CEE247: Earthquake Hazard Mitigation

Group 5

Date Submitted: 6/13/2019

~Khalid Alsadhan~

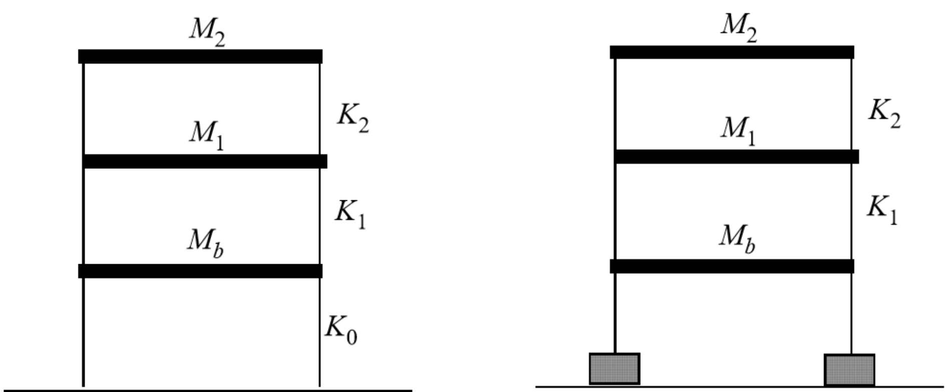
~Antoine Hascoat~

~Emilien Marotte~

~Thomas Viguier~

Project Statement

A three-story shear-type RC building is to be retrofitted with isolation devices at its first story. The un-retrofitted building can be idealized into a three-degree freedom linear system, whose mass and stiffness properties are shown in Figure below. The structural damping is approximated with Rayleigh damping, whose parameters are determined based on 2% modal damping for the first two modes. The footprint of the building is about 18m by 18m and the total height of the building is 12m with 4m story height. *Use SI units for all your calculations and results*.



Fixed-Base Structure **Isolated Structure**

$M_1 = M_2 = M_b = 325000kg$ $K_1 = K_2 = K_0 = 250 MN/m$

1. The design earthquake motions include 7 ground motions (listed in Table 1 below) recorded in past earthquakes in California. Their data files as well as plot files are posted on CCLE. Construct 5% damping response spectrum (relative displacement and total acceleration) for each individual earthquake and the averaged response spectrum based on all earthquakes. Use the response spectrum as guidance to choose the target isolation period of the base-isolated building. When isolation is used, the base story is replaced with the isolation devices whose behavior is nonlinear.

2. Formulate the equation of motion for the fixed-base building using the relative displacements to ground as unknowns (i.e., u_{s0}, u_{s1} and u_{s2} represents the relative displacement of base story, first story and the second story to the ground respectively). Obtain the natural periods and natural modes of the fixed-base building. Construct a linear time history analysis program in MATLAB for the fixed-base building using the built in ODE solver. Verify the solutions of your program by comparing to results by modal analysis or Newmark method. Compute the inter-story drifts, floor total accelerations and base shears under the design earthquake motions.

Table 1. Earthquake records selected for design and simulation

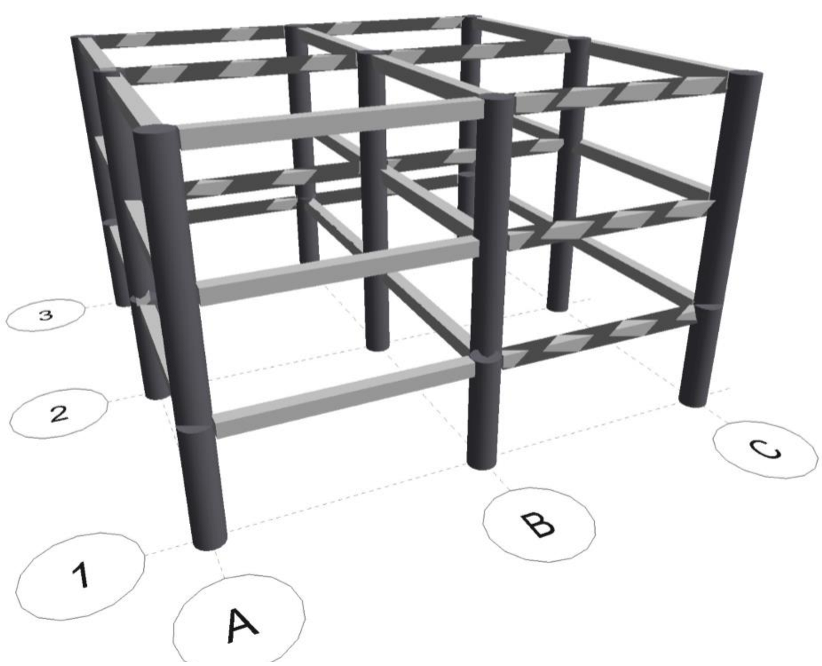
Record Station	**Earthquake**	**Magnitude M_w**	**Distance to Fault (km)**	**Peak Acceleration (g)**	**Peak Velocity (m/s)**
Pacoima Dam	1971 San Fernando	6.6	8.5	1.17	1.14
El Centro Array #5	1979 Imperial Valley	6.4	30.4	0.38	0.99
El Centro Array #7	1979 Imperial Valley	6.4	29.4	0.46	1.13
Lucerne Valley	1992 Landers	7.3	42.0	0.71	1.36
Rinaldi	1994 Northridge	6.7	9.9	0.89	1.75
Sylmar	1994 Northridge	6.7	12.3	0.73	1.22
Newhall	1994 Northridge	6.7	20.2	0.59	0.96

3. Construct a program (in Excel or Matlab) to conduct bearing design for the building based on the target isolation period and the design displacement from Part 1. Present in details a sample bearing design that includes the number of bearings needed, type of bearing, dimensions of bearing and nonlinear mechanical properties of the bearing for numerical analysis along with all applicable model parameters.

4. Formulate the equations of motion for the base-isolated building using the relative displacements to ground as unknowns (i.e., u_{s1} and u_{s2} represents the relative displacement of the first story and the second story to the ground respectively while u_b represents the relative displacement in isolator to ground). Construct a nonlinear time history analysis program in MATLAB for base-isolated building where Bouc-Wen model is used to model the nonlinear behavior of isolation devices. Use your program to compute the inter-story drifts, floor total accelerations and the base shears of the building above the isolation layer and compare with the fixed-base building case. Compute the displacement and shear in the isolation unit.

5. In order to achieve the optimum design, the design objective is to minimize a force quantity, which is a function of both the peak top floor absolute acceleration and bearing displacement as defined by $f(\ddot{U}_2, u_b) = Q + 2K_p|u_b| + M_2|\ddot{U}_2|$, where Q is the characteristic strength of the isolator, K_p is the post-yielding stiffness of the isolator and $\ddot{U}_2$ is the total acceleration at the second floor. Adjust the mechanical properties of the bearing and conduct the nonlinear time history analysis to improve your building performance. Report the best design you can come up with including their mechanical properties and the associated force functions based on computed response quantities under each earthquake.

6. Prepare a written report and submit the hard copy before 5PM on **June 12, 2019**. The main body of the report should be written like a technical paper, including abstract, problem statement, approach, representative results and summary or conclusion. Appendices may be used to provide supporting data, hand calculations and MATLAB programs developed. The final grade will be based on: 1) accuracy and completeness of the work (65%); 2) efficiency of the design (i.e. how good is your design compared with others) (15%); and 3) presentation and organization of the material (20%). Submit all your MATLAB and Excel codes in a single zip file on CCLE along with your report.



3D-Model for Aesthetic Purposes

Project Engineers

Khalid Alsadhan, M.Sc. Structural Mechanics

Role:
Project Leader and point-of-contact between team members. Wrote and oversaw all the MATLAB code on the project

Emilien Marotte, M.Sc. Structural Mechanics

Role:
Facilitated project support for developing a framework for code execution.

Thomas Viguier, M.Sc. Structural Mechanics

Role:
Oversaw the bearing design and offered support for the optimization process of the project.

Antoine Hascoat, M.Sc. Earthquake/Structural Engineering

Role:
Developed Excel Framework for designing the bearing and also provided support on increasing the efficiency of the project

PART 1

Objective: Develop a 5 % damping response spectrum (relative displacement and total acceleration) for each individual earthquake and the averaged response spectrum based on all earthquakes. **MATLAB FILES:** Part_A_Final.m and ResponseSpectra.m (function)

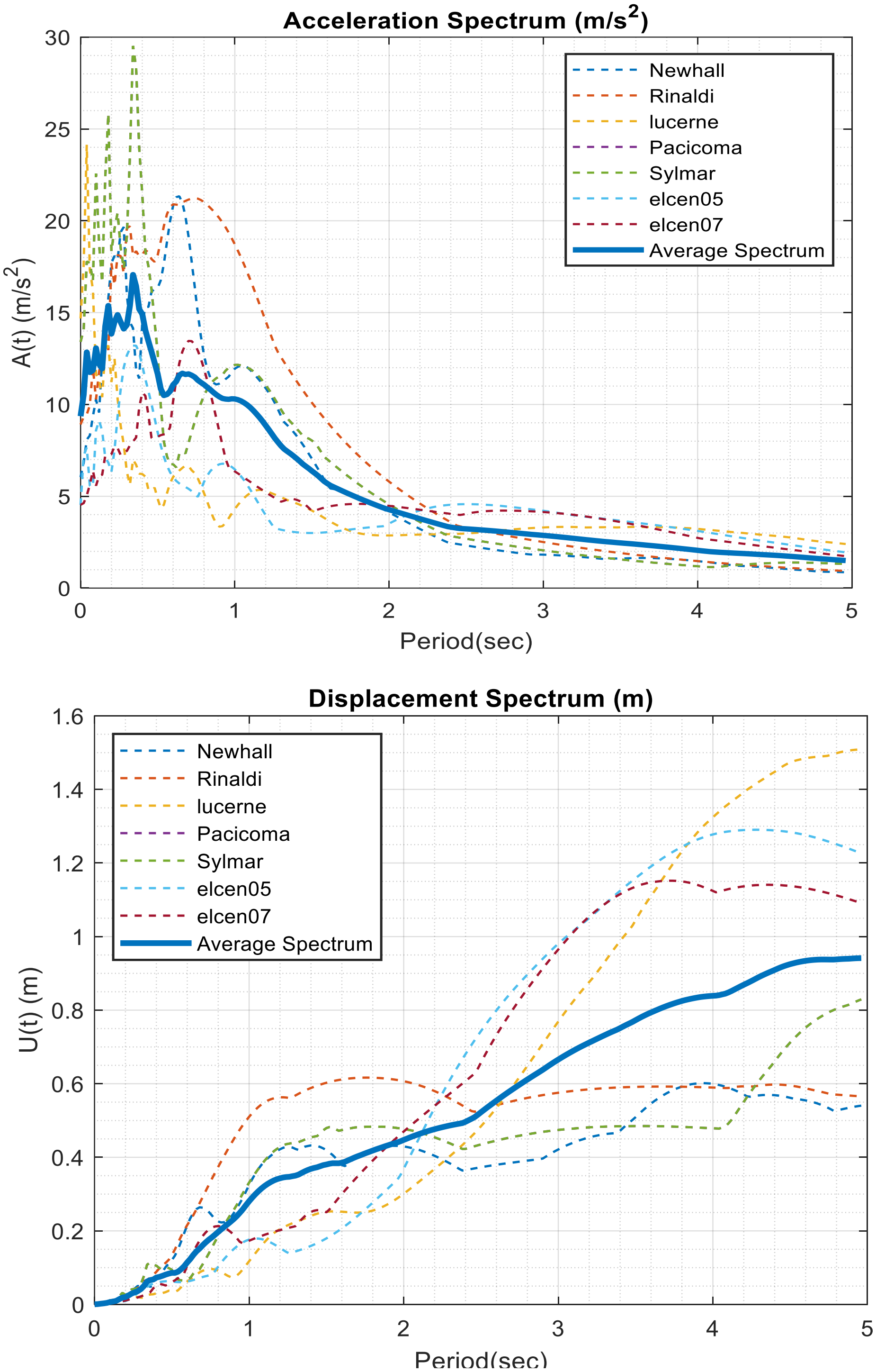


Figure 1- Acceleration (top) Displacement (bottom) Spectra

PART 2

Objective: Develop the Equation of Motion for a fixed base building using relative displacements to grounds as unknowns. Obtain the natural periods and natural modes of the fixed base building. Use and ODE solver and compare with the Newmark Method. Compute Inter-story drifts, floor total accelerations, and base-shears under design earthquake motions.

Relevant MATLAB Files: Part_B_Final.m and ResponseSpectra.m (function)

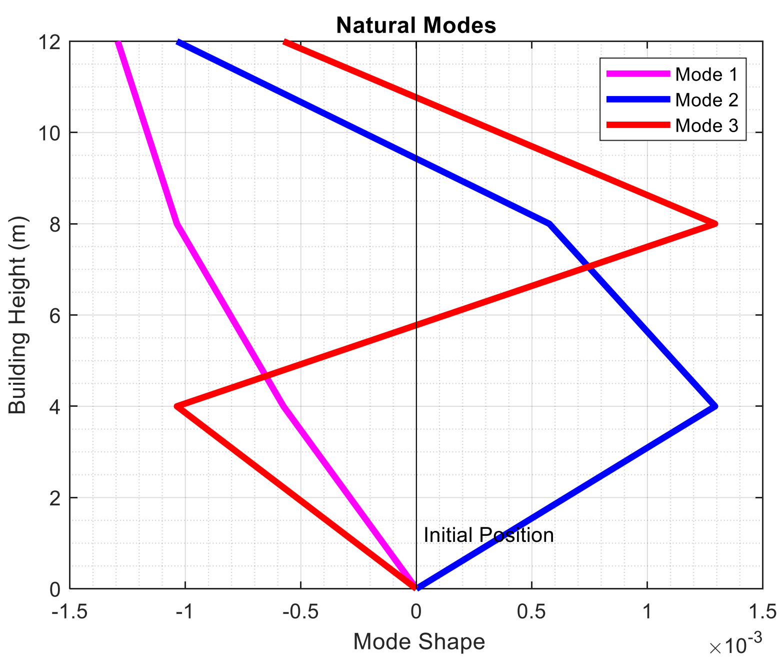
The following graphs are provided for the Newhall Station. To confirm the results, proceed to open "Part_B_Final.m" and make sure the argument for the line containing the ResponseSpectrum function is 1. This will generate all the necessary results. Input numbers 1-7 for different design Earthquakes given in the problem statement.

Equation of Motion for 3-DOF Structure:

$$\begin{bmatrix} M_b & 0 & 0 \\ 0 & M_1 & 0 \\ 0 & 0 & M_2 \end{bmatrix} \begin{bmatrix} \ddot{u_b} \\ \ddot{u_1} \\ \ddot{u_2} \end{bmatrix} + [C] \begin{bmatrix} \dot{u_b} \\ \dot{u_1} \\ \dot{u_2} \end{bmatrix} + \begin{bmatrix} (K_1 + K_2) & -K_2 & 0 \\ -K_2 & (K_2 + K_3) & -K_3 \\ 0 & -K_3 & K_3 \end{bmatrix} \begin{bmatrix} u_b \\ u_1 \\ u_2 \end{bmatrix} = -\begin{bmatrix} M_b \\ M_1 \\ M_2 \end{bmatrix} [\ddot{u_g}]$$

The damping matrix C was determined using the Rayleigh Damping method with a 2% modal damping for the first two modes.

Structural Modes for Fixed-Base Case:



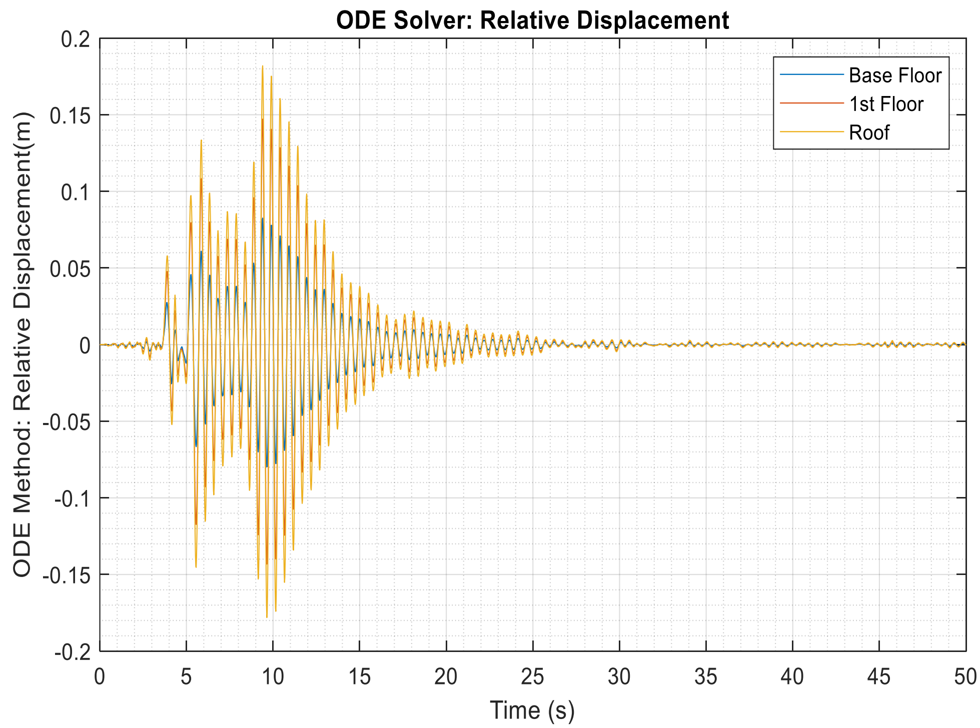
Periods:

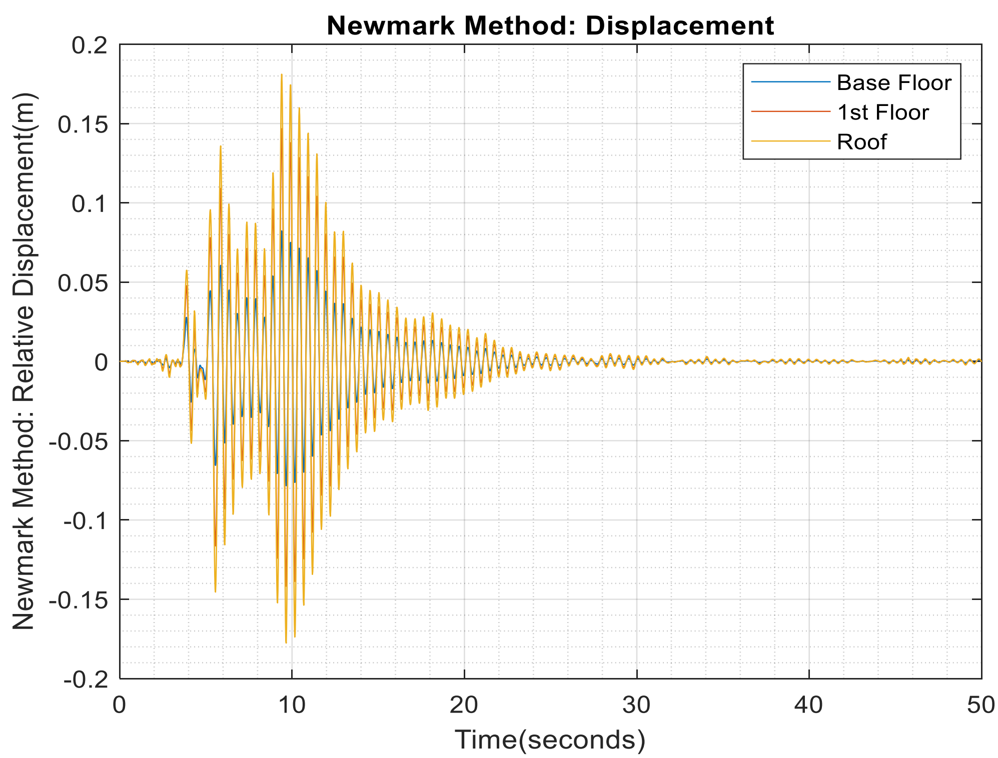
T1 = 0.509 seconds

T2 = 0.182 seconds

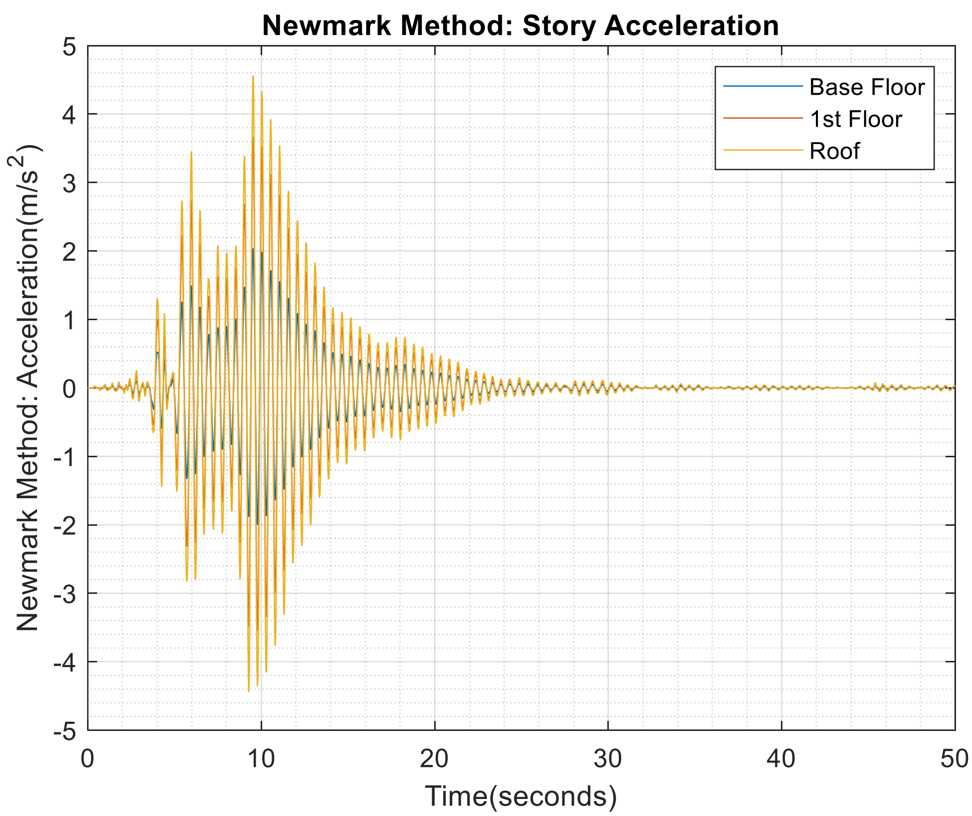
T3 = 0.1257 seconds

The following graphs are the Displacement graphs under the **Newhall Earthquake Station** performed using and ODE Solver and the Newmark Method.






ODE Solver: Story Acceleration
Base Floor
1st Floor
Roof
Acceleration(m/s^2)
Time (s)
5
4
3
2
1
0
-1
-2
-3
-4
-5
0
10
20
30
40
50

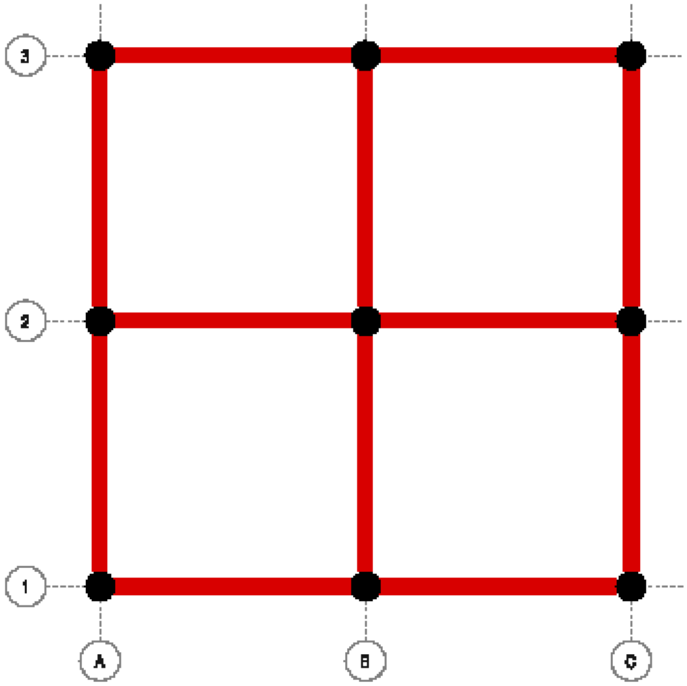

Newmark Method: Story Acceleration
Base Floor
1st Floor
Roof
Newmark Method: Acceleration(m/s^2)
Time(seconds)
5
4
3
2
1
0
-1
-2
-3
-4
-5
0
10
20
30
40
50

Below is an Excel Table that illustrates all the following Earthquakes using the MATLAB code.

	ODE Solver						
Max Values	**Story Drifts (m)**			**Floor Acclerations(m/s^2)**			Base Shear (N)
	Roof	1st Floor	Base Floor	Roof	1st Floor	Base Floor	
1.Newhall	0.0349	0.0646	0.0826	4.6275	3.6839	2.0197	3357600.159
2. Rinaldi	0.0257	0.0462	0.0585	3.0752	2.4776	1.3834	2254277.272
3. Lucerne	0.0426	0.0770	0.0976	5.0910	4.1084	2.2944	3734827.626
4. Pacoima	0.0380	0.0603	0.0665	4.0385	3.3091	1.9320	2983222.160
5. Sylmar	0.0204	0.0351	0.0434	2.5473	2.0508	1.1401	1864904.512
6. Elcen05	0.0100	0.0181	0.0239	0.9339	0.7550	0.4264	686797.869
7. Elcen07	0.0123	0.0230	0.0312	1.5335	1.2150	0.6618	1088194.559

	Newmark Method						
Max Values	**Interstory Drifts (m)**			**Floor Acclerations (m/s^2)**			Base Shear (N)
	Roof	1st Floor	Base Floor	Roof	1st Floor	Base Floor	
1.Newhall	0.0369	0.0644	0.0823	4.5563	3.6578	2.0339	3330635.400
2. Rinaldi	0.0256	0.0466	0.0593	3.1039	2.4904	1.3830	2267633.308
3. Lucerne	0.0435	0.0788	0.1008	5.2701	4.1745	2.2946	3750631.110
4. Pacoima	0.0361	0.0585	0.0707	4.1682	3.3243	1.8637	2996651.452
5. Sylmar	0.0199	0.0347	0.0430	2.5045	1.9855	1.0823	1810963.837
6. Elcen05	0.0105	0.0186	0.0246	0.9521	0.7802	0.4460	707954.865
7. Elcen07	0.0124	0.0227	0.0314	1.5361	1.2105	0.6608	1091417.351

Parts 3 & 5- Bearing Design and Optimization



Preliminary Bearing Layout

Preliminary Bearing Properties:

-High Damping Elastomeric Bearing

- 9 Bearings at the base of the columns.

- G = 0.30 MPa, k = 2000 MPa

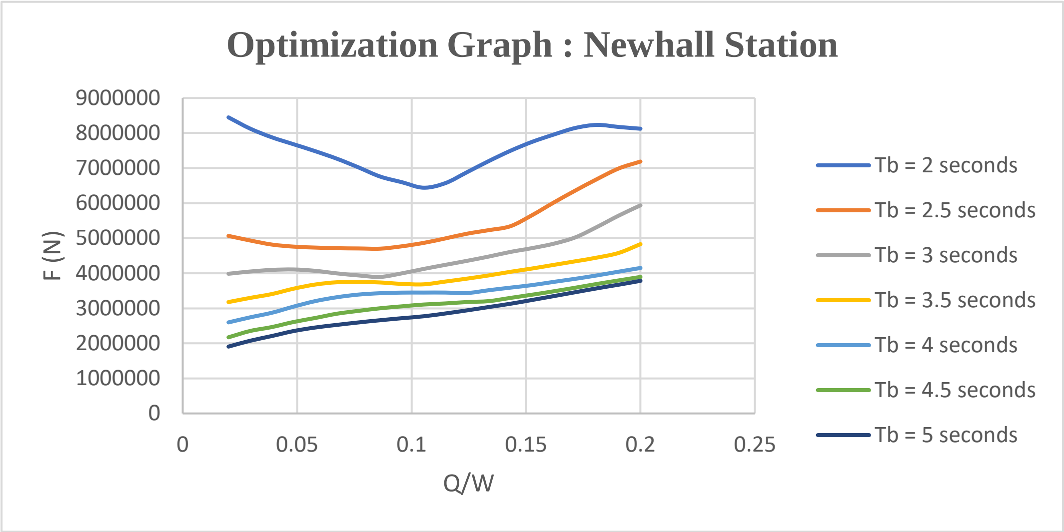
-γ = 150%

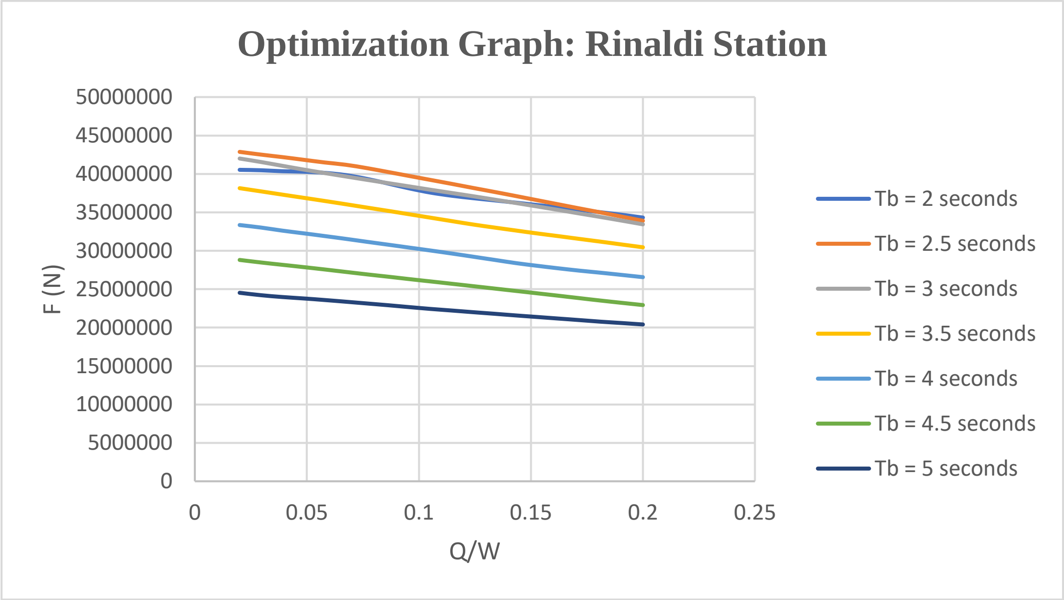
Methodology and Optimization Approach:

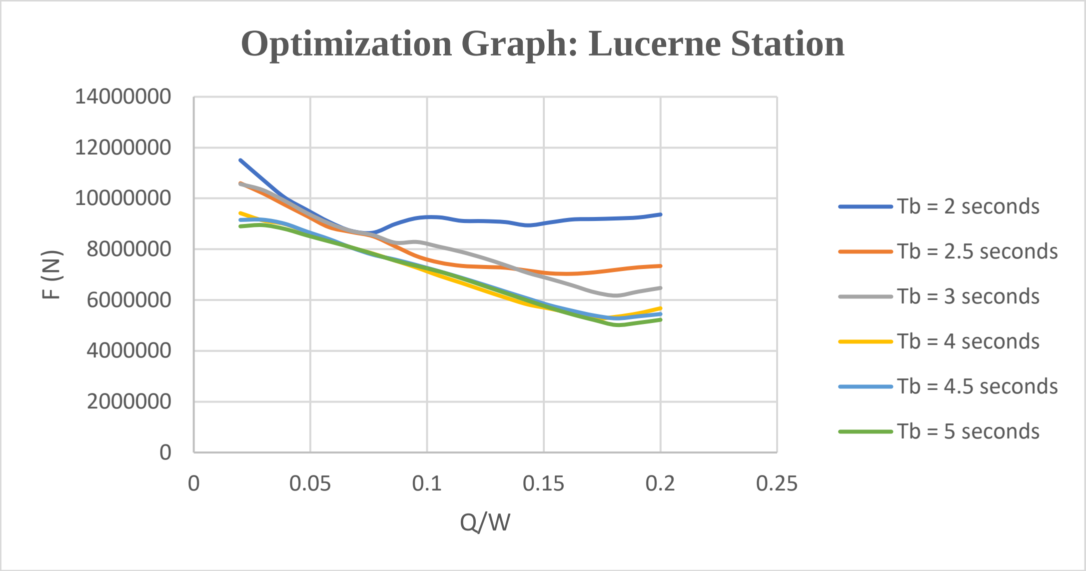
To approach the design of the High Damping Elastomeric devices that will be used to extend the period of the structure, and thus reduce the effects of the earthquakes, we must first begin with selecting a design period. After selecting a T_b value, we compute our maximum displacement D_m, and our total maximum displacement D_{TM}. Afterwards, we can calculate our post-yielding stiffness K_p and Area of our specific bearing. Other important factors can later be solved such that we can verify that our bearing design can satisfy the three checks which are: 1. Buckling, 2. Rollout Displacement 3. Global Overturning. To come up with the most efficient bearing design, we must first find the necessary $K_{p,}$ K_e , and Q values such that the following equation can be minimized:

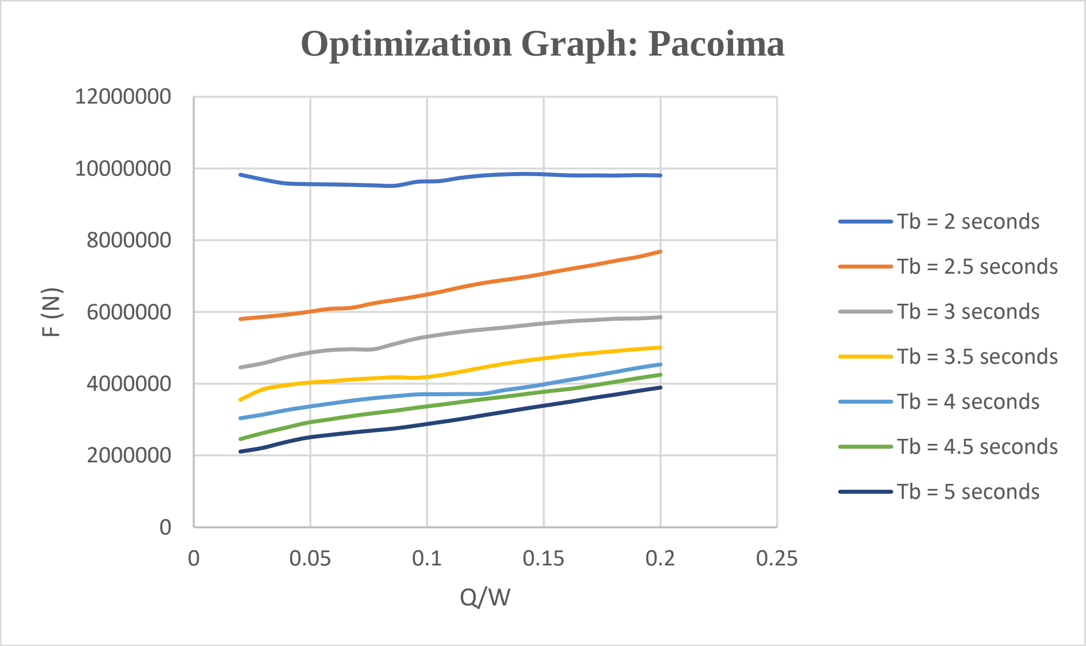
$$f(\ddot{U}_2, u_b) = Q + 2K_p|u_b| + M_2|\ddot{U}_2|$$

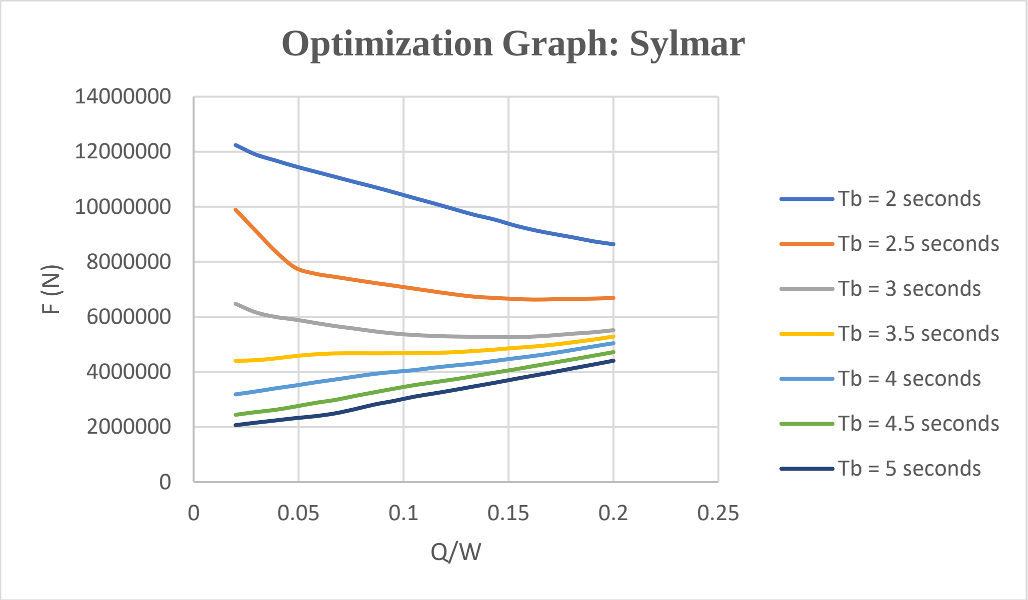
From our Excel file, “Optimization Graphs” Sheet 1, we have plotted the F vs Q/W values for all Earthquakes at seven Targeted Isolation Periods.

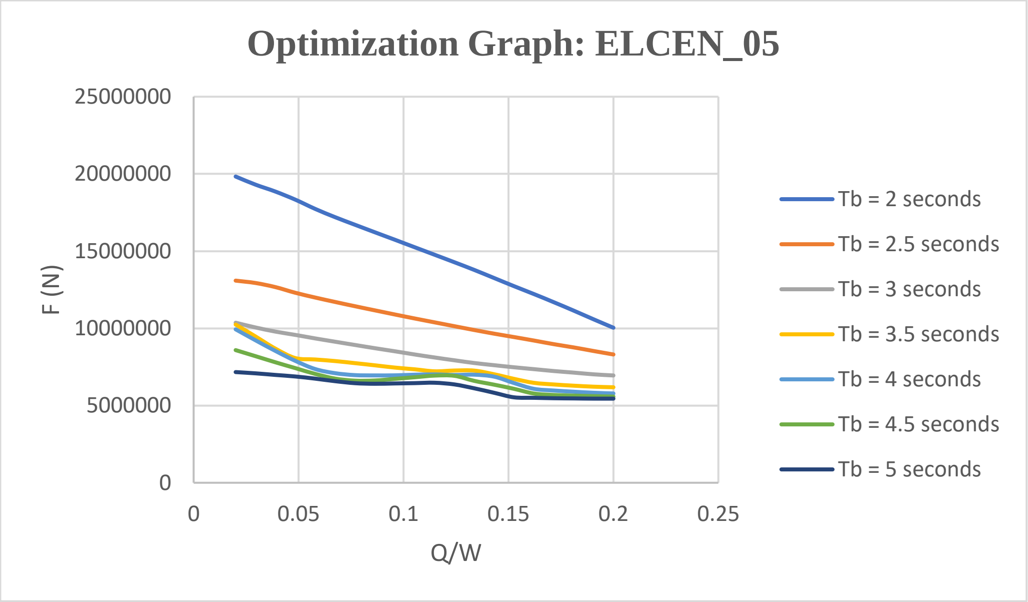


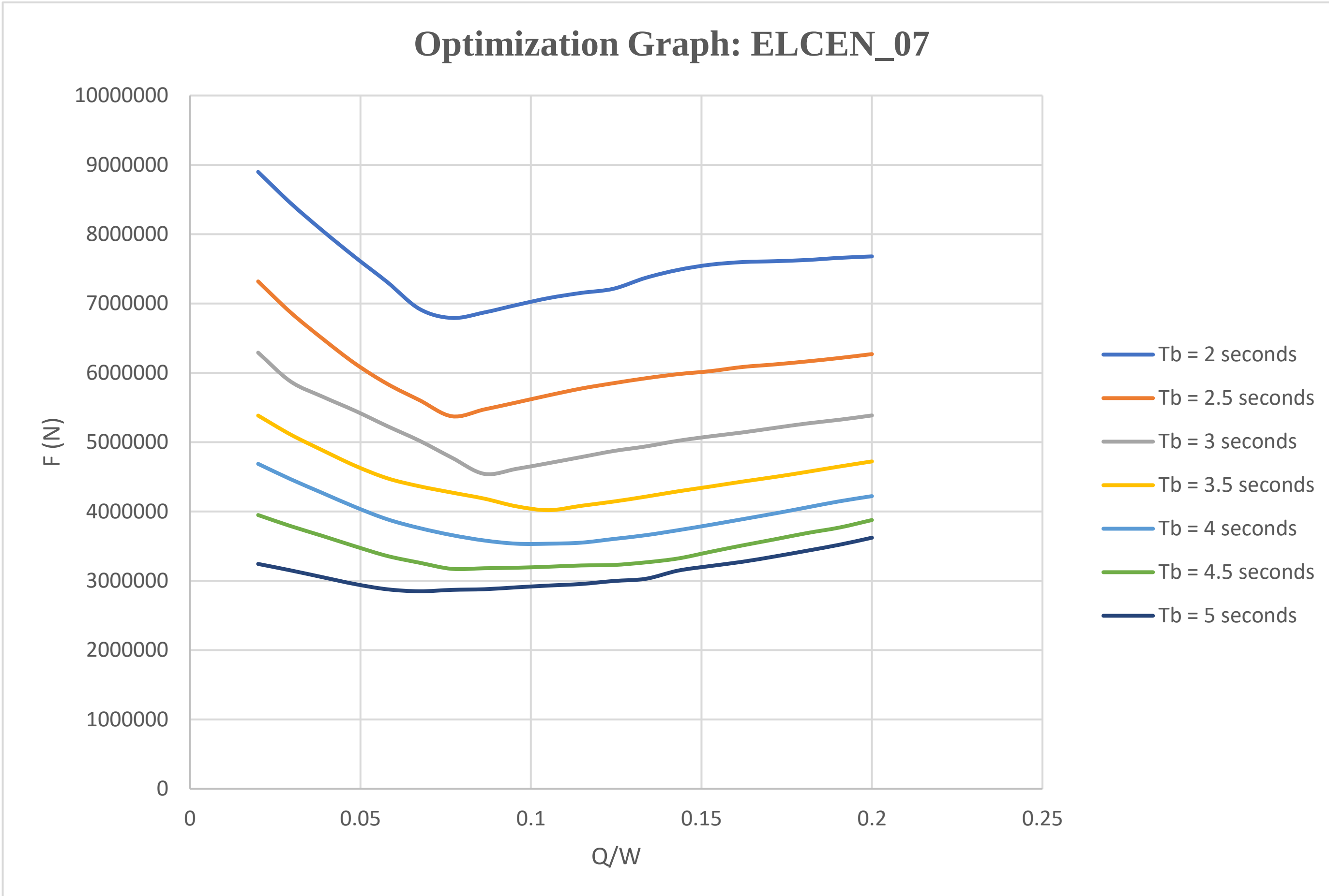



Optimization Graph: Lucerne Station
F (N)
Q/W
14000000
12000000
10000000
8000000
6000000
4000000
2000000
0
0
0.05
0.1
0.15
0.2
0.25
Tb = 2 seconds
Tb = 2.5 seconds
Tb = 3 seconds
Tb = 4 seconds
Tb = 4.5 seconds
Tb = 5 seconds

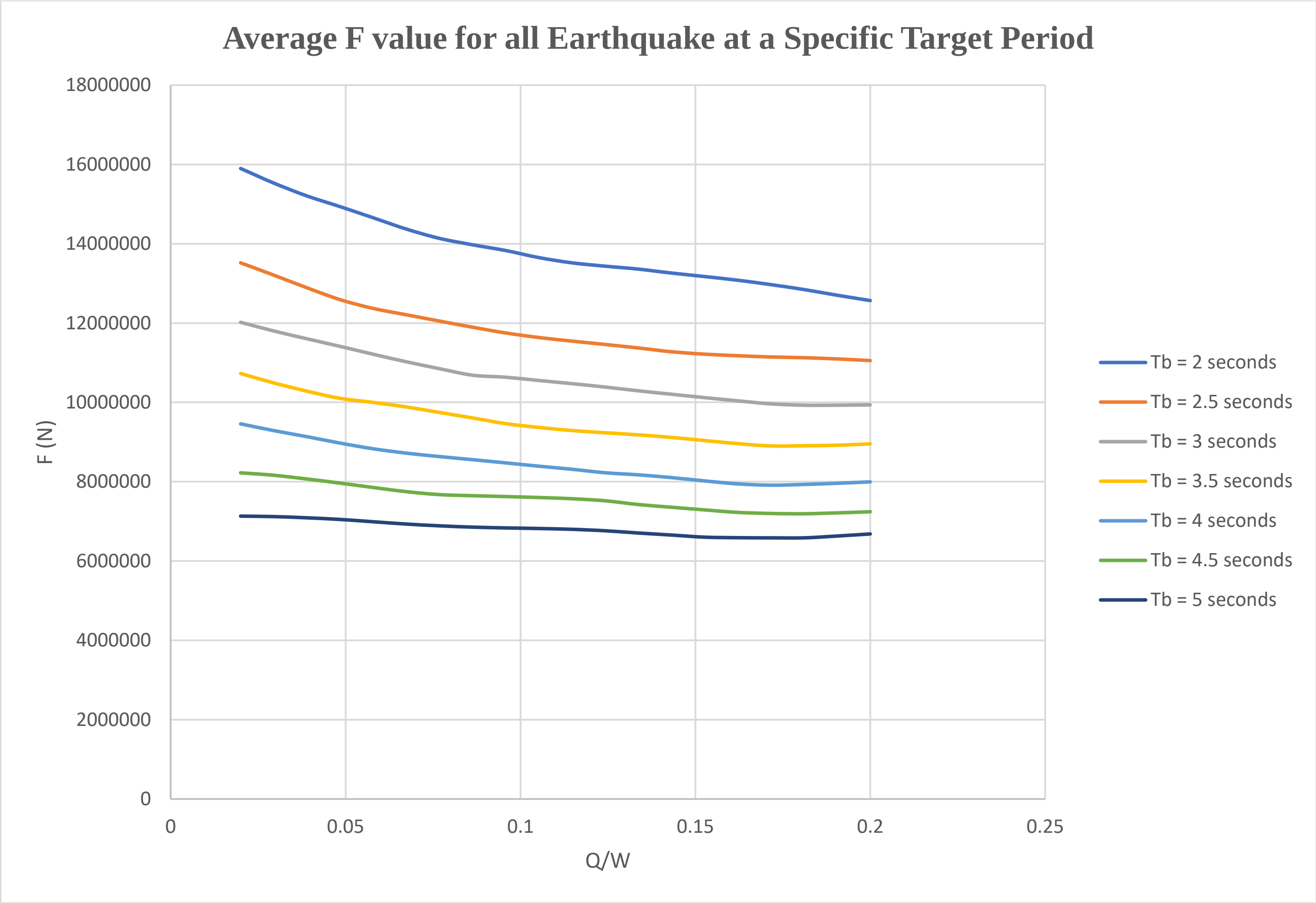

Optimization Graph: Pacoima
F (N)
Q/W
12000000
10000000
8000000
6000000
4000000
2000000
0
0
0.05
0.1
0.15
0.2
0.25
Tb = 2 seconds
Tb = 2.5 seconds
Tb = 3 seconds
Tb = 3.5 seconds
Tb = 4 seconds
Tb = 4.5 seconds
Tb = 5 seconds


Optimization Graph: Sylmar
14000000
12000000
10000000
8000000
6000000
4000000
2000000
0
F (N)
0
0.05
0.1
0.15
0.2
0.25
Q/W
Tb = 2 seconds
Tb = 2.5 seconds
Tb = 3 seconds
Tb = 3.5 seconds
Tb = 4 seconds
Tb = 4.5 seconds
Tb = 5 seconds


Optimization Graph: ELCEN_05
25000000
20000000
15000000
10000000
5000000
0
F (N)
0
0.05
0.1
0.15
0.2
0.25
Q/W
Tb = 2 seconds
Tb = 2.5 seconds
Tb = 3 seconds
Tb = 3.5 seconds
Tb = 4 seconds
Tb = 4.5 seconds
Tb = 5 seconds

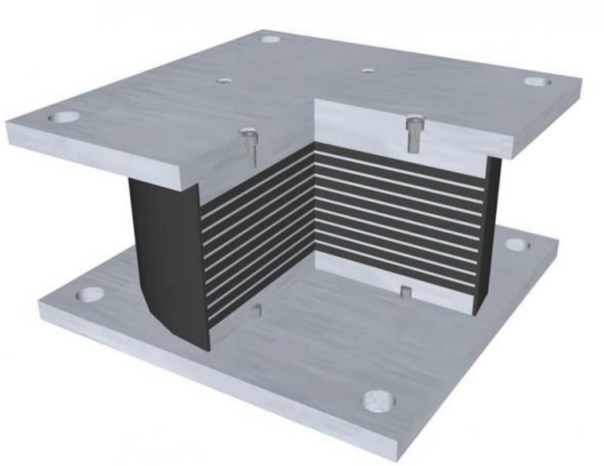


Once we graphed the F vs Q/W values for the selected time periods for all the earthquake stations, it is clear to see that our target range is within 2-3.5 seconds. On the next page, we show our average curves for each Tb value of all Earthquakes.



To optimize our design, we must pick a Q/W value that corresponds to a minimum F value. From our excel file, “Optimization Graphs” sheet 1, we first select a Q/W value of a particular Tb value and obtain all the necessary bearing parameters to conduct the design which is detailed in “Optimization Graphs” sheet 2.

Bearing Design:



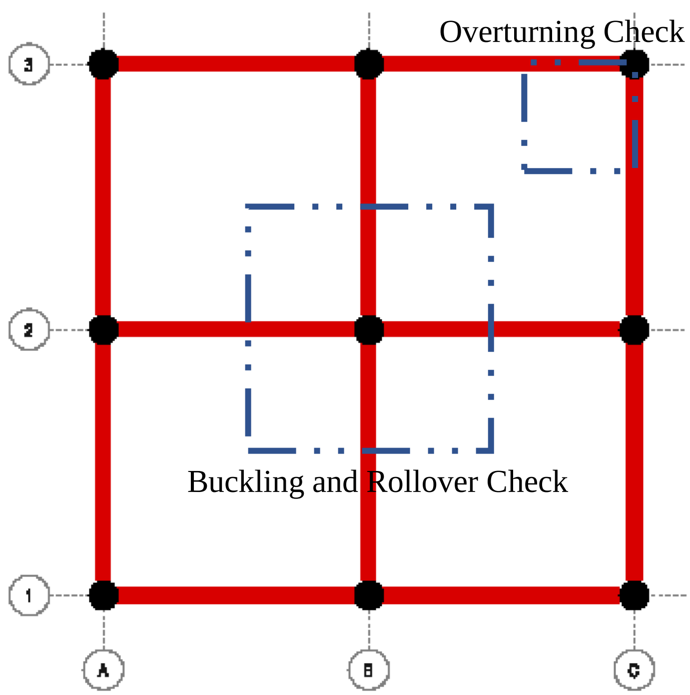
High Damping Elastomeric Bearing

After conducting multiple iterations on Excel to find the most optimized bearing characteristics, the following parameters are used in the following design:

Building Data		
Total Mass	975000	kg
Total Weight	9564750	N
G	300000	N/m^2
k	2000	N/mm^2
Target Period and Displacement		
Tb	3	seconds
Dm	0.666	meters
γ_{strain}	150	%
sigma_tr	0.444	meters
D_tm	0.766	meters
γ_{max}	172.5	%

BEARING PROPERTIES		
nb	9	bearings
Kp	475203.175	N/m
Ke	4752031.749	N/m
Q	192413.684	N
Dy	0.045	meters
Area	0.703	meters
Diameter	1.000	meters
Radius	0.473	meters
S_factor	20	assumption
t_rubber	12.000	mm
n_rubber	37	layers
S_recalculated	20	
n_steel	36	layers
t_steel	2.5	mm
t_end	3	mm
h_total	540.000	mm
Ec_infinity	720000000	N/m
Ec	529411764.7	N/m
I_s	0.015957387	m^4

Using these parameters, it is imperative to check the design against buckling, rollout displacement and global over turning. As mentioned earlier, the properties represented in the table above are for a single bearing. Therefore, when calculating the required checks for this design, it is judicious to design for the bearing with the highest tributary area which is the center bearing in the figure below. However, when checking for global overturning, we use the corner bearings as we must check that the there is no net tension force.



Buckling, Rollover Displacement, and Global Overturning Checks

To calculate the maximum displacement, we utilized our displacement spectrum and picked a distance of 0.666 m that corresponds to an isolation period of 3 seconds. The total thickness, Kp, Ke, Q and Dy values were obtained by the following equations:

$$\sum t_r = \frac{D_m}{\gamma_{\text{strain}}}$$

$$K_p = m_{total}\left(\frac{2\pi}{T_b}\right)^2$$

$$K_e = 10 * K_p$$

$$Q = W_{total} * [0.02\sim0.2]$$

$$D_y = \frac{Q}{K_e - K_p}$$

Once we determined the above values we can proceed to determine the Area of a single bearing.

$$A = \left(\sum t_r\right) * \frac{K_p}{G}$$

Assuming an S factor, we can then calculate the thickness of a single rubber bearing, and calculate the elastic modulus with compressibility effects.

$$t_r = \frac{R}{2S}$$

$$E_c^{\infty} = 6GS^2$$

$$E_C = \frac{(E_c^{\infty} k)}{E_c^{\infty} + k} \rightarrow ad - hoc\ assumption$$

From the lecture notes, if $P_S << P_E$ and S > 5, the critical buckling load can be calculated as:

$$P_{critical} = \frac{G\pi^2 R^3 S}{\sqrt{2}\Sigma t_r}$$

<u>The Safety check for buckling can then be checked by the equation below:</u>

$$S.F = \frac{P_{critical}}{P_{applied}} > 3$$

The Rollover Displacement can be computed as:

$$D_{TM} = 1.15D_m$$

$$\delta = \frac{\emptyset}{1 + \frac{k_H}{P}} > D_{TM}$$

$\emptyset$ is the diameter of the bearing and P is the applied force on the bearing.

Finally, Global overturning is checked by the following equation:

$$E = \frac{\frac{(Height_{building})}{2} * k_H * D_M}{3 * (Building\ width)}$$

E must be less than the applied force on the corner bearing of the floor plan.

Using the following Bearing Parameters, the following checks were calculated in an excel sheet and were deemed satisfactory.

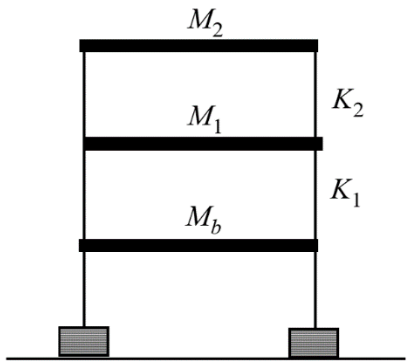
Buckling Load Check			
P_applied	2391187.5	Newtons	
P_e	285935173.8	Newtons	
Ps	210990.2096	Newtons	
Pcritical	7767208.139	Newtons	
S.F	3.25	O.K	3

Rollout Displacement					
Diameter	1	meters			
K_H	475203	N/m			
P	2391188	N			
h	0.54	meters			
Rollout Displacement	0.90309	O.K	D_tm	0.7659	meters

Global Overturning			
K_H	475203.17	N/m	
Dm	0.666	meters	
Building_Height	12	meters	
E	35165.035	N	
Corner_Force	597796.88	N	O.K

Parts 4- Isolated Building Structure using Optimized Bearings

The following relevant MATLAB files are: Final_Bearing_Design.m and IsolatedStructureSingular.m



Isolated Structure

$$K_1 = K_2 = K_0 = 250MN/m$$

The following structure is now retrofitted with high elastomeric rubber bearing as the base of its columns. The equation of the motion now becomes the following:

$$\begin{bmatrix} M_b & 0 & 0 \\ 0 & M_1 & 0 \\ 0 & 0 & M_2 \end{bmatrix} \begin{bmatrix} \ddot{u_b} \\ \ddot{u_1} \\ \ddot{u_2} \end{bmatrix} + [C] \begin{bmatrix} \dot{u_b} \\ \dot{u_1} \\ \dot{u_2} \end{bmatrix} + \begin{bmatrix} (K_1) & -K_1 & 0 \\ -K_1 & (K_1 + K_2) & -K_2 \\ 0 & -K_2 & K_2 \end{bmatrix} \begin{bmatrix} u_b \\ u_1 \\ u_2 \end{bmatrix} + (n_b) \begin{bmatrix} P(t) \\ 0 \\ 0 \end{bmatrix} = - \begin{bmatrix} M_b \\ M_1 \\ M_2 \end{bmatrix} [\ddot{u_g}]$$

Using the Bouc-Wen modeling technique, the following variables are defined:

$$P(t) = \alpha K_e u(t) + (1 - \alpha) K_e D_y z(t)$$

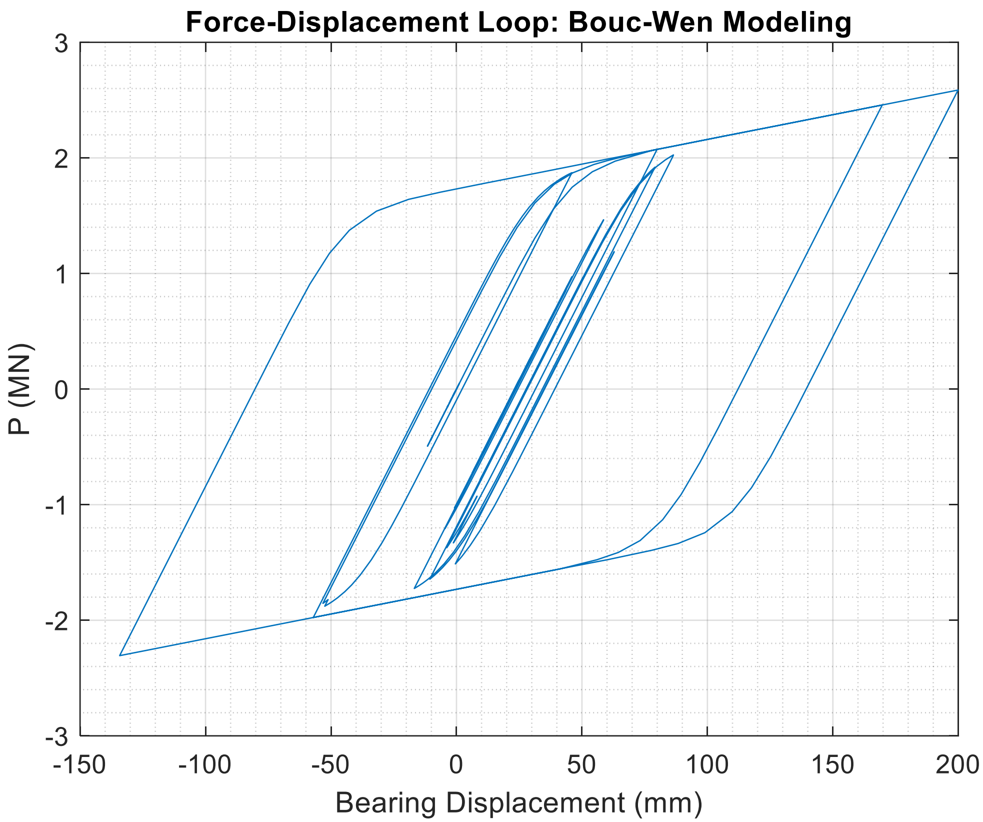
For our program, our numerical modeling parameter gamma, alpha and n are defined below:

$$\gamma = 0.5$$

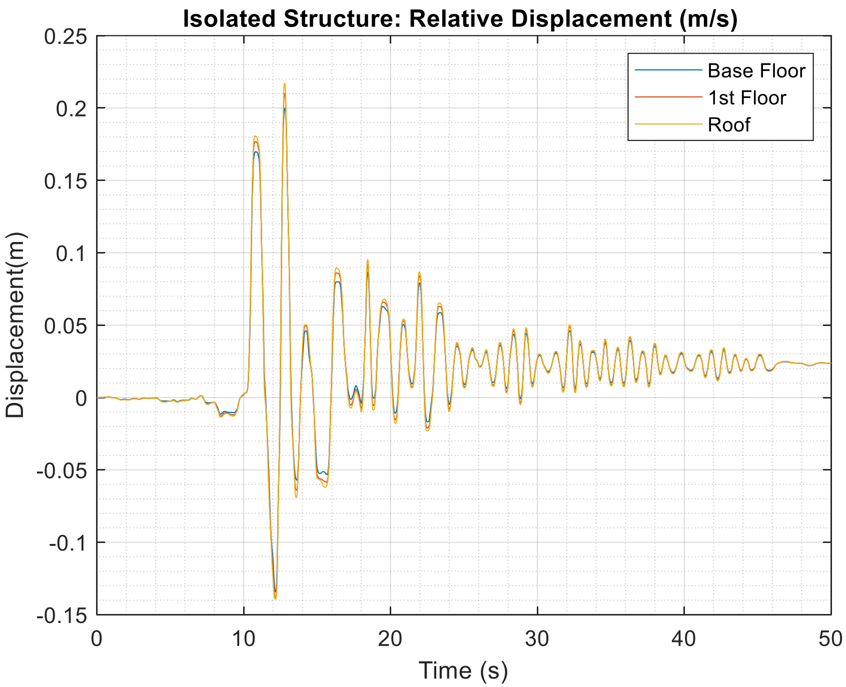
$$\alpha = 0.5$$

$$n = 5$$

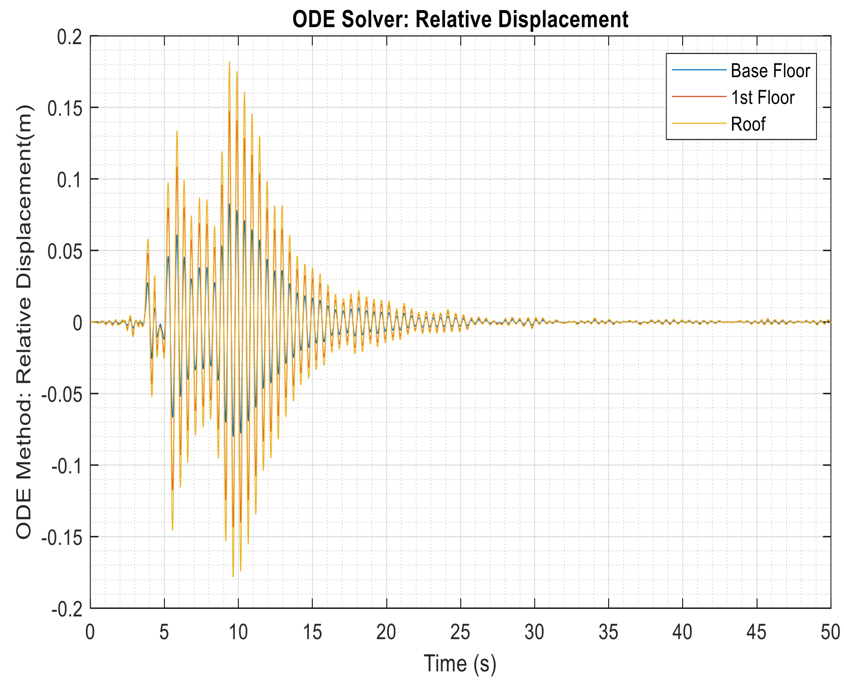
The following graphs were retrieved from running the code on the **Newhall Earthquake**. All Earthquakes were ran and exported into an Excel file called NTLAIsolated.xs



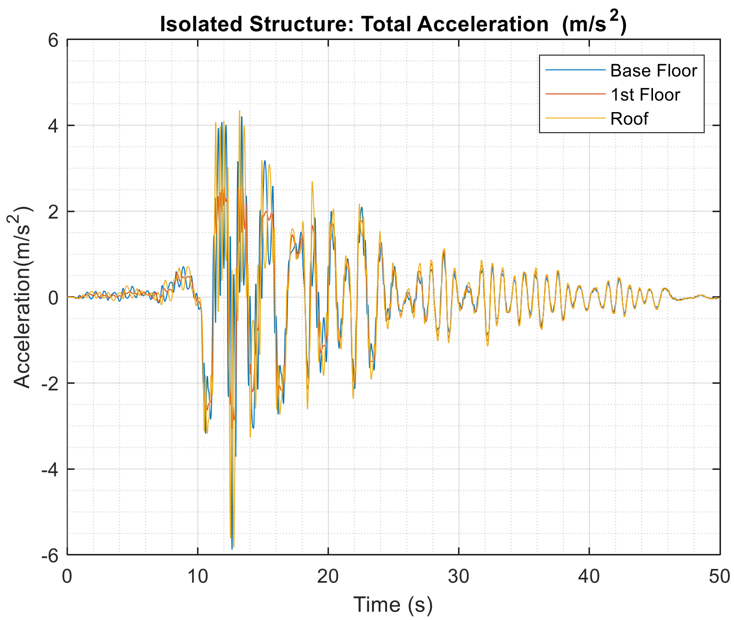
ISOLATED STRUCTURE



FIXED STRUCTURE



ISOLATED STRUCTURE



FIXED STRUCTURE

Max Values	ODE Solver						
	Story Drifts (m)			**Floor Acclerations(m/s^2)**			Base Shear (N)
	Roof	1st Floor	Base Floor	Roof	1st Floor	Base Floor	
1.Newhall	0.0349	0.0646	0.0826	4.6275	3.6839	2.0197	3357600.159
2. Rinaldi	0.0257	0.0462	0.0585	3.0752	2.4776	1.3834	2254277.272
3. Lucerne	0.0426	0.0770	0.0976	5.0910	4.1084	2.2944	3734827.626
4. Pacoima	0.0380	0.0603	0.0665	4.0385	3.3091	1.9320	2983222.160
5. Sylmar	0.0204	0.0351	0.0434	2.5473	2.0508	1.1401	1864904.512
6. Elcen05	0.0100	0.0181	0.0239	0.9339	0.7550	0.4264	686797.869
7. Elcen07	0.0123	0.0230	0.0312	1.5335	1.2150	0.6618	1088194.559

Non-Linear Time History Analysis: Isolated Structure									
Earthquake Station	Max Interstory Drifts (m)			Max Floor Acclerations (m/s^2)			Base Shear (N)	Isolator Displacement (m)	F (N)
	Roof	1st Floor	Base Floor	Roof	1st Floor	Base Floor			
1.Newhall	0.0075	0.0111	0.1999	5.8252	3.0578	5.8721	357249.322	0.1999	3035995
2. Rinaldi	0.0221	0.0419	3.1625	17.2353	15.9077	17.2013	2072915.344	3.1625	6744285
3. Lucerne	0.0054	0.0096	0.3609	4.1616	3.5117	4.2191	296609.096	0.3609	2495355
4. Pacoima	0.0064	0.0090	0.2848	5.0632	3.1524	4.8774	264114.048	0.2848	2788350
5. Sylmar	0.0046	0.0085	0.2951	3.4984	3.1755	3.5425	225706.132	0.2951	2279801
6. Elcen05	0.0057	0.0107	0.4609	4.4825	3.8497	4.4480	411949.571	0.4609	2599647
7. Elcen07	0.0052	0.0091	0.2555	4.1639	3.1014	4.0502	306271.886	0.2555	2496078
Average F value = 3205644.5 N									

To check our code these are the required results you will need to input in your program

Number of Bearings	9
Kp	475203.2 N/m
Ke	4752032 N/m
Q	192413.7 N
n (bouc-wen modeling)	5
gamma	0.5
alpha	0.5

~END~

Appendix: MATLAB CODE

ResponseSpectra.m

```
function [S_a,S_d,Time,ug2dot] = ResponseSpectra(n)
if n == 1
newhall_plot
close all
elseif n == 2
    clear all
    rinaldi_plot
    close all
elseif n == 3
    clear all
    lucerne_plot
    close all
elseif n == 4
    clear all
    pacoima_plot
    close all
elseif n == 5
    clear all
    sylmar_plot
    close all
elseif n == 6
    clear all
    elcen05_plot
    close all
elseif n == 7
clear all
elcen07_plot
close all
end

ug2dot = Ag;
neq = length(ug2dot);
np = 2^(ceil(log(neq)/log(2))+3);
z = zeros(1,(np-neq));
ug2dot = [ug2dot z];
nfft = length(ug2dot);
delta_w = (2*pi)/((nfft-1)*dt);
w_1 = [0:delta_w:(nfft/2)*delta_w];
w_2 = [-(nfft/2-1)*delta_w:delta_w:-delta_w];
w = [w_1 w_2];
u2dot_w = fft(ug2dot);
t_period=[0.04:0.02:5];

xi5 = 0.05;

for i = 1:length(t_period)
    wn(i) = 2.*pi()/t_period(i);
```

```
for j = 1:nfft
    beta(j) = w(j)/wn(i) ;
    h(j) = -1/(wn(i)^2*((1-beta(j)^2)+2*xi5*beta(j)*sqrt(-1)));
    uNew2(j) = h(j)*u2dot_w(j) ;
    vNew2(j) = sqrt(-1)*w(j)*uNew2(j);
    aNew2(j) = -w(j)^2*uNew2(j);
end
uNew2=ifft(uNew2);
vNew2=ifft(vNew2);
aNew2=ifft(aNew2);
a_total=aNew2+ug2dot;
S_d(i)=max(abs(uNew2));
S_a(i)=max(abs(a_total));
Time = Time;


end
end
```

Part_A_Final.m

```
clear all; clc ;
%% Part A
t=[0:0.02:30];
t_period=[0.04:0.02:5];
[S_a1,S_d1,Time,ug2dot] = ResponseSpectra(1); %newhall
[S_a2,S_d2,Time,ug2dot] = ResponseSpectra(2); %rinaldi
[S_a3,S_d3,Time,ug2dot] = ResponseSpectra(3); %lucerne
[S_a4,S_d4,Time,ug2dot] = ResponseSpectra(4); %pacoima
[S_a5,S_d5,Time,ug2dot] = ResponseSpectra(4); %sylmar
[S_a6,S_d6,Time,ug2dot] = ResponseSpectra(6); %elcen05
[S_a7,S_d7,Time,ug2dot] = ResponseSpectra(7); %elcen07
avg_Sa = (S_a1+S_a2+S_a3+S_a4+S_a5+S_a6+S_a7)/7;
avg_Sd = (S_d1+S_d2+S_d3+S_d4+S_d5+S_d6+S_d7)/7;


figure(1)
plot(t(1:length(t_period)),S_d1(1:length(t_period)),'--
',t(1:length(t_period)),S_d2(1:length(t_period)),'--',...
    t(1:length(t_period)),S_d3(1:length(t_period)),'--
',t(1:length(t_period)),S_d4(1:length(t_period)),'--',...
    t(1:length(t_period)),S_d5(1:length(t_period)),'--
',t(1:length(t_period)),S_d6(1:length(t_period)),'--',...
    t(1:length(t_period)),S_d7(1:length(t_period)),'--
',"LineWidth",1);
hold on ;
plot(t(1:length(t_period)),avg_Sd(1:length(t_period)),'Linewidth',2.5)
;
xlabel('Period(sec)')
ylabel('U(t) (m)')
grid on ; grid minor
title("Displacement Spectrum (m)")
legend("Newhall","Rinaldi","lucerne","Pacicoma","Sylmar","elcen05","el
cen07","Average Spectrum","LineWidth",1,'Location','northwest')
hold off

figure(2)
plot(t(1:length(t_period)),S_a1(1:length(t_period)),'--
',t(1:length(t_period)),S_a2(1:length(t_period)),'--',...
    t(1:length(t_period)),S_a3(1:length(t_period)),'--
',t(1:length(t_period)),S_a4(1:length(t_period)),'--',...
    t(1:length(t_period)),S_a5(1:length(t_period)),'--
',t(1:length(t_period)),S_a6(1:length(t_period)),'--',...
    t(1:length(t_period)),S_a7(1:length(t_period)),'--
',"LineWidth",1);
title("Acceleration Spectrum (m/s^2)")
hold on
plot(t(1:length(t_period)),avg_Sa(1:length(t_period)),'Linewidth',2.5)
;
```

```
xlabel('Period(sec)')
ylabel('A(t) (m/s^2)')
legend("Newhall","Rinaldi","lucerne","Pacicoma","Sylmar","elcen05","el
cen07","Average Spectrum","LineWidth",1,'Location','northeast')
grid on ; grid minor
hold off
```

Part_B_Final.m

```
%% Part B
clc; close all
K1 = 250 ;
K2 = K1 ;
K3 = K2 ;
[S_a1,S_d1,Time,ug2dot] = ResponseSpectra(1); %Newhall
global M C K ug2dot Time
M = [ 325 0 0 ;
      0 325 0 ;
       0 0 325]*10^3 ; % kg

K = [ K1+K2 -K2 0 ;
      -K2 K2+K3 -K3;
      0 -K3 K3;] *10^6; %N/m

[Mode_Shapes,D] = eig(K,M);
[W] = sqrt(D) ;
Mode_Shapes ;

z = figure();
draw_line = vertcat([0,0,0],Mode_Shapes);
node = (0:3)*4;
color = {'m','b','r','g','y'};
for i = 1:3
    plot(draw_line(:,i),node,'color',color{i},'LineWidth',3);
    hold on ;
end
grid on ; grid minor;
h = vline(0,'k','Initial Position')
xlabel(' Mode Shape');
ylabel('Building Height (m)');
legend('Mode 1','Mode 2','Mode 3','Initial Position')
title("Natural Modes");

[T] = (2*pi)*[1/W(1,1) 1/W(2,2) 1/W(3,3)]; % Natural Periods.
T1 = T(1); T2 = T(2); T3 = T(3);
Natural_Periods_Fixed_Base = table(T1, T2, T3);
%Rayleigh Structural Damping
A = inv([W(1,1)^-1 W(1,1); W(2,2)^-1 W(2,2)])*2*[0.02; 0.02];
C =A(1)*M + A(2)*K ;
%Modal Analysis
Ag_vector = length(ug2dot);
r_factor = ones(3,1);
dt = 0.02;
beta = 1/4; gamma = 1/2;
```

```
Mmodal = Mode_Shapes'*M*Mode_Shapes;
Modal_M_diagonal = diag(Mmodal);
Cmodal = Mode_Shapes'*C*Mode_Shapes;
Modal_C_diagonal = diag(Cmodal);
Kmodal = Mode_Shapes'*K*Mode_Shapes;
Modal_K_diagonal = diag(Kmodal);
Time = 0:dt:(length(ug2dot)-1)*dt ;

%% Begin ODE Analysis for the Fixed Base Structure
[t,output] = ode23(@(t,y) LinearTimeHistory(t,y), Time,
zeros(1,6)) ;

us0 = output(:,1) ; %Relative Displacement Base Floor
udots0 = output(:,2);   %Relative Velocity Base Floor


us1 = output(:,3) ; %Relative Displacement 1st Floor
udots1 = output(:,4);   %Relative Velocity 1st Floor

us2 = output(:,5) ; %Relative Displacement Roof Floor
udots2 = output(:,6);   %Relative Velocity Roof Floor

% Accelerations
r = ones(3,1);
ODE_Displacements = [us0,us1,us2];
ODE_Velocities = [udots0,udots1,udots2];
ODE_Acc_Total = -K\M*ODE_Displacements'-C\M*ODE_Velocities';
StoryShear = M*ODE_Acc_Total ;
ODE_Acc_Total = ODE_Acc_Total' ;
Base_ODEAcc = max(abs(ODE_Acc_Total(:,1)));
First_ODEAcc = max(abs(ODE_Acc_Total(:,2)));
Roof_ODEAcc = max(abs(ODE_Acc_Total(:,3)));
baseShear = max(sum(StoryShear)) ;

base_accODE = ODE_Acc_Total(:,1);
first_accODE = ODE_Acc_Total(:,2);
roof_accODE = ODE_Acc_Total(:,3);

% Calculate Drifts
drift_Roof_to_FirstFloor = max(abs(us2-us1));
drift_FirstFloor_to_Base = max(abs(us1-us0)) ;
drift_Base_to_Ground = max(abs(us0));

figure()
if length(Time) < length(us0)
```

```
plot(Time,us0(1:length(Time)),Time,us1(1:length(Time)),Time,us2(
1:length(Time)));
else

plot(Time(1:length(us0)),us0,Time(1:length(us1)),us1,Time(1:leng
th(us2)),us2);
end
legend("Base Floor","1st Floor","Roof ");
ylabel("Displacement(m)")
xlim([0 50]);
xlabel("Time (s)")
title("ODE Solver: Relative Displacement")
grid on ; grid minor

figure()
if length(Time) < length(us0)

plot(Time,base_accODE(1:length(Time)),Time,first_accODE(1:length
(Time)),Time,roof_accODE(1:length(Time)));
else

plot(Time(1:length(base_accODE)),base_accODE,Time(1:length(first
_accODE)),first_accODE,Time(1:length(roof_accODE)),roof_accODE);
end
legend("Base Floor","1st Floor","Roof ");
ylabel("Acceleration(m/s^2)")
xlim([0 50]);
xlabel("Time (s)")
title("ODE Solver: Story Acceleration")
grid on ; grid minor



BaseFloor = max(abs(us0));
FirstFloor = max(abs(us1));
Roof = max(abs(us2));
ODEResults =
table(drift_Roof_to_FirstFloor,drift_FirstFloor_to_Base,drift_Ba
se_to_Ground,...
    Roof_ODEAcc,First_ODEAcc,Base_ODEAcc,baseShear)

%% Check ODE method with Newmark Method

L = zeros(3,3); q = zeros(Ag_vector,3);
qdot = zeros(Ag_vector,3); q2dot = zeros(Ag_vector,3);
disp_NEWMARKmodal = zeros(3,Ag_vector);
```

```
vel_NEWMARKmodal = zeros(3,Ag_vector);
acc_NEWMARKmodal = zeros(3,Ag_vector);
for i=1:3

    L(i) = Mode_Shapes(:,i)'*M*r_factor/Modal_M_diagonal(i);
%participation factor
    [q(:,i),qdot(:,i),q2dot(:,i)] =
NewmarkIntegration(gamma,beta,Modal_M_diagonal(i),...
        Modal_C_diagonal(i),Modal_K_diagonal(i),L(i)*ug2dot,dt);

    disp_NEWMARKmodal =
disp_NEWMARKmodal+Mode_Shapes(:,i)*q(:,i)'; %relative
displacement
    vel_NEWMARKmodal =
vel_NEWMARKmodal+Mode_Shapes(:,i)*qdot(:,i)';%relative velocity
    acc_NEWMARKmodal =
acc_NEWMARKmodal+Mode_Shapes(:,i)*q2dot(:,i)'; %relative
acceleration
end
Accl_BaseNM = acc_NEWMARKmodal(1,:)';    %Total Acceleration Base
Floor
Accl_FirstNM = acc_NEWMARKmodal(2,:)';  %Total Acceleration 1st
Floor
Accl_RoofNM = acc_NEWMARKmodal(3,:)'; %Total Acceleration Roof
vel_NEWMARKmodal = vel_NEWMARKmodal';


Newmark_Roof = disp_NEWMARKmodal(3,:);
Newmark_First = disp_NEWMARKmodal(2,:);
Newmark_Base = disp_NEWMARKmodal(1,:);
NewmarkDisplacements =
[Newmark_Base',Newmark_First',Newmark_Roof'];
Newmark_Velocities =
[vel_NEWMARKmodal(:,1),vel_NEWMARKmodal(:,2),vel_NEWMARKmodal(:,
3)];
Newmark_Acc_Total = -K\M*NewmarkDisplacements'-
C\M*Newmark_Velocities';
StoryShearNWMrk = M*Newmark_Acc_Total ;
BaseShearNewMark = max(sum(StoryShearNWMrk));
Newmark_Acc_Total = Newmark_Acc_Total' ;
Base_Acc_Newmark = max(abs(Newmark_Acc_Total(:,1)));
First_Acc_Newmark = max(abs(Newmark_Acc_Total(:,2)));
Roof_Acc_Newmark  = max(abs(Newmark_Acc_Total(:,3)));


Max_BaseNewmark = max(abs(Newmark_Base));
Max_FirstFloorNewmark = max(abs(Newmark_First-Newmark_Base));
```

```
Max_RoofNewmark = max(abs(Newmark_Roof-Newmark_First));
DriftsNewMark =
table(Max_RoofNewmark,Max_FirstFloorNewmark,Max_BaseNewmark,...

Roof_Acc_Newmark,First_Acc_Newmark,Base_Acc_Newmark,BaseShearNew
Mark)


%% Export to Excel
%filename = 'Part2EXCEL.xlsx' ;
%writetable(ODEResults,filename,'Sheet',1,'WriteVariableNames',f
alse,'Range','D17')
%filename = 'Part2EXCEL.xlsx' ;
%writetable(DriftsNewMark,filename,'Sheet',1,'WriteVariableNames
',false,'Range','L17')



figure()
plot(Time(1:length(Newmark_Base)),Newmark_Base,Time(1:length(New
mark_First)),...

Newmark_First,Time(1:length(Newmark_Roof)),Newmark_Roof);
xlim([0, 50]);
legend("Base Floor", "1st Floor", "Roof");
title("Newmark Method: Relative Displacement");
ylabel("Newmark Method: Displacement(m)");
xlabel("Time(seconds)");
grid on; grid minor

figure()
plot(Time(1:length(Newmark_Acc_Total(:,1))),Newmark_Acc_Total(:,
1),Time(1:length(Newmark_Acc_Total(:,2))),...

Newmark_Acc_Total(:,2),Time(1:length(Newmark_Acc_Total(:,3))),Ne
wmark_Acc_Total(:,3));
xlim([0, 50]);
legend("Base Floor", "1st Floor", "Roof");
title("Newmark Method: Storey Acceleration");
ylabel("Newmark Method: Acceleration(m/s^2)");
xlabel("Time(seconds)");
grid on; grid minor
```

LinearTimeHistory.m

```
function [dy] = LinearTimeHistory(t,y)
global ug2dot M C K Time
dy = zeros(6,1);
ag = interp1(Time, ug2dot, t,'linear');

dy(1) = y(2) ;
dy(2) = (-C(1,1)*y(2)-C(1,2)*y(4)-K(1,1)*y(1)-K(1,2)*y(3))/M(1,1)-ag;
dy(3) = y(4);
dy(4) =(1/M(2,2))*(-C(2,1)*y(2)-C(2,2)*y(4)-C(2,3)*y(6)-K(2,1)*y(1)-
K(2,2)*y(3)-K(2,3)*y(5))-ag;
dy(5) = y(6) ;
dy(6) =(-C(3,2)*y(4)-C(3,3)*y(6)-K(3,2)*y(3)-K(3,3)*y(5))/M(3,3)-ag ;
end
```

NewmarkIntegration.m

```
function [d,v,a] = NewmarkIntegration(gamma,beta,m,c,k,ag,dt)
a = zeros(1,length(ag)+1);
v = zeros(1,length(ag)+1);
d = zeros(1,length(ag)+1);
d(1) = 0;
v(1) = 0;
a(1) = 0;
for i = 1:length(ag);

    d_init = d(i)+dt*v(i)+0.5*(1-2*beta)*dt^2*a(i);
    v_init = v(i)+(1-gamma)*dt*a(i);
    m_init = m+gamma*dt*c+beta*dt^2*k;
    Pnot = -m*ag(i)-c*v_init-k*d_init;
    a(i+1) = Pnot/m_init;
    d(i+1) = d_init+beta*dt^2*a(i+1);
    v(i+1) = v_init+gamma*dt*a(i+1);
end
d = d(2:end);
v = v(2:end);
a = a(2:end);
end
```

Part_B_Final.m

```
%% Part B
clc; close all
K1 = 250 ;
K2 = K1 ;
K3 = K2 ;
[S_a1,S_d1,Time,ug2dot] = ResponseSpectra(1); %Newhall
global M C K ug2dot Time
M = [ 325 0 0 ;
      0 325 0 ;
       0 0 325]*10^3 ; % kg

K = [ K1+K2 -K2 0 ;
      -K2 K2+K3 -K3;
      0 -K3 K3;] *10^6; %N/m

[Mode_Shapes,D] = eig(K,M);
[W] = sqrt(D) ;
Mode_Shapes ;

z = figure();
draw_line = vertcat([0,0,0],Mode_Shapes);
node = (0:3)*4;
color = {'m','b','r','g','y'};
for i = 1:3
    plot(draw_line(:,i),node,'color',color{i},'LineWidth',3);
    hold on ;
end
grid on ; grid minor;
h = vline(0,'k','Initial Position')
xlabel(' Mode Shape');
ylabel('Building Height (m)');
legend('Mode 1','Mode 2','Mode 3','Initial Position')
title("Natural Modes");

[T] = (2*pi)*[1/W(1,1) 1/W(2,2) 1/W(3,3)]; % Natural Periods.
T1 = T(1); T2 = T(2); T3 = T(3);
Natural_Periods_Fixed_Base = table(T1, T2, T3);
%Rayleigh Structural Damping
A = inv([W(1,1)^-1 W(1,1); W(2,2)^-1 W(2,2)])*2*[0.02; 0.02];
C =A(1)*M + A(2)*K ;
%Modal Analysis
Ag_vector = length(ug2dot);
r_factor = ones(3,1);
dt = 0.02;
beta = 1/4; gamma = 1/2;
Mmodal = Mode_Shapes'*M*Mode_Shapes;
Modal_M_diagonal = diag(Mmodal);
Cmodal = Mode_Shapes'*C*Mode_Shapes;
```

```
Modal_C_diagonal = diag(Cmodal);
Kmodal = Mode_Shapes'*K*Mode_Shapes;
Modal_K_diagonal = diag(Kmodal);
Time = 0:dt:(length(ug2dot)-1)*dt ;

%% Begin ODE Analysis for the Fixed Base Structure
[t,output] = ode23(@(t,y) LinearTimeHistory(t,y), Time, zeros(1,6)) ;

us0 = output(:,1) ; %Relative Displacement Base Floor
udots0 = output(:,2);   %Relative Velocity Base Floor


us1 = output(:,3) ; %Relative Displacement 1st Floor
udots1 = output(:,4);   %Relative Velocity 1st Floor

us2 = output(:,5) ; %Relative Displacement Roof Floor
udots2 = output(:,6);   %Relative Velocity Roof Floor

% Accelerations
r = ones(3,1);
ODE_Displacements = [us0,us1,us2];
ODE_Velocities = [udots0,udots1,udots2];
ODE_Acc_Total = -K\M*ODE_Displacements'-C\M*ODE_Velocities';
StoryShear = M*ODE_Acc_Total ;
ODE_Acc_Total = ODE_Acc_Total' ;
Base_ODEAcc = max(abs(ODE_Acc_Total(:,1)));
First_ODEAcc = max(abs(ODE_Acc_Total(:,2)));
Roof_ODEAcc = max(abs(ODE_Acc_Total(:,3)));
baseShear = max(sum(StoryShear)) ;

base_accODE = ODE_Acc_Total(:,1);
first_accODE = ODE_Acc_Total(:,2);
roof_accODE = ODE_Acc_Total(:,3);

% Calculate Drifts
drift_Roof_to_FirstFloor = max(abs(us2-us1));
drift_FirstFloor_to_Base = max(abs(us1-us0)) ;
drift_Base_to_Ground = max(abs(us0));

figure()
if length(Time) < length(us0)

plot(Time,us0(1:length(Time)),Time,us1(1:length(Time)),Time,us2(1:leng
th(Time)));
else

plot(Time(1:length(us0)),us0,Time(1:length(us1)),us1,Time(1:length(us2
)),us2);
end
legend("Base Floor","1st Floor","Roof ");
ylabel("Displacement(m)")
xlim([0 50]);
```

```
xlabel("Time (s)")
title("ODE Solver: Relative Displacement")
grid on ; grid minor

figure()
if length(Time) < length(us0)

plot(Time,base_accODE(1:length(Time)),Time,first_accODE(1:length(Time)
),Time,roof_accODE(1:length(Time)));
else

plot(Time(1:length(base_accODE)),base_accODE,Time(1:length(first_accOD
E)),first_accODE,Time(1:length(roof_accODE)),roof_accODE);
end
legend("Base Floor","1st Floor","Roof ");
ylabel("Acceleration(m/s^2)")
xlim([0 50]);
xlabel("Time (s)")
title("ODE Solver: Story Acceleration")
grid on ; grid minor



BaseFloor = max(abs(us0));
FirstFloor = max(abs(us1));
Roof = max(abs(us2));
ODEResults =
table(drift_Roof_to_FirstFloor,drift_FirstFloor_to_Base,drift_Base_to_
Ground,...
    Roof_ODEAcc,First_ODEAcc,Base_ODEAcc,baseShear)

%% Check ODE method with Newmark Method

L = zeros(3,3); q = zeros(Ag_vector,3);
qdot = zeros(Ag_vector,3); q2dot = zeros(Ag_vector,3);
disp_NEWMARKmodal = zeros(3,Ag_vector);
vel_NEWMARKmodal = zeros(3,Ag_vector);
acc_NEWMARKmodal = zeros(3,Ag_vector);
for i=1:3

    L(i) = Mode_Shapes(:,i)'*M*r_factor/Modal_M_diagonal(i);
%participation factor
    [q(:,i),qdot(:,i),q2dot(:,i)] =
NewmarkIntegration(gamma,beta,Modal_M_diagonal(i),...
        Modal_C_diagonal(i),Modal_K_diagonal(i),L(i)*ug2dot,dt);

    disp_NEWMARKmodal = disp_NEWMARKmodal+Mode_Shapes(:,i)*q(:,i)';
%relative displacement
    vel_NEWMARKmodal =
vel_NEWMARKmodal+Mode_Shapes(:,i)*qdot(:,i)';%relative velocity
    acc_NEWMARKmodal = acc_NEWMARKmodal+Mode_Shapes(:,i)*q2dot(:,i)';
%relative acceleration
```

```
end
Accl_BaseNM = acc_NEWMARKmodal(1,:)';   %Total Acceleration Base Floor
Accl_FirstNM = acc_NEWMARKmodal(2,:)';  %Total Acceleration 1st Floor
Accl_RoofNM = acc_NEWMARKmodal(3,:)'; %Total Acceleration Roof
vel_NEWMARKmodal = vel_NEWMARKmodal';


Newmark_Roof = disp_NEWMARKmodal(3,:);
Newmark_First = disp_NEWMARKmodal(2,:);
Newmark_Base = disp_NEWMARKmodal(1,:);
NewmarkDisplacements = [Newmark_Base',Newmark_First',Newmark_Roof'];
Newmark_Velocities =
[vel_NEWMARKmodal(:,1),vel_NEWMARKmodal(:,2),vel_NEWMARKmodal(:,3)];
Newmark_Acc_Total = -K\M*NewmarkDisplacements'-
C\M*Newmark_Velocities';
StoryShearNWMrk = M*Newmark_Acc_Total ;
BaseShearNewMark = max(sum(StoryShearNWMrk));
Newmark_Acc_Total = Newmark_Acc_Total' ;
Base_Acc_Newmark = max(abs(Newmark_Acc_Total(:,1)));
First_Acc_Newmark = max(abs(Newmark_Acc_Total(:,2)));
Roof_Acc_Newmark  = max(abs(Newmark_Acc_Total(:,3)));


Max_BaseNewmark = max(abs(Newmark_Base));
Max_FirstFloorNewmark = max(abs(Newmark_First-Newmark_Base));
Max_RoofNewmark = max(abs(Newmark_Roof-Newmark_First));
DriftsNewMark =
table(Max_RoofNewmark,Max_FirstFloorNewmark,Max_BaseNewmark,...

Roof_Acc_Newmark,First_Acc_Newmark,Base_Acc_Newmark,BaseShearNewMark)


%% Export to Excel
%filename = 'Part2EXCEL.xlsx' ;
%writetable(ODEResults,filename,'Sheet',1,'WriteVariableNames',false,'
Range','D17')
%filename = 'Part2EXCEL.xlsx' ;
%writetable(DriftsNewMark,filename,'Sheet',1,'WriteVariableNames',fals
e,'Range','L17')



figure()
plot(Time(1:length(Newmark_Base)),Newmark_Base,Time(1:length(Newmark_F
irst)),...
            Newmark_First,Time(1:length(Newmark_Roof)),Newmark_Roof);
xlim([0, 50]);
legend("Base Floor", "1st Floor", "Roof");
title("Newmark Method: Relative Displacement");
ylabel("Newmark Method: Displacement(m)");
xlabel("Time(seconds)");
grid on; grid minor
```

```
figure()
plot(Time(1:length(Newmark_Acc_Total(:,1))),Newmark_Acc_Total(:,1),Tim
e(1:length(Newmark_Acc_Total(:,2))),...

Newmark_Acc_Total(:,2),Time(1:length(Newmark_Acc_Total(:,3))),Newmark_
Acc_Total(:,3));
xlim([0, 50]);
legend("Base Floor", "1st Floor", "Roof");
title("Newmark Method: Storey Acceleration");
ylabel("Newmark Method: Acceleration(m/s^2)");
xlabel("Time(seconds)");
grid on; grid minor
```

Optimization.m (to do parts 3 and 5).

```
%% Optimization Code
clc; clear all
global Time M K C ug2dot Kp Ke Dy gamma beta n alpha K_0 nb
[~,~,~,ug2dot] = ResponseSpectra(2);
m = 325*10^3;
K1 = 250*10^6 ;
K2 = K1 ;
K3 = K1;

m_total = m*3 ;
Tb = [2,2.5,3,3.5,4,4.5,5];
Target_array = {[],[],[],[],[],[],[]};
W_total = m_total*9.81 ;
Q_W = linspace(0.02,0.2,20);
M = [m 0 0 ;
    0 m 0 ;
    0 0 m] ; %Isolated Mass Matrix (kg).

K = [K1 -K1 0 ;
    -K1 (K1+K2) -K2;
       0 -K2 K2]; %Isolated Stiffness Matrix (N/m)

K_0 =  [ K1+K2 -K2 0 ;
        -K2 K2+K3 -K3;
        0 -K3 K3;] ; %Original Stiffness Matrix (N/M)

[mode_shapes,w_sq] = eig(K_0,M) ;
W = sqrt(w_sq); % natural frequencies (rad/s)
A = inv([W(1,1)^-1 W(1,1); W(2,2)^-1  W(2,2)])*2*[0.02; 0.02];
C =A(1)*M + A(2)*K ;

dt = 0.02;
lag = length(ug2dot);
Time = 0:dt:(lag-1)*dt ;


%% Bouc-Wen Modeling Parameters
beta=0.5;
gamma=0.5;
n=5;
nb = 1 ;
%% Optimization

for i = 1:length(Tb)
    Kp = m_total*(2*pi/Tb(i))^2;
    Ke = (10*Kp);
    for j = 1:length(Q_W);
        Q = W_total*Q_W(j) ;
```

```
        Dy = Q/(Ke-Kp);
        alpha = Kp/Ke ;
        [t,output] = ode23(@(t,y) IsolatedStructure(t,y), [0 60],
zeros(1,7)) ;

        base_floor = output(:,1) ;
        base_floor_v = output(:,2);

        middle_floor = output(:,3) ;
        middle_floor_v = output(:,4);


        top_floor = output(:,5) ;
        top_floor_v = output(:,6);
        P = (alpha*Ke*output(:,1) + (1-alpha)*Ke*Dy*output(:,7));

        Displacements = [base_floor,middle_floor,top_floor];
        Velocities = [base_floor_v,middle_floor_v,top_floor_v];

        acc = (-C/M*(Velocities')-K/M*(Displacements'));
        acc = acc';


        top_acc = max(abs(acc(:,3)));
        base_drift = max(abs(base_floor)) ;
        f = Q + 2*Kp*base_drift+m*top_acc ;
        %clear t y output P ;
        Target_array{1,i}(j,1) = f ;
    end

end
plot(Q_W,Target_array{1,1},Q_W,Target_array{1,2},Q_W,Target_array{1,3}
,Q_W,Target_array{1,4},Q_W,Target_array{1,5},...
     Q_W,Target_array{1,6},Q_W,Target_array{1,7});
 grid on; grid minor;
 title("Optimization Graph")
```

IsolatedStructure.m (this is different from the IsolatedStructureSingular.m in that this does not have an “nb” variable).

```
function [dy] = IsolatedStructure(t,y)
global Time ug2dot M C K Ke Kp Dy gamma beta n nb
%% Arrange the state-space vectors for the isolated structure.
%-------------------------------------------------------------------
%[y] = [ub... y(1)
%      ubdot...y(2)
%      u1...y(3)
%      u1dot...y(4)
%      u2...y(5)
%      u2dot...y(6)
%      z(t)...y(7)]  7-state vectors we need to solve for.
%-------------------------------------------------------------------
dy = zeros(7,1);
ag = interp1(Time,ug2dot(1:length(Time)),t,'linear');
alpha = Kp/Ke ;
P = (alpha*Ke*y(1) + (1-alpha)*Ke*Dy*y(7));
%P = (alpha*Ke*y(1) + (1-alpha)*Ke*Dy*y(7))*nb;

dy(1) = y(2) ;
dy(2) = -ag-
(C(1,1)*y(2)+C(1,2)*y(4)+K(1,1)*y(1)+K(1,2)*y(3)+P)/M(1,1);
dy(3) = y(4) ;
dy(4) = -ag-
(C(2,1)*y(2)+C(2,2)*y(4)+C(2,3)*y(6)+K(2,1)*y(1)+K(2,2)*y(3)+K(2,3)*y(
5))/M(2,2);
dy(5) = y(6);
dy(6) = -ag-(C(3,2)*y(4)+C(3,3)*y(6)+K(3,2)*y(3)+K(3,3)*y(5))/M(3,3);
dy(7) = (y(2)-gamma*abs(y(2))*y(7)*(abs(y(7)))^(n-1)-
beta*y(2)*(abs(y(7)))^n)/Dy;


end
```

Final_Bearing_Design.m

```
%% Isolated Building
clc ;   close all ; clear all
global Time M K C ug2dot Kp Ke Dy gamma beta n alpha K_0 nb
[S_a1,S_d1,Time,ug2dot] = ResponseSpectra(1);
mb = 325*10^3 ;
m1 = mb ;
m2 = m1;
K1 = 250*10^6 ;
K2 = K1 ;
K3 = K1;

%% Bouc-Wen Modeling Parameters
beta=0.5;
gamma=0.5;
n=5;
%% Bearing Parameters

Ke =4752031.749; Kp = (1/10)*Ke ;
alpha = Kp/Ke ;
Dy = 0.045;
Q = Dy*(Ke-Kp);
nb = 9 ;

%% Assemble the Matricies

M = [mb 0 0 ;
    0 m1 0 ;
    0 0 m2] ; %Isolated Mass Matrix (kg).

K = [K1 -K1 0 ;
    -K1 (K1+K2) -K2;
       0 -K2 K2]; %Isolated Stiffness Matrix (N/m)

K_0 =  [ K1+K2 -K2 0 ;
        -K2 K2+K3 -K3;
        0 -K3 K3;] ; %Original Stiffness Matrix (N/M)

[mode_shapes,w_sq] = eig(K_0,M) ;
W = sqrt(w_sq); % natural frequencies (rad/s)
A = inv([W(1,1)^-1 W(1,1); W(2,2)^-1  W(2,2)])*2*[0.02; 0.02];
C =A(1)*M + A(2)*K ;

dt = 0.02;
lag = length(ug2dot);
Time = 0:dt:(lag-1)*dt ;
%% Begin ODE Nonlinear Analysis
[t,output] = ode23(@(t,y) IsolatedStructureSingular(t,y), [0 60],
zeros(1,7)) ;
```

```
base_floor = output(:,1) ;
base_floor_v = output(:,2);

middle_floor = output(:,3) ;
middle_floor_v = output(:,4);


top_floor = output(:,5) ;
top_floor_v = output(:,6);

P = (alpha*Ke*output(:,1) + (1-alpha)*Ke*Dy*output(:,7))*nb;
Displacements = [base_floor,middle_floor,top_floor];
Velocities = [base_floor_v,middle_floor_v,top_floor_v];

acc = (-C/M*(Velocities')-K/M*(Displacements'));
StoryShear = M*acc ;
acc = acc';

base_acc1 = acc(:,1)-P/M(1,1) ;
middle_acc1 = acc(:,2) ;
top_acc1 = acc(:,3) ;

base_acc = max(abs(base_acc1)) ;
middle_acc = max(abs(acc(:,2))) ;
top_acc = max(abs(acc(:,3)));

accelerations = table(base_acc,middle_acc,top_acc)

P_MN = P/1000000; %to change to MN
figure() ;
plot(base_floor*1000,P_MN);
grid on; grid minor ;
hold on
xlabel("Bearing Displacement (mm)");
ylabel("P (MN)");
title("Force-Displacement Loop: Bouc-Wen Modeling")


max(abs(top_floor)) ;
roof_drift = max(abs((top_floor-middle_floor)))
middle_drift = max(abs((middle_floor-base_floor)))
base_drift = max(abs(base_floor))

baseShear = max(sum(StoryShear))

%% Plot Relative Displacements
figure()
if length(Time) < length(base_floor)

plot(Time,base_floor(1:length(Time)),Time,middle_floor(1:length(Time))
,Time,top_floor(1:length(Time)));
else
```

```
plot(Time(1:length(base_floor)),base_floor,Time(1:length(middle_floor)
),middle_floor,Time(1:length(top_floor)),top_floor);
end
legend("Base Floor","1st Floor","Roof ");
ylabel("Displacement(m)")
xlim([0 50]);
xlabel("Time (s)")
title("Isolated Structure: Relative Displacement (m/s) ")
grid on ; grid minor

%% Plot Total Accelerations
figure()
if length(Time) < length(base_acc1)

plot(Time,base_acc1(1:length(Time)),Time,middle_acc1(1:length(Time)),T
ime,top_acc1(1:length(Time)));
else

plot(Time(1:length(base_acc1)),base_acc1,Time(1:length(middle_acc1)),m
iddle_acc1,Time(1:length(top_acc1)),top_acc1);
end
legend("Base Floor","1st Floor","Roof ");
ylabel("Acceleration(m/s^2)")
xlim([0 50]);
xlabel("Time (s)")
title("Isolated Structure: Total Acceleration  (m/s^2)")
grid on ; grid minor


%% Shear and Displacement in isolation unit.
sigma_tr = 0.444;  % m
u_base = base_drift;  %m
shear_strain = (u_base/sigma_tr)*100;  %in percent

%% Tables
NTLA = table(roof_drift,middle_drift,base_drift,...
    top_acc,middle_acc,base_acc,baseShear)



%% Export to Excel
%filename = 'NTLAIsolated.xlsx' ;
%writetable(NTLA,filename,'Sheet',1,'WriteVariableNames',false,'Range'
,'E11')
```

IsolatedStructureSingular.m

```
function [dy] = IsolatedStructureSingular(t,y)
global Time ug2dot M C K Ke Kp Dy gamma beta n nb
%% Arrange the state-space vectors for the isolated structure.
%-------------------------------------------------------------------
%[y] = [ub... y(1)
%      ubdot...y(2)
%      u1...y(3)
%      u1dot...y(4)
%      u2...y(5)
%      u2dot...y(6)
%      z(t)...y(7)]  7-state vectors we need to solve for.
%-------------------------------------------------------------------
dy = zeros(7,1);
ag = interp1(Time,ug2dot(1:length(Time)),t,'linear');
alpha = Kp/Ke ;
P = (alpha*Ke*y(1) + (1-alpha)*Ke*Dy*y(7))*nb;

dy(1) = y(2) ;
dy(2) = -ag-
(C(1,1)*y(2)+C(1,2)*y(4)+K(1,1)*y(1)+K(1,2)*y(3)+P)/M(1,1);
dy(3) = y(4) ;
dy(4) = -ag-
(C(2,1)*y(2)+C(2,2)*y(4)+C(2,3)*y(6)+K(2,1)*y(1)+K(2,2)*y(3)+K(2,3)*y(
5))/M(2,2);
dy(5) = y(6);
dy(6) = -ag-(C(3,2)*y(4)+C(3,3)*y(6)+K(3,2)*y(3)+K(3,3)*y(5))/M(3,3);
dy(7) = (y(2)-gamma*abs(y(2))*y(7)*(abs(y(7)))^(n-1)-
beta*y(2)*(abs(y(7)))^n)/Dy;


end
```