Nonlinear Frame Analysis- Project 2

Benjamin Enssle
Franco Sola

Civil, Environmental and Architectural Engineering
UNIVERSITY OF COLORADO BOULDER

Structural Engineering & Structural Mechanics

CVEN 5111- Structural Dynamics

Professor Dr. P. Sideris

December 9th, 2016

Abstract

Analysis to predict the response of structures to ground motion is the basis of earthquake engineering. It allows for direct representation on how a system will behave under loading conditions caused by the moving ground. Ground motions vary greatly between different earthquakes, and depend on the location of site with respect to fault lines and epicenter, geologic formations, and soil composition. Still, valuable insight regarding the behavior of a simple shear "building" can be obtained by analyzing its response to a particular ground motion shaking, in this case, the 1940 El Centro earthquake in the Southern California Imperial Valley.

Table of Contents

Abstract 2
List of Figures 2
Introduction 3
Code Development 4
Linear Elastic Analysis 6
Nonlinear Analysis –Retrofitted System – Seismic Loading: 16
Conclusion 31
Appendix 32

List of Figures

Modal Shapes........7
Response to Harmonic Excitation-Analytical Response........13
Response to Harmonic Exciation........14
Ground Shaking Accelogram........15
Base Shear Time History........16
Relative Displacement at First Floor........17
Relative Displacement at Second Floor........18
Relative Displacement at Third Floor........19
Inter-story Displacement at First Floor........20
Inter-story Displacement at Second Floor........21
Inter-story Displacement at Third Floor........22
Inter-story Force vs. Inter-story Displacement at First Floor........23
Inter-story Force vs. Inter-story Displacement at Second Floor........24
Inter-story Force vs. Inter-story Displacement at Third Floor........25
Non-Linear Damping Force vs Inter-Story Velocity at First Floor........26
Non-Linear Damping Force vs Inter-Story Velocity at Second Floor........27
Non-Linear Damping Force vs Inter-Story Displacement at First Floor........28
Non-Linear Damping Force vs Inter-Story Displacement at Second Floor........29

Introduction

The analysis presented in this report is based on a 3-story shear building with two retrofitted nonlinear dampers located in the first and second story. Several simplifying assumptions (axially rigid columns and axially and flexurally rigid beam-slabs) were made to model a three-story "shear building" as a three-degree-of-freedom system which consists of horizontal displacement at each floor level, where each concrete rigid slab is modelled as a point mass. The displacement (u), velocity ($\dot{u}$) and acceleration ($\ddot{u}$) of each slab are thereby constrained along the local horizontal x-axes, which simplifies the problem while maintaining an accurate representation of the actual response of a structure under limited amplitude ground shaking. The columns' deformation response is based on the nonlinear hysteretic model, which further simplifies the system. The accuracy of this model decreases substantially when predicted plastic deformation substantially exceeds the assumed plastic inter-story force deformation limit.

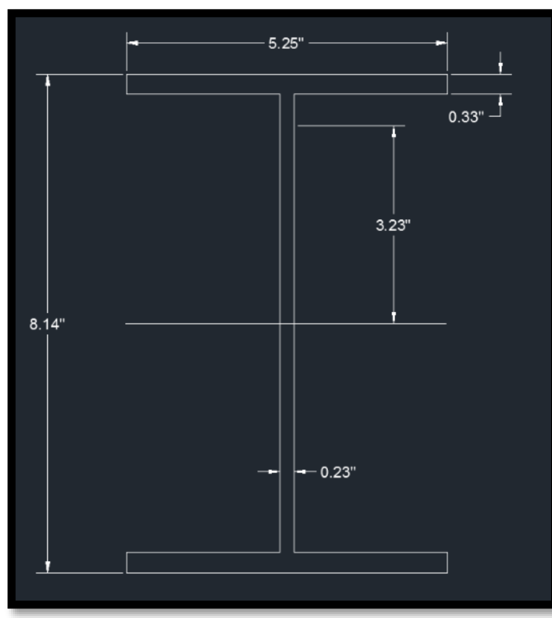
Matlab was used to numerically solve the equation of motion for the system of nonlinear ordinary differential equations using the Newton-Raphson method with trapezoidal integration. The script has been written to analyze the response of the structure to arbitrary ground accelerometer data. The predicted displacement, velocity and acceleration time histories, as well as force response from the columns can be easily manipulated to produce plots that graphically represent the relationship between these response characteristics with respect to time or to other response parameters.

W8x18 Beam Properties:

Area=5.26 in^2

I=61.9 in^4

$$d = \frac{0.33*5.25*3.905+.23*3.74*1.87}{0.33*5.25+0.23*3.74} = 3.23\ in$$

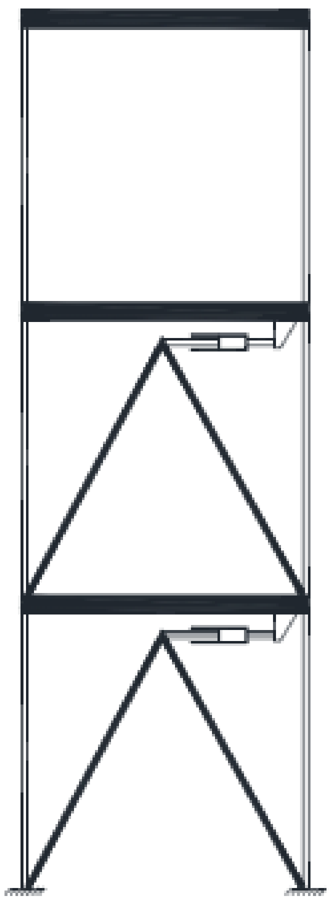


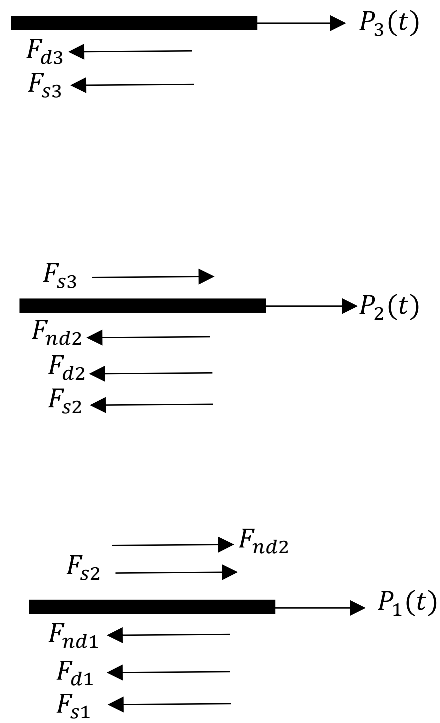
The column stiffness is calculated using information from above. Since columns are oriented to bend about the strong axis alone, the second moment of area about the principal axis is used to calculate column flexural stiffness. The lateral design force for each column is assumed to be the minimum force that will cause fully plastic flexural yielding at some point(s) in the column.

Code Development

1.

FBD





Equation of Motion

$$\boldsymbol{m}\ddot{\vec{u}} + \boldsymbol{c}\dot{\vec{u}} + \vec{F}_{nd} + \vec{F}_{s} = -\boldsymbol{m}\vec{r}_g \ddot{u}_g(t)$$

Where:

$$\vec{F}_{nd} = \begin{bmatrix} F_{nd1} - F_{nd2} \\ F_{nd2} \\ 0 \end{bmatrix}$$

$$F_{nd}(\dot{u}) = \begin{cases} c_{nd}\dot{u} & |\dot{u}| \leq 1\ in/sec \\ c_{nd}|\dot{u}|^{0.6} sgn(\dot{u}) & |\dot{u}| \geq 1\ in/sec \end{cases}$$

$$\vec{F_s} = \begin{bmatrix} F_{s1} - F_{s2} \\ F_{s2} - F_{s3} \\ F_{s3} \end{bmatrix}$$

$$F_{s1} = k_{str} * u_1 \qquad |F_{s1}| \leq F_{y,str}$$

$$F_{s2} = k_{str} * (u_2 - u_1) \qquad |F_{s2}| \leq F_{y,str}$$

$$F_{s2} = k_{str} * (u_3 - u_2) \qquad |F_{s3}| \leq F_{y,str}$$

$$\vec{r_g} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Newton Raphson

The implicit Trapezoidal Rule was used for the numerical integration of the ordinary differential equation described in the equation of motion. Since it is a second order method, it provides high accuracy compared to explicit and implicit Euler's methods (both first order) and it proves to be numerically stable compared with the first-order methods. The Newton-Raphson iteration scheme was utilized to solve the EOM using the Trapezoidal Rule, as it is a robust method that converges well at the cost of computational time. The additional computational requirements of the implicit Euler solution with trapezoidal integration are not cumbersome in this case since the system in relatively simple, and because the Jacobian, which must be inverted, can be written in terms of the displacement vector alone, making it a 3 x 3 matrix in this case study.

$$\vec{u}_i = \vec{u}_{i-1} + \frac{h}{2}(\dot{\vec{u}}_i + \dot{\vec{u}}_{i-1})$$

$$\dot{\vec{u}}_i = \dot{\vec{u}}_{i-1} + \frac{h}{2}(\ddot{\vec{u}}_i + \ddot{\vec{u}}_{i-1})$$

$$\ddot{\vec{u}}_i = \boldsymbol{m}^{-1}(\vec{P}_i - \vec{F}_{s,i} - \vec{F}_{nd,i-1} - \boldsymbol{c}\dot{\vec{u}}_i)$$

Where:

$$\vec{P}_i = -\boldsymbol{m}\vec{r}_g \ddot{u}_g(t_i)$$

Trapezoidal Rule

$$\dot{\vec{u}}_i = \dot{\vec{u}}_{i-1} + \frac{h}{2} * \boldsymbol{m}^{-1}(\vec{P}_i + \vec{P}_{i-1} - \vec{F}_{s,i} - \vec{F}_{s,i-1} - \boldsymbol{c}\dot{\vec{u}}_i - \boldsymbol{c}\dot{\vec{u}}_{i-1} - \vec{F}_{nd,i} - \vec{F}_{nd,i-1})$$

$$\dot{\vec{u}}_i = \frac{2}{h}(\vec{u}_i) - \frac{2}{h}(\vec{u}_{i_{i-1}}) - \dot{\vec{u}}_{i-1}$$

Newton Raphson

$$R = -\frac{2}{h}(\vec{u}_i) + \frac{2}{h}(\vec{u}_{i-1}) + 2(\dot{\vec{u}}_{i-1}) + \frac{h}{2} * \boldsymbol{m}^{-1}(\vec{P}_i + \vec{P}_{i-1} - \vec{F}_{s,i} - \vec{F}_{s,i-1} - \vec{F}_{nd,i} - \vec{F}_{nd,i-1} - c\dot{\vec{u}}_i - c\dot{\vec{u}}_{i-1})$$

$$\frac{\partial R}{\partial \vec{u}_i} = \frac{4}{h^2} * \boldsymbol{m} + (\boldsymbol{k_{tan}} + \boldsymbol{c}\frac{2}{h} + \boldsymbol{c_{tan}}\frac{2}{h})$$

Where:

$$\boldsymbol{c_{tan}} = \begin{bmatrix} c_{tan,1} + c_{tan,2} & -c_{tan,2} & 0 \\ -c_{tan,2} & c_{tan,2} & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \boldsymbol{k_{tan}} = \begin{bmatrix} k_{tan,1} + k_{tan,2} & -k_{tan,2} & 0 \\ -k_{tan,2} & k_{tan,2} + k_{tan,3} & -k_{tan,3} \\ 0 & -k_{tan,3} & k_{tan,3} \end{bmatrix}$$

Tolerance:

A tolerance of |R|<1e-6 was used for Newton Raphson iterations convergence

Linear Elastic Analysis

3.

Equation of Motion

$$\boldsymbol{m}\ddot{\vec{u}} + \boldsymbol{c}\dot{\vec{u}} + \mathbf{k}\vec{u} = -\boldsymbol{m}\ddot{u}_g(t)$$

Stiffness Matrix

$$k = 2\left(\frac{12EI}{h^3}\right) = \frac{24EI}{h^3} = \frac{24(29000ksi)(61.9in^4)}{(120\ in)^3} = 24.93\frac{kips}{in}$$

$$\boldsymbol{k} = k * \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

Mass Matrix

$$m = \frac{w}{g} = \frac{20}{386.4} = 0.05176\ \frac{kips - sec^2}{in}$$

$$\boldsymbol{m} = m * \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

4.

$$(\boldsymbol{k} - \omega^2 \boldsymbol{m})\,\vec{\varphi} = \vec{0}$$

$$\left(k * \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} - \omega^2 m * \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right)\vec{\varphi} = \vec{0}$$

$$\left(\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right)\vec{\varphi} = \vec{0}$$

Where $\lambda = \frac{\omega^2 m}{k}$

Matlab function [phi_vect, Lam] = eig (k,m) used to calculate Eigenvectors and Eigenvalues:

$$\lambda = \begin{bmatrix} 0.1981 \\ 1.5550 \\ 3.2470 \end{bmatrix}$$

$$\vec{\varphi} = \begin{bmatrix} -0.3280 & 0.7370 & -0.5910 \\ -0.5910 & 0.3280 & 0.7370 \\ -0.7370 & -0.5910 & -0.3280 \end{bmatrix}$$

Mode Shapes Normalized:

$$\vec{\varphi}_1 = \begin{bmatrix} 0.4450 \\ 0.8019 \\ 1.0000 \end{bmatrix} \qquad \vec{\varphi}_2 = \begin{bmatrix} -1.2470 \\ -0.5550 \\ 1.0000 \end{bmatrix} \qquad \vec{\varphi}_3 = \begin{bmatrix} 1.8019 \\ -2.2469 \\ 1.0000 \end{bmatrix}$$

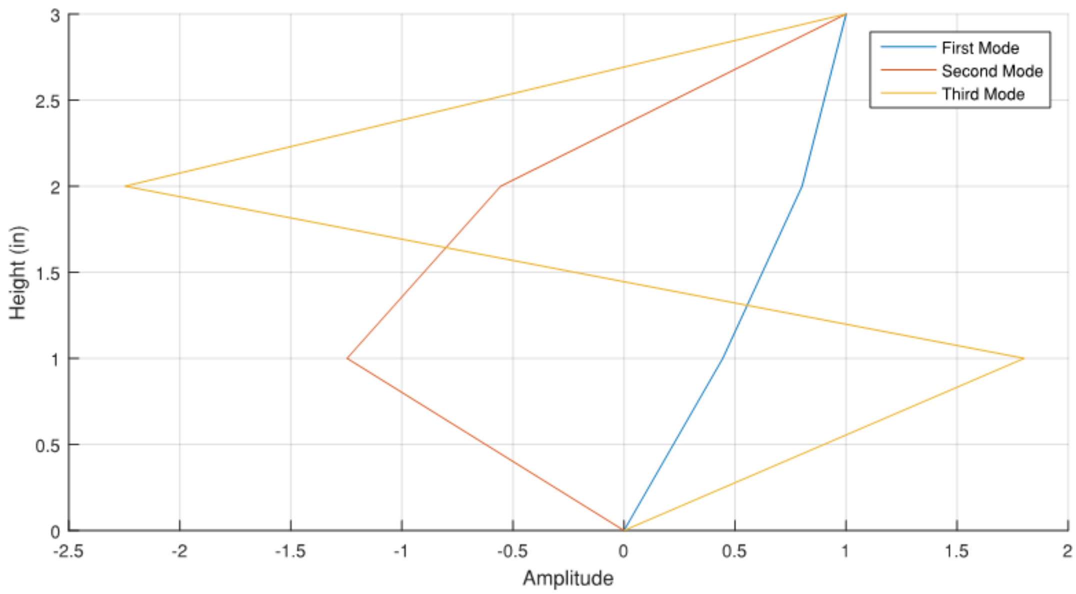


FIGURE 1- MODAL SHAPES

Undamped Natural Frequencies

$$\lambda = \frac{\omega^2 m}{k}$$

$$\omega_1 = \sqrt{\frac{\lambda_1 k}{m}} = \sqrt{\frac{0.1981 * 24.93}{0.05176}} = 9.768\,\frac{rad}{sec}$$

$$\omega_2 = \sqrt{\frac{\lambda_2 k}{m}} = \sqrt{\frac{1.5550 * 24.93}{0.05176}} = 27.367\,\frac{rad}{sec}$$

$$\omega_3 = \sqrt{\frac{\lambda_3 k}{m}} = \sqrt{\frac{3.2470 * 24.93}{0.05176}} = 39.546\,\frac{rad}{sec}$$

5.

$$\frac{1}{2}\begin{bmatrix} \frac{1}{9.768} & 9.768 \\ \frac{1}{27.367} & 27.367 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix} = \begin{Bmatrix} 0.03 \\ 0.05 \end{Bmatrix}$$

$$\begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix} = 2 * \begin{Bmatrix} 0.03 \\ 0.05 \end{Bmatrix} * \begin{bmatrix} \frac{1}{9.768} & 9.768 \\ \frac{1}{27.367} & 27.367 \end{bmatrix}^{-1}$$

$$\begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix} = \begin{Bmatrix} 0.2721 \\ 0.0033 \end{Bmatrix}$$

$$\boldsymbol{c} = a_0 \boldsymbol{m} + a_1 \boldsymbol{k}$$

$$\boldsymbol{c} = 0.2721 * 0.05176 * \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + 0.0033 * 24.93 * \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

$$\boldsymbol{c} = \begin{bmatrix} 0.1786 & -0.0823 & 0 \\ -0.0823 & 0.1786 & -0.0823 \\ 0 & -0.0823 & 0.0964 \end{bmatrix}$$

$$\xi_3 = \frac{\vec{\varphi}_3^{\,T} \boldsymbol{c}\, \vec{\varphi}_3}{2\left(\vec{\varphi}_3^{\,T} \boldsymbol{m}\, \vec{\varphi}_3\right) \omega_3} = 0.0687$$

$$\omega_{1d} = \omega_1 \sqrt{1 - {\xi_1}^2} = 9.768\sqrt{1 - 0.03^2} = 9.763 \frac{rad}{sec}$$

$$\omega_{2d} = \omega_2 \sqrt{1 - {\xi_2}^2} = 27.367\sqrt{1 - 0.05^2} = 27.333 \frac{rad}{sec}$$

$$\omega_{3d} = \omega_3 \sqrt{1 - {\xi_3}^2} = 39.546\sqrt{1 - 0.0687} = 39.423 \frac{rad}{sec}$$

$$T_{1d} = \frac{2\pi}{9.763} = 0.644\ sec$$

$$T_{2d} = \frac{2\pi}{27.333} = 0.230\ sec$$

$$T_{3d} = \frac{2\pi}{39.423} = 0.160\ sec$$

6.

First Mode

$$M_1 = \vec{\varphi}_1{}^T \boldsymbol{m}\, \vec{\varphi}_1 = \begin{bmatrix} 0.4450 \\ 0.8019 \\ 1.0000 \end{bmatrix}^T \begin{bmatrix} 0.05176 & 0 & 0 \\ 0 & 0.05176 & 0 \\ 0 & 0 & 0.05716 \end{bmatrix} \begin{bmatrix} 0.4450 \\ 0.8019 \\ 1.0000 \end{bmatrix} = 0.0953\ \frac{kips - sec^2}{in}$$

$$K_1 = \vec{\varphi}_1{}^T \boldsymbol{k}\, \vec{\varphi}_1 = \begin{bmatrix} 0.4450 \\ 0.8019 \\ 1.0000 \end{bmatrix}^T * 24.93 \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 0.4450 \\ 0.8019 \\ 1.0000 \end{bmatrix} = 9.0906\ \frac{kips}{in}$$

$$C_1 = \vec{\varphi}_1{}^T \boldsymbol{c}\, \vec{\varphi}_1 = \begin{bmatrix} 0.4450 \\ 0.8019 \\ 1.0000 \end{bmatrix}^T \begin{bmatrix} 0.1786 & -0.0823 & 0 \\ -0.0823 & 0.1786 & -0.0823 \\ 0 & -0.0823 & 0.0964 \end{bmatrix} \begin{bmatrix} 0.4450 \\ 0.8019 \\ 1.0000 \end{bmatrix} = 0.0559$$

Second Mode

$$M_2 = \vec{\varphi}_2{}^T \boldsymbol{m}\, \vec{\varphi}_2 = \begin{bmatrix} -1.2470 \\ -0.5550 \\ 1.0000 \end{bmatrix}^T \begin{bmatrix} 0.05176 & 0 & 0 \\ 0 & 0.05176 & 0 \\ 0 & 0 & 0.05716 \end{bmatrix} \begin{bmatrix} -1.2470 \\ -0.5550 \\ 1.0000 \end{bmatrix} = 0.1482\ \frac{kips - sec^2}{in}$$

$$K_2 = \vec{\varphi}_2{}^T \boldsymbol{k}\, \vec{\varphi}_2 = \begin{bmatrix} -1.2470 \\ -0.5550 \\ 1.0000 \end{bmatrix}^T * 24.93 \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} -1.2470 \\ -0.5550 \\ 1.0000 \end{bmatrix} = 110.9858\ \frac{kips}{in}$$

$$C_2 = \vec{\varphi}_2{}^T \boldsymbol{c}\, \vec{\varphi}_2 = \begin{bmatrix} -1.2470 \\ -0.5550 \\ 1.0000 \end{bmatrix}^T \begin{bmatrix} 0.1786 & -0.0823 & 0 \\ -0.0823 & 0.1786 & -0.0823 \\ 0 & -0.0823 & 0.0964 \end{bmatrix} \begin{bmatrix} -1.2470 \\ -0.5550 \\ 1.0000 \end{bmatrix} = 0.4066$$

Third Mode

$$M_3 = \vec{\varphi}_3{}^T \boldsymbol{m}\, \vec{\varphi}_3 = \begin{bmatrix} 1.8019 \\ -2.2469 \\ 1.0000 \end{bmatrix}^T \begin{bmatrix} 0.05176 & 0 & 0 \\ 0 & 0.05176 & 0 \\ 0 & 0 & 0.05716 \end{bmatrix} \begin{bmatrix} 1.8019 \\ -2.2469 \\ 1.0000 \end{bmatrix} = 0.4811\ \frac{kips - sec^2}{in}$$

$$K_3 = \vec{\varphi}_3{}^T \boldsymbol{k}\, \vec{\varphi}_3 = \begin{bmatrix} 1.8019 \\ -2.2469 \\ 1.0000 \end{bmatrix}^T * 24.93 \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1.8019 \\ -2.2469 \\ 1.0000 \end{bmatrix} = 752.4369\ \frac{kips}{in}$$

$$C_3 = \vec{\varphi}_3{}^T \boldsymbol{c}\, \vec{\varphi}_3 = \begin{bmatrix} 1.8019 \\ -2.2469 \\ 1.0000 \end{bmatrix}^T \begin{bmatrix} 0.1786 & -0.0823 & 0 \\ -0.0823 & 0.1786 & -0.0823 \\ 0 & -0.0823 & 0.0964 \end{bmatrix} \begin{bmatrix} 1.8019 \\ -2.2469 \\ 1.0000 \end{bmatrix} = 2.6142$$

Modal Forces

$$P(t) = -\boldsymbol{m} * g \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \sin(\omega * t)$$

Where:

$$\omega = 0.7 * \omega_1$$

$$P_1 = \vec{\varphi}_1{}^T P(t) = \begin{bmatrix} -1.2470 \\ -0.5550 \\ 1.0000 \end{bmatrix}^T \boldsymbol{m} * g \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}^T = -44.9381 \sin(\omega * t)$$

$$P_2 = \vec{\varphi}_2{}^T P(t) = \begin{bmatrix} -1.2470 \\ -0.5550 \\ 1.0000 \end{bmatrix}^T \boldsymbol{m} * g \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}^T = 16.0401 \sin(\omega * t)$$

$$P_3 = \vec{\varphi}_3{}^T P(t) = \begin{bmatrix} 1.8019 \\ -2.2469 \\ 1.0000 \end{bmatrix}^T \boldsymbol{m} * g \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}^T = -11.1000 \sin(\omega * t)$$

Modal Equation of Motion

$$M_n \ddot{q}_n + C_n \dot{q}_n + K_n q_n = P_n(t)$$

$$\ddot{q}_n + 2\xi_n \omega_n \dot{q}_n + \omega_n{}^2 q_n = \frac{P_n(t)}{M_n}$$

Steady State Response:

$$q_n(t) = \frac{P_{n,o}}{K_n} R_{dn} \sin(wt - \varphi_n)$$

Where:

$$R_{dn} = \frac{1}{\left[\left(1 - \left(\frac{\omega}{\omega_n}\right)^2\right)^2 + \left(2\xi_n \left(\frac{\omega}{\omega_n}\right)\right)^2\right]^{1/2}}$$

$$\varphi_n = COS^{-1}\left(\frac{1 - \left(\frac{\omega}{\omega_n}\right)^2}{\left[\left(1 - \left(\frac{\omega}{\omega_n}\right)^2\right)^2 + \left(2\xi_n \left(\frac{\omega}{\omega_n}\right)\right)^2\right]^{1/2}}\right)$$

First Mode:

$$q_1(t) = \frac{P_{n,1}}{K_1} R_{d1} \sin(wt - \varphi_1)$$

$$R_{d1} = \frac{1}{\left[\left(1-\left(\frac{6.8376}{9.768}\right)^2\right)^2 + \left(2*0.03\left(\frac{6.8376}{9.768}\right)\right)^2\right]^{1/2}} = 1.954$$

$$\varphi_1 = COS^{-1}\left(\frac{1-\left(\frac{6.8376}{9.768}\right)^2}{\left[\left(1-\left(\frac{6.8376}{9.768}\right)^2\right)^2 + \left(2*0.03\left(\frac{6.8376}{9.768}\right)\right)^2\right]^{1/2}}\right) = 0.0822$$

$$q_1(t) = \frac{-44.9381}{9.0906}(1.954)\sin(6.8376t - 0.0822)$$

Second Mode:

$$q_2(t) = \frac{P_{n,2}}{K_2} R_{d2} \sin(wt - \varphi_2)$$

$$R_{d2} = \frac{1}{\left[\left(1-\left(\frac{6.8376}{27.367}\right)^2\right)^2 + \left(2*0.05\left(\frac{6.8376}{27.367}\right)\right)^2\right]^{1/2}} = 1.0662$$

$$\varphi_2 = COS^{-1}\left(\frac{1-\left(\frac{6.8376}{27.367}\right)^2}{\left[\left(1-\left(\frac{6.8376}{27.367}\right)^2\right)^2 + \left(2*0.05\left(\frac{6.8376}{27.367}\right)\right)^2\right]^{1/2}}\right) = 0.0266$$

$$q_2(t) = \frac{16.0401}{110.9858}(1.0662)\sin(6.8376t - 0.0266)$$

Third Mode:

$$q_3(t) = \frac{P_{n,3}}{K_3} R_{d3} \sin(wt - \varphi_2)$$

$$R_{d3} = \frac{1}{\left[\left(1 - \left(\frac{6.8376}{39.546}\right)^2\right)^2 + \left(2 * 0.0687 \left(\frac{6.8376}{39.546}\right)\right)^2\right]^{1/2}} = 1.0305$$

$$\varphi_3 = COS^{-1}\left(\frac{1 - \left(\frac{6.8376}{27.367}\right)^2}{\left[\left(1 - \left(\frac{6.8376}{39.546}\right)^2\right)^2 + \left(2 * 0.0687 \left(\frac{6.8376}{39.546}\right)\right)^2\right]^{1/2}}\right) = 0.0245$$

$$q_3(t) = \frac{-11.1}{752.4369}(1.0305)\sin(6.8376t - 0.0245)$$

$$\vec{u}(t) = \vec{\varphi}_1 q_1(t) + \vec{\varphi}_2 q_2(t) + \vec{\varphi}_3 q_3(t)$$

$$\begin{aligned}\vec{u}(t) = \quad & \begin{bmatrix} 0.4450 \\ 0.8019 \\ 1.0000 \end{bmatrix} (-9.659) \sin(6.8376t - 0.0822) \\ & + \begin{bmatrix} -1.2470 \\ -0.5550 \\ 1.0000 \end{bmatrix} (0.1541) \sin(6.8376t - 0.0266) \\ & + \begin{bmatrix} 1.8019 \\ -2.2469 \\ 1.0000 \end{bmatrix} (-0.0152)\sin(6.8376t - 0.0245)\end{aligned}$$

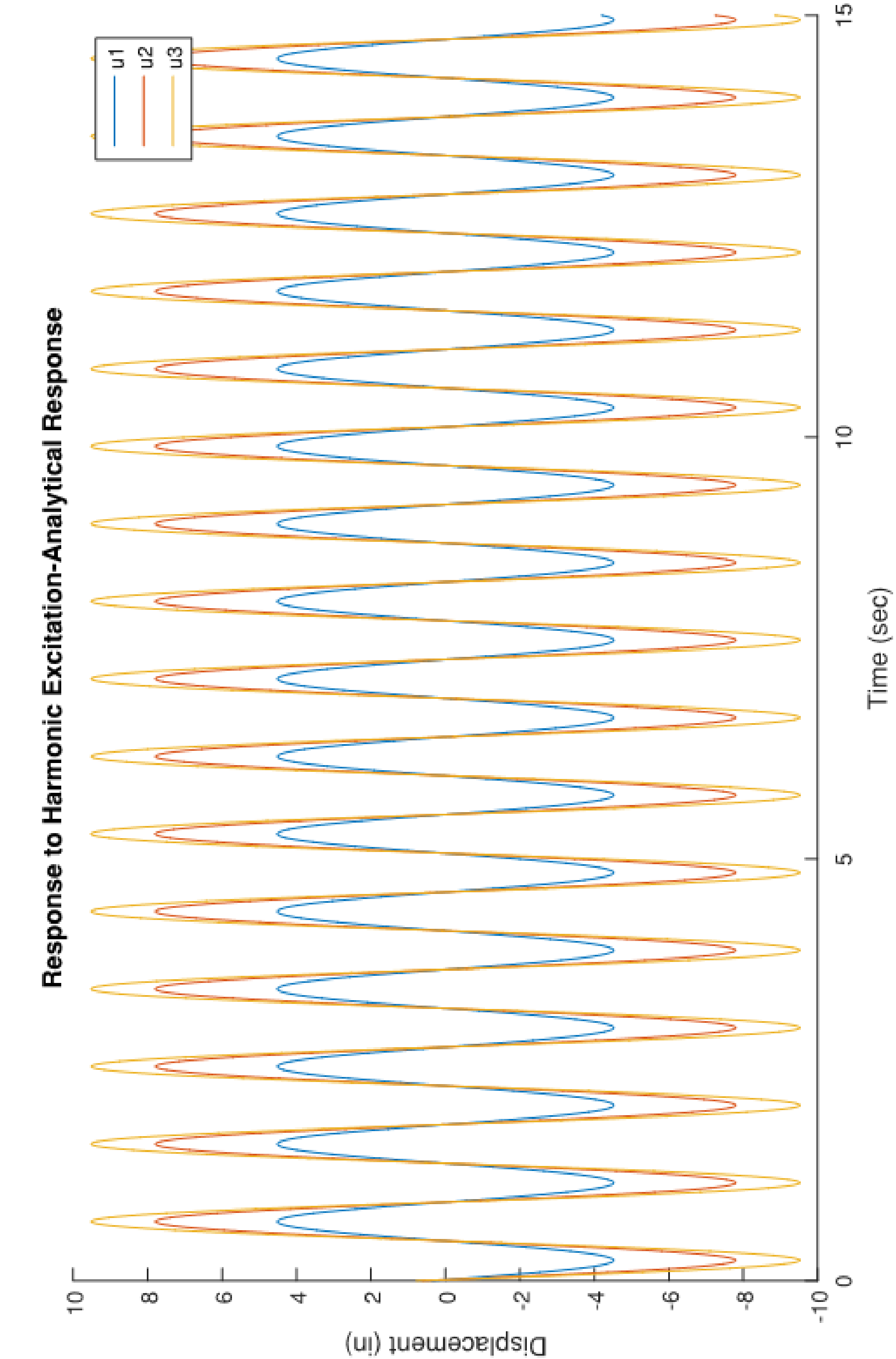
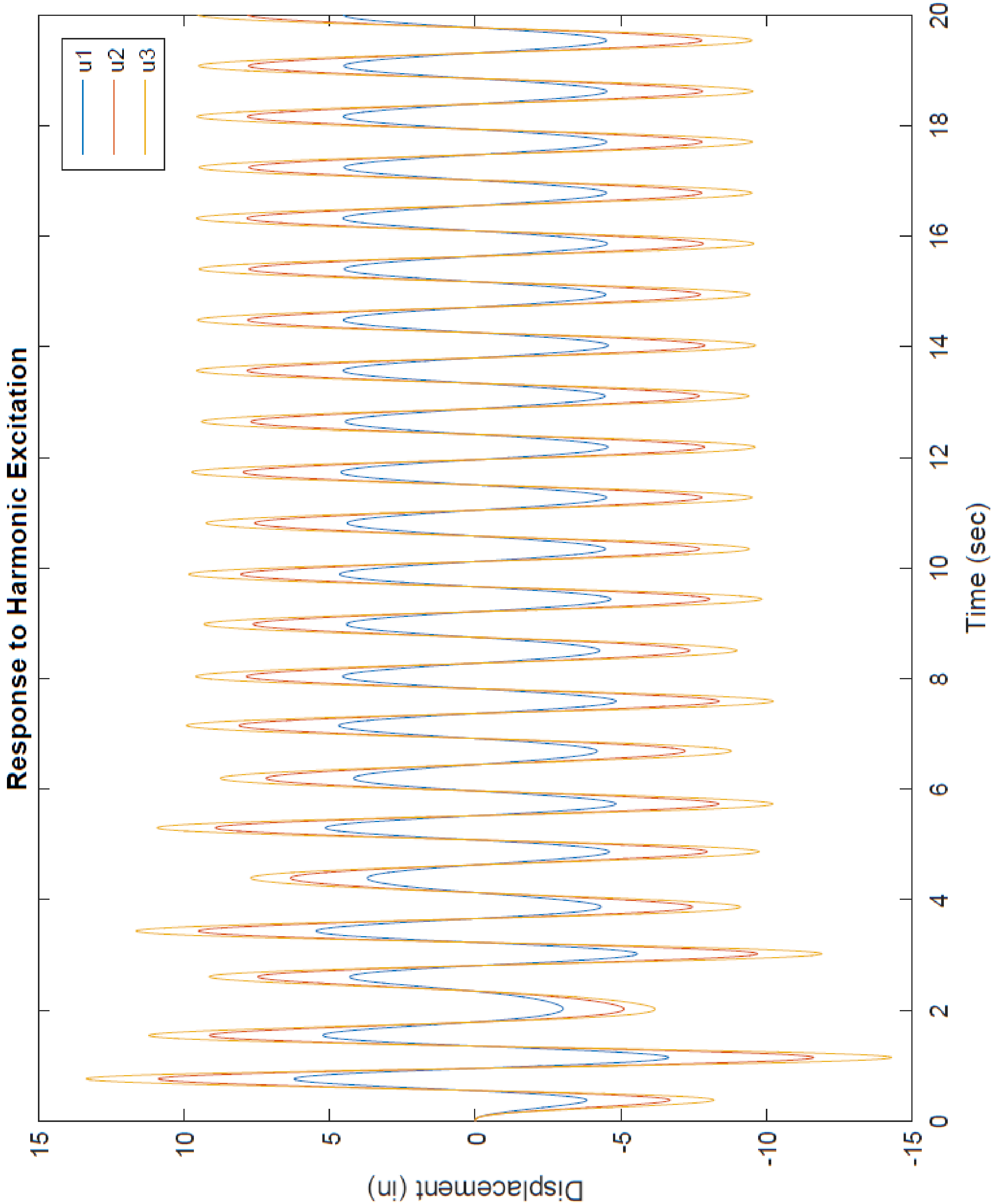

Response to Harmonic Excitation-Analytical Response
Displacement (in)
Time (sec)
u1
u2
u3
10
8
6
4
2
0
-2
-4
-6
-8
-10
0
5
10
15

FIGURE 2

7.



Response to Harmonic Excitation
u1
u2
u3
Displacement (in)
Time (sec)
15
10
5
0
-5
-10
-15
0
2
4
6
8
10
12
14
16
18
20

FIGURE 3

Nonlinear Analysis –Retrofitted System – Seismic Loading:

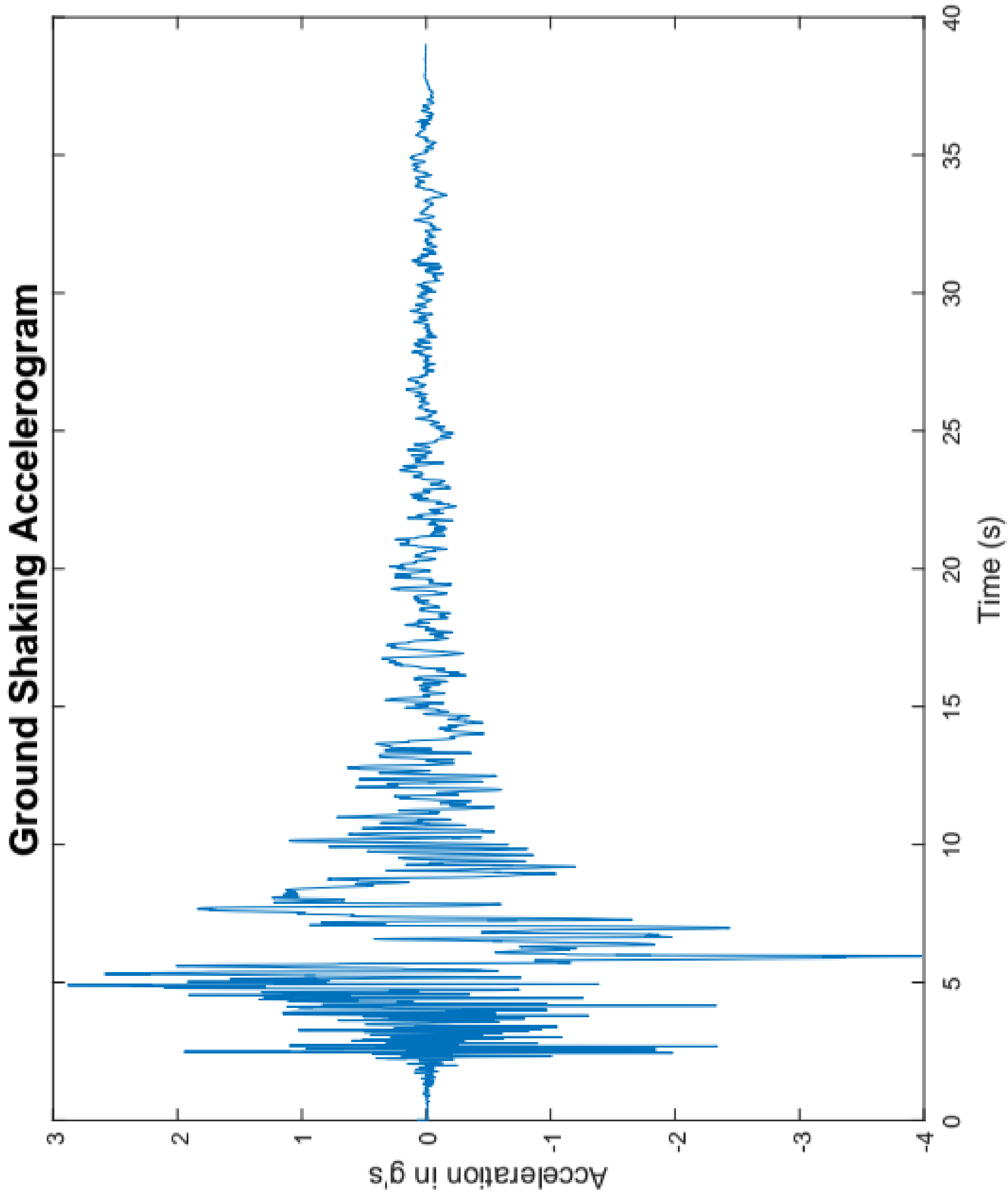


FIGURE 4

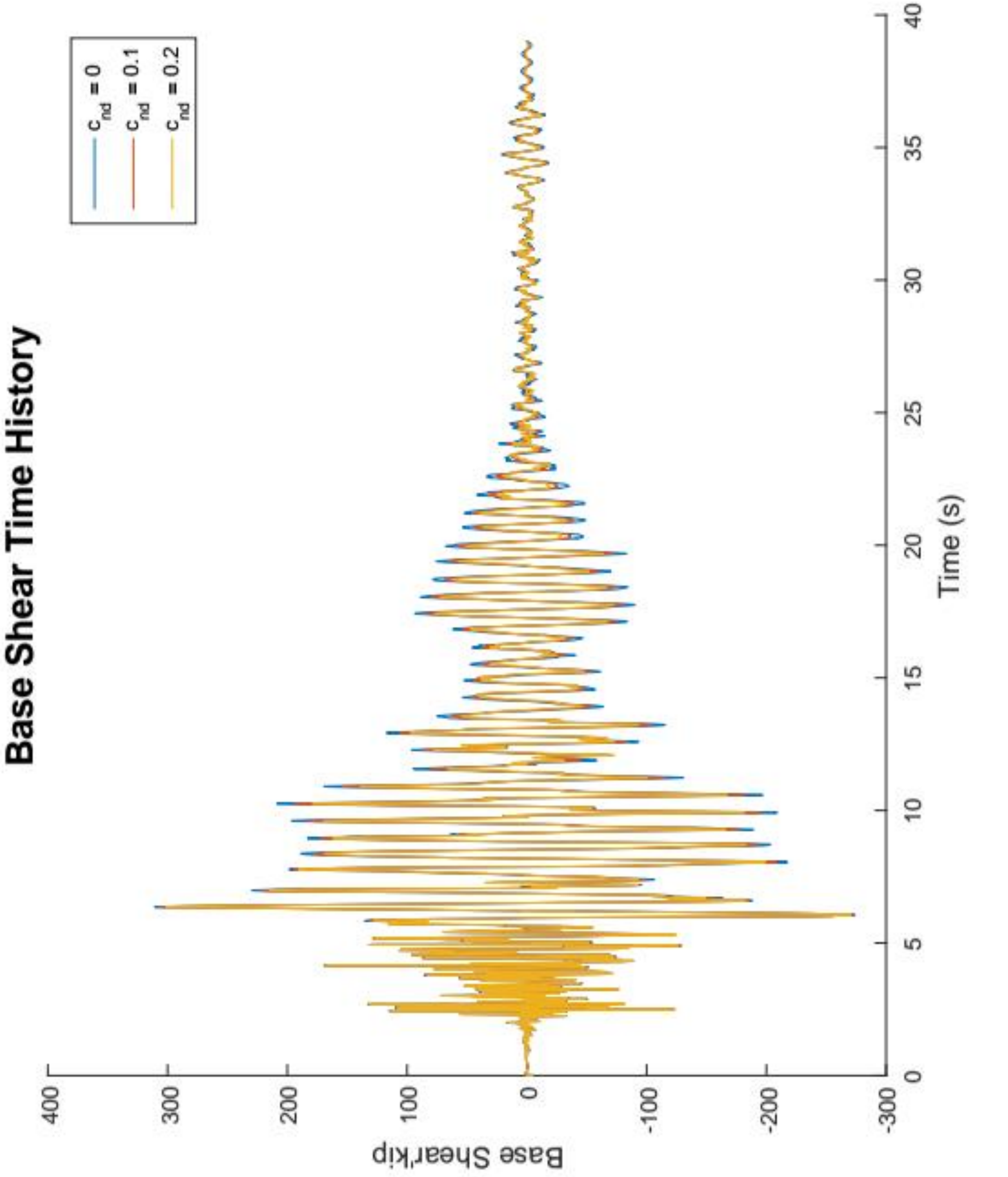

Base Shear Time History
$c_{nd} = 0$
$c_{nd} = 0.1$
$c_{nd} = 0.2$
Base Shear,kip
Time (s)
400
300
200
100
0
-100
-200
-300
0
5
10
15
20
25
30
35
40

FIGURE 5

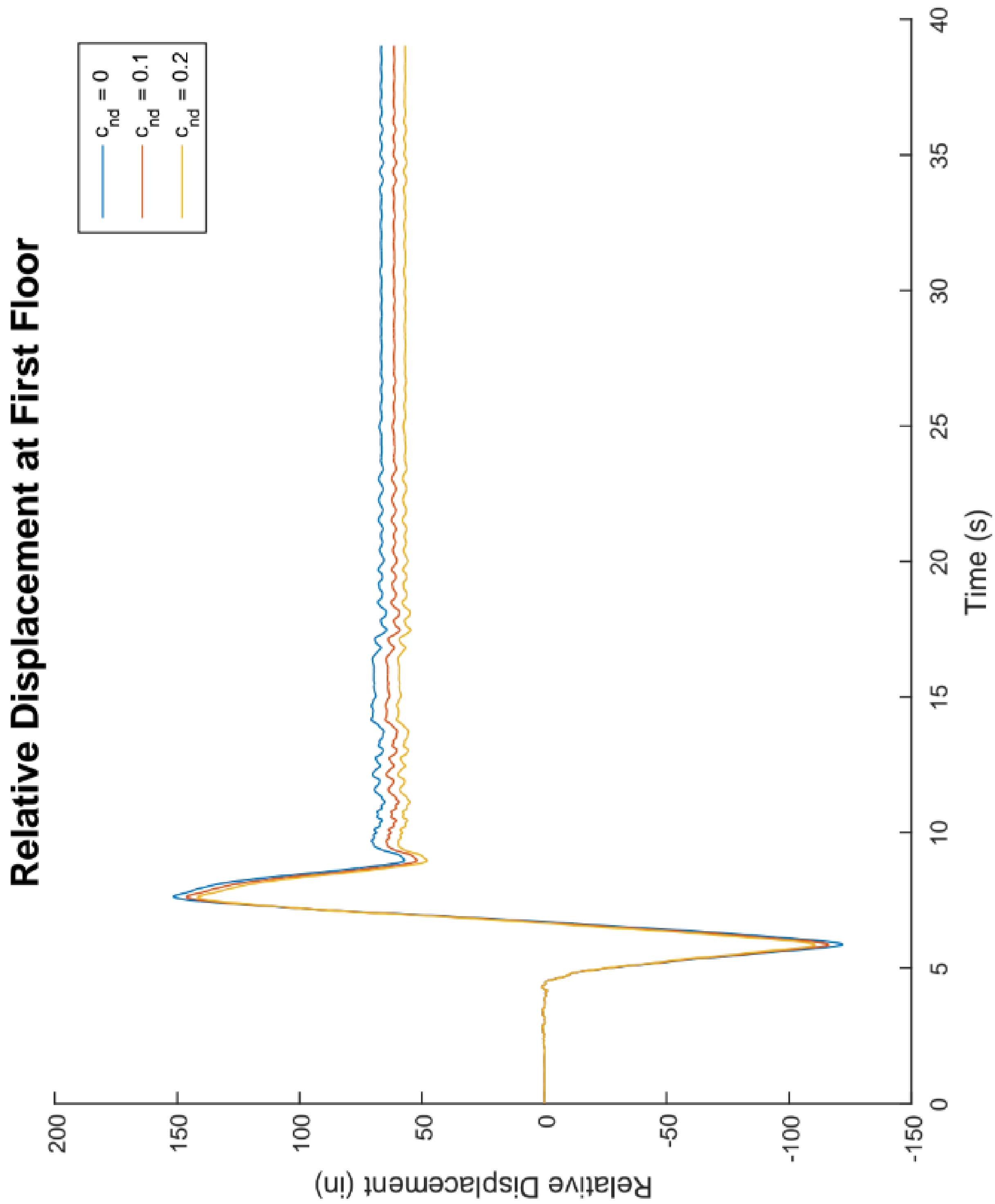

Relative Displacement at First Floor
c_nd = 0
c_nd = 0.1
c_nd = 0.2
Relative Displacement (in)
Time (s)
200
150
100
50
0
-50
-100
-150
0
5
10
15
20
25
30
35
40

FIGURE 6

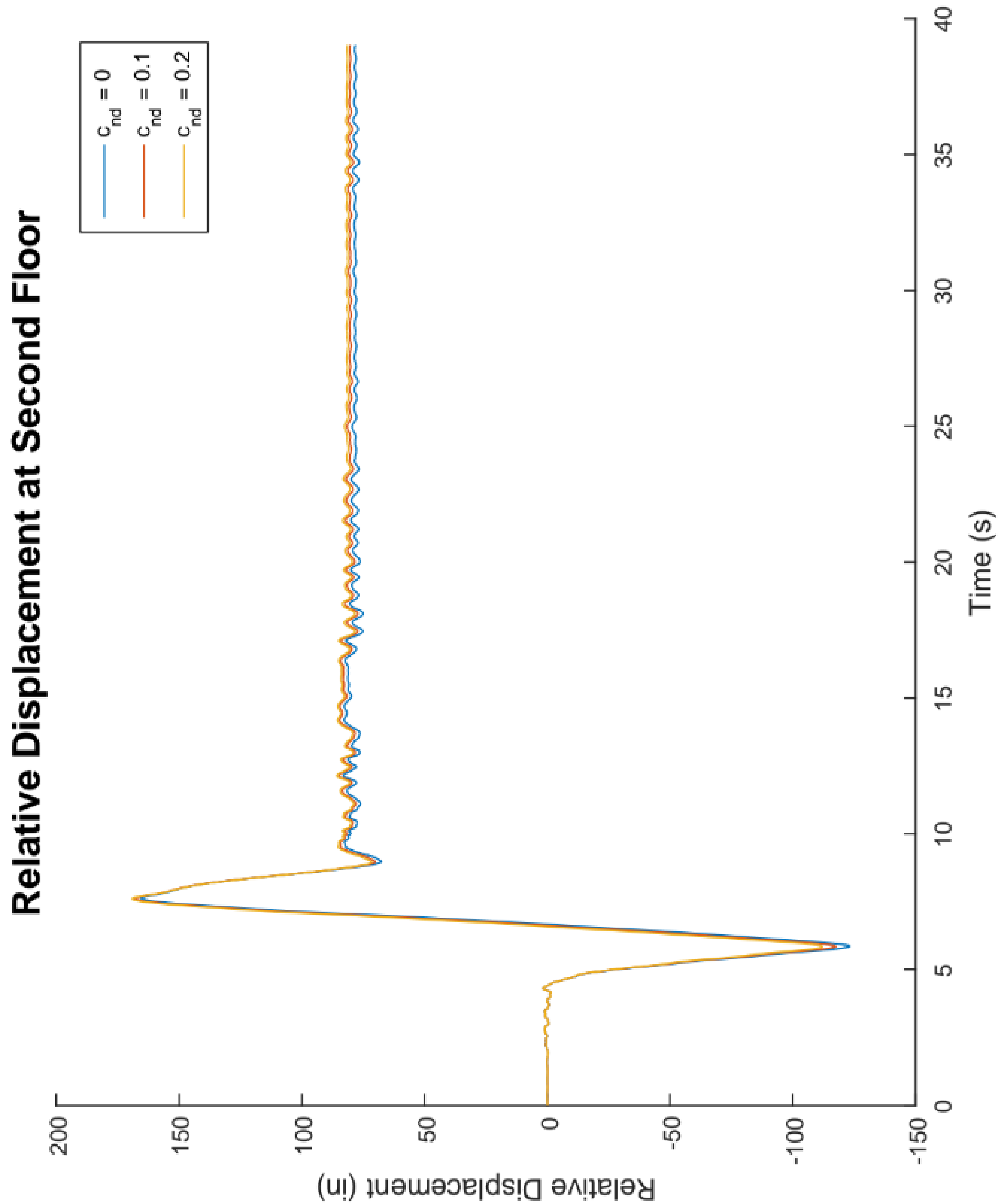

Relative Displacement at Second Floor
$c_{nd} = 0$
$c_{nd} = 0.1$
$c_{nd} = 0.2$
Relative Displacement (in)
Time (s)
200
150
100
50
0
-50
-100
-150
0
5
10
15
20
25
30
35
40

FIGURE 7

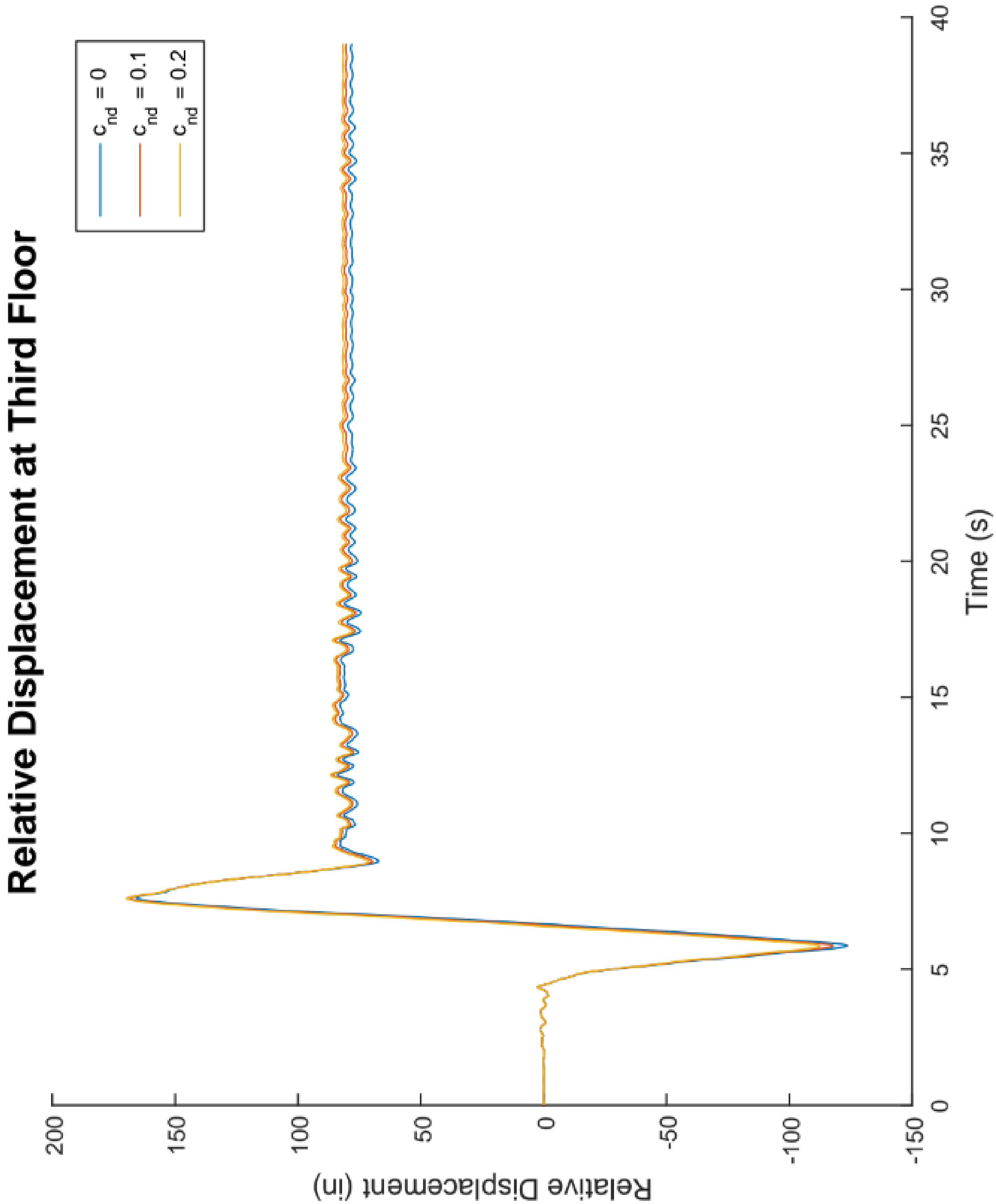

Relative Displacement at Third Floor
Relative Displacement (in)
Time (s)
c_{nd} = 0
c_{nd} = 0.1
c_{nd} = 0.2
200
150
100
50
0
-50
-100
-150
0
5
10
15
20
25
30
35
40

FIGURE 8

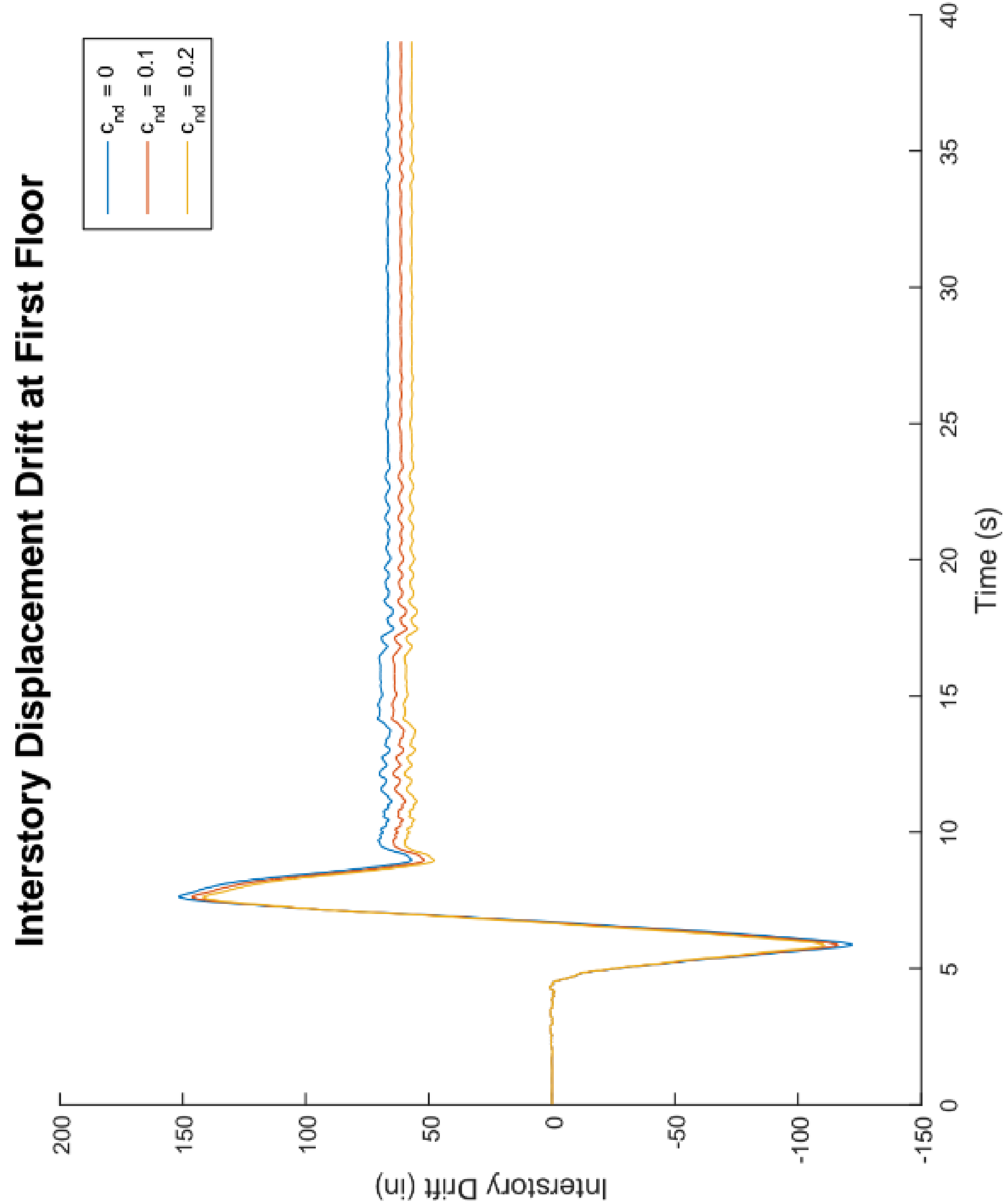

Interstory Displacement Drift at First Floor
c_{nd} = 0
c_{nd} = 0.1
c_{nd} = 0.2
Interstory Drift (in)
200
150
100
50
0
-50
-100
-150
Time (s)
0
5
10
15
20
25
30
35
40

FIGURE 9

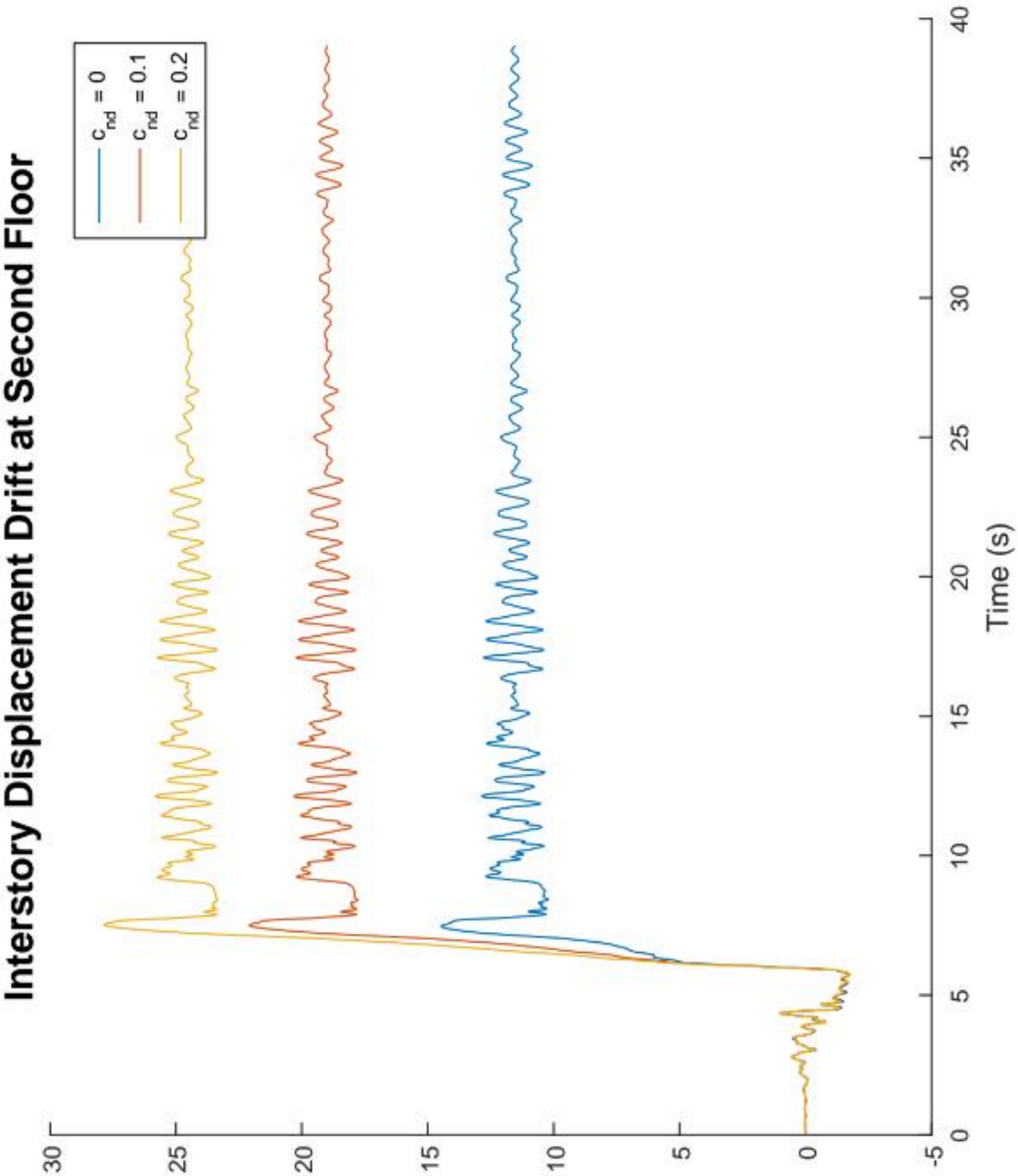

Interstory Displacement Drift at Second Floor
c_{nd} = 0
c_{nd} = 0.1
c_{nd} = 0.2
Interstory Drift (in)
Time (s)
-5
0
5
10
15
20
25
30
35
40

FIGURE 10

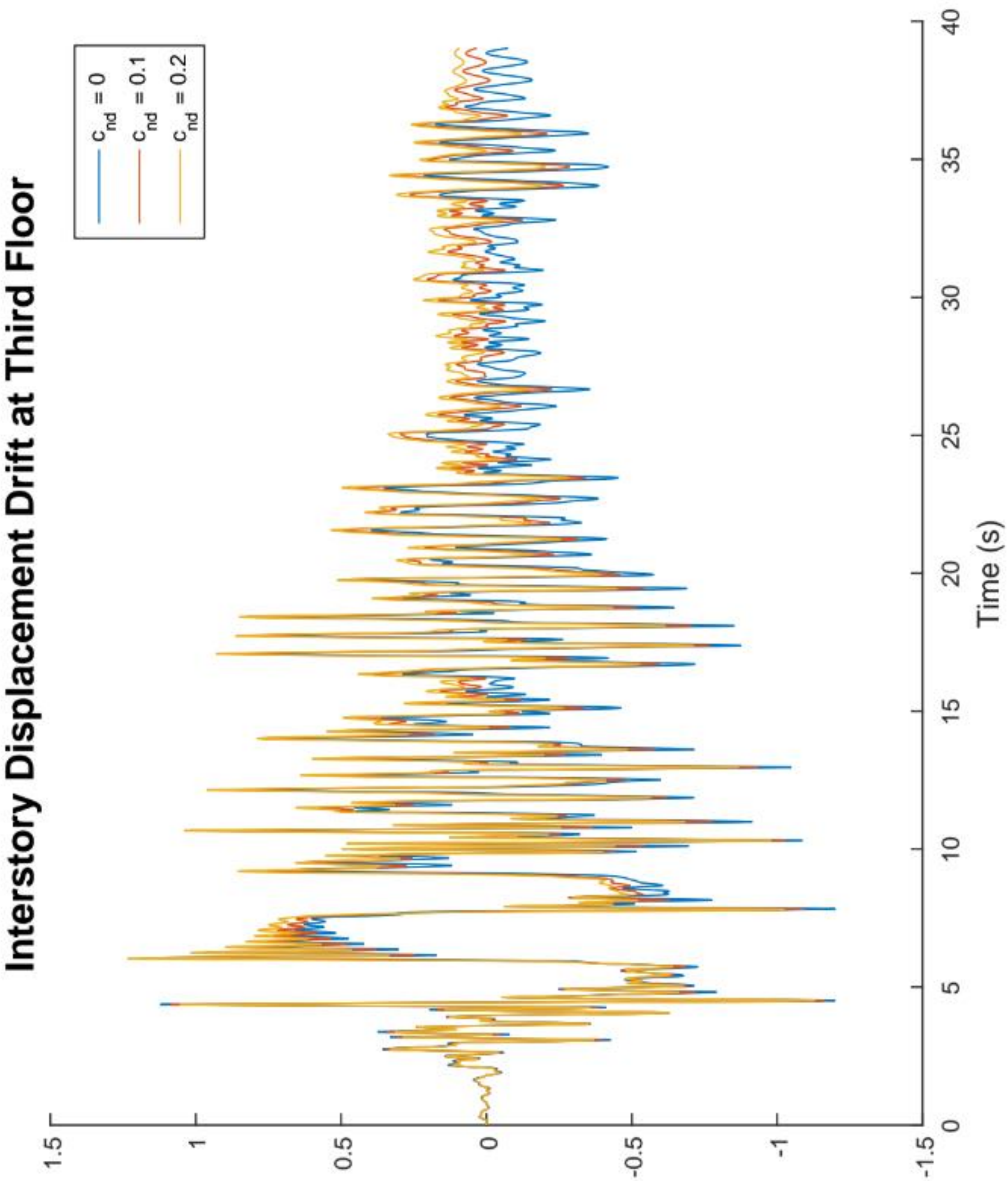

Interstory Displacement Drift at Third Floor
$c_{nd} = 0$
$c_{nd} = 0.1$
$c_{nd} = 0.2$
Interstory Drift (in)
Time (s)
1.5
1
0.5
0
-0.5
-1
-1.5
0
5
10
15
20
25
30
35
40

FIGURE 11

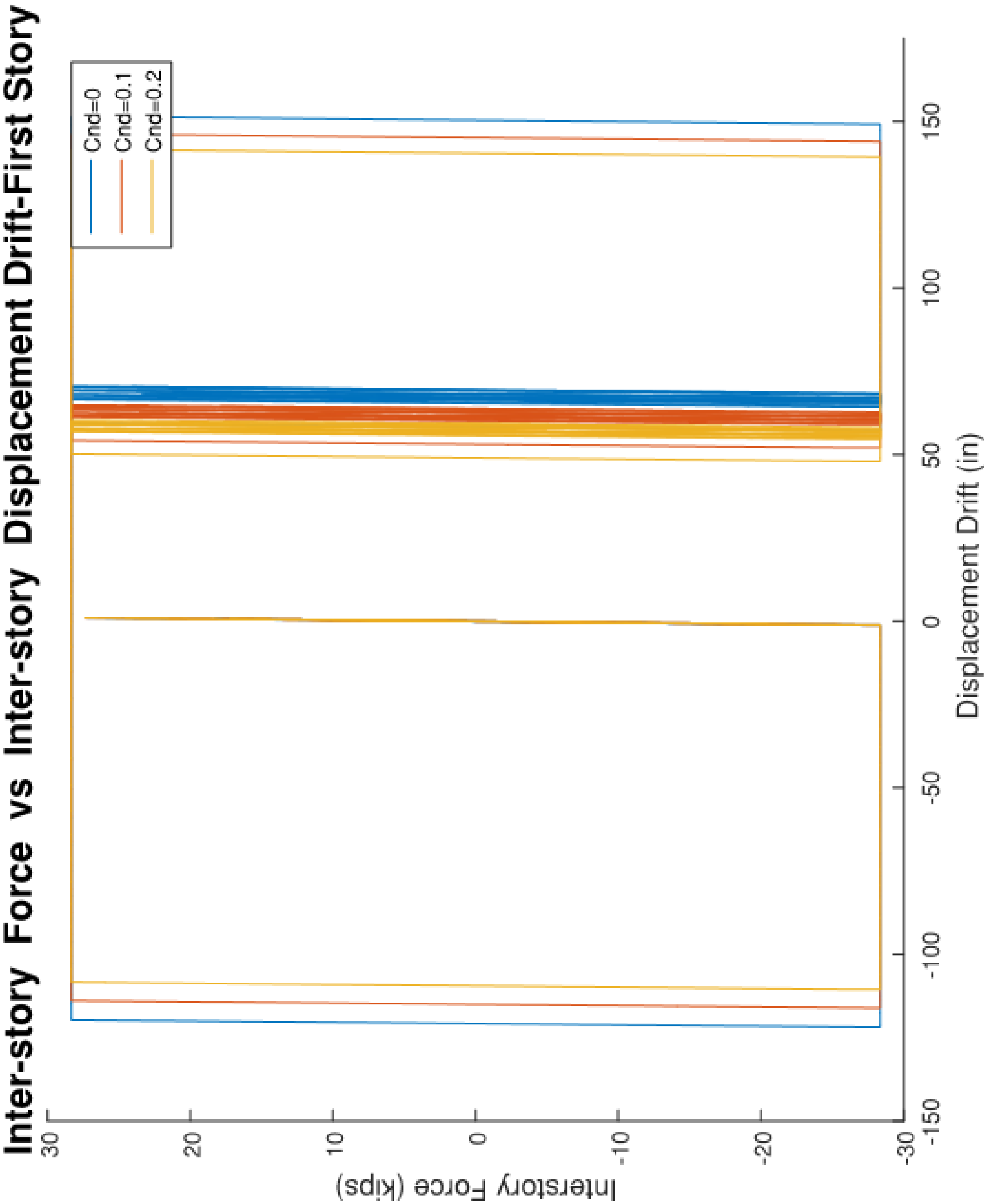

Inter-story Force vs Inter-story Displacement Drift-First Story
Cnd=0
Cnd=0.1
Cnd=0.2
Interstory Force (kips)
Displacement Drift (in)
30
20
10
0
-10
-20
-30
-150
-100
-50
0
50
100
150

FIGURE 12

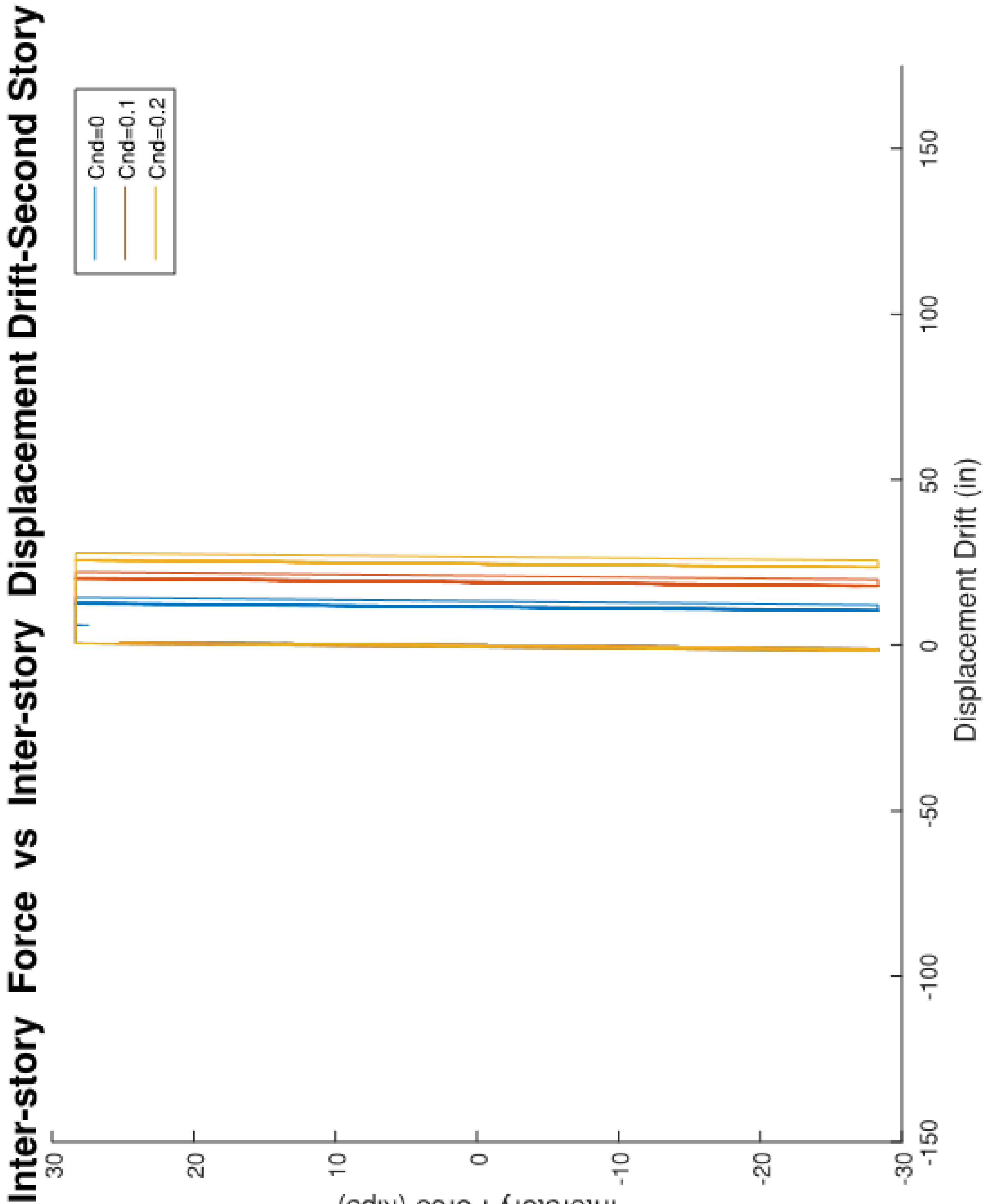

Inter-story Force vs Inter-story Displacement Drift-Second Story
Cnd=0
Cnd=0.1
Cnd=0.2
Interstory Force (kips)
Displacement Drift (in)
30
20
10
0
-10
-20
-30
-150
-100
-50
0
50
100
150

FIGURE 13

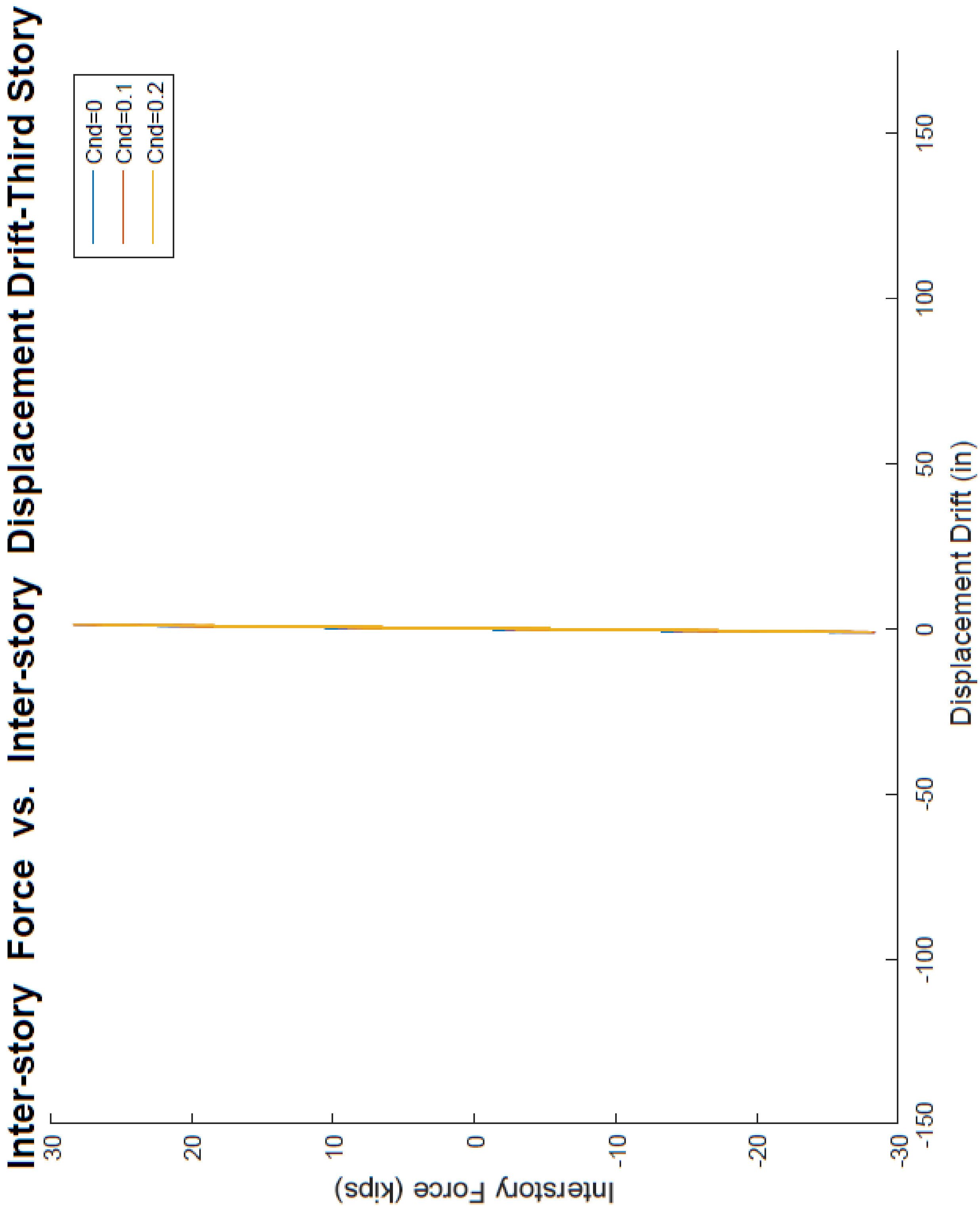

Inter-story Force vs. Inter-story Displacement Drift-Third Story
Cnd=0
Cnd=0.1
Cnd=0.2
Interstory Force (kips)
Displacement Drift (in)
30
20
10
0
-10
-20
-30
-150
-100
-50
0
50
100
150

FIGURE 14

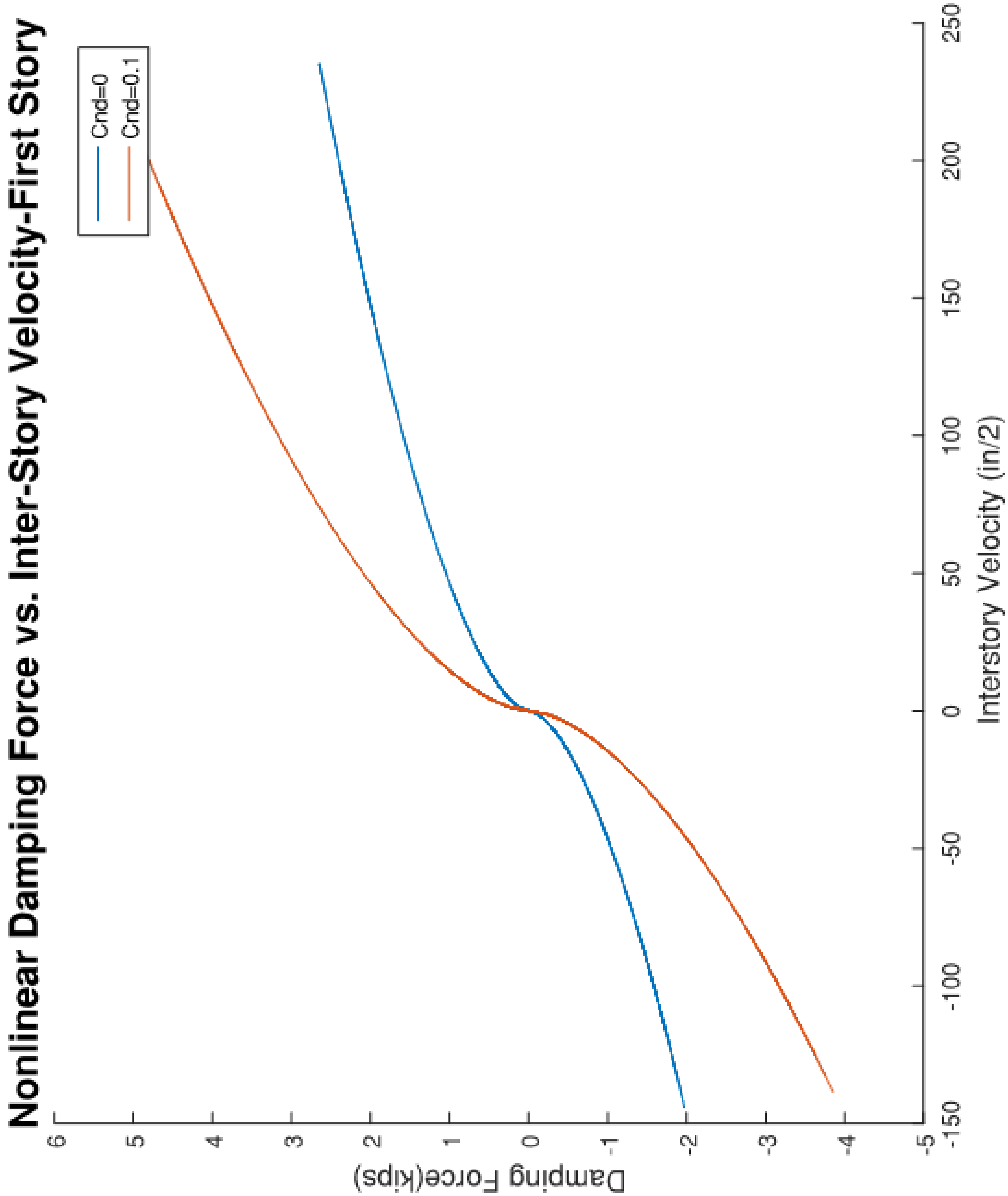

Nonlinear Damping Force vs. Inter-Story Velocity-First Story
Cnd=0
Cnd=0.1
Damping Force(kips)
Interstory Velocity (in/2)
6
5
4
3
2
1
0
-1
-2
-3
-4
-5
-150
-100
-50
0
50
100
150
200
250

FIGURE 15

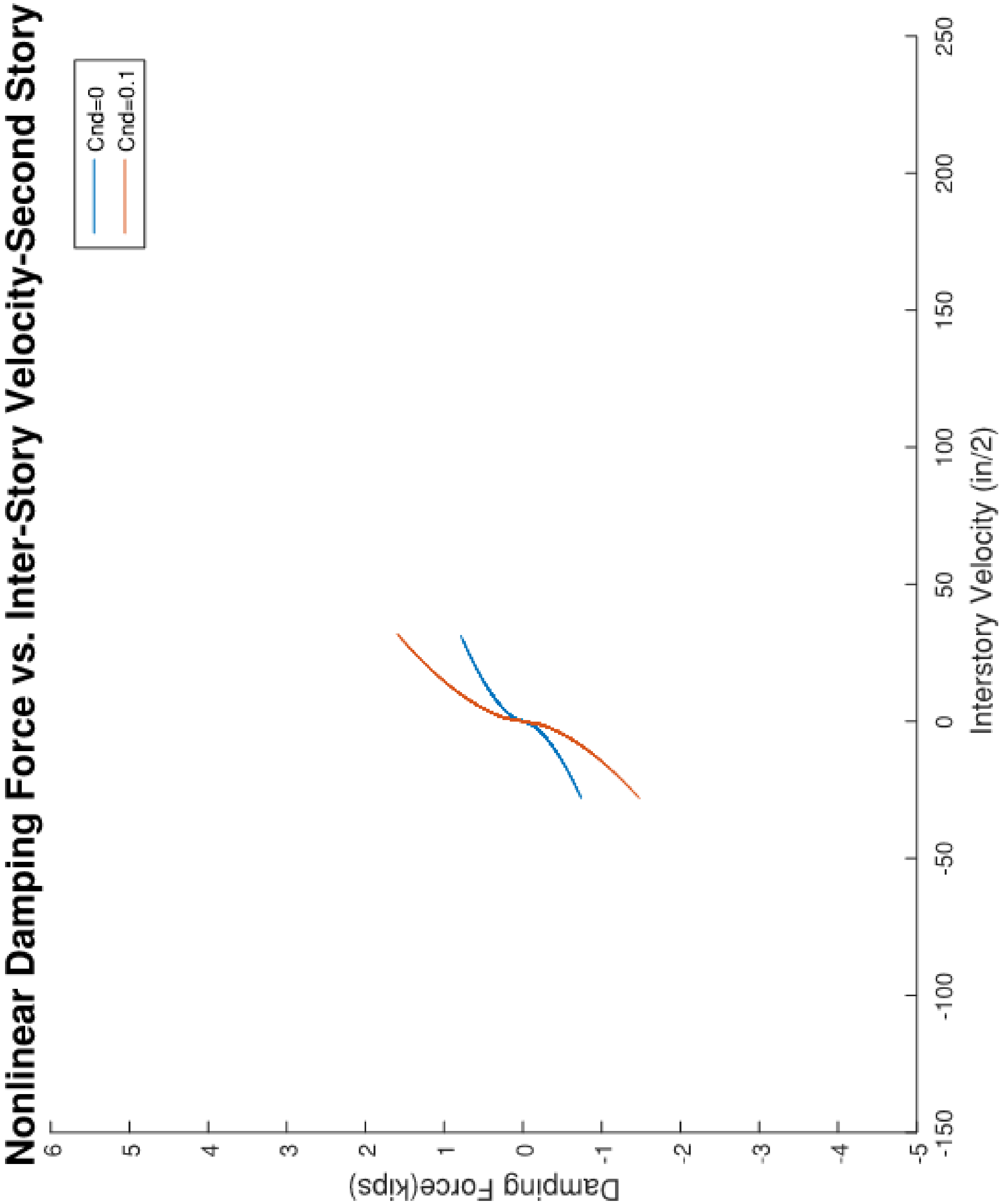

Nonlinear Damping Force vs. Inter-Story Velocity-Second Story
Cnd=0
Cnd=0.1
Damping Force(kips)
Interstory Velocity (in/2)
6
5
4
3
2
1
0
-1
-2
-3
-4
-5
-150
-100
-50
0
50
100
150
200
250

FIGURE 16

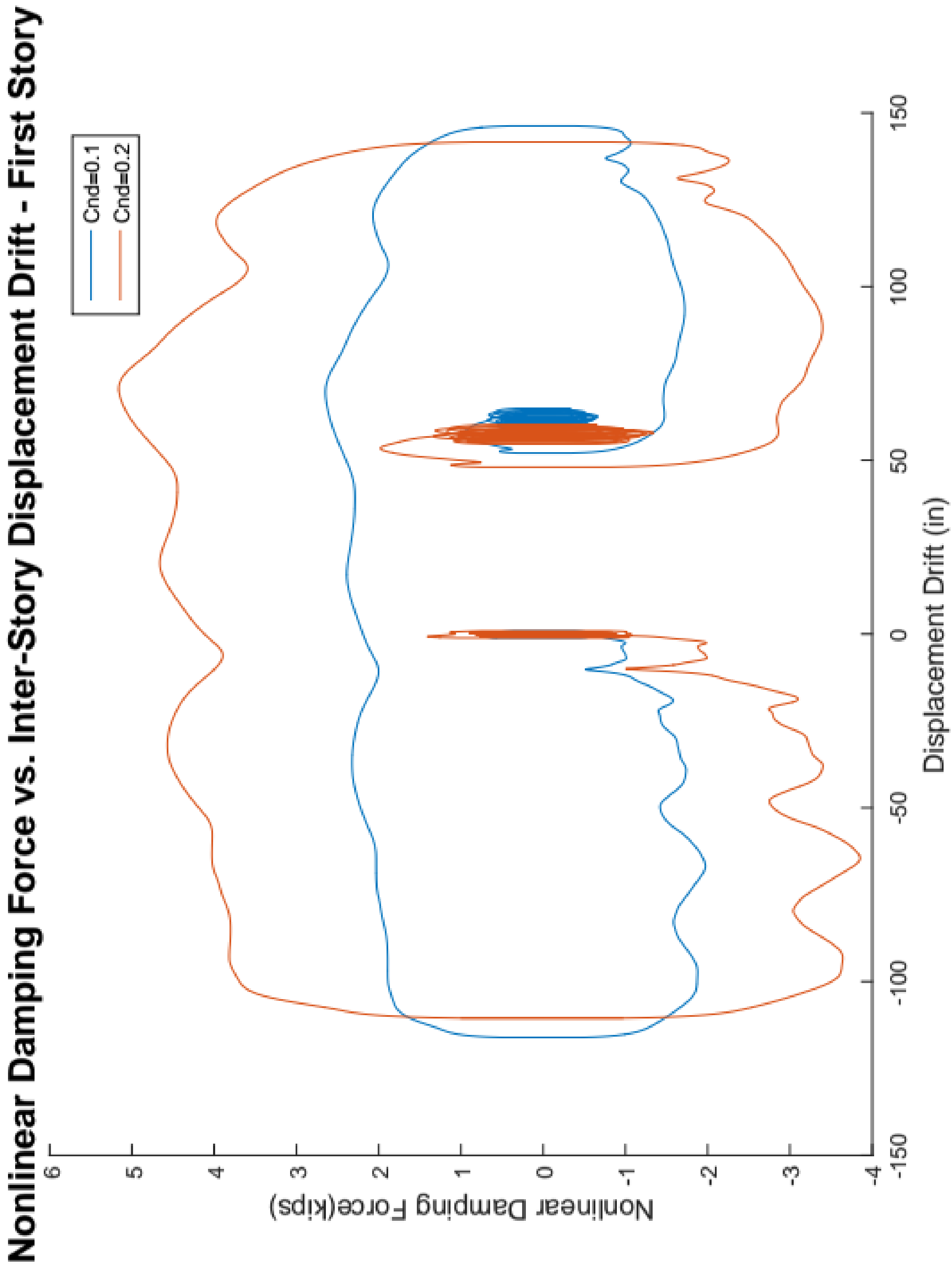
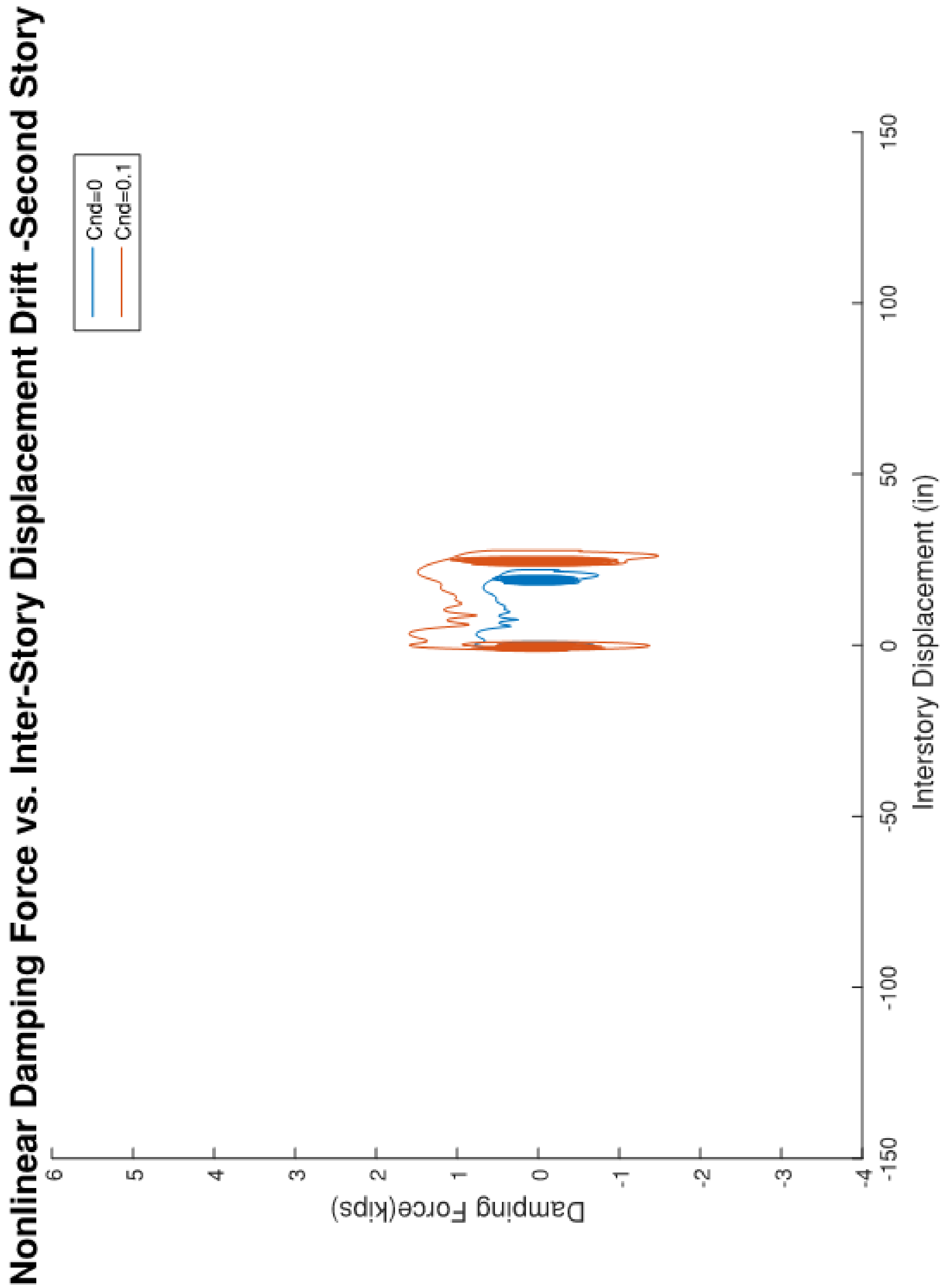

Nonlinear Damping Force vs. Inter-Story Displacement Drift - First Story
Cnd=0.1
Cnd=0.2
Nonlinear Damping Force(kips)
Displacement Drift (in)
6
5
4
3
2
1
0
-1
-2
-3
-4
-150
-100
-50
0
50
100
150

FIGURE 17



Nonlinear Damping Force vs. Inter-Story Displacement Drift -Second Story
Cnd=0
Cnd=0.1
Damping Force(kips)
Interstory Displacement (in)
6
5
4
3
2
1
0
-1
-2
-3
-4
-150
-100
-50
0
50
100
150

FIGURE 18

The damper generally does reduce the calculated displacement response of the structure for the given earthquake, but for the values of cnd given, the reduction was not very significant. Larger damping might result is more substantial reduction in deformation response and damage, but the damping would likely need to be substantially higher. Looking at the forces produced by the damper (less than 6 kips maximum for the first story and less than 2 kips maximum for the second story) and comparing it to the forces generated by the columns (28.33 kips), it is apparent that the damping forces generated are substantially less than the forces produced by the columns.

Conclusion

The analysis of structures under earthquake loads (ground acceleration) is imperative in understanding the response of systems under such conditions. In the simplified case of a single-degree of freedom shear building, the analysis conducted through Matlab programming is able to accurately demonstrate the structure's displacement and velocity time histories along with the forces experienced by the columns and the frame as a whole. Although as previously discussed this is a simplified system, it is still a huge insight into the response and behavior of actual scenarios encountered in field conditions. Overall the Matlab program developed allows the user to represent the response of any type of shear building through manipulation of different parameters, such as mass matrix (m), Rayleigh damping matrix (c, column stiffness (kstr), vector of the ground motion ($\ddot{u}_g$), time step of the ground motion (dtg) and time step of the analysis (dtan).

Appendix

```
function u = modeplots

h=[0 1 2 3]
phi1= [0;0.4450;0.8019;1];
phi2= [0;-1.247; -0.555; 1];
phi3= [0;1.8019;-2.2469;1];

hold on
grid on
plot(phi1,h)
plot(phi2,h)
plot(phi3,h)
ylabel('Height (in)')
xlabel('Amplitude')
legend('First Mode','Second Mode','Third Mode')

function u = ElasticAnaly

t=0:0.001:15;

m= 0.05176*[1 0 0; 0 1 0; 0 0 1];
k=24.93*[2 -1 0; -1 2 -1; 0 -1 1];
c=[0.1786 -0.0823 0;-0.0823 0.1786 -0.0823;0 -0.0823 0.0964];

w1=9.768;
w2=27.367;
w3=39.546;
zita1=0.03;
zita2=0.05;
zita3=0.0687;
w=0.7*w1;

phi1= [0.4450;0.8019;1];
phi2= [-1.247; -0.555; 1];
phi3= [1.8019;-2.2469;1];

po=-0.05176*386.4;
p=[po;po;po];

pt= p*sin(w*t);

M1=phi1'*m*phi1;
K1=phi1'*k*phi1;
C1=phi1'*c*phi1;

M2=phi2'*m*phi2;
K2=phi2'*k*phi2;
C2=phi2'*c*phi2;

M3=phi3'*m*phi3;
K3=phi3'*k*phi3;
C3=phi3'*c*phi3;
```

```
P1=phi1'*p;
P2=phi2'*p;
P3=phi3'*p;

Pt1=phi1'*pt;
Pt2=phi2'*pt;
Pt3=phi3'*pt;

Q1=(-9.659)*sin(6.8376*t-0.0822);
Q2=(0.1541)*sin(6.8376*t-0.0266);
Q3=(-0.0152)*sin(6.8376*t-0.0245);

u1=phi1(1)*Q1+phi2(1)*Q2+phi3(1)*Q3;
u2=phi1(2)*Q1+phi2(2)*Q2+phi3(2)*Q3;
u3=phi1(3)*Q1+phi2(3)*Q2+phi3(3)*Q3;

hold on
plot(t,u1)
plot(t,u2)
plot(t,u3)
 xlabel('Time (sec)')
 ylabel('Displacement (in)')
 title('Response to Harmonic Excitation-Analytical Response')
legend('u1','u2','u3')
clear; close all; clc;
% load earthquake data: ug_dd, earthquake shaking data in g's and dtg,
% interval between earthquake data points
% Three-story frame, each rigid slab of same weight, 20kip
ndof=3;     % Number of degrees of freedom
mstr=20/12/32.2;%kip.s^2/in         Building damping matrix
mmat=eye(3);
h=120;%in       % interstory height
% Given damping ratio, xi, for first and second modes:
xi=zeros(3); xi(1,1)=.03; xi(2,2)=.05;   % Third xi TBD assuming Raleigh
Damping
% A992 wide flange column
fy=50;%ksi
E=29000;%ksi
% (2) W8x18 columns per story
Ixx=61.9;%in4
Zxx=17.0;%in3
% Max plastic moment available
Mmax=fy*Zxx;%kip.in
fystr=1e1664*Mmax/h;%kip         For fix-fix columns boundary conditions
kstr=2*(12*E*Ixx/h^3);
k=[ 2  -1  0
   -1   2 -1
    0  -1  1];
[Phi, Lamda]=eig(k,mmat);
% Scale Phi vector such that last row = 1
for ii=1:length(Phi); Phi(:,ii)=Phi(:,ii)*1/(Phi(length(Phi),ii)); end
% Natural modal frequencies
wn=sqrt(Lamda*kstr/mstr);
% Define Rayleigh damping constants, a0 and a1
a=rref([1   wn(1,1)^2   2*xi(1,1)*wn(1,1);
```

```
        1   wn(2,2)^2   2*xi(2,2)*wn(2,2)]); a0=a(1,3); a1=a(2,3); clear a;
cmat=a0*mstr*mmat+a1*kstr*k;
% define xi as third mode
xi(3,3)=(a0+a1*wn(3,3)^2)/(2*wn(3,3));
mmat=mstr*mmat;   %NOTE: need to re-define m before running function.

 %7. Steady State Response (no non-linear damper) using function
 dtg=.005;
 dtan=dtg/1;
 cnd=0;
 tg=[0:dtg:20];
 ug_dd=sin(.7*wn(1,1)*tg);
 [u,ud,udd,fstr,fnd,tan] =
Sola_Enssle_solver(mmat,cmat,cnd,kstr,fystr,dtg,ug_dd,dtan);
 ff(1)=figure(1);
 plot(tan,u);
 xlabel('Time (sec)')
 ylabel('Displacement (in)')
 title('Response to Harmonic Excitation')
 legend('u1','u2','u3');
 print(figure(1),'7_ Numerical Solution Displacement Response','-dpdf');

%% 8. El Centro Response Analysis (Nonlinear damping)
load('acceleration.mat')   % Loads ug_dd (El Centro shaking acceleration in
g's)
                           % and dtg = .005
dtan=dtg/5;
cnd=0;
tg=[0:1:length(ug_dd)-1]*dtg;
% Plot accelerogram in g's
% plot(tg,ug_dd);
% fft=title('Ground Shaking Accelerogram'); fft.FontSize=16;
% ffxl=xlabel('Time (s)'); ffyl=ylabel(['Acceleration in g' char(39) 's']);
ffxl.FontSize=12; ffyl.FontSize=12;
% % legend('u1','u2','u3');
% print(figure(1),'8_ Accelerogram','-dpdf');

% % Plot time history of base shear
% for cnd=0:.1:.2
% [u,ud,udd,fstr,fnd,tan] =
Sola_Enssle_solver(mmat,cmat,cnd,kstr,fystr,dtg,ug_dd,dtan);
% a=mmat*udd; FV_base=a(1,:)+a(2,:)+a(3,:); clear a;
% ff(2)=figure(2);
% hold on
% plot(tan,FV_base);
% fft=title('Base Shear Time History'); fft.FontSize=16;
% ffxl=xlabel('Time (s)'); ffyl=ylabel(['Base Shear (kip)']);
ffxl.FontSize=12; ffyl.FontSize=12;
% end
% legend('c_{nd} = 0', 'c_{nd} = 0.1', 'c_{nd} = 0.2')
% print(figure(2),'8_ Base Shear','-dpdf');

% Plot time history of relative displacement at each floor
cnd_=0:.1:.2;
asdf={'r' 'g' 'b'};
for i=1:length(cnd_)
```

```
cnd=cnd_(i);
[u,ud,udd,fstr,fnd,tan] =
Sola_Enssle_solver(mmat,cmat,cnd,kstr,fystr,dtg,ug_dd,dtan);
u_cnd(:,:,i)=u;
end
% (plot figures)
ffii={'First' 'Second' 'Third'};
for i=1:length(cnd_)
ff(i)=figure(i);
hold on
plot(tan,[u_cnd(i,:,1);u_cnd(i,:,2);u_cnd(i,:,3)]);
fft=title(['Relative Displacement at ' char(ffii(i)) ' Floor']);
fft.FontSize=16;
ffxl=xlabel('Time (s)'); ffyl=ylabel(['Relative Displacement (in)']);
ffxl.FontSize=12; ffyl.FontSize=12;
legend('c_{nd} = 0', 'c_{nd} = 0.1', 'c_{nd} = 0.2')
print(figure(i),['8_ Relative Displacement' char(ffii(i)) 'Floor'],'-dpdf');
end
%% Plot interstory displacement drifts
for i=1:length(cnd_)
ff(i+3)=figure(i+3);
hold on
if i==1
plot(tan,[u_cnd(i,:,1);u_cnd(i,:,2);u_cnd(i,:,3)]);
elseif i>=2
plot(tan,[u_cnd(i,:,1)-u_cnd(i-1,:,1);u_cnd(i,:,2)-u_cnd(i-
1,:,2);u_cnd(i,:,3)-u_cnd(i-1,:,3)]);
end
fft=title(['Interstory Displacement Drift at ' char(ffii(i)) ' Floor']);
fft.FontSize=16;
ffxl=xlabel('Time (s)'); ffyl=ylabel(['Interstory Drift (in)']);
ffxl.FontSize=12; ffyl.FontSize=12;
legend('c_{nd} = 0', 'c_{nd} = 0.1', 'c_{nd} = 0.2')
print(figure(i+3),['8_ Interstory Drift' char(ffii(i)) 'Floor'],'-dpdf');
end
%%

%% 8. Inter-story  Force  vs.  Inter-story  Displacement Drift-First Story
load('acceleration.mat')   % Loads ug_dd (El Centro shaking acceleration in
g's)
                           % and dtg = .005
dtan=dtg/5;
cnd=0;
cndprime=0.1;
cndalpha=0.2;
tg=[0:1:length(ug_dd)-1]*dtg;
% Plot accelerogram in g's
[u,ud,udd,fstr,fnd,tan] =
Sola_Enssle_solver(mmat,cmat,cnd,kstr,fystr,dtg,ug_dd,dtan);
[uprime,udprime,uddprime,fstrprime,fndprime,tan] =
Sola_Enssle_solverprime(mmat,cmat,cndprime,kstr,fystr,dtg,ug_dd,dtan);
[ualpha,udalpha,uddalpha,fstralpha,fndalpha,tan] =
Sola_Enssle_solveralpha(mmat,cmat,cndalpha,kstr,fystr,dtg,ug_dd,dtan);
figure(1)
hold on
plot(u(1,:),fstr(1,:));
plot(uprime(1,:),fstrprime(1,:));
```

```
plot(ualpha(1,:),fstralpha(1,:));
fft=title('Inter-story  Force  vs.  Inter-story  Displacement Drift-First
Story'); fft.FontSize=16;
ffxl=xlabel(' Displacement Drift (in)'); ffyl=ylabel(['Interstory Force
(kips)']); ffxl.FontSize=12; ffyl.FontSize=12;
xlim([-150 175])
ylim([-30 30])

legend('Cnd=0','Cnd=0.1','Cnd=0.2');
print(figure(1),'Force  vs. Displacement 3','-dpdf');

%% 8. Inter-story  Force  vs.  Inter-story  Displacement Drift-Second Story
load('acceleration.mat')   % Loads ug_dd (El Centro shaking acceleration in
g's)
                           % and dtg = .005
dtan=dtg/5;
cnd=0;
cndprime=0.1;
cndalpha=0.2;
tg=[0:1:length(ug_dd)-1]*dtg;
% Plot accelerogram in g's
[u,ud,udd,fstr,fnd,tan] =
Sola_Enssle_solver(mmat,cmat,cnd,kstr,fystr,dtg,ug_dd,dtan);
[uprime,udprime,uddprime,fstrprime,fndprime,tan] =
Sola_Enssle_solverprime(mmat,cmat,cndprime,kstr,fystr,dtg,ug_dd,dtan);
[ualpha,udalpha,uddalpha,fstralpha,fndalpha,tan] =
Sola_Enssle_solveralpha(mmat,cmat,cndalpha,kstr,fystr,dtg,ug_dd,dtan);
figure(1)
hold on
plot(u(2,:)-u(1,:),fstr(2,:));
plot(uprime(2,:)-uprime(1,:),fstrprime(2,:));
plot(ualpha(2,:)-ualpha(1,:),fstralpha(2,:));
fft=title('Inter-story  Force  vs.  Inter-story  Displacement Drift-Second
Story'); fft.FontSize=16;
ffxl=xlabel(' Displacement Drift (in)'); ffyl=ylabel(['Interstory Force
(kips)']); ffxl.FontSize=12; ffyl.FontSize=12;
xlim([-150 175])
ylim([-30 30])
legend('Cnd=0','Cnd=0.1','Cnd=0.2');
print(figure(1),'Force  vs. Displacement 3','-dpdf');


%% 8. Inter-story  Force  vs.  Inter-story  Displacement Drift-Third Story
load('acceleration.mat')   % Loads ug_dd (El Centro shaking acceleration in
g's)
                           % and dtg = .005
dtan=dtg/5;
cnd=0;
cndprime=0.1;
cndalpha=0.2;
tg=[0:1:length(ug_dd)-1]*dtg;
% Plot accelerogram in g's
[u,ud,udd,fstr,fnd,tan] =
Sola_Enssle_solver(mmat,cmat,cnd,kstr,fystr,dtg,ug_dd,dtan);
[uprime,udprime,uddprime,fstrprime,fndprime,tan] =
Sola_Enssle_solverprime(mmat,cmat,cndprime,kstr,fystr,dtg,ug_dd,dtan);
```

```
[ualpha,udalpha,uddalpha,fstralpha,fndalpha,tan] =
Sola_Enssle_solveralpha(mmat,cmat,cndalpha,kstr,fystr,dtg,ug_dd,dtan);
figure(1)
hold on
plot(u(3,:)-u(2,:),fstr(3,:));
plot(uprime(3,:)-uprime(2,:),fstrprime(3,:));
plot(ualpha(3,:)-ualpha(2,:),fstralpha(3,:));
fft=title('Inter-story  Force  vs.  Inter-story  Displacement Drift-Third
Story'); fft.FontSize=16;
ffxl=xlabel(' Displacement Drift (in)'); ffyl=ylabel(['Interstory Force
(kips)']); ffxl.FontSize=12; ffyl.FontSize=12;
xlim([-150 175])
ylim([-30 30])
legend('Cnd=0','Cnd=0.1','Cnd=0.2');
print(figure(1),'Force  vs. Displacement 3','-dpdf');


%% 8. Nonlinear Damping Force vs. Inter-Story Velocity-First Story'
load('acceleration.mat')   % Loads ug_dd (El Centro shaking acceleration in
g's)
                           % and dtg = .005
dtan=dtg/5;
cnd=0.1;
cndprime=0.2;

tg=[0:1:length(ug_dd)-1]*dtg;
% Plot accelerogram in g's
[u,ud,udd,fstr,fnd,tan] =
Sola_Enssle_solver(mmat,cmat,cnd,kstr,fystr,dtg,ug_dd,dtan);
[uprime,udprime,uddprime,fstrprime,fndprime,tan] =
Sola_Enssle_solverprime(mmat,cmat,cndprime,kstr,fystr,dtg,ug_dd,dtan);
figure(1)
hold on
plot(ud(1,:),fnd(1,:));
plot(udprime(1,:),fndprime(1,:));
fft=title('Nonlinear Damping Force vs. Inter-Story Velocity-First Story');
fft.FontSize=16;
ffxl=xlabel(' Interstory Velocity (in/2)'); ffyl=ylabel(['Damping
Force(kips)']); ffxl.FontSize=12; ffyl.FontSize=12;
xlim([-150 250])
ylim([-5 6])
legend('Cnd=0','Cnd=0.1');

%% 8. Nonlinear Damping Force vs. Inter-Story Velocity-Second Story'
load('acceleration.mat')   % Loads ug_dd (El Centro shaking acceleration in
g's)
                           % and dtg = .005
dtan=dtg/5;
cnd=0.1;
cndprime=0.2;

tg=[0:1:length(ug_dd)-1]*dtg;
% Plot accelerogram in g's
[u,ud,udd,fstr,fnd,tan] =
Sola_Enssle_solver(mmat,cmat,cnd,kstr,fystr,dtg,ug_dd,dtan);
```

```
[uprime,udprime,uddprime,fstrprime,fndprime,tan] =
Sola_Enssle_solverprime(mmat,cmat,cndprime,kstr,fystr,dtg,ug_dd,dtan);
figure(1)
hold on
plot(ud(2,:)-ud(1,:),fnd(2,:));
plot(udprimte(2,:)-udprime(1,:),fndprime(2,:));
fft=title('Nonlinear Damping Force vs. Inter-Story Velocity-Second Story');
fft.FontSize=16;
ffxl=xlabel(' Interstory Velocity (in/2)'); ffyl=ylabel(['Damping
Force(kips)']); ffxl.FontSize=12; ffyl.FontSize=12;
xlim([-150 250])
ylim([-5 6])
legend('Cnd=0','Cnd=0.1');

%% 8. Nonlinear Damping Force vs. Inter-Story Displacement-First Story'
load('acceleration.mat')   % Loads ug_dd (El Centro shaking acceleration in
g's)
                           % and dtg = .005
dtan=dtg/5;
cnd=0.1;
cndprime=0.2;

tg=[0:1:length(ug_dd)-1]*dtg;
% Plot accelerogram in g's
[u,ud,udd,fstr,fnd,tan] =
Sola_Enssle_solver(mmat,cmat,cnd,kstr,fystr,dtg,ug_dd,dtan);
[uprime,udprime,uddprime,fstrprime,fndprime,tan] =
Sola_Enssle_solverprime(mmat,cmat,cndprime,kstr,fystr,dtg,ug_dd,dtan);
figure(1)
hold on
plot(u(1,:),fnd(1,:));
plot(uprime(1,:),fndprime(1,:));
fft=title('Nonlinear Damping Force vs. Inter-Story Displacement Drift -First
Story'); fft.FontSize=16;
ffxl=xlabel(' Interstory Displacement (in)'); ffyl=ylabel(['Damping
Force(kips)']); ffxl.FontSize=12; ffyl.FontSize=12;
xlim([-150 150])
ylim([-4 6])
legend('Cnd=0','Cnd=0.1');

%% 8. Nonlinear Damping Force vs. Inter-Story Displacement-Second Story'
load('acceleration.mat')   % Loads ug_dd (El Centro shaking acceleration in
g's)
                           % and dtg = .005
dtan=dtg/5;
cnd=0.1;
cndprime=0.2;

tg=[0:1:length(ug_dd)-1]*dtg;
% Plot accelerogram in g's
[u,ud,udd,fstr,fnd,tan] =
Sola_Enssle_solver(mmat,cmat,cnd,kstr,fystr,dtg,ug_dd,dtan);
[uprime,udprime,uddprime,fstrprime,fndprime,tan] =
Sola_Enssle_solverprime(mmat,cmat,cndprime,kstr,fystr,dtg,ug_dd,dtan);
figure(1)
hold on
```

```
plot(u(2,:)-u(1,:),fnd(2,:));
plot(uprimte(2,:)-uprime(1,:),fndprime(2,:));
fft=title('Nonlinear Damping Force vs. Inter-Story Displacement Drift -Second
Story'); fft.FontSize=16;
ffxl=xlabel(' Interstory Displacement (in)'); ffyl=ylabel(['Damping
Force(kips)']); ffxl.FontSize=12; ffyl.FontSize=12;
xlim([-150 150])
ylim([-4 6])
legend('Cnd=0','Cnd=0.1');
```

Enssle_Sola_solver function

```
function [u,ud,udd,fstr,fnd,tan] =
Sola_Enssle_solver(mmat,cmat,cnd,kstr,fystr,dtg,ug_dd,dtan)

% Output Variables:
% u,ud,udd are the output vectors of the computed displacement,
    % velocity and acceleration with size of 3×Nt, where Nt
    % is the number of discrete time instants
% f is the output vector of the computed interstory shear from both
    % columns together. Its size is 3×Nt, where Nt is the number
    % of discrete time instants
% Input Variables:
% m, c are the 3×3 mass and damping matrices
% cnd is the constant of the nonlinear dampers (same for both dampers)
% kstr is the elastic interstory stiffness from both columns together
    % (same for all stories)
% fystr is the interstory yield force (same for all stories)
% dtg is the (scalar) time step of the ground motion,
% ugdd is the ground acceleration time history in units of g,
% dtan is the (scalar) time step of the analysis.
err=.000001;                                   % tolerance in norm of R_z
vector during NR iterations
jmax=100;                                      % maximum number of NR
iterations
ndof=length(mmat);
g=32.2*12;%in/s^2
ug_dd=g*ug_dd;%in/s^2                          % re-define ground
acceleration in in/s^2
tg=[0:1:length(ug_dd)-1]*dtg;                  % tg=time vector corresponding
to ug_dd (s)
tan=0:dtan:tg(end);                            % tan=vector of time for
analysis (s)
ugddan=interp1(tg,ug_dd,tan);                  % ug_ddan = vector of
interpolated ground
                                               % accelerations at analysis
time steps
% Initialize marices
u=zeros(ndof,length(tan));
ud=u;
udd=u; udd(:,1)=-ugddan(1);
fstr=u;
Fs=u;
Fd=u;
fnd=u;
Fnd=u;
ktan=zeros(ndof,1);
for i = 1:length(tan)-1
    u_pre=u(:,i);      % u_pre = displacement vector at i (ndofx1)
    ud_pre=ud(:,i);    % ud_pre = velocity vector at i     (ndofx1)
    u_cur=u(:,i);      % u_cur is the ndofx1 vector to become  u(:,i+1) after
convergence
    ud_cur=ud(:,i);    % ud_cur is the ndofx1 vector to become ud(:,i+1) after
convergence
    dud_du=2/dtan;
    for j=1:jmax
```

```
%%% Internal column shear force f (ndof x 1)
fstr(:,i+1)=fstr(:,i)+kstr*[-1  0  0  1  0  0
                             1 -1  0 -1  1  0
                             0  1 -1  0 -1  1]*[u_pre;u_cur];
for ii=1:ndof
    if abs(fstr(ii,i+1))>fystr
        ktan(ii)=0;
        fstr(ii,i+1)=sign(fstr(ii,i+1))*fystr;
    else
        ktan(ii)=kstr;
    end
end

ktan_=[ktan(1)+ktan(2)    -ktan(2)          0
       -ktan(2)         ktan(2)+ktan(3)   -ktan(3)
       0                   -ktan(3)       ktan(3)];

%%% Slab Shear Force Fs (ndof x 1)
Fs(:,i+1)=[fstr(1,i+1)-fstr(2,i+1);
           fstr(2,i+1)-fstr(3,i+1);
           fstr(3,i+1)];  %Fs(:,i+1)

%%% Rayleigh (Classical) damping force Fd (ndof x 1)
Fd(:,i+1)=cmat*ud_cur;
dFd_du=cmat*dud_du;

%%% Nonlinear damping force Fnd (ndof x 1)
if abs(ud_cur(1))<=1
    fnd1=cnd*ud_cur(1);
    ctan1=cnd;
else
    fnd1=cnd*abs(ud_cur(1))^.6*sign(ud_cur(1));
    ctan1=.6*cnd*abs(ud_cur(1))^(-.4);
end
ud_21=ud_cur(2)-ud_cur(1);   % define ud_12 for convenience
if abs(ud_21)<=1
    fnd2=cnd*ud_21;
    ctan2=cnd;
else
    fnd2=cnd*abs(ud_21)^.6*sign(ud_21);
    ctan2=.6*cnd*abs(ud_21)^(-.4);
end
Fnd(:,i+1)=[fnd1-fnd2; fnd2; 0]; %Fnd(:,i+1)

dFnd_du=[ctan1+ctan2        -ctan2          0
         -ctan2              ctan2          0
         0                    0             0]*dud_du;
R_u_cur=(-4/dtan)*mmat*ud_pre+(4/dtan^2)*mmat*(u_cur-u_pre)+ ...
    (Fd(:,i+1)+Fnd(:,i+1)+Fs(:,i+1)+mmat*ones(ndof,1)*ugddan(i+1) ...
    +Fd(:,i)+Fnd(:,i)+Fs(:,i)+mmat*ones(ndof,1)*ugddan(i));

if norm(R_u_cur) < err
    break;
elseif j>jmax-1
    error(['Newton Raphson Failed to converge after jmax = ' ...
        num2str(jmax) ' iterations.  Fail at i = ' num2str(i)])
```

```
        end
        dR_du=(4/dtan^2)*mmat+(ktan_+dFd_du+dFnd_du);
        u_cur=u_cur-dR_du\R_u_cur;
        ud_cur=-ud_pre+(2/dtan)*(u_cur-u_pre);
    end
fnd(:,i+1)=[fnd1; fnd2; 0];
u(:,i+1)=u_cur;
ud(:,i+1)=ud_cur;
udd(:,i+1)=mmat\(-Fd(:,i+1)-Fnd(:,i+1)-Fs(:,i+1)-
mmat*ones(ndof,1)*ugddan(i+1));
end
```