS\run.tcl");
% run_opensees(".\dynamic_EW_lin\run.tcl");
% run_opensees(".\dynamic_EW_dt\run.tcl");

```