T.C.
İSTANBUL KÜLTÜR UNIVERSITY
INSTITUTE OF GRADUATE STUDIES

AN OPENSEES GRAPHICAL USER INTERFACE FOR STRUCTURAL DYNAMICS INSTRUCTION

Master of Science Thesis

Bilal EIN LAROUZI

1800004561

Department: Civil Engineering

Programme: Structural Engineering

Supervisor: Assist. Prof Dr. Gökhan YAZICI

AUGUST 2020

T.C.
İSTANBUL KÜLTÜR UNIVERSITY
INSTITUTE OF GRADUATE STUDIES

AN OPENSEES GRAPHICAL USER INTERFACE FOR STRUCTURAL DYNAMICS INSTRUCTION

Master of Science Thesis

Bilal EIN LAROUZI

1800004561

Date of submission: 04 August 2020

Date of defence examination: 18 August 2020

Supervisor and Chairperson: **Assist. Prof Dr. Gökhan YAZICI**

Members of Examining Committee: **Asst. Prof.Dr. Erdal COŞKUN**
Asst. Prof.Dr. Cenk ÜSTÜNDAĞ

AUGUST 2020

Preamble:

I would to express my sincere gratitude to the İSTANBUL KÜLTÜR University for letting me fulfil my dream to get the master's degree. I would to express my thankful feelings to all the academic staff of civil engineering department who help and guided me through my graduated study in the past two years.

I would like to express my deepest gratitude and appreciation to my supervisor Dr. Gökhan YAZICI, who expertly guided me through my master's degree and shared the excitement and enthusiasm to keep me constantly engaged with my thesis topic. I believe that without his suggestions and instruction this thesis would not be completed.

I would thank my Family who were always supporting and encourage me through all my life stages this thesis would not have been possible without them. Finally, I would send me deepest gratitude to all my friends for their support.

Table of Contents

ABBREVIATIONS

GUI:	Graphical user interface
PA:	Pushover Analysis
LRDB:	Low rubber damping bearing
HRDB	High rubber damping bearing
SFP:	Single Friction Pendulum Bearing
TFP:	Triple Friction Pendulum Bearing

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

F_y:	Yield Strength
E:	Modulus of elasticity
F_c:	Compressive strength
F_{cu}:	Concrete crashing strength
A:	Area of the section
I_z:	Moment of inertia about Z-axiz
[M]:	Mass matrix
[C]:	Damping matrix
[K]:	Stiffness matrix
Ø:	Mode shape
ω:	Natural frequency
F_d:	Damping force
K_0, K_e, K_{intial}:	Initial stiffness
K_s:	Strain hardening stiffness
α:	Strain hardening ratio
F_c:	Maximum force
δ_c:	Deformation at maximum force
K_c:	Post capping stiffness
F_r:	Friction of the yield strength
T_n:	Natural time period
R, R_{eff}:	Effective radius
μ:	Friction coefficient
d:	Displacement controlled

Üniversite:	**İstanbul Kültür Üniversitesi**
Enstitüsü:	**Lisansüstü Eğitim Enstitüsü**
Anabilim Dalı:	**İnşaat Mühendisliği**
Programı:	**Yapı (İngilizce)**
Tez Danışmanı:	**Dr. Öğretim Üyesi Gökhan YAZICI**
Tez Türü ve Tarihi:	**Yüksek Lisans – Ağustos 2020**

KISA ÖZET

Yapı Dinamiği Eğitimi için OpenSEES Grafik Kullanıcı Arayüzü

Bilal EIN LAROUZI

Bu tez kapsamında, yapı dinamiği eğitiminde kullanmak amacıyla OpenSEES platformu için bir grafik kullanıcı ara yüzü oluşturulmuştur. Bu grafik kullanıcı ara yüzü, MATLAB App Designer yazılımı kullanılarak hazırlanmış olup, OpenSEES platformu için analiz girdi dosyalarının hazırlanmasını ve analiz sonuçlarının görselleştirmesini sağlamaktadır.

Hazırlanan grafik kullanıcı ara yüzü, iki araç içermektedir. İlk araç, farklı tipte sismik izolasyon sistemi atama seçeneği ile düzlem çerçevelerin zaman tanım alanında doğrusal analizinde kullanılabilmektedir. İkinci araç ise düzlem çerçevelerin itme analizi için kullanılabilmekledir. Geliştirilen araçlar ile inşaat mühendisliği öğrencilerinin yapı dinamiği ve deprem mühendisliği ile ilgili bilgilerini geliştirmeleri amaçlanmıştır.

Bu tez kapsamında geliştirilen grafik kullanıcı ara yüzü, açık kaynaklı bir analiz platformu olan OpenSEES ile düzlem çerçeve sistemlerin modellenmesi, analizi ve analiz sonuçlarının görselleştirilmesi için gerekli programlama yükünü azaltmakta ve öğrencilerin parametrik çalışma yapmasını kolaylaştırmaktadır.

Anahtar Kelimeler: Grafik kullanıcı ara yüzü, OpenSEES, Yapı dinamiği, Deprem Mühendisliği, Sismik izolasyon

University: İstanbul Kültür University

Institute: Institute of Graduate Studies

Department: Civil Engineering

Program: Structural Engineering

Supervisor: Asst. Prof. Dr. Gokhan YAZICI

Degree Awarded and Date: Master of Science – AUGUST 2020

ABSTRACT

AN OPENSEES GRAPHICAL USER INTERFACE FOR STRUCTURAL DYNAMICS INSTRUCTION

Bilal EIN LAROUZI

A Graphical User Interface (GUI) for OpenSEES platform for structural dynamics education was created within the scope of this thesis. The Graphical User Interface (GUI) was created using Matlab App Designer to create analysis input files and to visualize analysis results.

The graphical user interface includes two seperate tools. The first tool can be used to conduct linear time history analysis of plane frames with different seismic isolation systems. The second tool can be used to conduct pushover analysis of plane frames. These tools are aimed to improve the knowledge of çivil engineeering students in the fields of structural dynamics and earthquake engineering.

This thesis shows that the apps will help the structural engineering students to improve their knowledge about the dynamic and static problems. This thesis shows that the apps help the user also to save time by trying different modellings in short time with no need to learn the OpenSEES programming language.

The graphical user interface developed in this thesis significantly decreases the required programming load in the modelling, analysis and the visualization of the analysis results of plane frame systems with the open source analysis platform OpenSEES, making it easier for students to conduct parametric studies.

Key Words: Graphical user interface, OpenSEES, Structural Dynamics, Earthquake engineering, Seismic isolation

Science Code: 624.03.01

1. Introduction

1.1 Introduction:

Civil engineering strongly depending on the computer programs which allow the engineers to model, simulate, analyse and design the structures. Structural engineering programs used to do the complex analyses and calculation, which cannot be done by hand. These computer programs use different methods of analysis and some programs use integrated graphical user interfaces (GUI) like ETABS, SAP2000 and Robot while others such as OpenSEES require users to manually create text-based input files. OpenSEES program is an open source tool which used to model and analyse frames and structures, OpenSEES can be used to analyse structures under dynamic and earthquake loading. In order to help civil and structural engineering students to have better understanding and solidifying their knowledge about the structure behaviour from that the idea of creating a graphical user interface for the OpenSEES program is founded. The tools developed in this thesis are designed mainly for educational purposes act as pre-processors and post-processors for OpenSEES which does not have a GUI. OpenSEES users need a solid knowledge of OpenSEES commands and experience to be able to model and analyse the structure correctly because OpenSEES uses a text-based input files.

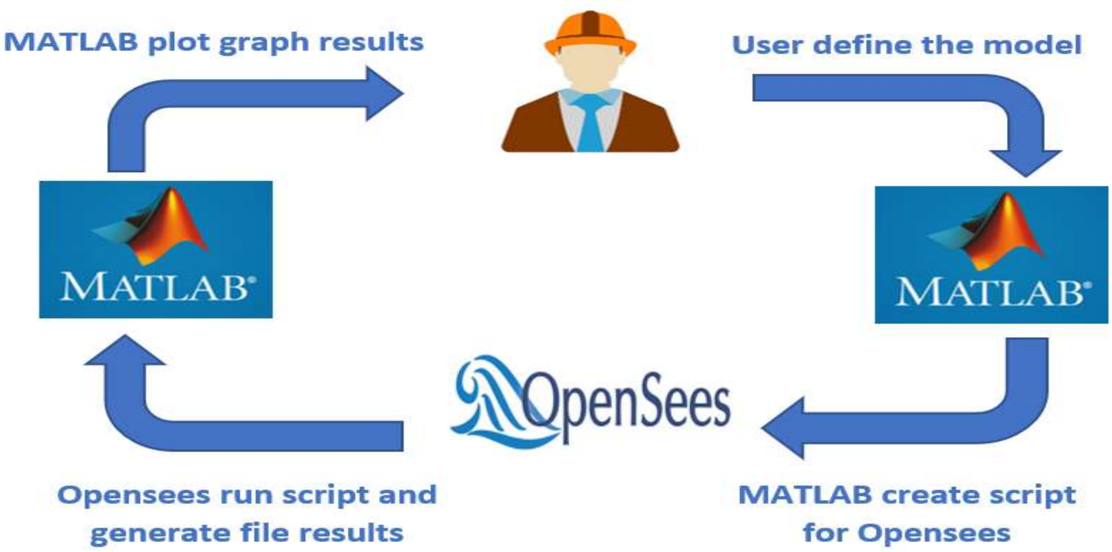

Figure 1.1:Main Idea of the Apps

1.2 Motivation

This thesis deals with creating two GUIs which have been designed with purpose mainly to help structural engineering student to improve their knowledge about structural dynamics and earthquake engineering. It will allow the students to have better understanding by trying different modelling properties in short time with a simple educational interface. The applications have the ability to generate coding to model the two-dimensional frames to be analysed for Eigen analysis, linear Time-History analysis and the first mode pushover analysis. Modal and linear time history analyses are important type of analyses for the frames to predict the frame behaviour under dynamic loading, while the pushover analysis is used to predict the strength capacity of frame by pushing the frame to its limit. These apps can be more generalized in future and can be used as the base for more apps for different type of analysis. For better and easier understanding this tool also designed to be as post-processing to plot the results which be written as table into graphs.

1.3 Literature Review

GID + OpenSEES:

In 2017, Vassilis K. Papanikolaou, Theocharis Kartalis-Kaounis, Evangelos Protopapadakis, and Theocharis Papadopoulos created a new graphical user interface for OpenSEES. It connects the OpenSEES solver to the pre and post processor GID. This graphical user interface provides extended user dialogs and tools to create modal geometry, assign materials, elements and boundary conditions. It also allows the user to choose the analysis options and preform the analysis using the OpenSEES solver. After the analysis the GID automatically transform the results from the row numerical to more understandable form. By GID post-processing the results can be presented as deformed shape viewer with animation capabilities. This GUI its main goal was to help the researchers and user who use the OpenSEES platform to analyse their work to save time which they spent by writing their own models [9].

DYANAS:

Dynamic analysis of single-degree-of-freedom systems (DYANAS) was created in 2018. Georgios Baltzopoulos, Roberto Baraschino, Iunio Iervolino and Dimitrios Vamvatsikos have created DYANAS as Graphical user interface for OpenSEES by using MATLAB processor. This GUI uses OpenSEES finite element platform to preform non-linear dynamic analysis for single degree of freedom oscillators. Main advantages for the DYANAS interface are ease in the definition of the required analysis parameters and corresponding seismic input, efficient execution of the analyses themselves and availability of a suite of convenient, built-in post-processing tools for the management and organization of the structural responses. The types of dynamic analysis frameworks supported are incremental, multiple-stripe and cloud. Simultaneous consideration of pairs of uncoupled dynamic systems gives the possibility for intensity measures to refer to bidirectional ground motion [10].

2. OpenSEES platform and Analyses

2.1 OpenSEES overview

The OPEN System of Earthquake Engineering Simulation which known as "OpenSEES" is a proprietary object-oriented software. OpenSEES was developed by the National Science Foundation as sponsored by Pacific Earthquake Engineering centre (PEER). It uses the C++ as primary programming language in addition to several Fortran numerical libraries for linear equation solver. It allows the user to create finite element projects to simulate the response of structural and geotechnical systems subjected to the earthquakes.

The main framework of OpenSEES (Figure 2.1) consists of:

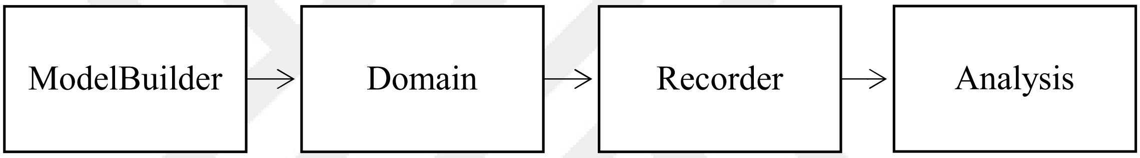

Figure 2.1:OpenSEES framework

Shortly the modelBuilder where the model is constructed and added to the domain, while the domain holds the model to given time. The recorder is used to define the parameters to be recorded from the model during the analysis. The analysis transfers the model from state at time t to the state at time (t+dt).

The domain (Figure 2.2) consists of:

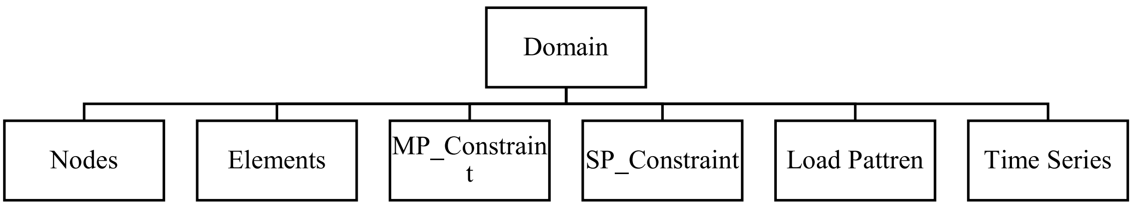

Figure 2.2: OpenSEES Domain's components

2.2 Modelling the structure in OpenSEES

The first step of simulating and analysis any type of structure is the modelling of the structure. It is an important step because the analysis result will be useless if the modelling has been defined wrong. Modelling the structure is consists of:

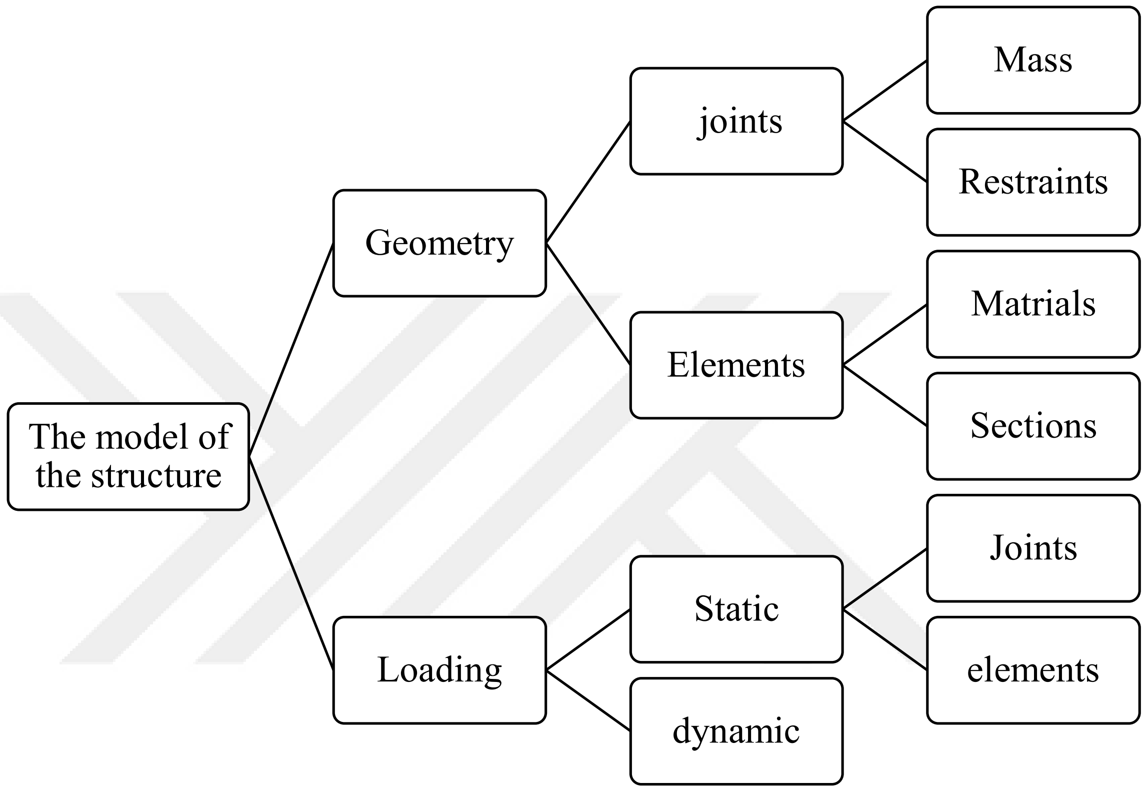

Figure 2.3: Modelling the structure by OpenSEES

2.2.1 Defining the Geometry of modelling:

Geometry of structure is defining the general shape of the model. In OpenSEES defining the model take place in domain of the problem it starts by defining model builder and joints coordinates.

<u>Model Builder:</u>

Model builder in OpenSEES is a command used to define the dimensional type of the model. Model builder is constructed by:

[model basic -ndm $ndm <-ndf $ndf>]

Where $ndm is dimension of problem and $ndf is number of degrees of freedom in each joint [1].

Node command:

Node command used to define joints and it's the coordinates. These joints can be used to connect the members, assign masses, define restraints and define joints loading. Joints of the structural system are defined by the node command:

[node $nodeTag (ndm $coords)]

Where the $nodeTag represent the unique number of the node. While the $coords represent the coordinates x, y, z respectively of the node [1].

The constructed nodes represent the joints of structure. These nodes used to define the restraints and masses.

Fix command:

To define the type of joint fix, bin, roller or free the Fix command is used.

[fix $nodeTag (ndf $constrValues]

Where $nodeTag represents the pre-defined nodes, ndf represent the degree of freedom and $constrValues represent the value of restraint (0 means free and 1 means fix) [1].

Equal DOF command:

EqualDOF is a type of restraint command. Which control the same value of deformation for two or more nudes.

[equalDOF $rNodeTag $cNodeTag $dof1 $dof2 ...]

Where $rNodeTag is tag of pre-defined node (retained node), $cNodeTag is tag of pre-defined node (constrained node) and $dof1, $dof2 are represent the degree of freedom which is constrained [1].

Mass command:

Joint masses are defined by mass command. These defined masses are used to construct the mass matrix which is used in modal analysis.

[mass $nodeTag (ndf $massValues)]

Where $nodeTag represent pre-defined nodes where the mass lumped, ndf represent the degree of freedom and $massValue the value of mass [1].

Materials

After defining the joints, the materials need to be defined. The materials are defined by the type of material. Some of the common materials command are listed below.

Elastic Isotropic Material

This type of material is defined by:

[nDMaterial ElasticIsotropic $matTag $E $v <$rho>]

Where $matTag represent the unique number of the material. $E is the modulus of elasticity, $v is the Poisson's ratio and $rho is the mass density [1].

Steel01 Material

Steel01 material is uniaxial bilinear material with kinematic hardening to represent the nonlinear behaviour of steel material (Figure 2.4).

[uniaxialMaterial Steel01 $matTag $Fy $E0 $b]

Where $matTag is unique number $Fy is the yield strength of steel $E is the modulus of elasticity of material and $b is the strain-hardening ratio [1].

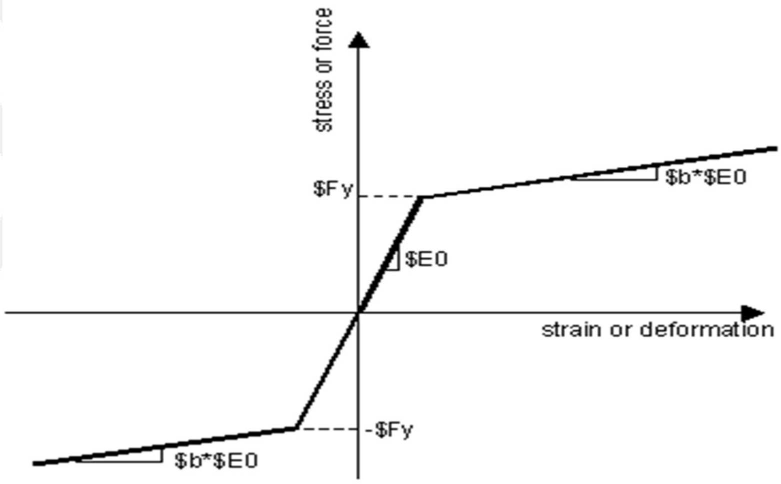

Figure 2.4: Steel 01 Material in OpenSEES [1]

Concrete01 Material

This material used to model uniaxial zero tensile strength concrete material with linear degraded unloading or reloading (Figure 2.5).

[uniaxialMaterial Concrete01 $matTag $fpc $epsc0 $fpcu $epsU]

Where $matTag is unique number of materials. $fpc represent the compressive strength of concrete. $epsc0 is the concrete strain at maximum strength. $fpcu is concrete crushing strength. While $epsU is the strain of concrete at crushing strength level [1].

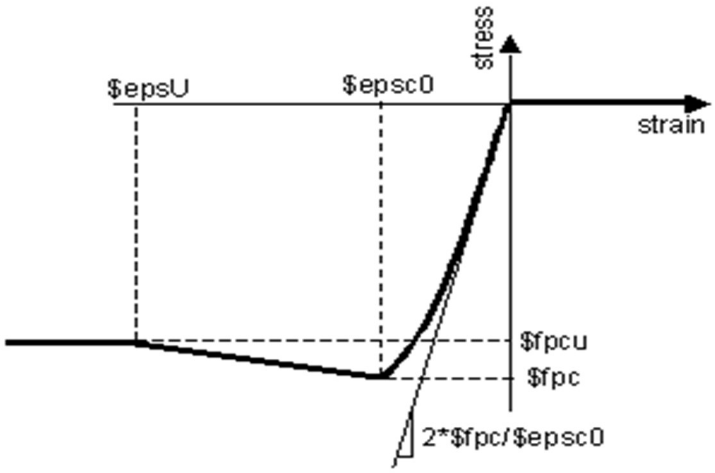

Figure 2.5: Concrete01 in OpenSEES [1]

Sections:

The defined materials are used to define the sections. OpenSEES has a reach library with many types of structural sections. These sections types allow the user to model the structure properly. The following section will define some of sections types:

Elastic Section:

This type used to construct an Elastic section which is used in elastic analysis of section. Elastic section creation doesn't need pre-defined material.

[section Elastic $secTag $E $A $Iz]

Where the $secTag is a unique integer tag, $E is a modulus of elasticity of the section, $A area of the section and $Iz is the moment of inertia on the Z-axis [1].

Wide Flange section:

Wide flange section command allows the user to define steel I-section in plane analysis (2D analysis).

[section WFSection2d $secTag $matTag $d $tw $bf $tf $Nfw $Nff]

Where $secTag is unique number tag section, $matTag is a tag for pre-defined material, $d is section depth, $tw is web thickness, $bf is flange width, $tf flange thickness, Nfw is the number of fibres in web and $Nff number of fibres in flange. Number of fibres used to define a mesh to analyse the section a greater number of fibres gives more proper analysis [1].

<u>Reinforced Concrete section:</u>

Reinforced concrete section command allows the user to create a plane rectangular section with defining the core and cover regions and steel layers which can be used in nonlinear analysis.

[section RCSection2d $secTag $coreTag $coverTag $steelTag $d $b $cover $Atop $Abot $Aside $Nfcore $Nfcover $Nfs]

Where $secTag is unique integer tag for the section, $coreTag is pre-defined material tag for core region, $coverTag is pre-defined material tag for cover region, $steelTag is predefined tag material for steel reinforcement bars. $d is the depth of section, $b is width of section. $cover is the depth of cover. $Atop is area of steel in top layer, $Abot is area of steel in bottom layer, $Aside is the area of steel in intermediate layer. $Nfcore is number of fibres in core, $Nfcover number of fibres in cover and $Nfs number of steel bar in top and bottom layers [1].

<u>Elements:</u>

Defined sections are used to create elements. OpenSEES has ability to construct several types of element that gives the user huge options to model the structure. Here are some of the type of beams columns and trusses elements.

<u>Truss Element:</u>

Truss element is the element which carry only the axial force. Truss element can be modelled by the following command.

For truss element which is defined by area and uniaxial materials:

[element truss $eleTag $iNode $jNode $A $matTag <-rho $rho>]

For truss element if pre-defined section is used

[element trussSection $eleTag $iNode $jNode $secTag <-rho $rho>]

Where $eleTag is unique integer, $iNode $jNode are end nodes of the element, $A is the section area, $matTag is predefined uniaxial material, $secTag is tag of predefined section, $rho is mass density [1].

<u>Elastic Beam Column Element:</u>

Elastic beam or column element used in modelling to catch the linear behaviour of the element. This type of element constructs by:

[element elasticBeamColumn $eleTag $iNode $jNode $A $E $Iz $transfTag <-mass $massDens> <-cMass>]

Where $eleTag is unique integer element tag, $iNode $jNode are end nodes of the element, $A is the cross sectional area of element, $E is modulus of elasticity, $Iz is moment of inertia of section, $transTag is predefined transformation geometry tag, $massDens is the mass density of section and -cMass is the mass of element which used to construct mass matrix [1].

Nonlinear Beam Column Element:

The nonlinear is used to get the nonlinear behaviour of the element. the nonlinear beam column element is constructed by:

[element nonlinearBeamColumn $eleTag $iNode $jNode $numIntgrPts $secTag $transfTag]

Where $eleTag is unique integer tag element. $iNode $jNode are end nodes of the element. $numIntgrPts number of integration points along the element, $secTag is tag for predefined section and $transTag is predefined transformation geometry tag [1].

Zero length element:

Zero length element is used to construct an element with zero length such as plastic hinges in modelling of concentrated plasticity.

[element zeroLength $eleTag $iNode $jNode -mat $matTag1 $matTag2 ... -dir $dir1 $dir2...]

Where $eleTag is unique integer of element, $iNode $jNode are end nodes of the element. $matTag1, $matTag2 are predefined uniaxial materials, $dir1, $dir2 are material direction [1].

Geometry Transformation:

Geometry transformation is used to construct a coordinate transformation object. To transfer the stiffness of beam element stiffness and resisting force from the basic system to global system.

[geomTransf transfType? arg1? ...]

Where transType? is type Linear, P-delta or Corotational transformation and arg1? Is argument dependes on the type of transformation and the laying direction of the element [1].

Rayleigh Damping:

Damping of the model in OpenSEES is defined by defining coefficients of mass and stiffness matrices. The damping is introduced by command:

[rayleigh $alphaM $betaK $betaKinit $betaKcomm]

Where $alphaM is factor applied for mass matrix, $betaK is factor applied for current stiffness matrix, $betaKinit is factor applied for initial stiffness matrix and $betaKcomm factor applied for committed element stiffness matrix [1].

2.2.2 Defining the loading on the model in OpenSEES

In OpenSEES the next step after creating the geometry of the model as all analysis programs to complete the model is assigning the static loads to joint or element and dynamic loading. OpenSEES use pattern commands to construct the load pattern.

Plain Pattern

Plain pattern is used to construct a pattern associated to pre-defined time series. Plain pattern is used to define the static loading.

[plain Pattern $patternTag $tsTag < -fact $cFactor > {
load…
eleLoad…
sp…
}]

Where $patternTag is unique integer for pattern, $tsTag pre-defined time series tag and $cFactor is constant factor. load is nodal load command, eleLoad is element load command and sp is single point load command [1].

Nodal Load:

Nodal load is used to assign static load to the pre-defined joint. The nodal load is constructed in the Plain Pattern command.

[load $nodeTag (ndf $LoadValues)]

Where $nodeTag is tag of pre-defined node where the load assigned. ndf represent degree of freedom represent the direction of loading and $LoadValues the value of loading [1].

Element Load:

To define the static load to the element; eleLoad command is used. The element load is constructed within Plain Pattern.

For element uniform loading:

[eleLoad -ele $eleTag1 <$eleTag2 ....> -type -beamUniform $Wy <$Wx>]

For the point loading:

[eleLoad -range $eleTag1 $eleTag2 -type -beamPoint $Py $xL <$Px>]

Where $eleTag represent the pre-defined element tag where the load assigned, $Wy is value of uniform loading in Y-axis direction, $Wx is value of uniform loading in X-axis direction. $Py is point load in Y-axis direction. $xL is element relative distance when the point load acting. $Px is point load in X-axis direction [1].

<u>Single Point constraint:</u>

Single point constraint command is used to define the initial deformation of the joint. this command is defined in enclosing load pattern:

[sp $nodeTag $dofTag $dofValue]

Where $nodeTag is pre-defined node tag where the deformation applied, $dofTag is degree of freedom at node to which constraint is applied and $dofValue is reference value constraint [1].

<u>Uniform Excitation Pattern:</u>

Applying uniform excitation to modal in a certain direction is defined by:

[pattern UniformExcitation $patternTag $dir -accel $tsTag <-vel0 $vel0> <-fact $cFactor>]

Where $patternTag is unique integer of the pattern, $dir is direction of ground motion, $tsTag is pre-defined time series tag, $vel0 initial velocity and $cFactor is constant factor [1].

<u>Time Series</u>

Time series command is used to define time series which can be later used in load pattern.

<u>Constant time series:</u>

Is used to construct time series where load factor is independent of the time.

[timeSeries Constant $tag <-factor $cFactor>]

Where the $tag is unique integer of time series and $cFactor is constant factor.

<u>Linear time series:</u>

Is used to construct a time series where the load factor has linear relationship with time.

[timeSeries Linear $tag <-factor $cFactor>]

Where the $tag is unique integer of time series and $cFactor is constant factor [1].

Path time series:

Path time series used to define a relationship between the load factor and time. For a load path where the factors are specified in a file for a constant time interval between points:

[timeSeries Path $tag -dt $dt -filePath $filePath <-factor $cFactor>]

For a load path where both time and values are specified in a list included in the command:

[timeSeries Path $tag -fileTime $fileTime -filePath $filePath <-factor $cFactor>]

Where $tag is unique integer for time series. $dt time interval between two respective point, $filePath is a file containing load factor value, $fileTime is file containing time steps value and $cFactor is constant factor [1].

2.2.3 Defining the recorder in OpenSEES:

The recorder is a type of command which is define the type of data will be record during the analysis. Recorder command is recording the data as result various time. OpenSEES has two type of recorder nodes recorder and element recorder.

Nodes Recorder:

Nodes recorder records the response of the node during the analysing time.

Node recorder and Envelope Node recorder:

Node recorder is used to record the response of a pre-defined node during the analysing time. when envelope recorder is defined the max and min values are recorded.

For node recorder:

[recorder Node <-file $fileName> <-time> <-node $node1 $node2 ...> <-nodeRange $startNode $endNode> -dof ($dof1 $dof2 ...) $respType]

For node envelope recorder:

[recorder EnvelopeNode <-file $fileName><-time> <-node $node1 $node2 ...> <-nodeRange $startNode $endNode>-dof ($dof1 $dof2 ...) $respType]

Where $fileName is a name of file where data will be recored, $node1, $node2 are pre-defined nodes, $startNode first node in the range, $endNode final node in the range and $dof1,$dof2 are the degree of freedom whose response requested [1] .

$respType is response type request to record:

disp	displacement*
vel	velocity*
accel	acceleration*
incrDisp	incremental displacement
"eigen i"	eigenvector for mode i
reaction	nodal reaction
rayleighForces	damping forces

<u>Drift Node Recorder:</u>

Drift recorder is used to catch the drift value between two nodes. by using this command story drift can be determined:

[recorder Drift <-file $fileName> <-time> -iNode $inode1 $inode2 ... -jNode $jnode1 $jnode2 ... -dof $dof1 $dof2 ... -perpDirn $perpDirn1 $perpDirn2 ...]

Where $fileName is a file name where the data record $inode1, $inode2 tags of set of i nodes for which drift is being recorded $jnode1, $jnode2 tags of set of j nodes for which drift is being $dof1 $dof2 is nodal degree of freedom for which drift is being recorded $perpDirn1 $perpDirn2 are set of perpendicular global directions [1].

<u>Element Recorder:</u>

Element recorder is used to record the response of element or group of elements. For element envelope recorder the max, min and absolute max value will be recorded. For element recorder

[recorder Element <-file $fileName <-time> <-ele ($ele1 $ele2 ...)> <-eleRange $startEle $endEle> $arg1 $arg2 ...]

For envelope element recorder

[recorder EnvelopeElement <-file $fileName <-time> <-ele ($ele1 $ele2 ...)> <-eleRange $startEle $endEle> $arg1 $arg2 ...]

Where $filename is representing the file name where the data are recorded. $ele1, $ele2 are tag of element whose response is being recorded. $startEle, $endEle tags for the start and end elements whose response is being recorded. $arg1, $arg2 are argument which depends on the type of recorder [1].

2.2.4 Defining the analysis:

The final step to run a model is defining the analysis. The analysis in OpenSEES is consist of many commands as listed below:

Constraints Command:

Constrain command is used to determine how the constraint equation are enforced in analysis [1].

Numberer Command:

Numberer command is used to determine the mapping between equation number and degree of freedom and the number of degrees of freedom.

System command:

The system command is used to construct linear system of equation and linear solver. There are many types of the system command some of these types are listed and defined as following:

Band General: is used to construct band general system of equation object and used for matrix systems which have a banded profile

Band SPD: is used to construct linear system of equation object and used the positive definite matrix system

Full General: is used to construct linear system of equation and used the full matrix system which means there is no space saving technique to minimize to amount of memory to be used [1].

Algorithm Command:

The algorithm command is used to create algorithm solution object to determine the sequence of steps taken to solve nonlinear equation. There are many types of algorithm such as: Linear algorithm, Newton algorithm, Modified Newton algorithm and Secant Newton algorithm [1].

Integrator Command

The integrator command is used to construct integrator object. Integrator type depends on the type of analysis whether static or transient.

Static analysis could use load control or displacement control integrator which can be used in pushover analysis.

Transient analysis could use central difference or Newmark method integrator which can be used to solve the equation of motion [1].

Analysis Command

The analysis command is used to determine the type of analysis to be performed whether static, transient or variable transient [1].

Analyze Command

This command is used to let the OpenSEES app to start the analysis progress.

2.3 Dynamic Analysis

Dynamic analysis is a branch of structural analysis which determine the behaviour of the structure subjected to dynamic loading. Dynamic loading is a load which is various with time domain. Dynamic analysis has a many type of analysis such as Modal analysis and Time History analysis.

2.3.1 Modal Analysis:

Modal analysis or Eigen analysis is used to determine the dynamic properties of the structure. It is defining as the estimation of the mode shapes of the undamped structure. It is generally the first step in analysis for non-loaded structure in other word it depends on the physical properties of the structure itself. The modal analysis of structure uses the mass and stiffness matrices; changing the stiffness or mass matrices will result changing in the modal analysis results. This analysis gives us a prediction of the structure how it will vibrate. The structure has a complex movement consist of participation percentage from each mode shape. The modal analysis allows us to get the natural frequency, natural periods, the mode shapes and the participation factor of each mode of the structure (Figure 2.6). Multi degree of freedom systems are analysed as a series of single degree of freedom systems [2].

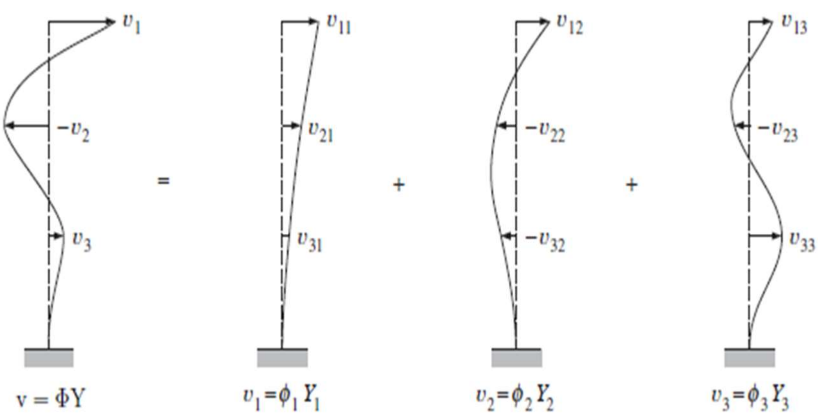

Figure 2.6:Mode shape by modal analysis [2]

The modal analysis can be calculated by the following equation [2]:

$$[M]\ddot{U}(t) + [C]\dot{U}(t) + [K]U(t) = f(t) \qquad \text{Equation 2.3.1.1}$$

Where

$$[M] = \begin{bmatrix} M1 & 0 & 0 \\ 0 & M2 & 0 \\ 0 & 0 & Mn \end{bmatrix} \qquad \text{Equation 2.3.1.2}$$

$$[C] = damping\ matrix \qquad \text{Equation 2.3.1.3}$$

$$[K] = \begin{bmatrix} K1 + K2 & -K2 & 0 \\ -K2 & K2 + K(n) & -K(n) \\ 0 & -K(n) & K(n) \end{bmatrix} \qquad \text{Equation 2.3.1.4}$$

Assuming that:

$$U = \emptyset Sin\omega t \quad \text{and} \quad \ddot{U} = -\omega^2 \emptyset\, Sin\omega t \qquad \text{Equation 2.3.1.5}$$

Since the structure is not loaded and undamped structure and for the orthogonality of mode shapes the frequency equation can be generated as following:

$$|[k] - \omega^2[M]|\emptyset = \{0\} \qquad \text{Equation 2.3.1.6}$$

This equation has (n) solutions for (n) degrees of freedom. Each solution is a distinct eigen vector which represents a natural frequency (ω). The mode shape can be determined as substituting the natural frequency in the equation to get modal shape.

2.3.2 Time History Analysis

Time history analysis is a study of the dynamic response of the structure exposed to a dynamic loading such as earthquake loading. Linear dynamic analysis of the multi degree of freedom system is modelled by mass, linear stiffness and linear damping matrices. The time history analysis uses an actual ground motion records to find the response of the structure and its internal forces by linear elastic analysis for each time step. In linear time history analysis, the structure analysed for time domain and interval time steps. The linear time history analysis takes into consideration only the elastic linear properties of the structure. The dynamic factors of analysis such a time interval and damping are affecting directly to obtained results.

Solving time history by direct integration by Newmark β:

Newmark β is one of the methods that used to solve nonlinear equations such as the equation of motion in the time domain (Time history analysis) [2].

$$M\ddot{U}(t_i) + C\dot{U}(t_i) + KU(t_i) = F(t_i) \qquad \text{Equation 2.3.2.1}$$

Where U represent the displacement as function of time, $\dot{U}$ and $\ddot{U}$ are represent the velocity and the acceleration as function of time respectively. While M the mass matrix, C the damping matrix and K is the stiffness matrix. While F is the external excitation force (Dynamic loading) as function of time. i notation refers to the time step number. Where the equation of motion between two respective time steps (i and i+1 steps) is given as follow:

$$\Delta M\ddot{U}(t_i) + \Delta C\dot{U}(t_i) + \Delta KU(t_i) = \Delta F(t_i) \qquad \text{Equation 2.3.2.2}$$

In this method the β and γ coefficient are define the change in acceleration for one step these coefficients are specifies the stability and accuracy of the result. Generally, β = 0.5 and γ = 0.25 which gives accuracy in results. It is explained in two different ways Average acceleration and linear acceleration:

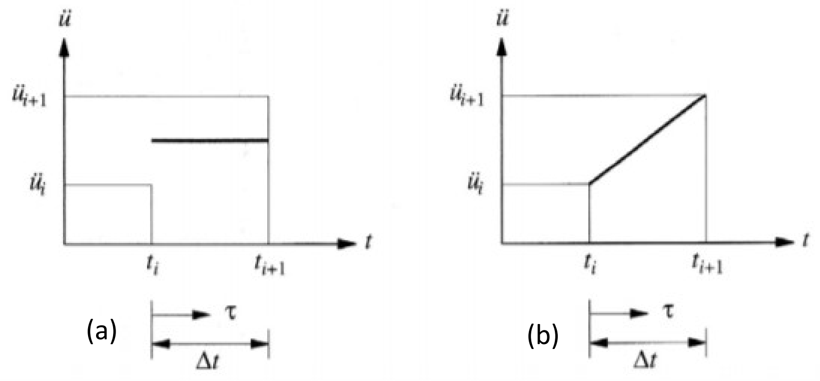

Figure 2.7: Newmark β method. (a) average acceleration (b) linear acceleration [2]

For calculation:

From the basis equation of Newmark:

$$\Delta\dot{U}_i = [(1-\gamma)\,\Delta t]\,\ddot{U}_i + (\gamma\,\Delta t)\,\ddot{U}_{i+1} = (\Delta t)\ddot{U}_i + (\gamma\,\Delta t)\,\Delta\ddot{U}_i \qquad \text{Equation 2.3.2.3}$$

And

$$\Delta U_i = (\Delta t)\dot{U}_i + [(0.5-\beta)(\Delta t)^2]\ddot{U}_i + [\beta(\Delta t)^2]\ddot{U}_{i+1} \qquad \text{Equation 2.3.2.4}$$

$$= (\Delta t)\dot{U}_i + \frac{(\Delta t)^2}{2}\ddot{U}_i + \beta(\Delta t)^2\,\Delta\ddot{U}_i \qquad \text{Equation 2.3.2.5}$$

Solving for $\Delta\ddot{U}_i$

$$\Delta\ddot{U}_i = \frac{1}{\beta(\Delta t)^2}\Delta U_i - \frac{1}{\beta\Delta t}\dot{U}_i - \frac{1}{2\beta}\ddot{U}_i \qquad \text{Equation 2.3.2.6}$$

By substitution $\Delta\ddot{U}_i$ in equation for $\Delta\dot{U}_i$

$$\Delta\dot{U}_i = \frac{\gamma}{\beta\Delta t}\Delta U_i - \frac{\gamma}{\beta}\dot{U}_i + \Delta t(1 - \frac{\gamma}{2\beta})\ddot{U}_i \qquad \text{Equation 2.3.2.7}$$

Then we substitute the $\Delta\dot{U}_i$ and $\Delta\ddot{U}_i$ in the equation pf motion.

$$m\Delta\ddot{U}_i + c\Delta\dot{U}_i + k\Delta U_i = \Delta F_i \qquad \text{Equation 2.3.2.8}$$

$$\hat{k}\Delta U_i = \Delta\hat{F}_i$$

Where

$$\hat{k} = k + \frac{\gamma}{\beta\Delta t}c + \frac{1}{\beta(\Delta t)^2}m \qquad \text{Equation 2.3.2.9}$$

$$\Delta\hat{F}_i = \Delta F_i + (\frac{1}{\beta\Delta t}m + \frac{\gamma}{\beta}c)\dot{U}_i + [\frac{1}{2\beta}m + \Delta t(\frac{\gamma}{2\beta} - 1)c]\ddot{U}_i \qquad \text{Equation 2.3.2.10}$$

Once the ΔU_i is computed we can use:

$$U_{i+1} = U_i + \Delta U_i \qquad \dot{U}_{i+1} = \dot{U}_i + \Delta\dot{U}_{i+1} \qquad \ddot{U}_{i+1} = \ddot{U}_i + \Delta\ddot{U}_{i+1} \qquad \text{Equation 2.3.2.11}$$

2.3.3 Damping

Damping is known as the dissipation of energy in structural system during dynamic loading on the structure. Damping force $(\boldsymbol{fd})$ can be due to several reasons and sometimes more than one user can be contributed at same time. The dissipation in energy can be due to several reason such as internal friction in material when structural member deformed, steel connection, opening and closing microcracks in concrete members or friction between structural and nonstructural elements. The damping coefficient in structural cannot be calculated from dimension and properties of structure also cannot be determined accurately due to unknown mechanism which structure will perform during an earthquake. the damping in the structure can be modeled in many methods such as Viscous damping or Rayleigh damping [2].

Viscous damping coefficient:

Viscous damping is the simplest way to model the damping force in building. In viscous damping modelling the force dissipated during the dynamic loading is proportional to the velocity of the structure (Figure 2.8). Viscous damping coefficient value is recommended in building codes depending on the structure properties. These values are determined from experiments and studies on the actual structure [2].

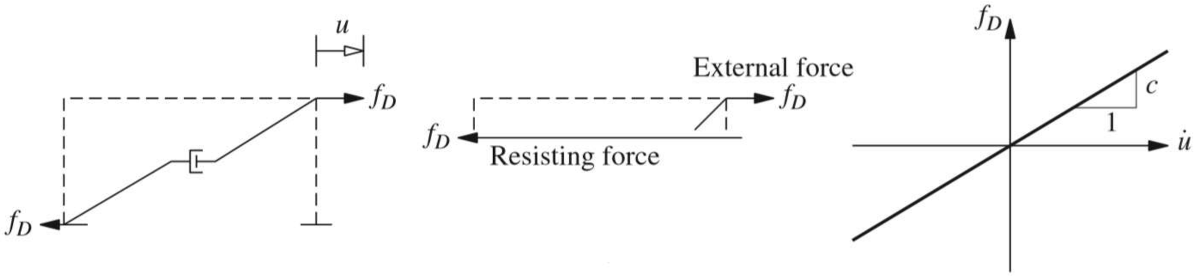

Figure 2.8: Viscous Damping [2]

$$fd = c\dot{u}$$ Equation 2.3.3.1

Where c is the viscous damping coefficient.

The viscous damper coefficient is intended to model the energy dissipation within the linear elastic limit of the structure. Over elastic range of deformations, the damping coefficient is determined from experiments may vary with the deformation amplitude.

Stress Level	Type and Condition of Structure	Damping Ratio (%)
Working stress, no more than about $\frac{1}{2}$ yield point	Welded steel, prestressed concrete, well-reinforced concrete (only slight cracking)	2–3
	Reinforced concrete with considerable cracking	3–5
	Bolted and/or riveted steel, wood structures with nailed or bolted joints	5–7
At or just below yield point	Welded steel, prestressed concrete (without complete loss in prestress)	5–7
	Prestressed concrete with no prestress left	7–10
	Reinforced concrete	7–10
	Bolted and/or riveted steel, wood structures with bolted joints	10–15
	Wood structures with nailed joints	15–20

Figure 2.9: Recommended damping value [2]

Rayleigh damping:

Rayleigh damping is known as one of the classical approaches for construction the damping matrix of the structure. Classical damping matrix is appropriate idealization if similar damping mechanism is distributed through the structure. In this approach the damping is defined as mass-proportional damping and stiffness-proportional damping (Figure 2.10). The damping matrix is defined as following [2] :

$$C = a0M + a1K \qquad \text{Equation 2.3.3.2}$$

Where a0 is coefficient proportional to the mass matrix and a1 is coefficient. proportional to the stiffness matrix. The damping ratio for nth mode is calculated by:

$$\zeta n = \frac{a0}{2\omega n} + \frac{a1}{2}\omega n \qquad \text{Equation 2.3.3.3}$$

The damping ratio ζi and ζj for i and j modes can be determined

$$\frac{1}{2}\begin{bmatrix} 1/\omega_i & \omega_i \\ 1/\omega_j & \omega_j \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix} = \begin{Bmatrix} \zeta_i \\ \zeta_j \end{Bmatrix}$$

If assumed the two modes will have the same ratio of damping the coefficient can be determined as following

$$a0 = \zeta \frac{2\,\omega i\,\omega j}{\omega i + \omega j} \quad and \quad a1 = \zeta \frac{2}{\omega i + \omega j} \qquad \text{Equation 2.3.3.4}$$

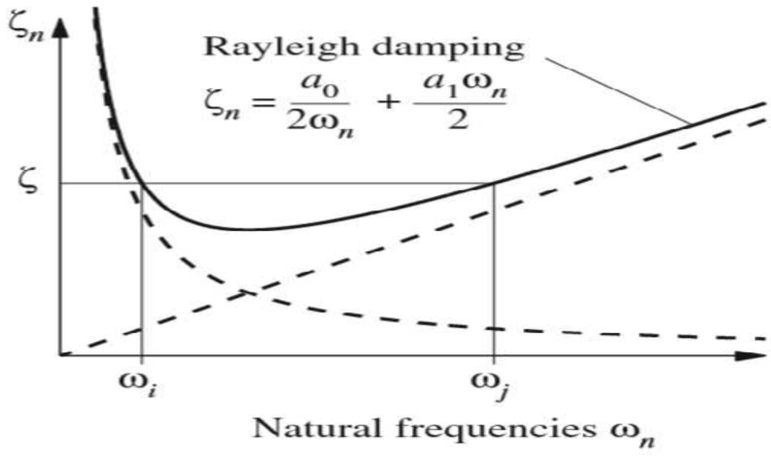

Figure 2.10: Rayleigh Damping [2]

2.4 Pushover Analysis

Pushover analysis (PA) is a nonlinear static analysis which is used in performance-based design. PA used to determine the strength capacity of the structure. Pushover analysis help us to know the nonlinear behaviour of the structure. Pushover analysis is carried under constant gravity loading with gradually increasing the lateral loading. The lateral load is assigned to the masses location in the structure and defined to be proportional to the multiplication of the mode shape (depending on the type of pushover analysis) and the mass of each story. The strength capacity of the structure can be presented by pushover curve (Performance curve) which is plotted by tracking the base share and the roof displacement (Figure 2.11). The nonlinearity of the structure can be represented by defining the concentrated plasticity region (Plastic hinges) or defining the element (material and section) nonlinearly (Finite element).

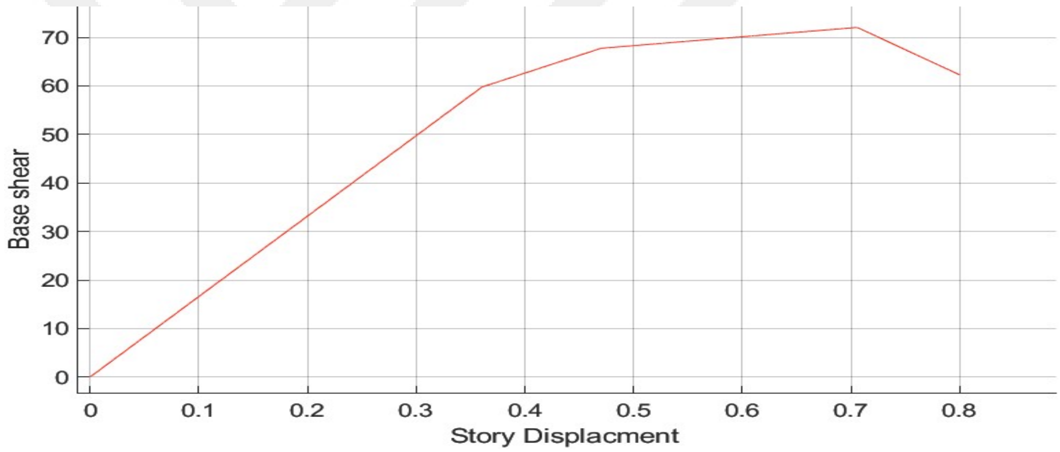

Figure 2.11: Pushover curve

Plastic Hinges:

Plastic hinges are zones in structure which has potential to be yielded and cracked. Plastic hinge model is simplest way to express the nonlinear behaviour. The hinges are defined by defining the force-displacement or moment-rotation relationship. In this approach, the nonlinear behaviour is generally occurred in the area where the plastic hinge is defined where the rest of the element is defined linearly. Plastic hinge is defined by zero length element. the plastic hinges are generally defined in the end of the beam or element columns (Figure 2.12) and if the element has long span the plastic hinges might be defined in the middle region of the span.

Ibarra-Krawinkler deterioration model:

There are many methods of modelling the plastic hinges and the Ibarra-Krawinkler deterioration model (Figure 2.12) is one of the most complex and realistic models to define the plastic hinges behaviours [3].

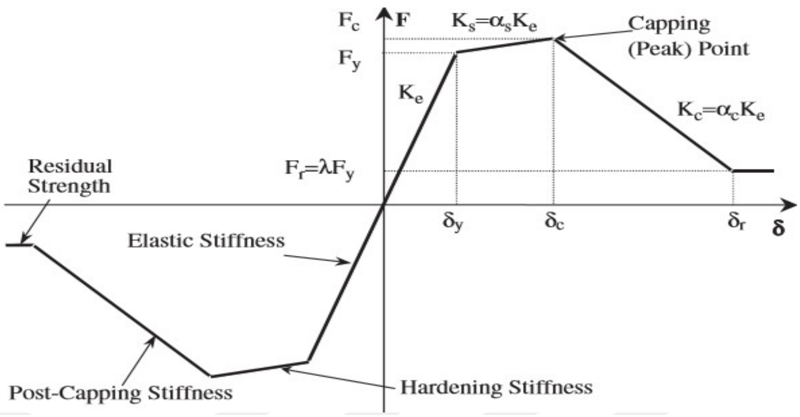

Figure 2.12: Ibarra-Krawinkler model [3]

The backbone curve is defined by the following properties:

- The yielding stress F_y where the elastic range in end and the plastic deformation occurs.
- The initial stiffness K_e which know is the stiffness from the unloading section to the yielding point.
- The strain hardening stiffness K_s where it is representing the slope between the yielding point and capping point it is defined as $K_s = \alpha_s K_e$ where α_s known as the stiffness ratio.
- The F_c and δ_c which represent the maximum force and the deformation at that force before the force starting to deteriorate. Where F_c can be calculated as following:

$$F_c = F_y + K_s(\delta_s - \delta_y)$$

- The post capping stiffness K_c which represents the slope between the capping point and the fracture. It expressed by $K_c = \alpha_c K_e$.
- Fraction of the yield strength F_r which represents the fraction of the yield strength of the component that is preserved once a given deterioration threshold is achieved. where the $F_r = \lambda F_y$

- The parameter α_s, δ_s, δ_c and α_c are obtained from the analytical prediction or from calibration of the hysteresis loop. Where δ_c/δ_s which known as the ductility capacity.

If no deterioration exists the backbone curve is defined by three parameters only which are the initial elastic stiffness K_e, yield strength F_y and the strain hardening stiffness K_s.

Description of hysteresis model without cyclic deterioration for bilinear model:

This model is based on the bilinear hysteretic model with strain hardening (Figure 2.13). These basic rules are preserved once post-capping and residual strength branches are included. However, it is necessary to introduce the 'strength limit' when the backbone curve includes a branch with negative slope. According to kinematic rules, the loading segment that starts at point 5 should continue up to intersect point 6'. However, this loading segment ends when it intersects the 'strength limit' at point 6. The limit corresponds to the strength of point 3, which is the smallest strength in the nonlinear range of the backbone curve in earlier cycles. If this condition were not established, the strength in the loading path could increase in later stages of deterioration [3].

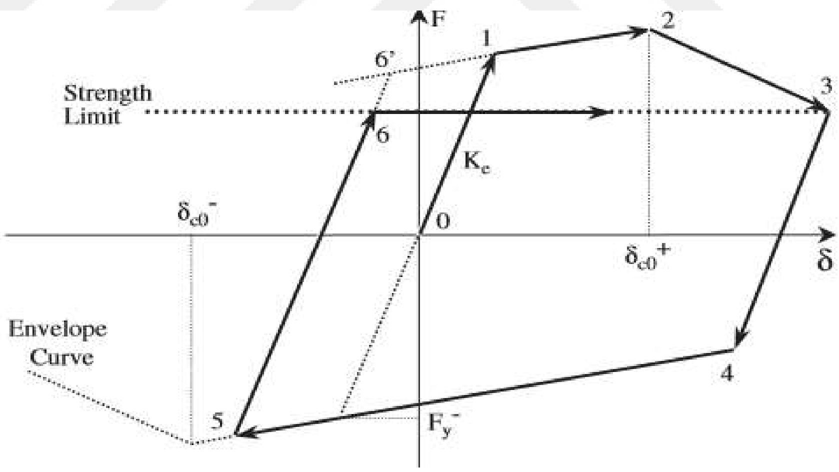

Figure 2.13: Bilinear material hysteresis model without cyclic deterioration [5]

Defining Bilinear material in OpenSEES:

The bilinear materials which used to define to the Plastic hinges is defined by Ibarra-Krawinkler deterioration model (Figure 2.14) as uniaxial materials:

"uniaxialMaterial Bilin $matTag $K0 $as_Plus $as_Neg $My_Plus $My_Neg $Lamda_S $Lamda_C $Lamda_A $Lamda_K $c_S $c_C $c_A $c_K $theta_p_Plus $theta_p_Neg $theta_pc_Plus $theta_pc_Neg $Res_Pos $Res_Neg $theta_u_Plus $theta_u_Neg $D_Plus $D_Neg <$nFactor>"

Table 2.1: Parameters of Bilinear materials in OpenSEES

$matTag	integer tag identifying material
$K0	elastic stiffness
$as_Plus	strain hardening ratio for positive loading direction
$as_Neg	strain hardening ratio for negative loading direction
$My_Plus	effective yield strength for positive loading direction
$My_Neg	effective yield strength for negative loading direction (negative value)
$Lamda_S	Cyclic deterioration parameter for strength deterioration
$Lamda_C	Cyclic deterioration parameter for post-capping strength deterioration
$Lamda_A	Cyclic deterioration parameter for acceleration reloading stiffness deterioration
$Lamda_K	Cyclic deterioration parameter for unloading stiffness deterioration
$c_S	rate of strength deterioration.
$c_C	rate of post-capping strength deterioration.
$c_A	rate of accelerated reloading deterioration.
$c_K	rate of unloading stiffness deterioration.
$theta_p_Plus	pre-capping rotation for positive loading direction (often noted as plastic rotation capacity)
$theta_p_Neg	pre-capping rotation for negative loading direction (often noted as plastic rotation capacity)
$theta_pc_Plus	post-capping rotation for positive loading direction
$theta_pc_Neg	post-capping rotation for negative loading direction
$Res_Pos	residual strength ratio for positive loading direction
$Res_Neg	residual strength ratio for negative loading direction
$theta_u_Plus	ultimate rotation capacity for positive loading direction
$theta_u_Neg	ultimate rotation capacity for negative loading direction
$D_Plus	rate of cyclic deterioration in the positive loading for symmetric hysteretic response use 1.0.
$D_Neg	rate of cyclic deterioration in the negative loading direction for symmetric hysteretic response use 1.0.
$nFactor	elastic stiffness amplification factor, mainly for use with concentrated plastic hinge elements.

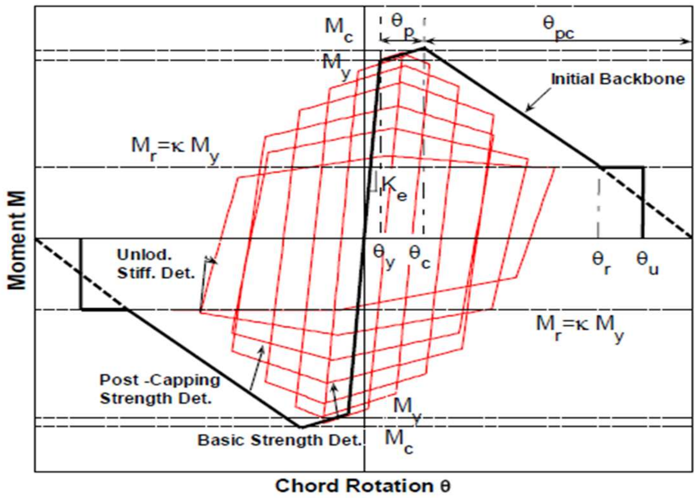

Figure 2.14: Modified Ibarra Krawinkler Deterioration Model [1]

2.5 Seismic isolation bearings

Seismic isolation bearing is used to protect the structure during extreme seismic loading. There are many types of seismic isolators such as elastomeric, single friction pendulum and triple friction pendulum bearings. The aim of using seismic isolators is to reduce the seismic demand on the structural members during earthquakes. The main objective of using these isolators is to elongate the natural period of the structure. Seismic isolators work efficiently in structure short to moderate natural vibration periods. Shifting the natural period to longer period (Figure 2.15) will result in an increase in displacements at the isolation level and reduction of the accelerations of the floors, keeping the structure in the elastic or near elastic range [6].

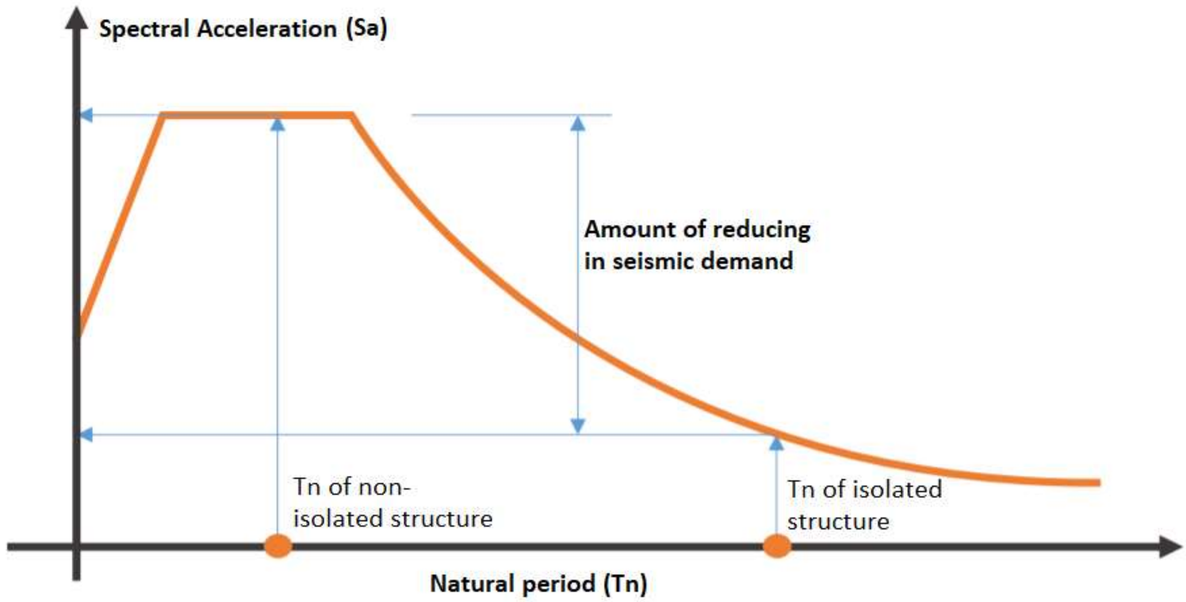

Figure 2.15: Working principles of seismic isolation [6]

2.5.1 Elastomeric bearing isolation:

Elastomeric isolators consist of rubber bonded with steel plates carrying the vertical loads (Figure 2.16). Elastomeric isolators allow the horizontal deformations. The isolators are installed between the super-structure and sub-structure. Elastomeric bearing is considered as a good solution for heavy structures located in low seismicity regions. It has the ability to resist the temperature changing and creep. There are two types of elastomeric bearing: Low-Damping rubber bearing (LDRB), High Damping rubber bearing (HDRB). Low damping rubber bearing is designed to have 2%-5% of critical damping. LDRB literal displacement is linearly due to the elastic stiffness of the rubber. LDRB works efficiently if the supplemental damper installed parallel to these bearings. High damping rubber bearings have damping ratios between 10% to 20% of the critical damping. HDRB is developed to eliminate the using of dampers so the rubber itself can provide the desired damping level. HDRB has non-linear behaviour of lateral force with displacement [4].

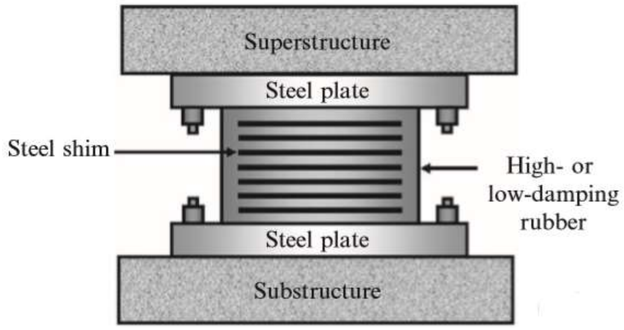

Figure 2.16: Elastomeric Bearing [4]

The important parameters in modelling and designing are horizontal stiffness, vertical stiffness and the axial loading to plot the Hysteresis loop (Figure 2.17).

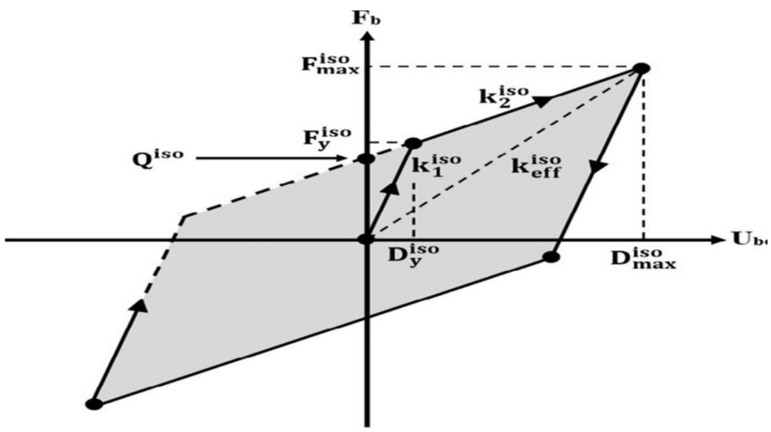

Figure 2.17: Hysteresis loop of elastomeric bearing [4]

Modelling the elastomeric using OpenSEES:

Elastomeric bearing has many types of modelling definition. OpenSEES uses the Bouc-Wen modelling (Figure 2.18). Elastomeric element is defined as unidirectional element for 2D models.

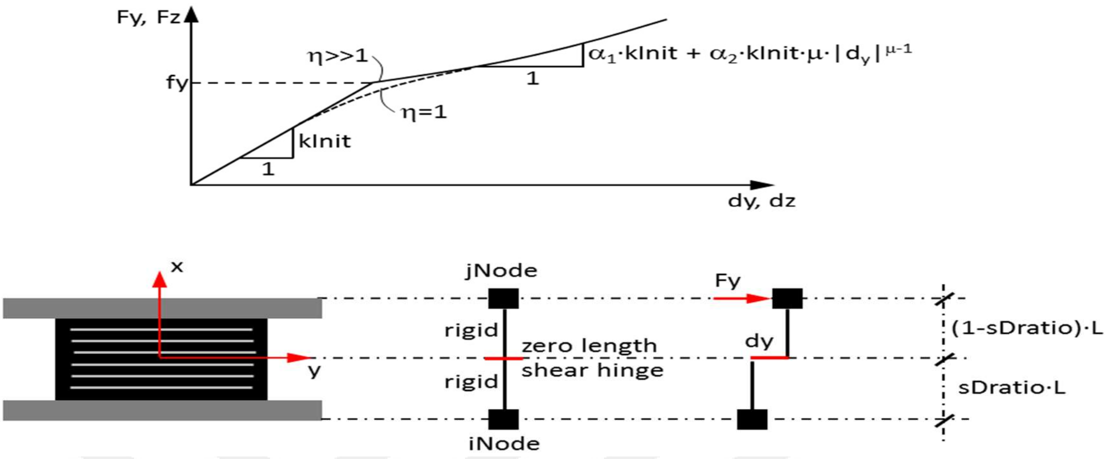

Figure 2.18: Elastomeric element by OpenSEES [1]

<u>Elastomeric element command for 2D modelling:</u>

element elastomericBearingBoucWen $eleTag $iNode $jNode $kInit $qd $alpha1 $alpha2 $mu $eta $beta $gamma -P $matTag -Mz $matTag <-shearDist $sDratio> <-doRayleigh> <-mass $m>

Table 2.2: Parameters used to define the Elastomeric bearing in OpenSEES

Parameter	Definition
eleTag	Unique tag for the element
iNode , jNode	Start and the end of the element
kInit	Initial stiffness
qd	Characteristic strength
alpha1	Post yield hardening linear ratio
alpha2	Post yield hardening non-linear ratio
mu	Exponent of non-linear hardening
eta	Yield exponent
beta, gamma	First and second hysteric parameters
P, Mz	Pre-defined material in axial and moment about z
sDratio	Shear distance coefficient
doRayleigh	Including Rayleigh damping
m	Mass of the element

2.5.2 Single friction pendulum bearing

Single friction pendulum bearing (SFP) is a type of bearing which uses a pendulum action to increase the period and friction to increase damping. SFP mainly consists of a steel slider and a curved surface (Figure 2.19). When the ground motion is weak the friction between slider and curved surface doesn't allow the slider to move therefore the structure behave like fixed base structure. For moderate and strong earthquakes, the slider can overcome the friction and moves thereby reducing the seismic demand on the structure element until the slider reaches the displacement (d) [8].

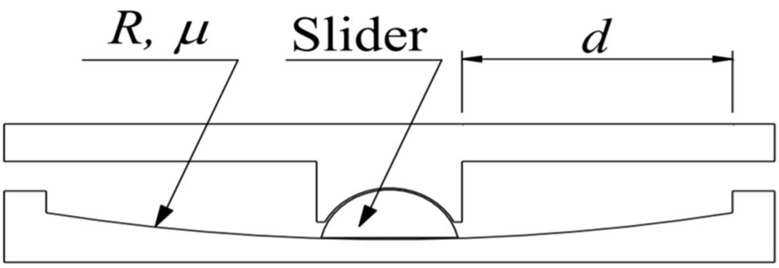

Figure 2.19: Section of single friction pendulum bearing [8]

The most important properties of modelling the SFP bearing is the radius of curvature (R) of the curved surface and the coefficient of friction between the slider and curved surface μ and the loading acting on the bearing. These properties are used to create the hysterics loop (Figure 2.20) [8].

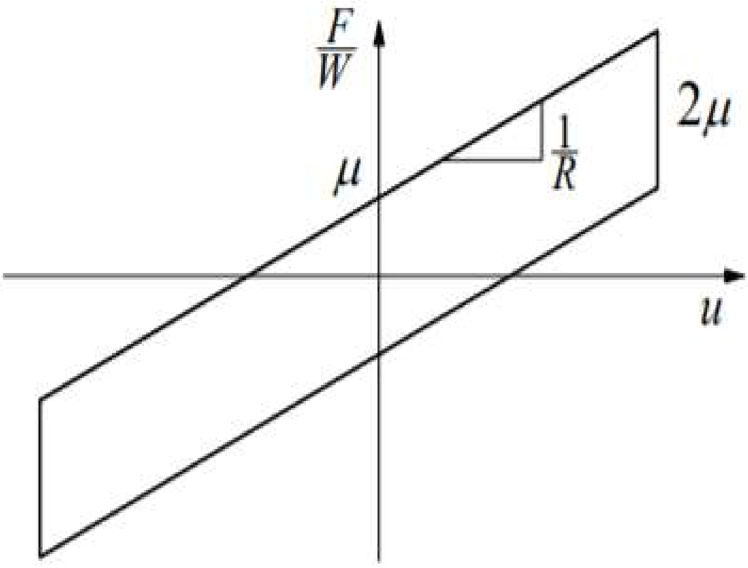

Figure 2.20: Hysteresis loop of SFP [8]

Modelling SFP bearing using OpenSEES:

In OpenSEES, singleFPBearing element object can be used to model SFP (Figure 2.21), which is defined by two nodes. The iNode represents the concave sliding surface and the jNode represents the articulated slider. The element can have zero length or the appropriate bearing height. The bearing has unidirectional (2D) friction properties (with post-yield stiffening due to the concave sliding surface) for the shear deformations, and force-deformation behaviours defined by Uniaxial Materials in the remaining two (2D)[1].

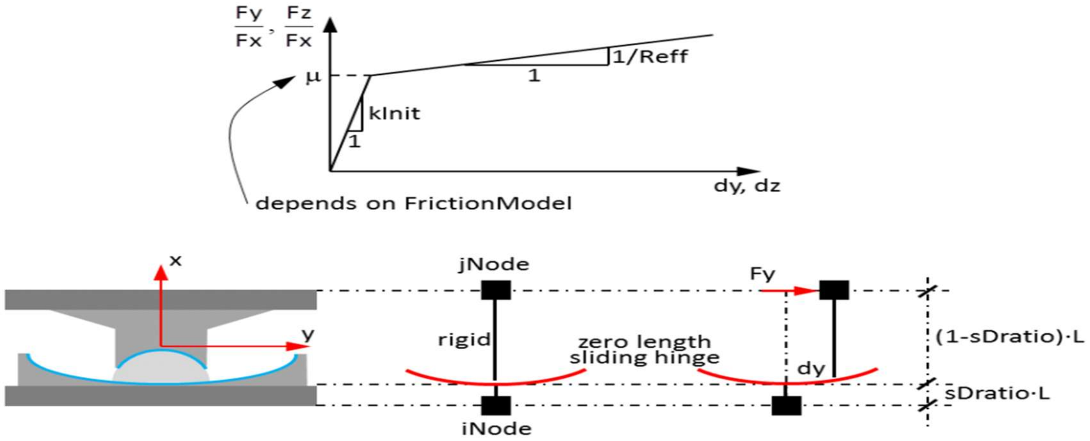

Figure 2.21: SFP element by OpenSEES [1]

<u>SFP element command for 2D modelling:</u>

element singleFPBearing $eleTag $iNode $jNode $frnMdlTag $Reff $kInit -P $matTag -Mz $matTag <-shearDist $sDratio> <-doRayleigh> <-mass $m>

Table 2.3: Parameters used to define the SFP bearing in OpenSEES

Parameter	Definition
eleTag	Unique tag for the element
iNode, jNode	Start and the end of the element
frnMdlTag	Pre-defined friction coefficient
Reff	Effective radius of curved surface
kInit	Initial stiffness
P, Mz	Pre-defined material in axial and moment about z
sDratio	Shear distance coefficient
doRayleigh	Including Rayleigh damping
m	Mass of the element

2.5.3 Triple friction pendulum bearing

Triple friction pendulum bearing (TFP) consists of three pendulums, two curved surfaces separated by articulated slider. The properties of each pendulum are chosen to make each pendulum active at different earthquake strength levels. The inner pendulum (slider) is free to slide along two inner spherical surfaces. The main objective of the slider to reduce the peak of ground acceleration. The two concaves curved surfaces are more independent pendulum. The first curved surface is designed to minimize the share force acting on structure while the other designed to minimize the isolator displacement during strong earthquakes (Figure 2.22).

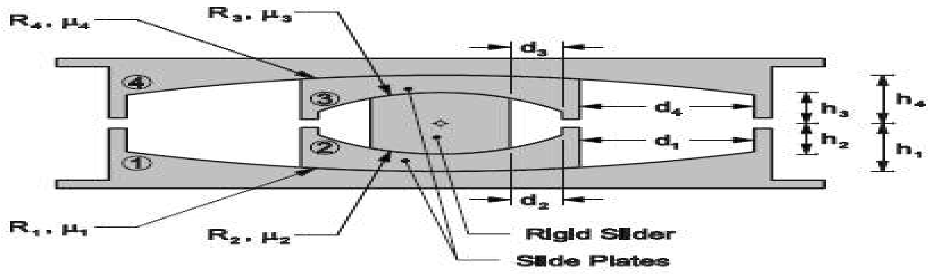

Figure 2.22: TFP element section [7]

The most important parameters to model and design the TFP are the stiffnesses of pendulums, friction coefficient of each pendulum and its effective radiuses. The hysteresis loop is consisting of five regimes (Figure 2.23)

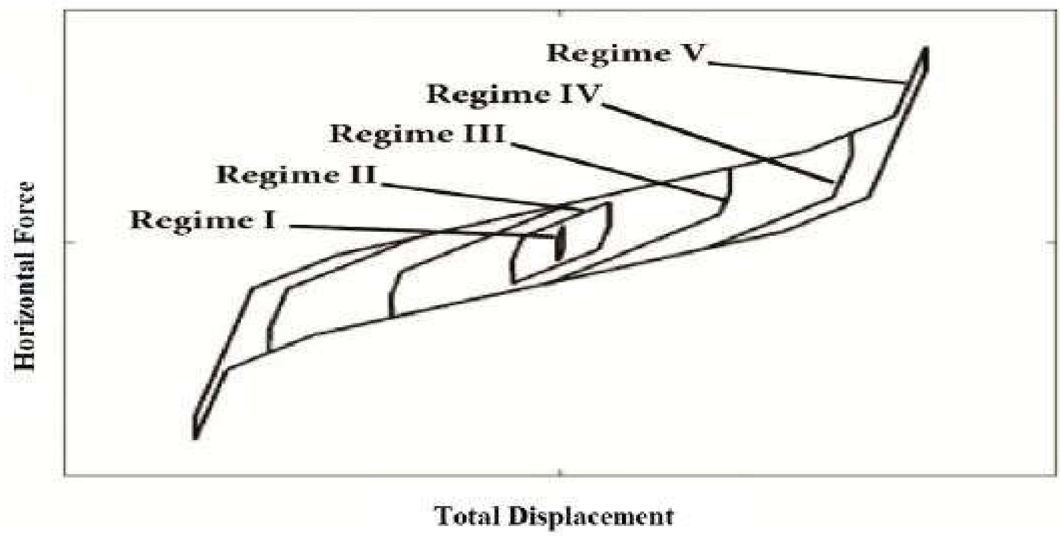

Figure 2.23: TFP hysteresis loop [7]

Modelling TFP bearing using OpenSEES:

In order to model TFP in OpenSEES we can use a Triple Friction Pendulum Bearing element object. The element is a 3-dimensional element. The element accounts for the vertical-horizontal coupling and the bidirectional coupling in horizontal behaviour. The friction coefficient model is a general model that accounts for the variation of friction coefficient on velocity and vertical force (Figure 2.24) [1].

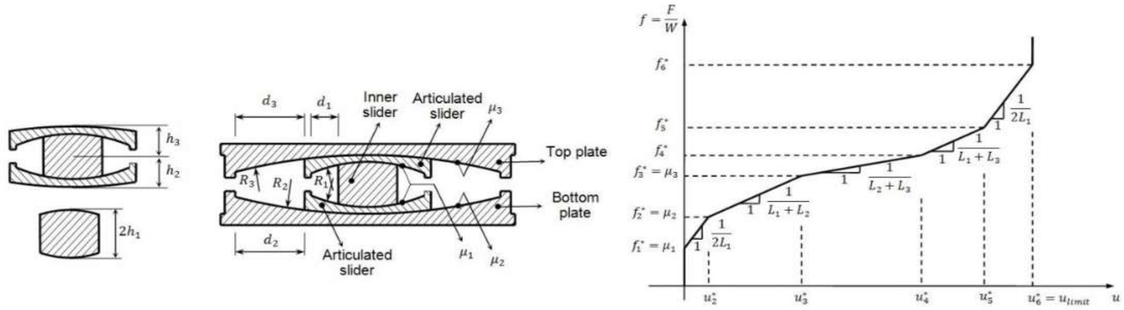

Figure 2.24:TFP element in OpenSEES [1]

TFP element command:

element TripleFrictionPendulum $eleTag $iNode $jNode $frnTag1 $frnTag2 $frnTag3 $vertMatTag $rotZMatTag $rotXMatTag $rotYMatTag $L1 $L2 $L3 $d1 $d2 $d3 $W $uy $kvt $minFv $tol

Table 2.4: Parameters to define TFP bearing in OpenSEES

Parameter	Definition
eleTag	Unique tag for the element
iNode, jNode	Start and the end of the element
frnTag1, frnTag2 and frnTag3	Pre-defined friction coefficients
VerMatTag, rotZMatTag	Pre-defined material in vertical and moment about z
rotXMatTag, rotYMatTag	Pre-defined material in rotational moment about x and Y
L1, L2 and L3	Effective radiuses of pendulums
d1, d2 and d3	Displacement of each pendulum
W	Axial force (used only for first iteration)
uy	Literal displacement where sliding of bearing start
kvt	Tension stiffness of the bearing
minFv	Minimum vertical compression in the bearing
tol	relative tolerance for checking the convergence

3. Overview of the Graphical User Interface

3.1 App overview

The apps have been designed by MATLAB APP Designer processor. The apps consist of three main steps, defining the model, running the analysis using OpenSEES solver and finally plotting the result. The user needs to define the model using editable fields and the MATLAB processor will generate the code according to pre-defined functions the user has to run the code using the OpenSEES solver. The result files will be generated directly after running the analysis in the same the location of OpenSEES in the PC.

3.2 Linear time history analysis app guide

Defining the geometry of the model:

In the first step of using the program the user has to define the model geometrically, materials, elements and loadings. This step can be done using the frame properties and loading and isolators properties tabs.

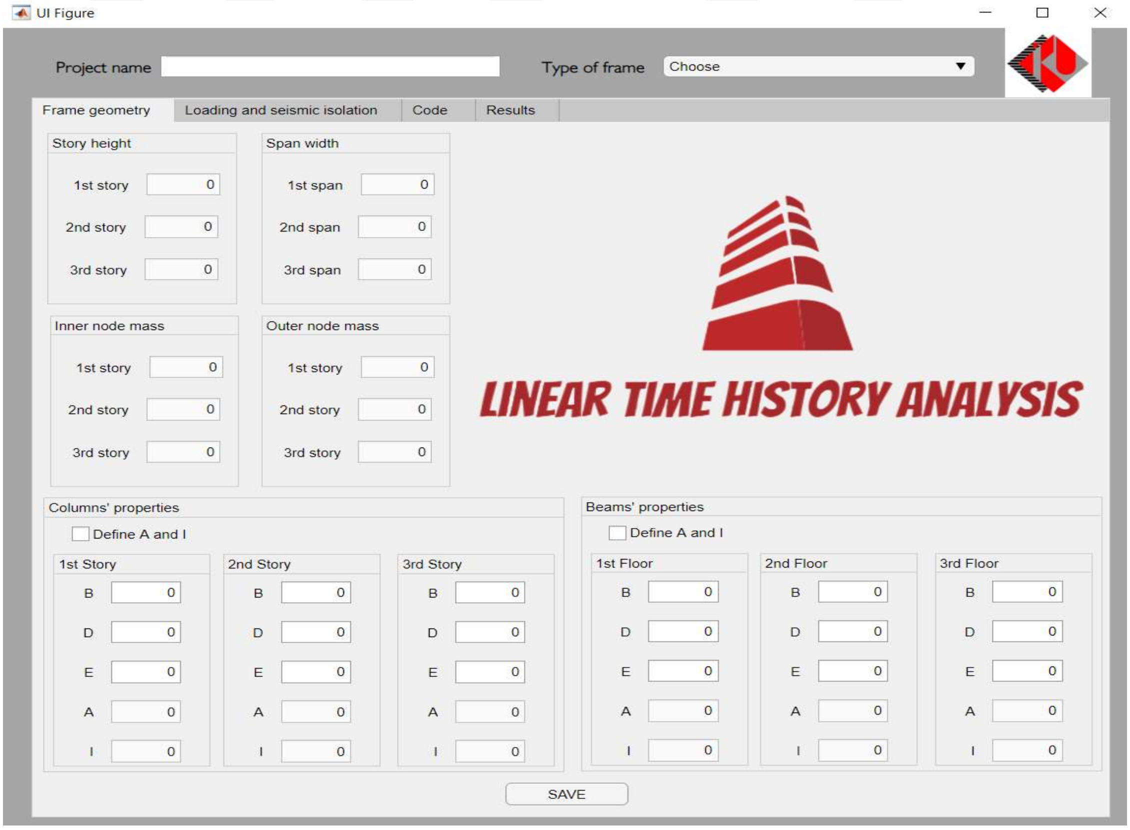

Figure 3.1: Main interface of the linear time history analysis app

From the Frame properties tab the name of project can be defined and will be used as file name when the result generated.

Figure 3.2: Project name editable filed

From the same tab the type of frame needs to be define. Form the type of frame drop down menu the geometry frame can be selected form pre-defined types.

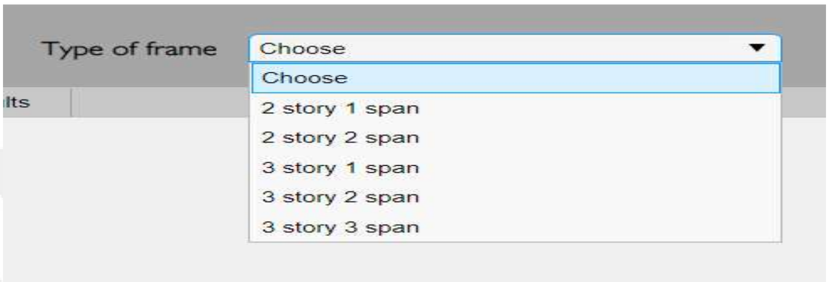

Figure 3.3: Type of frame dropdown menu

The height of each floor and the width of each bay width is defined in the first tab from the height of floor and width of bay width panels.

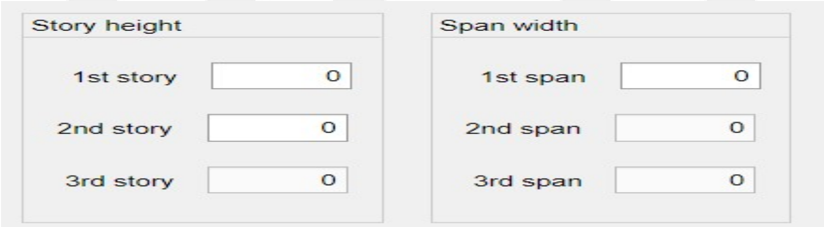

Figure 3.4: Story height and span width editable fields

To assign mass for each node we use the inner masses and outer masses from the first tab. the app will assign the mass to the node depending on the node location e.g. the border node in the first floor will take the defined mass in the editable field (1'st) in outer mass panel.

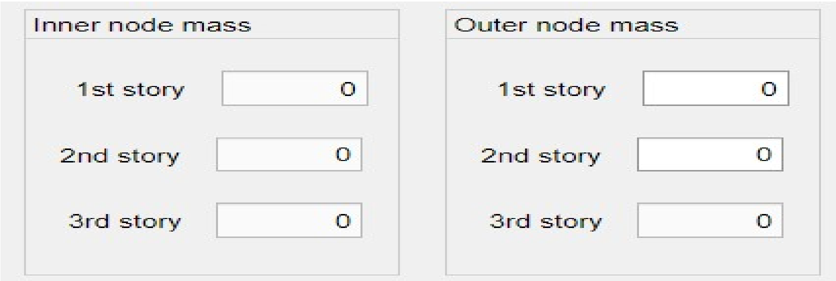

Figure 3.5: Nodes masses panels

As defining the model, the user has to define the elements' sections geometry. The program is capable with linear analysis; the user has to define the modulus of elasticity, area and moment of inertia in Z direction. Also, the user has the ability to define the depth and height of section and the app will calculate the area and the moment of inertia.

*Note: the app will consider the moment of inertia Iz as following:

$$Iz = \frac{b * d^3}{12}$$

After defining these steps, the user has to click on save button to save these properties.

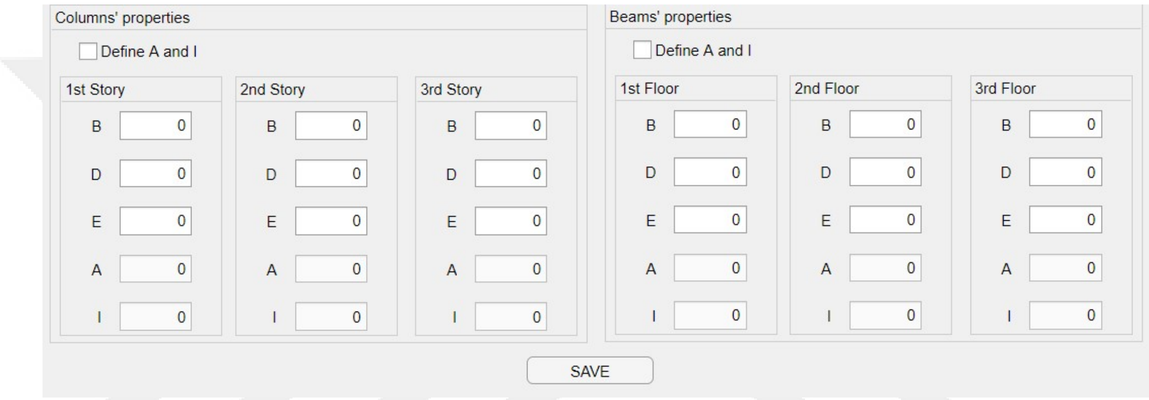

Figure 3.6:Columns and beams properties editable field

Defining the loading and damping:

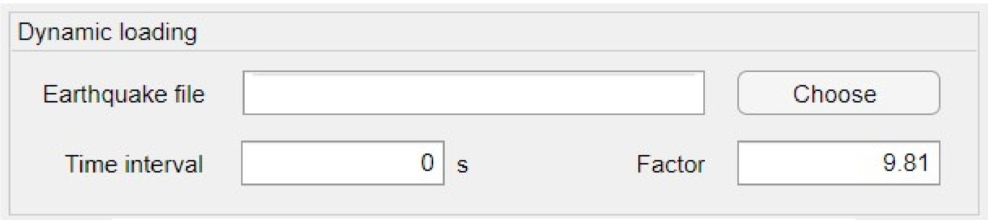

Figure 3.7: Dynamic loading panel

The app is capable only with one path for dynamic loading for time history analysis. The dynamic loading is defined by choosing the ground acceleration record file (.AT2) which can be downloaded from the "PEER Ground motion database" website. The time interval between two records must be defined.

*Note: The accleration record file must be saved in the same location of OpenSEES.exe on the PC.

*Note: since the OpenSEES can deal only with numbers, the user has to edit the accleration record file and delete the lines of the describation lines from the file.

For applying the gravity loading. In the same tab the user can define the gravity loading in the distributed loading on floor and concentrated loading on the columns at each floor.

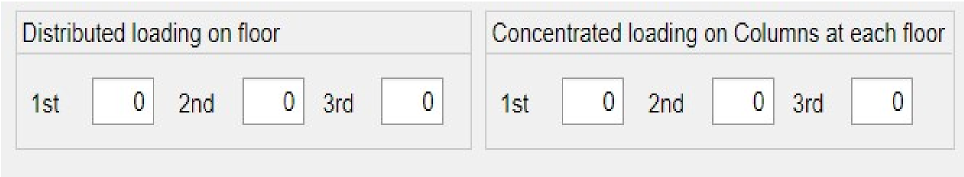

Figure 3.8: Gravity loading panels

To define the damping, the app is using Rayleigh damping to construct the damping matrix. The user needs to define the damping ratio ζ and the first and second natural periods which obtained from the modal analysis T1 and T2. The app will calculate the damping coefficient (a_0 and a_1).

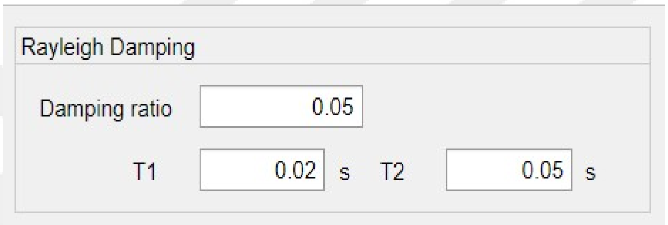

Figure 3.9: Rayleigh damping panel

For defining the parameter of transient analysis, the user can define the number of points and the time interval between two points for “Loading and seismic isolation” tab.

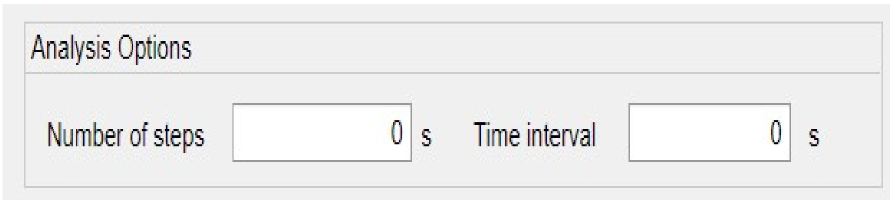

Figure 3.10:Analysis panel

Defining the seismic isolation bearings:

From the “Loading and Seismic isolation” tab the user can define the type of isolation bearing and its properties. The user has to define the type of bearing first.

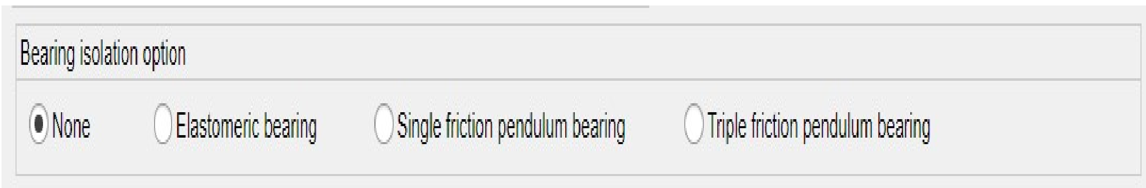

Figure 3.11: Bearing isolation option buttons

Elastomeric bearing:

After choosing the "Elastomeric bearing" button the new panel will appear, and the user has to define the properties of bearing material and its geometry. The elastomeric properties are defined in the section 2.5.1.

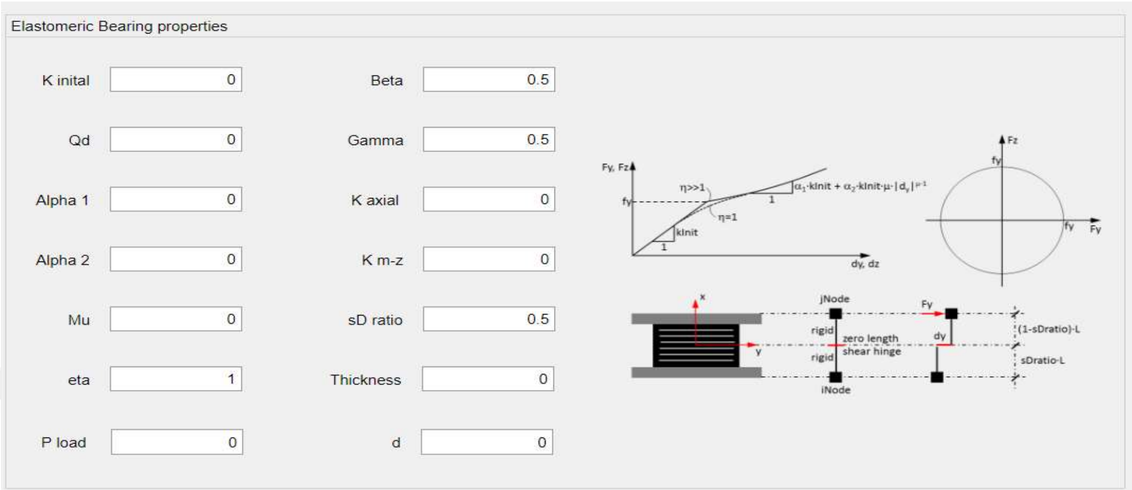

Figure 3.12: Elastomeric bearing panel

Single friction pendulum bearing:

After choosing the single friction pendulum bearing button, new panel will appear to allow the user to define the bearing properties. SFP bearing properties are defined previously in section 2.5.2.

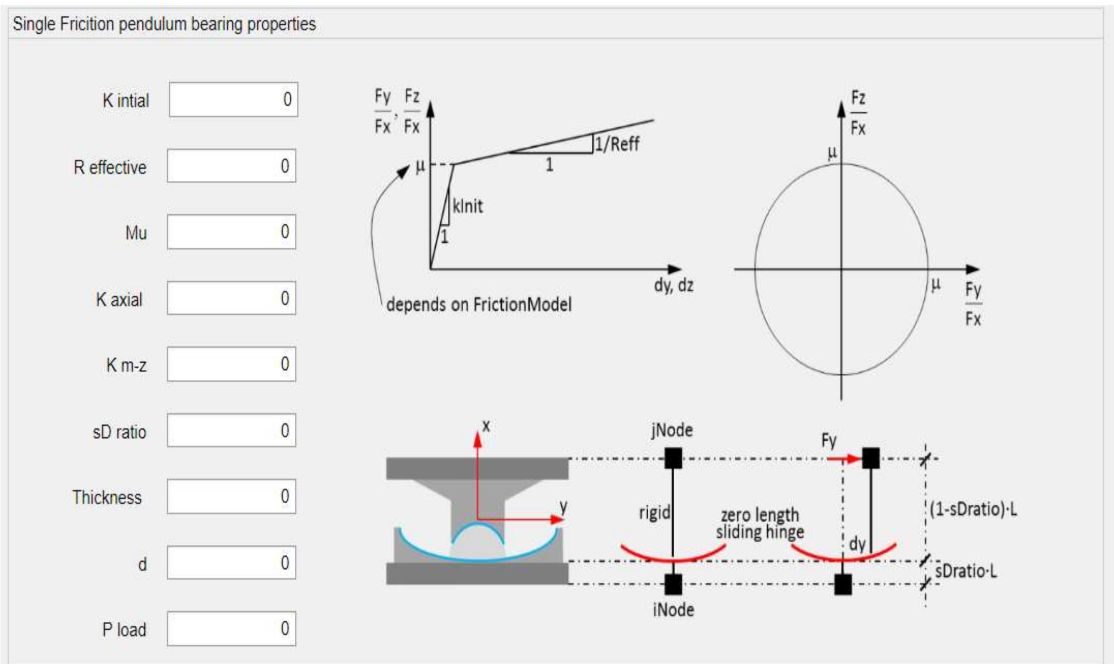

Figure 3.13: Single friction pendulum bearing panel

Triple friction pendulum bearing:

To assign the triple friction pendulum bearing. The user has to choose the Triple friction pendulum bearing, the new panel will appear, and the user has to define the bearing's properties. TFP bearing properties are defined in the section 2.5.3.

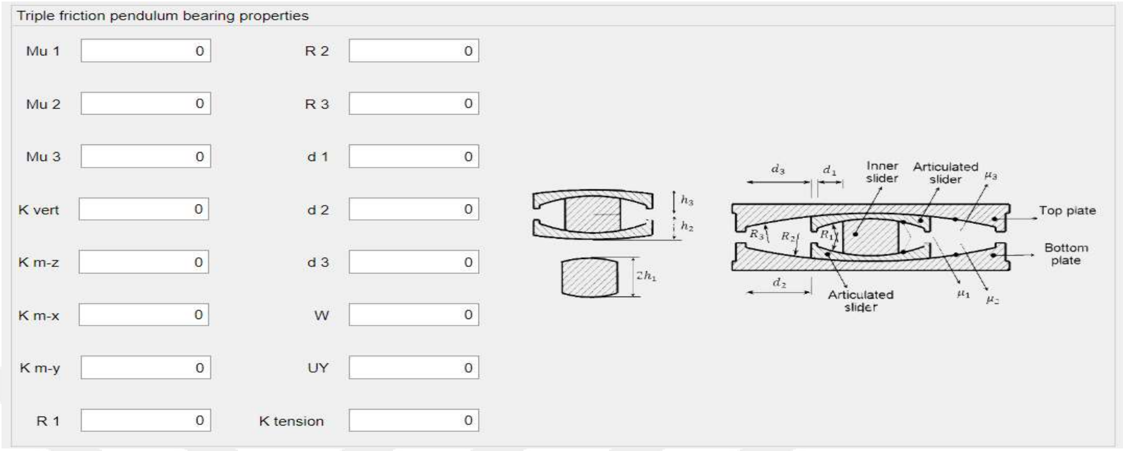

Figure 3.14:Triple friction pendulum bearing panel

Generating the code and running the analysis:

The code can be generated from the "Code" tab. The user has only to choose the type of analysis "Modal analysis", "Time History analysis" or "Bearing test" by clicking on the button and the code will be generated in the text box. The user has to copy and run the code in the OpenSEES platform, and the result files will be generated in the same location of OpenSEES in the PC.

Figure 3.15: Code tab

Displaying the results:

In the last tab "Results" tab the user can display and plot the result. The app has the ability to plot the story displacement, story acceleration, relative drift of story and the base shear various the time. Also, the user can plot the hysteresis loops for bearing and the load control loop for bearing. The user needs just to choose the type of result then choose the appropriate files and click on display and the tab will be displayed in the axes.

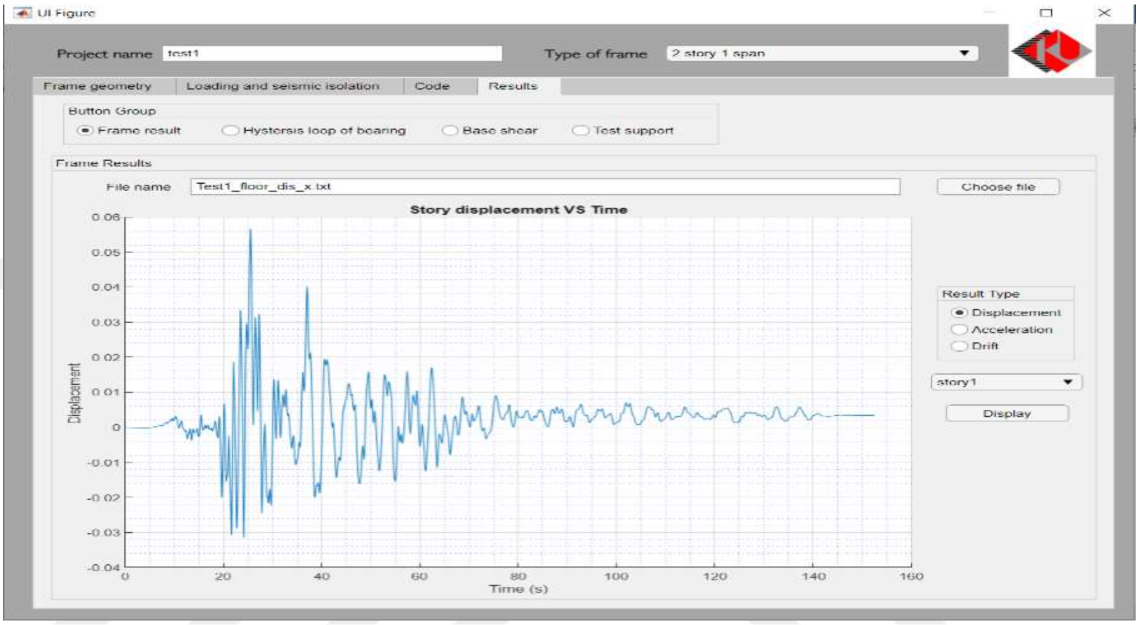

Figure 3.16: Result tab

Linear time history app's limitations:

- The project name must contain only letters, numbers or (_). Be careful don't use space even in the end of the name otherwise no result files will generate.
- The app capable to analyze only plane frames (X-Z plane frames).
- The app is capable only with 5 types of frames.
- The app is unitless, so the user needs to keep attention to the unit system to be used.
- The app is capable only running unidirectional input.
- The elements defined with no mass density, so the app will consider the lumped mass only as mass source.
- The damping can be defined only by Rayleigh damping method.
- The app is capable only with symmetric frames (loading and dimension symmetric)
- The columns and beams in each floor must have the same material and size.

3.3 Pushover analysis app guide

The app is designed to serve three type of frames to be analysed for pushover analysis. The app has three main stages start with modelling the frame, generating and running the code and finally plot the results which are the moment curvature diagram for hinges and the pushover curve.

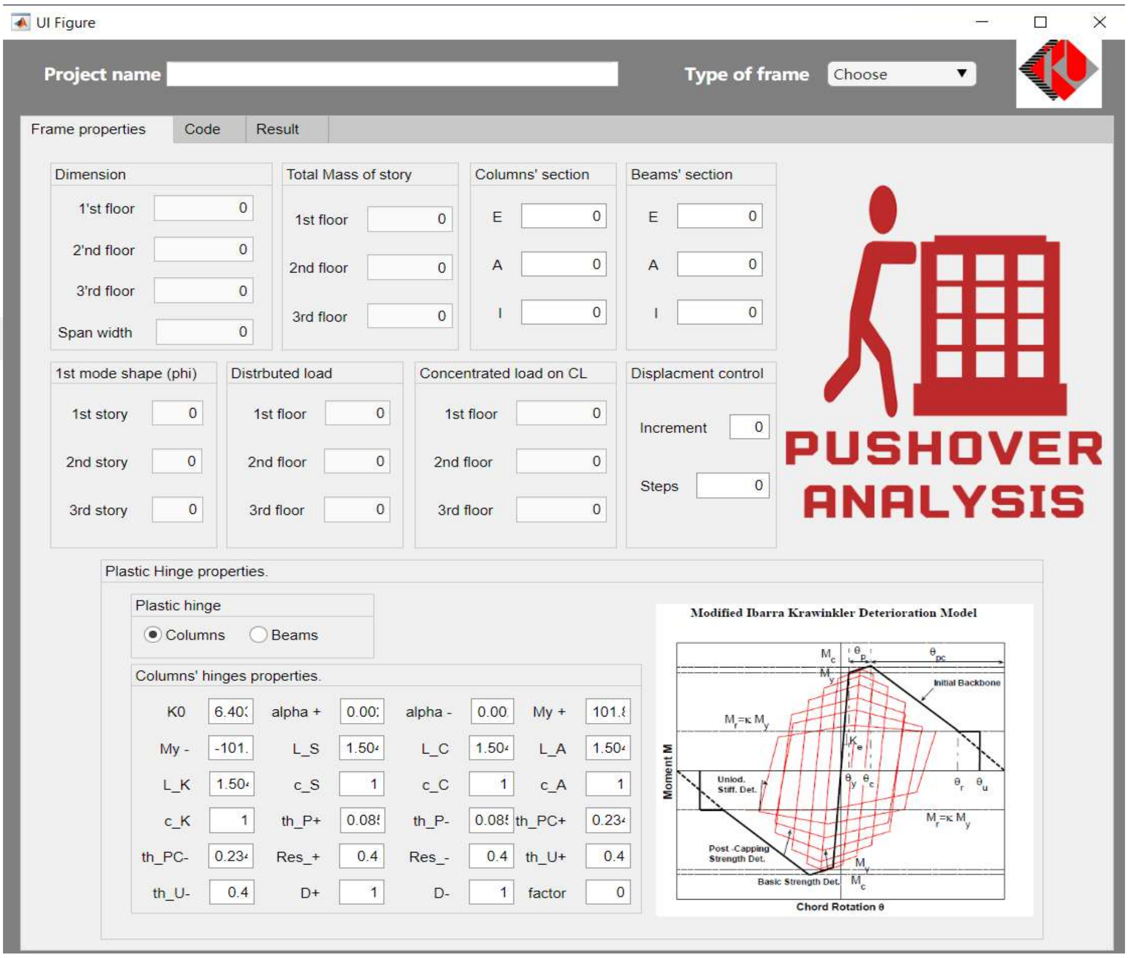

Figure 3.17:Pushover Analysis app's main interface

Defining the model start by defining the type of frame from the dropdown menu and name the project.

Figure 3.18: Project name editable field and type of frame dropdown menu

Defining the geometry of the structure, materials and loading can be done by the editable fields in the first tab of the app. The app is unitless which means the user can defined the structure with any unit system.

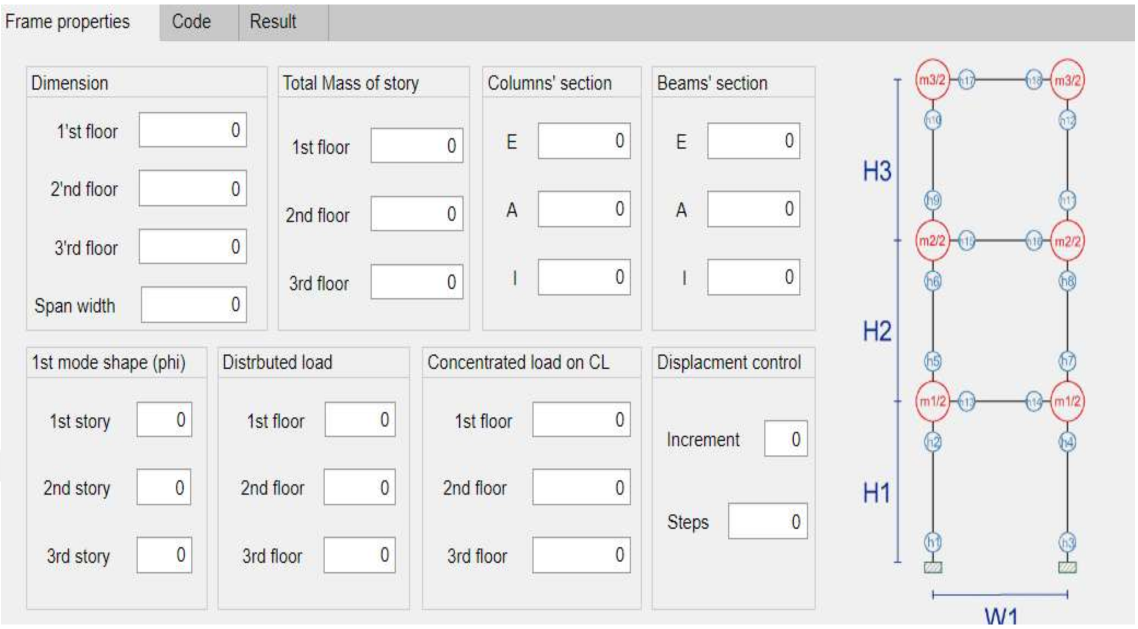

Figure 3.19: Editable fields used for defining the frame

To define the hinges properties the user has to define it by the editable fields in the first tab on the Plastic hinges properties panel. The plastic hinges are defined by the modified Ibarra Krawinkler deterioration model. which is defined in section 2.4.

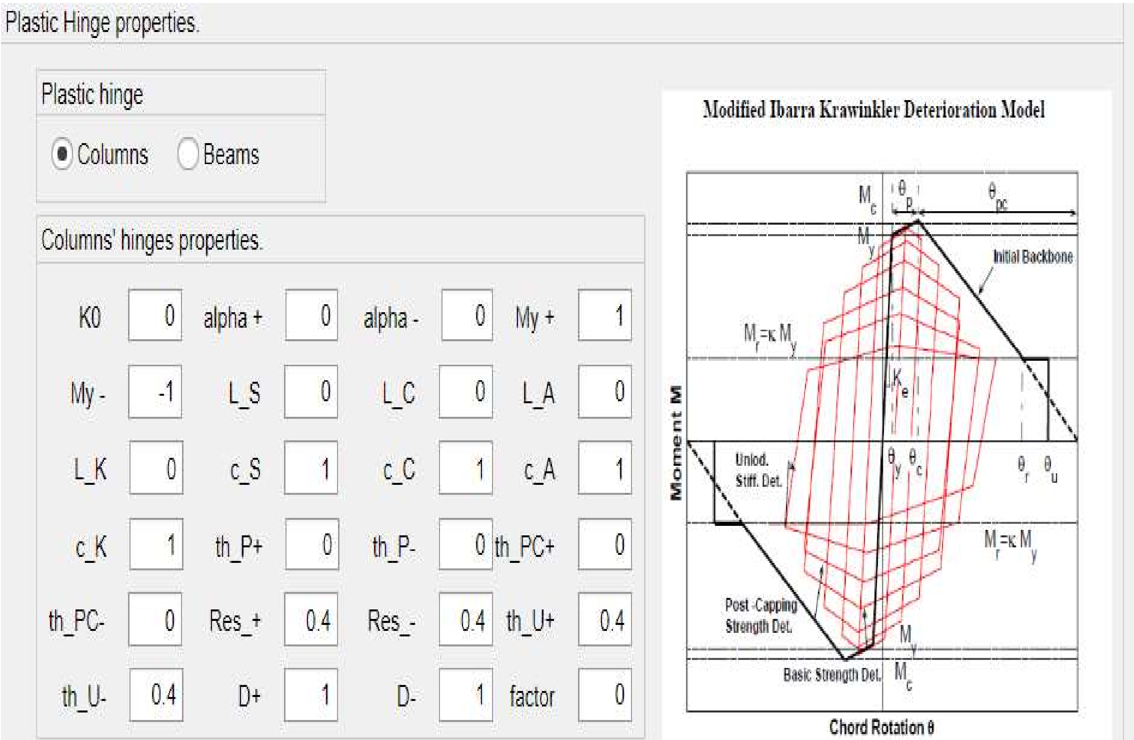

Figure 3.20: Plastic hinges properties panel

The next step is generating the code from the second tab then run the analysis using the OpenSEES solver. The code can be generated from the second tab “Code tab”. The modal analysis is used to define the mode shape of the frame which used in the pushover analysis. And the pushover analysis can be generated in the same tab and after running the code by the OpenSEES solver and the result file will be generated in the same location of OpenSEES.exe at the PC.

Figure 3.21: Code tab

After running the code, the user can plot the result from the result tab. For the moment-curvature diagram the user has to import the hinge deformation and hinge reaction and choose the hinge number. For plotting the pushover curve the base shear result and roof displacement files need to be imported.

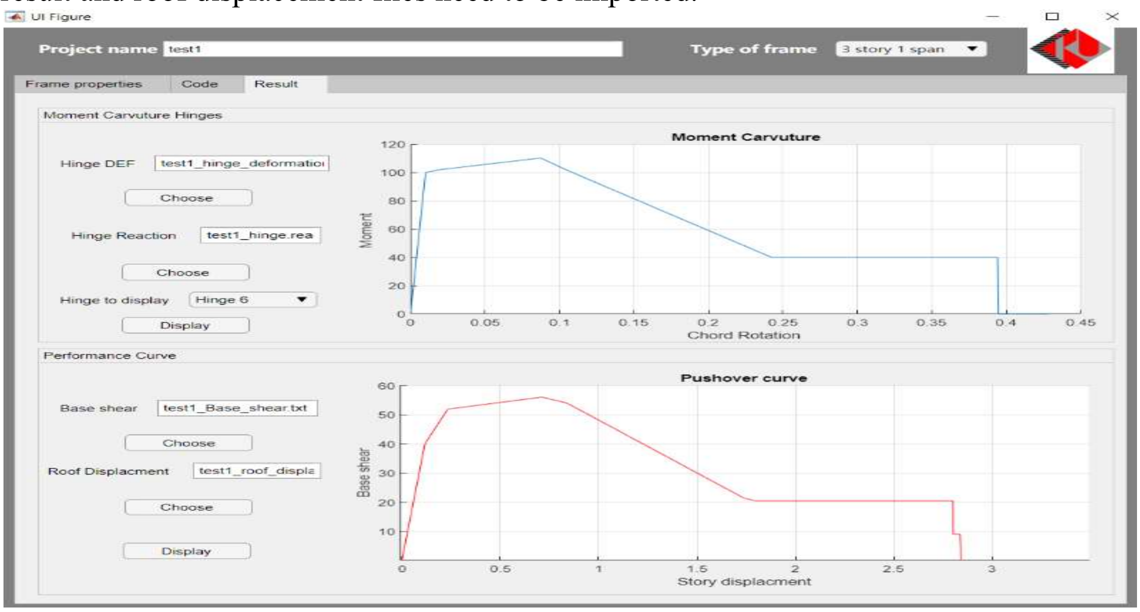

Figure 3.22: Result tab

Pushover analysis app’s limitations:

- The project name must contain only letters, numbers or (_). Be careful don’t use space even in the end of the name otherwise no result files will generate.
- The app is capable with plane frames only (X-Z plane frames).
- The app is capable with only three type of frames.
- The app is unitless, so the user needs to keep attention for unit system.
- The hinges are assigned at the end of members no space between end of member and the hinge.
- The hinge can be defined only by the modified Ibarra Krawinkler deterioration model.
- The hinges will be defined only by moment-curvature relationship only the app will not consider the force-displacement.
- The displacement controlled at the roof joint will be calculated as (increment displacement × Number of steps).

3.4 Verification examples

Modal analysis verification:

Two story with two spans frame has been analysed for modal analysis. The properties of the frame as following.

Table 3.1: Modal Analysis example frame properties

Story number	1st story	2nd story
Height (m)	3.4	3.4
Columns section (mm*mm)	150*150	100*100
Beams section (mm*mm)	100*100	
Beams and columns Material properties	E=200GPa	E=200GPa
Outer nodes mass X-direction (kg)	3000	2000
Inner nodes mass X-direction (kg)	2000	1500
Span number	1st span	2nd span
Width of span (m)	6	6

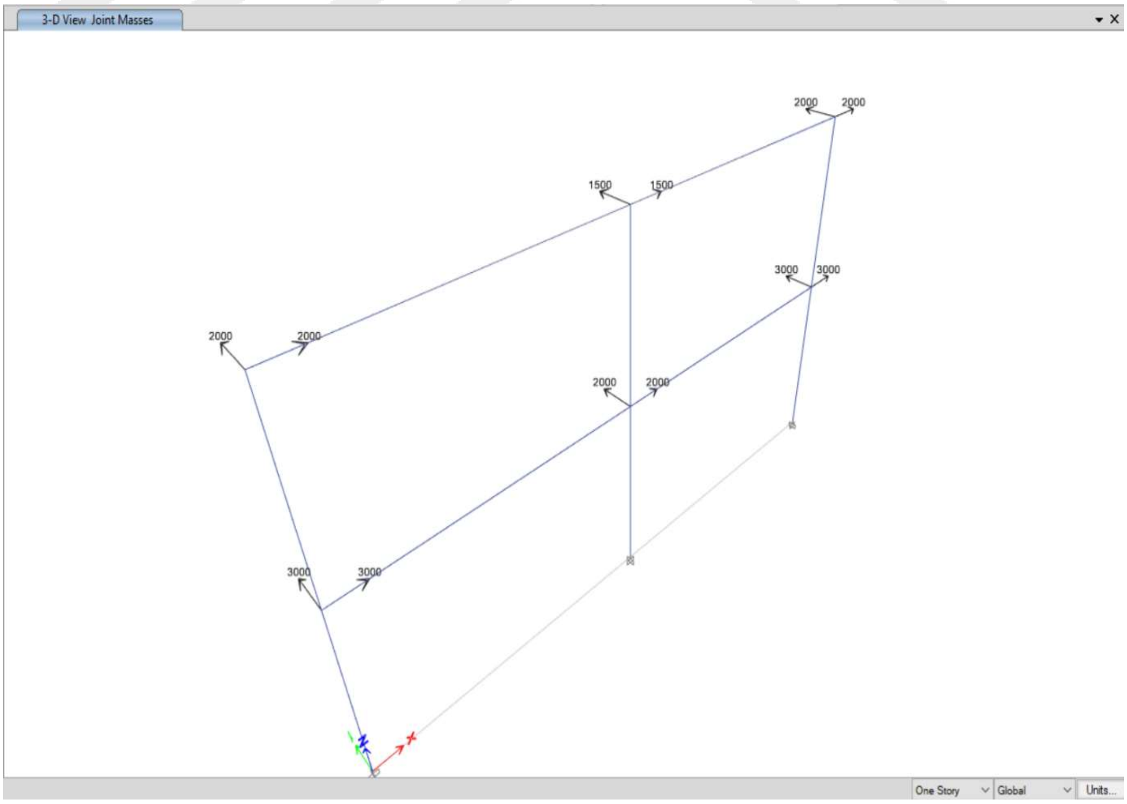

Figure 3.23: Verification example 1 frame ETABS interface

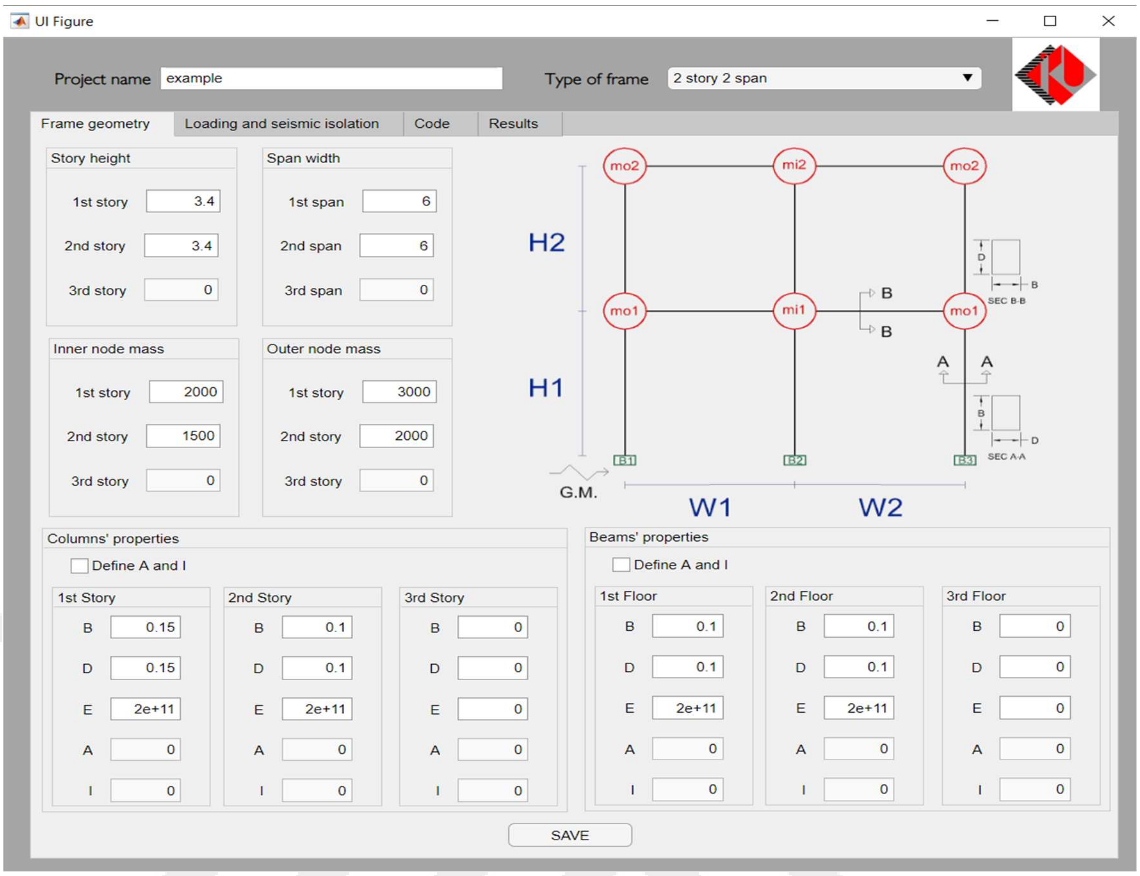

Figure 3.24: Verification example 1 time history analysis app's interface

The frame has been analysed in Eigen analysis by OpenSEES and Etabs software. The result which found is shown in the following table:

Table 3.2: Comparison of the results of the example 1

Mode number	Etabs				OpenSEES			
	Natural period (s)	Story number	Mode shape	Normalized mode shape	Natural period (s)	Story number	Mode shape	Normalized mode shape
1st	0.71	1st	-0.119	0.296	0.709	1st	-0.296	0.296
		2nd	-0.402	1		2nd	-1	1
2nd	0.224	1st	0.333	-2.378	0.2238	1st	1	2.331
		2nd	-0.143	1		2nd	-0.429	1

Linear time history analysis verification:

Three story one span frame has been analyzed for gravity and dynamic loading by OpenSEES and ETABS software. The properties of the frame as following.

* The modulus of elasticity E = 30 GPa.

* The Rayleigh damping coefficient: a_0 = 0.636 and a_1 = 0.0026

* Loading path: RSN1147_KOCAELI_ATS090. The dynamic analysis done for 30500 points was time interval 0.005.

Table 3.3: Time History analysis verification example frame properties

Story number	1^{st}	2^{nd}	3^{rd}
Height (m)	4	3.4	3.4
Span width (m)	5		
Columns size (mm×mm)	300 × 300	300 × 300	300 × 300
Beams size (mm×mm)	200 × 200	200 × 200	200 × 200
Joint loading (kN)	2	2	2
Distributed (kN/m)	3	3	3
Lumped mass (ton)	3.5	2.5	2

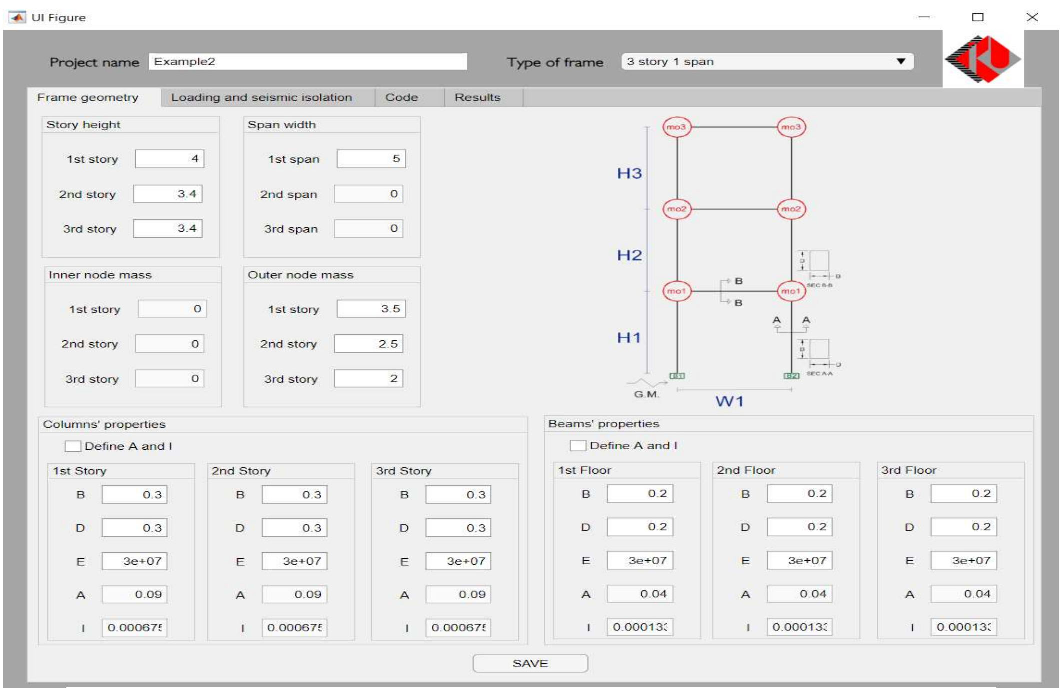

Figure 3.25: Verification example 2 interface of time history analysis app

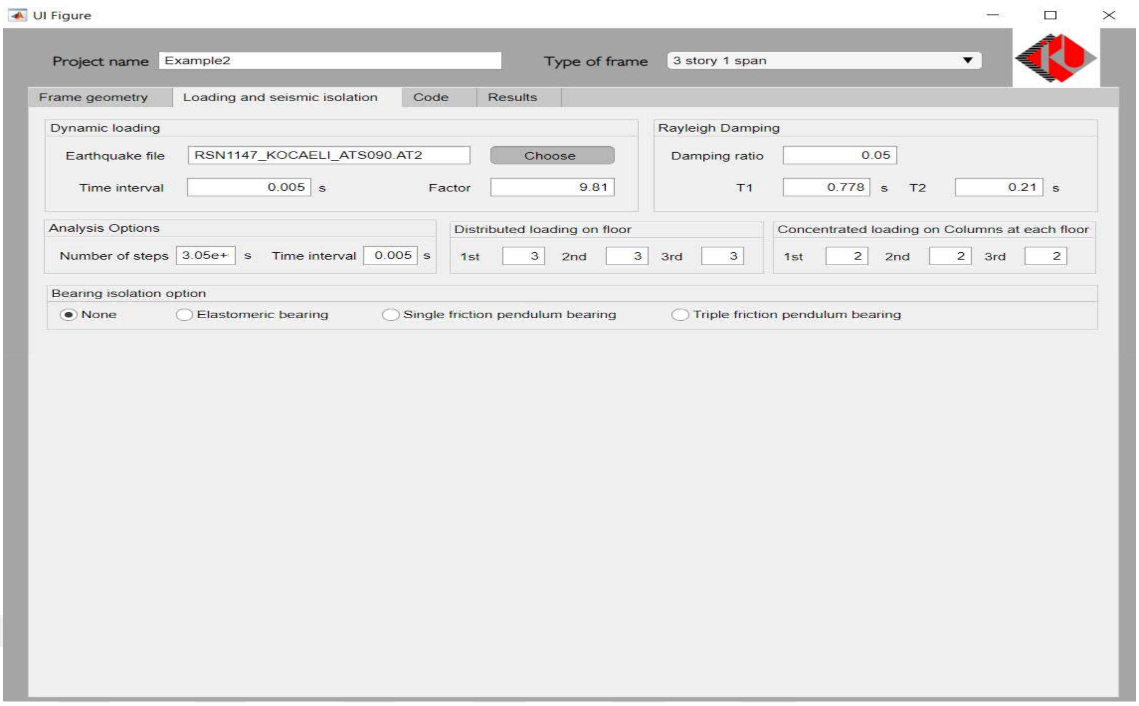

Figure 3.26: Verification example 2 loading tab

Results:

Table 3.4: Comparison of the results of the example 2

Result	OpenSEES	ETABS
Max displacement 1st floor	0.02705 mm, at 27.965s	0.02705 mm, at 27.965s
Max displacement 2nd floor	0.06442 mm, at 27.97s	0.06404 mm, at 27.97s
Max displacement 3rd floor	0.09536 mm, at 27.97s	0.09482 mm, at 27.97s
Max acceleration 1st floor	-1.725 mm/s^2 at 27.975s	-1.717 mm/s^2 at 27.975s
Max acceleration 2nd floor	-4.044 mm/s^2 at 27.98s	-4.0267 mm/s^2 at 27.98s
Max acceleration 3rd floor	-6.007mm/s^2 at 27.99s	-5.985 mm/s^2 at 27.99s
Max base shear force	58.777 kN at 27.535s	58.775 kN at 27.525s
Base reaction gravity analysis (Fx)	0.9578 kN	0.954 kN
Base reaction gravity analysis (Fy)	28.5 kN	28.5 kN

3.5 Case Studies

1st case: In this case the 3-story frame with 3 spans has been analyzed for time history analysis. First model was with no seismic isolation bearings while the second model was single friction pendulum bearing assigned. the modelling properties shown below:

* The Modulus of elasticity of element E = 30MPa.
* Rayleigh damping coefficients. $a_0 = 0.8123$ and $a_1 = 0.0022314$.
* Loading path: RSN1157_KOCAELI_CNA090.
* The analysis done for 30000 points with time interval 0.005s.

Table 3.5: Case study 1 frame properties

Story number	1st	2nd	3rd
Height (m)	4	3.5	3.5
Inner joint mass (ton)	5	4	3
Outer joint mass (ton)	2.5	2	1.5
Columns size (m×m)	0.3×0.3		
Beams size (m×m)	0.25×0.25		
Distributed loading (kN)	5	4	3
Concentrated loading (kN)	2	1.5	1
Span number	**1st**	**2nd**	**3rd**
Span width (m)	5		

The Single friction pendulum bearing properties:

Table 3.6: Case study 1 SFP bearing properties

Initial stiffness (kN/m)	362
Effective radius (m)	1.25
Frication coefficient (μ)	0.06
Thickness (m)	0.05
E_x and E_{m-z} (MPa)	2×10^9

Results:

Table 3.7: Case study 1 results

Result	Without bearing	With SFP bearing
1st natural period	0.589 s	1.13 s
Max 1st story displacement (m)	-0.011 at 21.88 s	-0.013 at 24.365 s
Max 2nd story displacement (m)	0.022 at 21.58 s	-0.014 at 24.34 s
Max 3rd story displacement (m)	0.030 at 21.555 s	-0.015 at 24.345 s
Max 1st story acceleration (m/s^2)	-3 at 21.615 s	-1.57 at 21.11 s
Max 2nd story acceleration (m/s^2)	-3.696 at 21.615 s	-1.62 at 21.105 s
Max 3rd story acceleration (m/s^2)	4.29 at 21.24 s	-1.68 at 21.11 s
Max 1st story drift	-0.00283 at 21.88 s	-0.0004 at 24.205 s
Max 2nd story drift	0.00338 at 21.555 s	-0.0004 at 24.205 s
Max 3rd story drift	0.00252 at 21.545 s	-0.0003 at 24.21 s
Max base shear force (kN)	87.65 at 21.875	-24.123 at 21.35 s

Discussion:

The single friction pendulum bearing enlarges the natural period of the frame by 91% while the absolute maximum roof displacement has been decreased to 0.015 m as 50 % lesser than the frame without SFP. The absolute maximum acceleration for the top floor has been decreased to 1.68 m/s^2 with 60% decreasing. Finally, the absolute maximum base shear force is minimized to 24.12 kN as 72 % lesser than the original frame.

2nd case: in this case study a triple friction pendulum bearing has been tested (displacement-controlled test) and the hysteresis loop of the TFP plotted using the linear time history analysis app. The properties of the bearing shown in the table below.

Table 3.8: Case study 2 TFP bearing properties

Effective stiffness (kN/m)	**1694**
Friction coefficient (μ_1, μ_2 and μ_3)	0.03, 0.07 and 0.13
Effective radiuses (R_1, R_2 and R_3) (m)	0.2, 2.1 and 2.1
Stopping distance (d_1, d_2 and d_3) (m)	0.05, 0.45 and 0.45
E_{m-x}, E_{m-y} and E_{m-z} (MPa)	10
Vertical stiffness (kN/m)	1
Axial loading (kN)	1

Result:

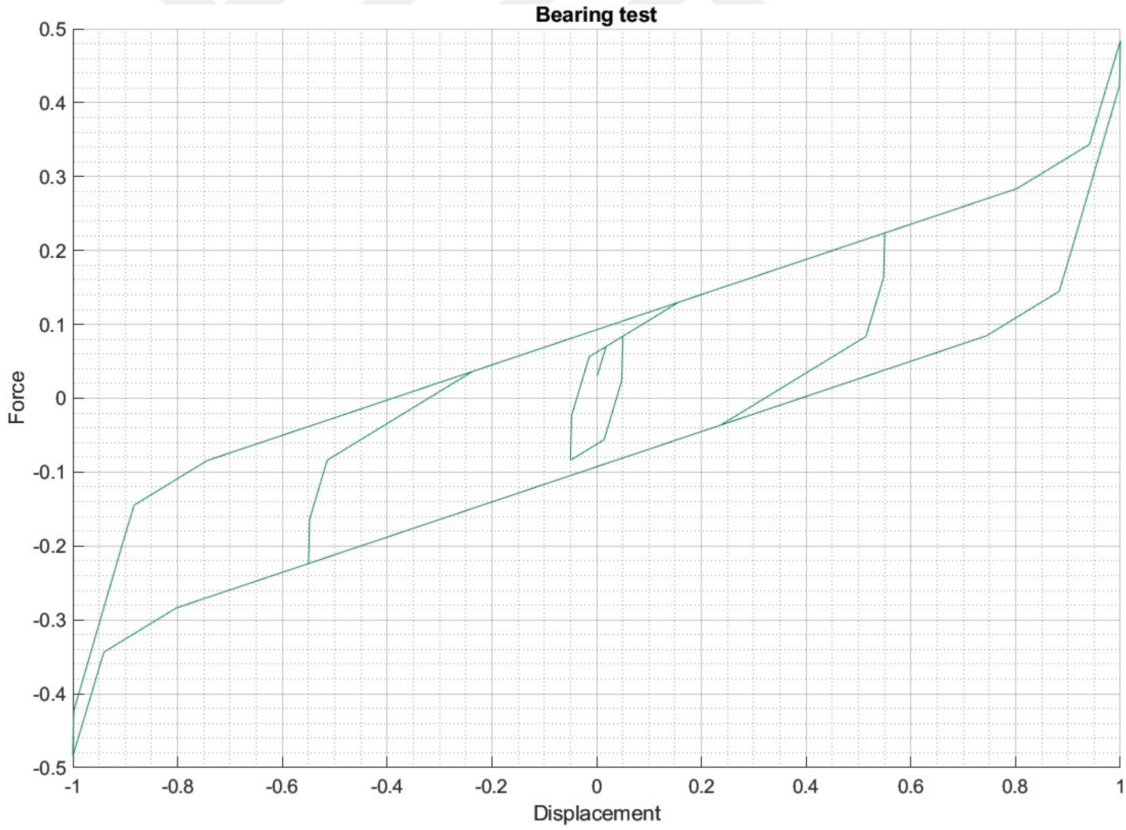

Figure 3.27: Case study 2 TFP hysteresis loop (displacement-controlled loop)

3rd case: This case study is for pushover analysis for 1 story frame with 1 span. The frame top floor has been controlled to reach 50 cm. the frame has been analyzed and pushover curve and the moment-curvature relationship has been done using pushover analysis app. The properties of the frame and hinges are shown below.

Table 3.9: Case study 3 frame properties

Height of frame (m)	3.4
Span width (m)	6
Mass floor (ton)	5
Distributed load (kN/m)	5
Concentrated load (kN)	2

The columns size W24×55 while the beam size W21×44 the hinge properties found from: http://resslabtools.epfl.ch/

Table 3.10:Case study 2 Column hinge properties

Initial stiffness (kN-mm/rad)	199791084
Strain hardening	0.002
My (kN-mm)	1070731
Lambda s, c, a and k	1.5
C s, c, a and k	1
$\boldsymbol{\theta}$p	0.017
$\boldsymbol{\theta}$pc	0.092
$\boldsymbol{\theta}$u	0.4
residual strength ratio	0.4

Table 3.11:Case study 2 beam hinge properties

Initial stiffness (kN-mm/rad)	79833187
Strain hardening	0.002
My (kN-mm)	848653
Lambda s, c, a and k	1.5
C s, c, a and k	1
$\boldsymbol{\theta}$p	0.026
$\boldsymbol{\theta}$pc	0.09
$\boldsymbol{\theta}$u	0.4
residual strength ratio	0.4

Note that: The hinges are assumed to be symmetrical with loading direction.

Result:

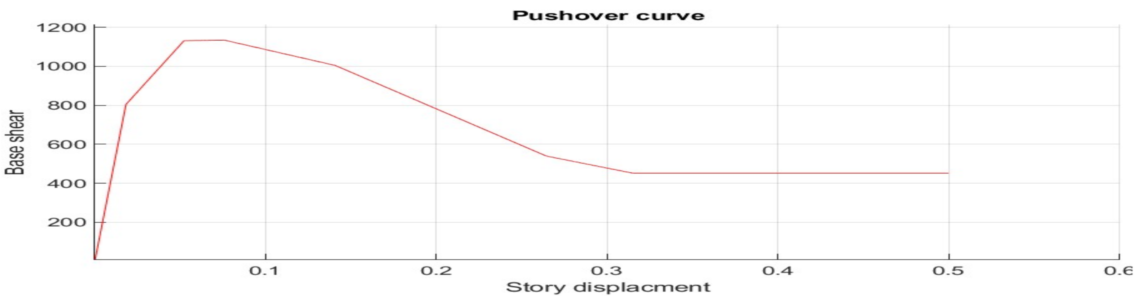

Figure 3.28: case study 3 Pushover curve

4. Conclusion

This thesis was about creating a graphical user interfaces for OpenSEES platform for educational purposes to help the civil and structural engineering to improve their knowledge about the linear dynamic analysis (Time history) and nonlinear static analysis (Pushover). Created apps mainly will help the structural engineering students by saving the time of learning OpenSEES language command and trying many modelling properties in short time, both apps have been designed as pre and post processesors. Time history analysis app has the ability to model five types of frames with option to assign the seismic isolation bearing and analyse the frame for modal and linear time history analysis and testing the support. THA app can plot and show the most important results which will help the user to see the frame how will act under dynamic excitation. The Pushover analysis app can deal with three types of frame to model and analyse the frame with the concept of the concentrated plasticity (Plastic hinges). The PA can plot the moment curvature relationship of the hinges and the pushover curve which help the user to know the overall strength of the frame. These apps can be used as the base for more analysis type app and can be more generalized with same functions for the more general purpose than educational aims.

References

[1] OpenSEES.berkeley.edu. Open System for Earthquake Engineering Simulation - Home Page. [online] Available at: <https://OpenSEES.berkeley.edu/>.

[2] Anil K. Chopra DYNAMICS OF STRUCTURES Theory and Applications to Earthquake Engineering, 4th edn., United States of America: Pearson education inc (2011).

[3] Dimitrios G. Lignos and Helmut Krawinkler 'Deterioration Modeling of Steel Components in Support of Collapse Prediction of Steel Moment Frames under Earthquake Loading', JOURNAL OF STRUCTURAL ENGINEERING (2011), 137(11), pp. 1291-1302.

[4] Agrawal, A.K. and Amjadian, M. 'Seismic component devices', in Alessio Pipinato (ed.) Innovative Bridge Design Handbook.: Elsevier Inc (2016)., pp. 531-553

[5] L. F. IBARRA, R. A. MEDINA AND H. KRAWINKLER 'Hysteretic models that incorporate strength and stiffness deterioration', Earthquake Engng Struct. Dyn (2005), 34, pp. 1489-1511.

[6] Roy, S.S. and Dash, S.R. Dynamic Behavior of Multi Span Continuous Girder Bridge with Isolation Bearings', International Journal of Bridge Engineering (2018), 06(2), pp. 01-23.

[7] Fenz, D. M and Constantinou, M. C. 'Modelling Triple Friction Pendulum Bearings for Response-History Analysis', Earthquake Engineering Research Institute (2008), 24(4), pp. 1011-1028.

[8] Nguyen, N.V., Nguyen C.H., Hoang, H.P. and Huong, K.T. 'Performance of Single Friction Pendulum bearing for isolated buildings subjected to seismic actions in Vietnam', *IOP Conf. Series: Earth and Environmental Science,* (2018), 143(012048)

[9] Vassilis K. Papanikolaou, Theocharis Kartalis-Kaounis, Evangelos Protopapadakis, and Theocharis Papadopoulos 'A New graphical user interface ', OpenSEES day Europe 2017 (2017), pp. 73-76.

[10] Georgios Baltzopoulosa , Roberto Baraschinoa, Iunio Iervolinoa , Dimitrios Vamvatsikos (2018) 'Dynamic analysis of single-degree-of-freedom systems (DYANAS): A graphical user interface for OpenSees', *Engineering Structures* (2018), 177 pp. 395-408

Appendices

Appendix A.

Verification example 1 OpenSEES code:

```
[wipe;
model basic -ndm 2 -ndf 3;
node 1 0. 0. ;
node 2 6 0. ;
node 3 12 0. ;
node 4 0. 3.4 ;
node 5 6 3.4 ;
node 6 12 3.4 ;
node 7 0. 6.8 ;
node 8 6 6.8 ;
node 9 12 6.8 ;
fix 1 1 1 1 ;
fix 2 1 1 1 ;
fix 3 1 1 1 ;
mass 1 0. 0. 0. ;
mass 2 0. 0. 0. ;
mass 3 0. 0. 0. ;
mass 4 3000 0. 0. ;
mass 5 2000 0. 0. ;
mass 6 3000 0. 0. ;
mass 7 2000 0. 0. ;
mass 8 1500 0. 0. ;
mass 9 2000 0. 0. ;
```

```
geomTransf Linear 1 ;

geomTransf PDelta 2 ;

element elasticBeamColumn 1 1 4 0.0225 200000000000 4.2187e-05 2 ;

element elasticBeamColumn 2 2 5 0.0225 200000000000 4.2187e-05 2 ;

element elasticBeamColumn 3 3 6 0.0225 200000000000 4.2187e-05 2 ;

element elasticBeamColumn 4 4 7 0.01 200000000000 8.3333e-06 2 ;

element elasticBeamColumn 5 5 8 0.01 200000000000 8.3333e-06 2 ;

element elasticBeamColumn 6 6 9 0.01 200000000000 8.3333e-06 2 ;

element elasticBeamColumn 7 4 5 0.01 200000000000 8.3333e-06 1 ;

element elasticBeamColumn 8 5 6 0.01 200000000000 8.3333e-06 1 ;

element elasticBeamColumn 9 7 8 0.01 200000000000 8.3333e-06 1 ;

element elasticBeamColumn 10 8 9 0.01 200000000000 8.3333e-06 1 ;

for { set k 1 } { $k <= 2 } { incr k } {

recorder Node -file [format "Verification_example_1mode%i.out" $k] -node 6 9 -dof 1
"eigen $k"

}

set lambda [eigen -fullGenLapack 2  ];

set omega {}

set f {}

set T {}

set pi 3.141593

foreach lam $lambda {

lappend omega [expr sqrt($lam)]

lappend f [expr sqrt($lam)/(2*$pi)]

lappend T [expr (2*$pi)/sqrt($lam)]

}

puts "periods are $T "

set period " Verification_example_1Periods.txt"

set Periods [ open $period "w" ]

foreach t $T {

puts $Periods "$t"
```

```
}
close $Periods
record ]
```

Verification example 2 OpenSEES Time history analysis code:

```
[ wipe;
model basic -ndm 2 -ndf 3;
file mkdir data;
node 1 0. 0. ;
node 2 5 0. ;
node 3 0. 3.4 ;
node 4 5 3.4 ;
node 5 0. 6.8 ;
node 6 5 6.8 ;
node 7 0. 6.8 ;
node 8 5 6.8 ;
fix 1 1 1 1 ;
fix 2 1 1 1 ;
mass 1 0. 0. 0. ;
mass 2 0. 0. 0. ;
mass 3 3.5 0. 0. ;
mass 4 3.5 0. 0. ;
mass 5 2.5 0. 0. ;
mass 6 2.5 0. 0. ;
mass 7 2 0. 0. ;
mass 8 2 0. 0. ;
geomTransf Linear 1 ;
geomTransf PDelta 2 ;
element elasticBeamColumn 1 1 3 0.09 30000000 0.000675 2 ;
element elasticBeamColumn 2 2 4 0.09 30000000 0.000675 2 ;
element elasticBeamColumn 3 3 5 0.09 30000000 0.000675 2 ;
```

```
element elasticBeamColumn 4 4 6 0.09 30000000 0.000675 2 ;
element elasticBeamColumn 5 5 7 0.09 30000000 0.000675 2 ;
element elasticBeamColumn 6 6 8 0.09 30000000 0.000675 2 ;
element elasticBeamColumn 7 3 4 0.04 30000000 0.00013333 1 ;
element elasticBeamColumn 8 5 6 0.04 30000000 0.00013333 1 ;
element elasticBeamColumn 9 7 8 0.04 30000000 0.00013333 1 ;
timeSeries Linear 2  -factor +1.000000E+00
pattern Plain 2 2  {
load  3 0 -2 0;
load  4 0 -2 0;
load  5 0 -2 0;
load  6 0 -2 0;
load  7 0 -2 0;
load  8 0 -2 0;
eleLoad -ele 7 -type -beamUniform -3;
eleLoad -ele 8 -type -beamUniform -3;
eleLoad -ele 9 -type -beamUniform -3;
}
recorder  Node  -file Verification_example_2_Gravity_Reactions.out  -time  -node 1 2  -dof
1 2 3 reaction ;
constraints Plain
numberer  RCM
system  ProfileSPD
test  NormDispIncr  +1.000000E-006   100  0   2;
algorithm  Newton
integrator  LoadControl  +0.1
analysis  Static
record
analyze 10
setTime 0.0
loadConst
```

```
remove recorders
wipeAnalysis
recorder Node -file Verification_example_2_floor_dis_x.out -time -node 4 6 8 -dof 1 disp ;
recorder Node -file Verification_example_2_floor_acc_x.out -time -node 4 6 8 -dof 1 accel ;
recorder Node -file Verification_example_2_Base_shear_x.out -time -node 1 2  -dof 1 reaction ;
recorder Drift -file Verification_example_2_story_drift.out -time -iNode 2 4 6 -jNode 4 6 8 -dof 1 -perpDirn 2
timeSeries Path 1 -dt 0.005 -filePath RSN1147_KOCAELI_ATS090.AT2 -factor 9.81 ;
pattern UniformExcitation 1 1 -accel 1;
rayleigh 0.63595 0.0026319 0. 0. ;
wipeAnalysis;
constraints Plain;
numberer Plain;
system BandGeneral;
algorithm Linear;
integrator Newmark 0.5 0.25 ;
analysis Transient
analyze 30500 0.005 ]
```

Verification example 2 Results:

Floor 1:

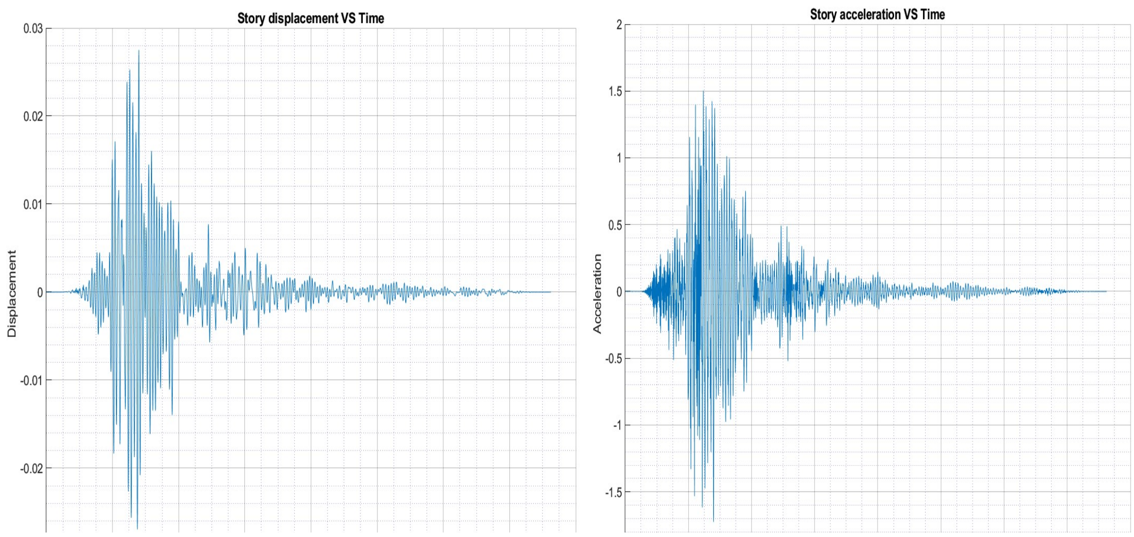

Figure A. 1: Verification example 2 results (Floor1).

Floor 2:

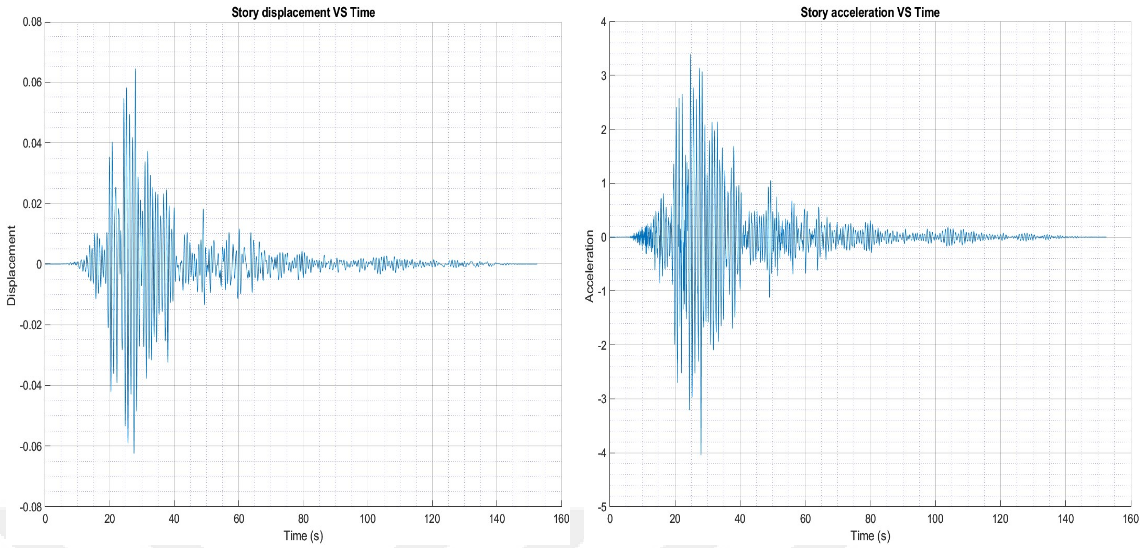

Figure A. 2:Verification example 2 results (Floor2).

Floor 3:

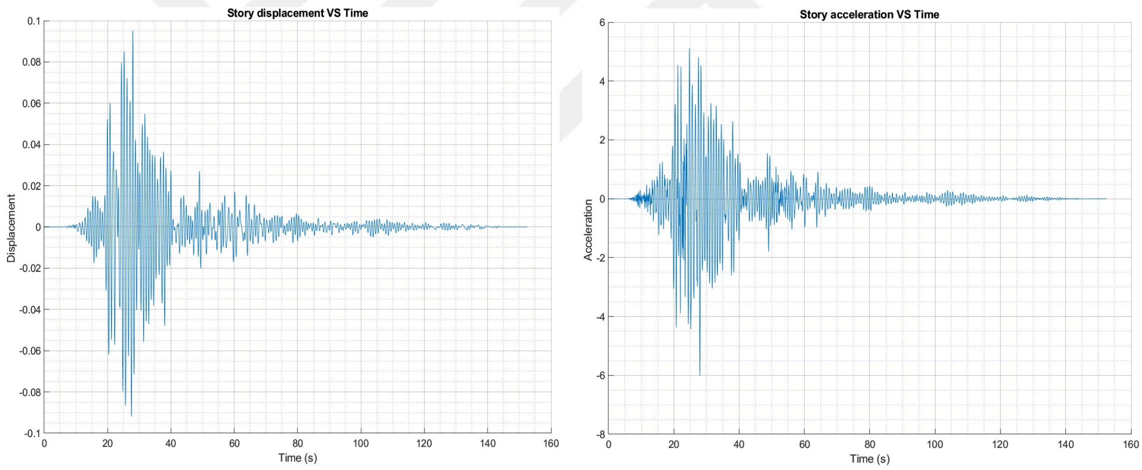

Figure A. 3: Verification example 2 results (Floor3).

Base shear force:

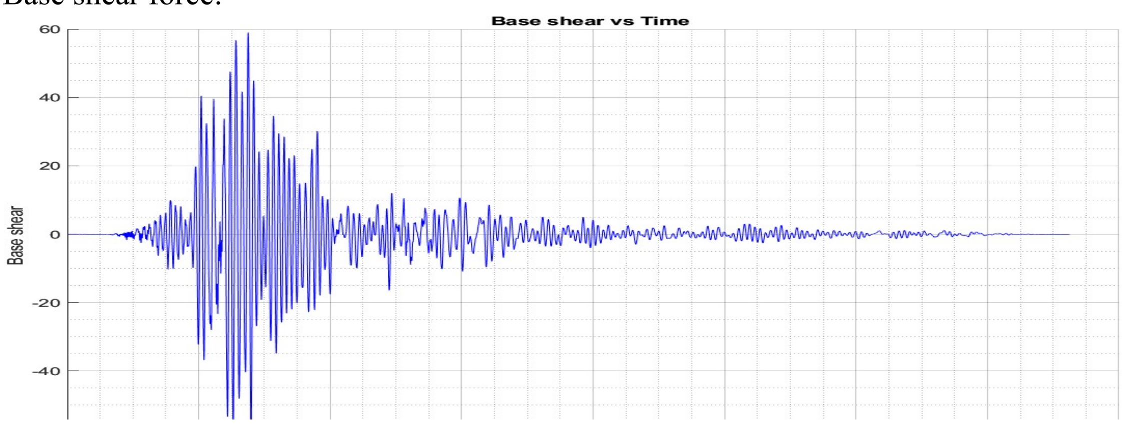

Figure A. 4:Verification example 2 Base shear force.

Appendix B.

Case study 1 Results:

Floor 1 displacement:

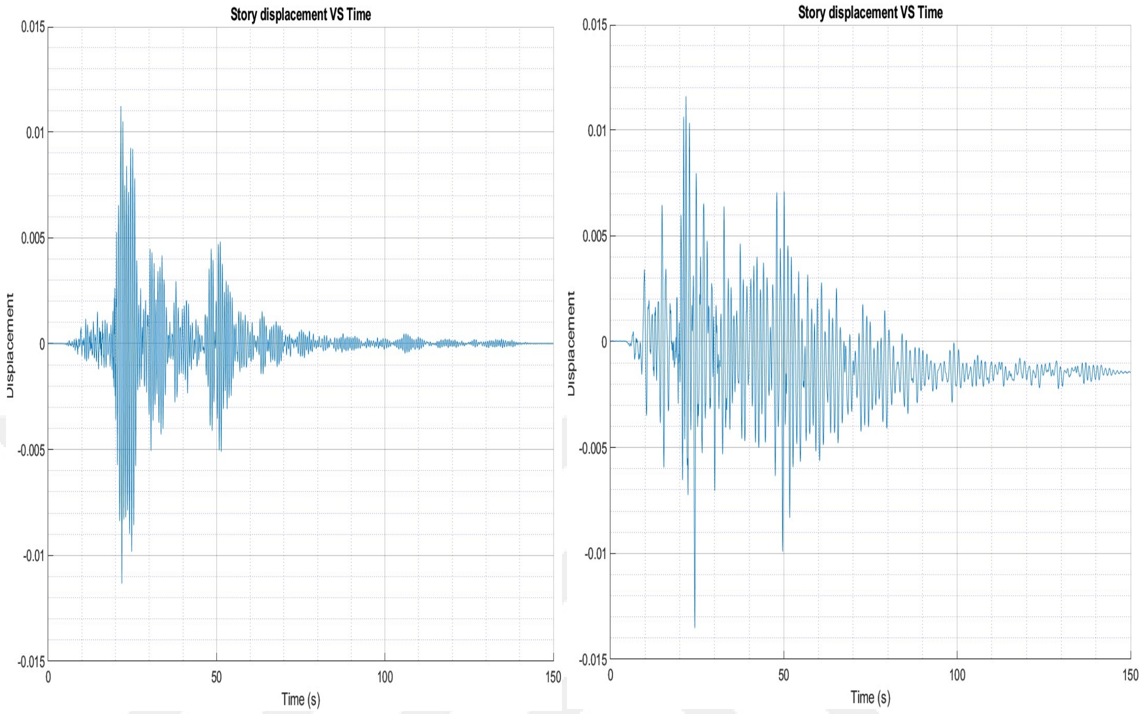

Figure B. 1: Case study 1 floor 1 displacement (Left: No bearing and Right: with SFP bearing)

Floor 2 displacement:

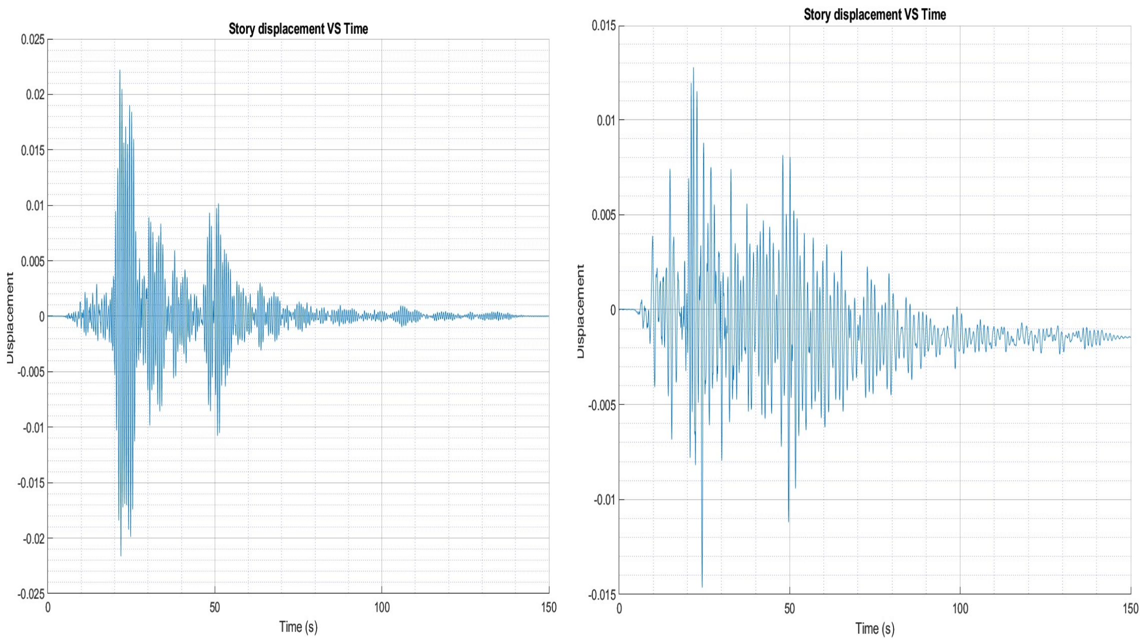

Figure B. 2: Case study 1 floor 2 displacement (Left: No bearing and Right: with SFP bearing)

Floor 3 displacement:

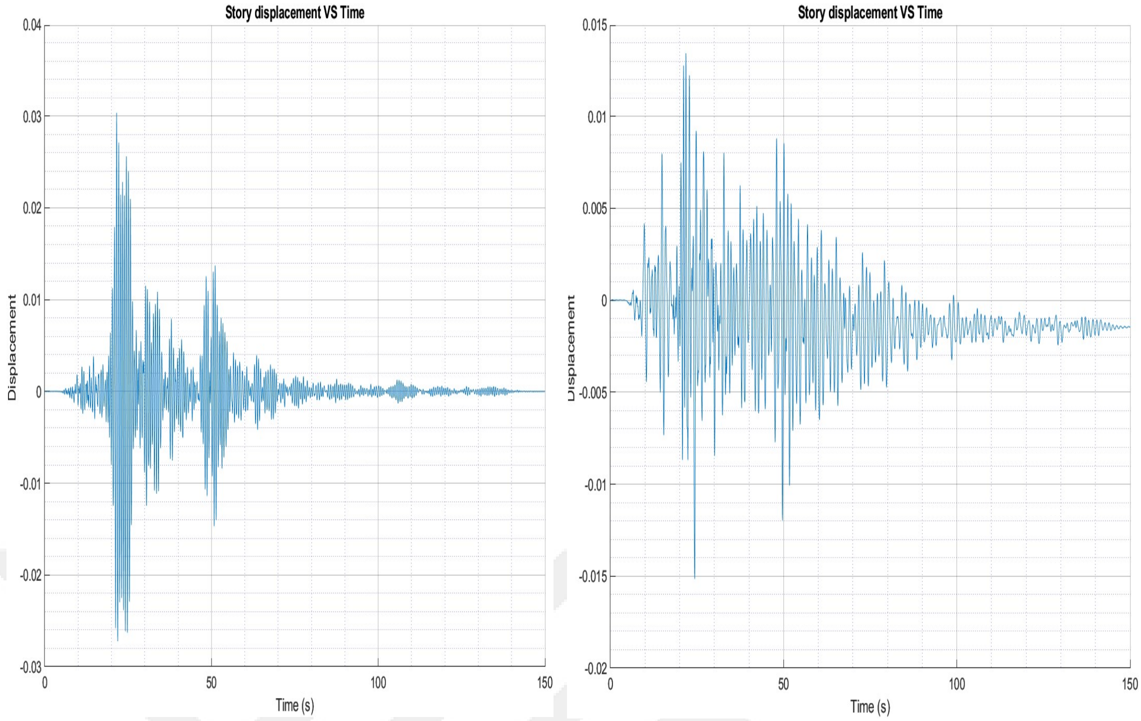

Figure B. 3:Case study 1 floor 3 displacement (Left: No bearing and Right: with SFP bearing)

Floor 1 acceleration:

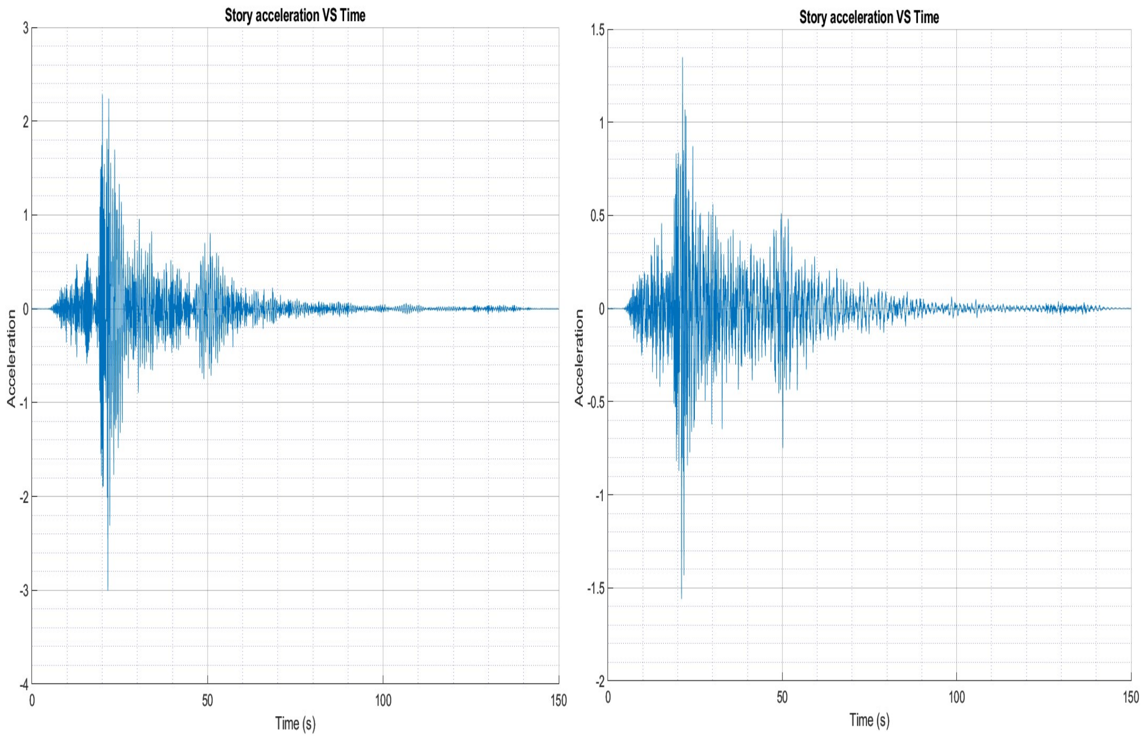

Figure B. 4:Case study 1 floor 1 acceleration (Left: No bearing and Right: with SFP bearing)

Floor 2 acceleration:

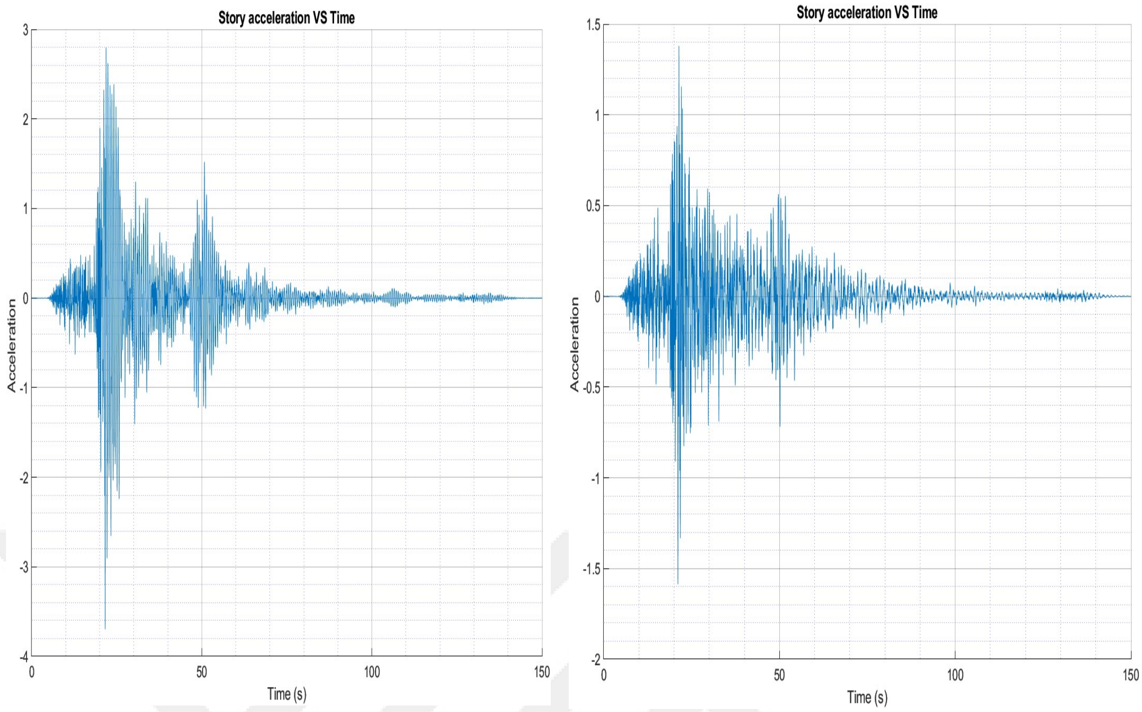

Figure B. 5: Case study 1 floor 2 acceleration (Left: No bearing and Right: with SFP bearing)

Floor 3 acceleration:

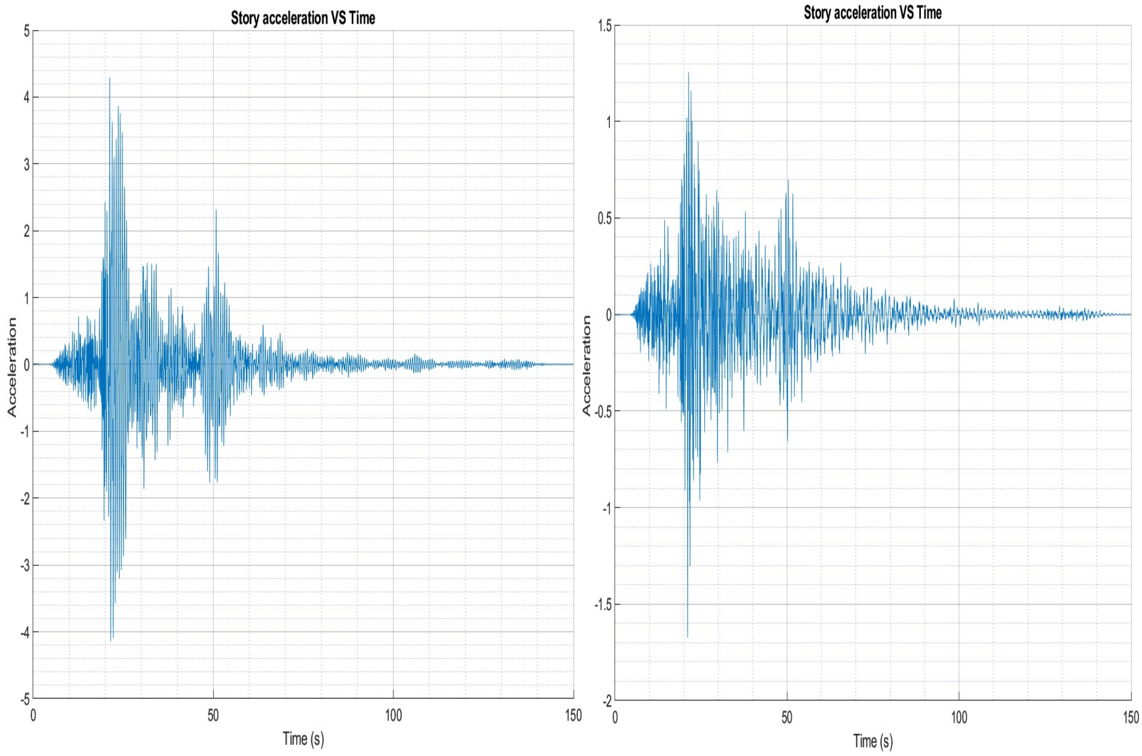

Figure B. 6:Case study 1 floor 3 acceleration (Left: No bearing and Right: with SFP bearing)

Base shear force X direction:

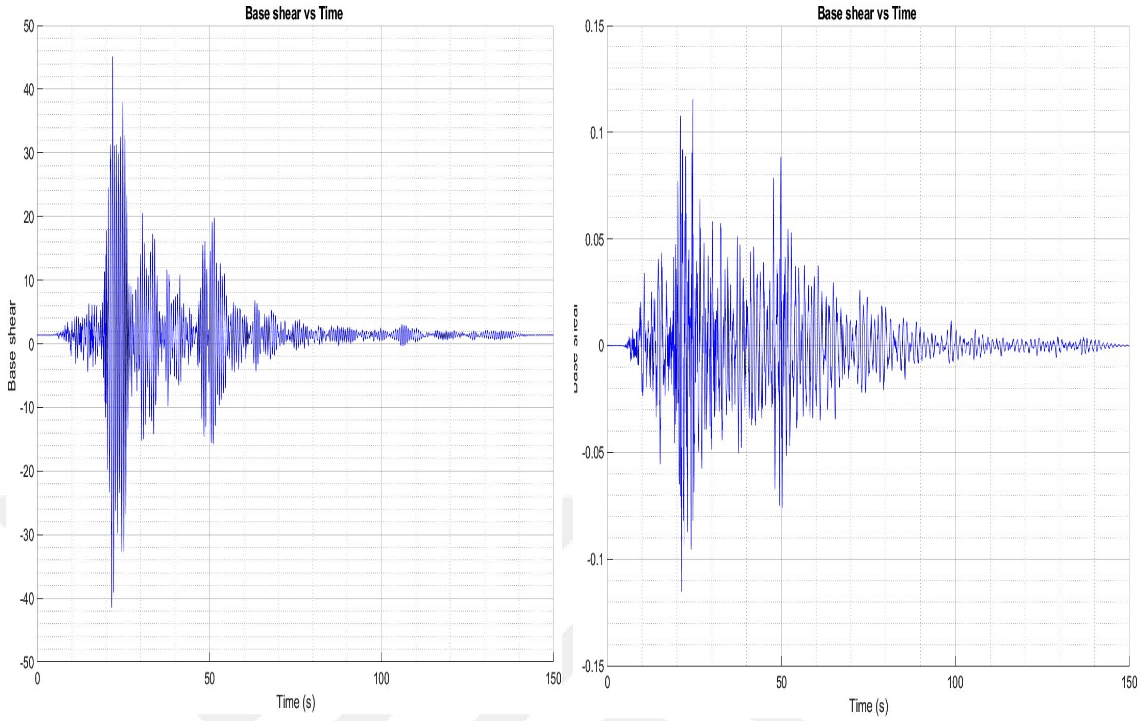

Figure B. 7:Case study 1 Base shear force (Left: No bearing and Right: with SFP bearing)

Case Study 2 code:

```
[model BasicBuilder -ndm 3 -ndf 6
node 1 0 0 0
node 2 0 0 0
fix 1 1 1 1 1 1 1
frictionModel Coulomb 1 0.03 ;
frictionModel Coulomb 2 0.07 ;
frictionModel Coulomb 3 0.13 ;
uniaxialMaterial Elastic 1 1694 ;
uniaxialMaterial Elastic 2 10 ;
uniaxialMaterial Elastic 3 10 ;
uniaxialMaterial Elastic 4 10 ;
element TripleFrictionPendulum 1 1 2 1 2 3 1 4 2 3 0.2 2.1 2.1 0.05 0.45 0.45 1 0.001 1 0.1 1.e-5 ;
pattern Plain 1 "Linear" {
load  2 0 0 -1 0 0 0
}
system BandGeneral
numberer Plain
constraints Plain
test NormDispIncr 1.0e-12 10
integrator LoadControl 0.1
algorithm Newton
analysis Static
analyze 10
loadConst -time 0.0;
pattern Plain 2 Linear {
load 2 1 0. 0. 0. 0. 0.
}
integrator DisplacementControl 2 1 .1;
analysis Static
```

```
recorder Node -file case_study_2TFP_Test.out -time -nodes 2 -dof 1  disp
foreach d { 0.05 -0.1 0.05 0.55 -1.1 0.55 1 -2 2  }   {
incr cnt 1;
puts "step: $cnt , d= $d" ;
if {$d >= 0} {
set du  .001;
} else {
set du  -.001;
}
set nstep [expr int($d/$du)];
set du [expr $du*1];
integrator DisplacementControl 2 1 $du;
set ok 0;
set temp 1;
while {$temp <= $nstep && $ok == 0} {
incr temp 1;
set ok [analyze 1]
if {$ok != 0} {
puts "Trying NewtonWithLineSearch .."
algorithm NewtonLineSearch
set ok [analyze 1]
algorithm Newton
}
if {$ok != 0} {
puts "Trying KrylovNewton .."
test EnergyIncr 1.0e-10 100 0
algorithm KrylovNewton
set ok [analyze 1]
test EnergyIncr 1.0e-10 100 0
algorithm Newton
}
```

```
if {$ok != 0} {
puts "Trying Broyden .."
algorithm Broyden 100
set ok [analyze 1]
algorithm Newton
}
}; # end while loop
}
Wipe]
```

Case study 3 Moment-Curvature relationship for hinges:

1st Hinge:

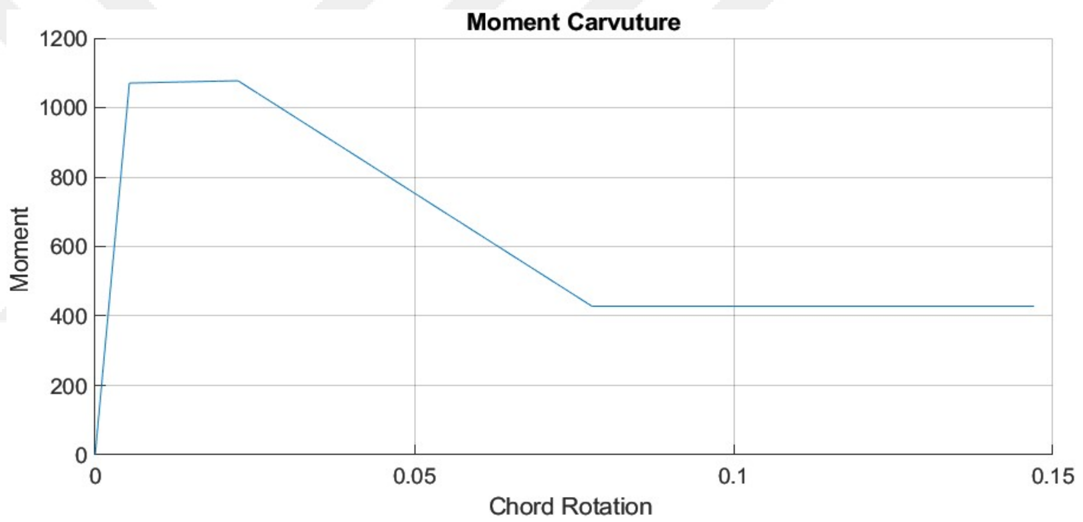

Figure B. 8:first hinge M-C relationship

2nd Hinge:

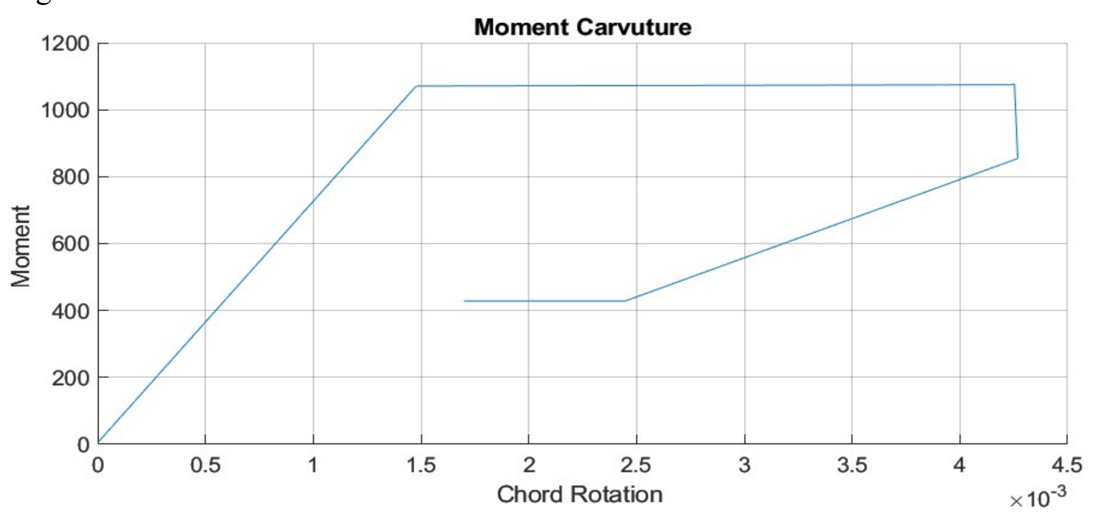

Figure B. 9: Second hinge M-C relationship

3rd Hinge:

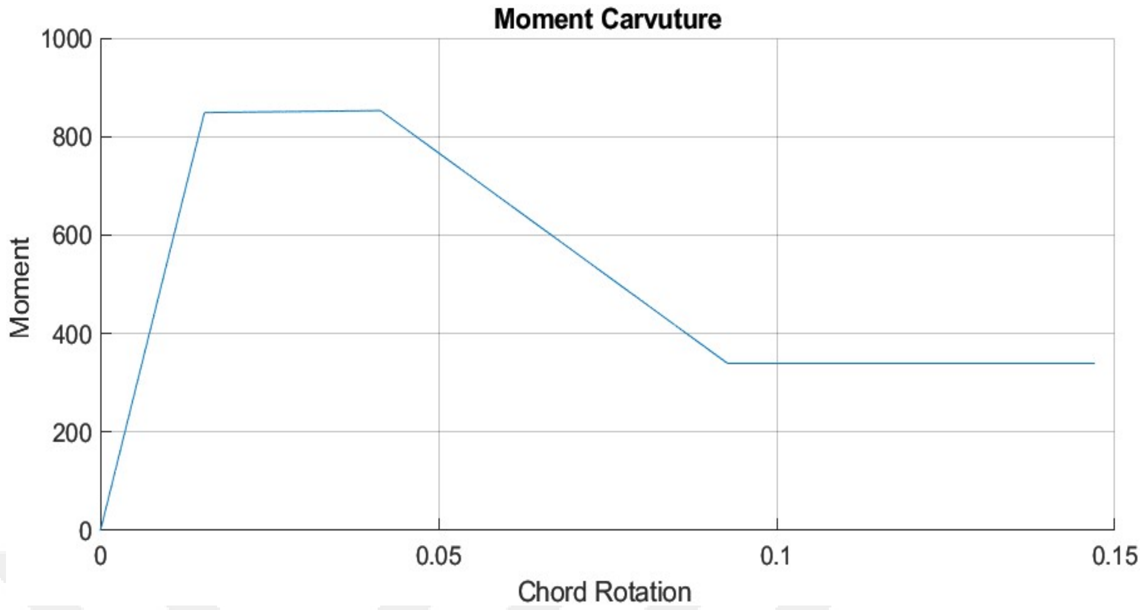

Figure B. 10:Third hinge M-C relationship

4th Hinge

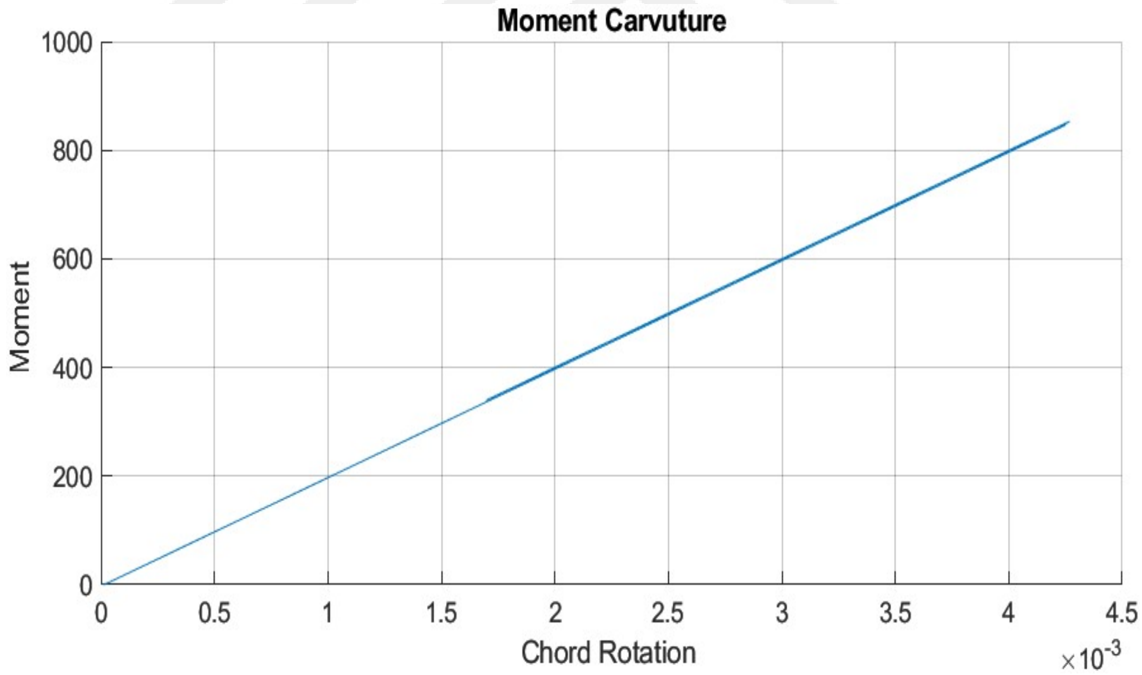

Figure B. 11:Fourth hinge M-C relationship

5^{th} Hinge:

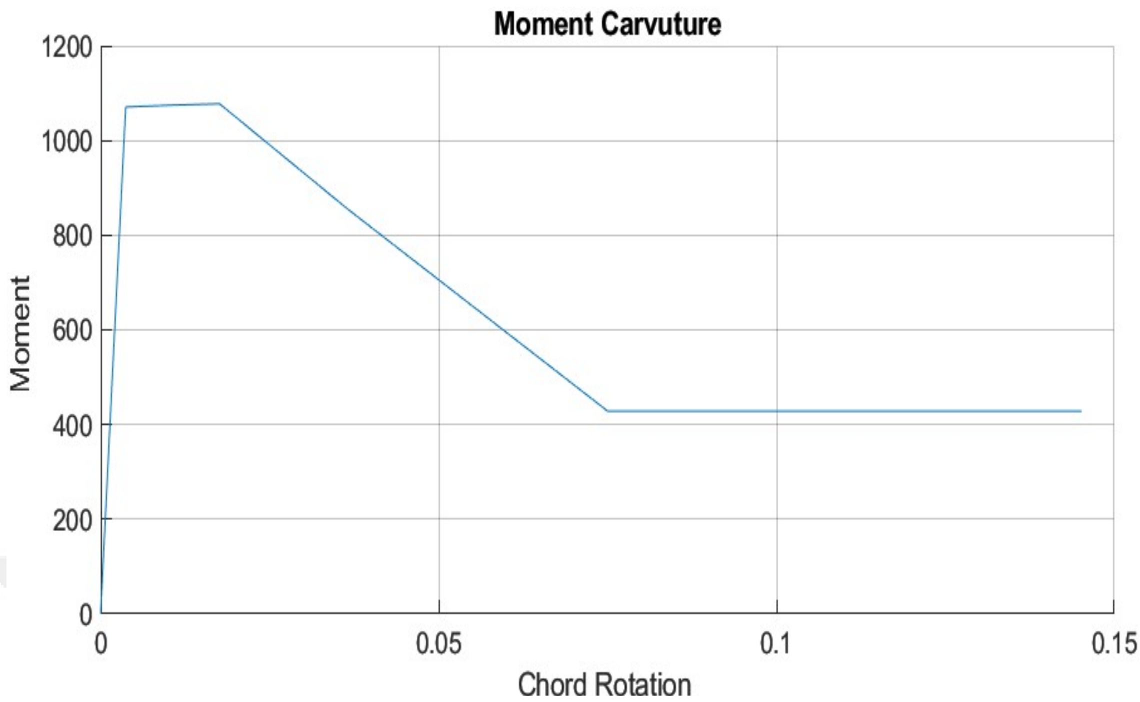

Figure B. 12:Fifth hinge M-C relationship

6^{th}Hinge:

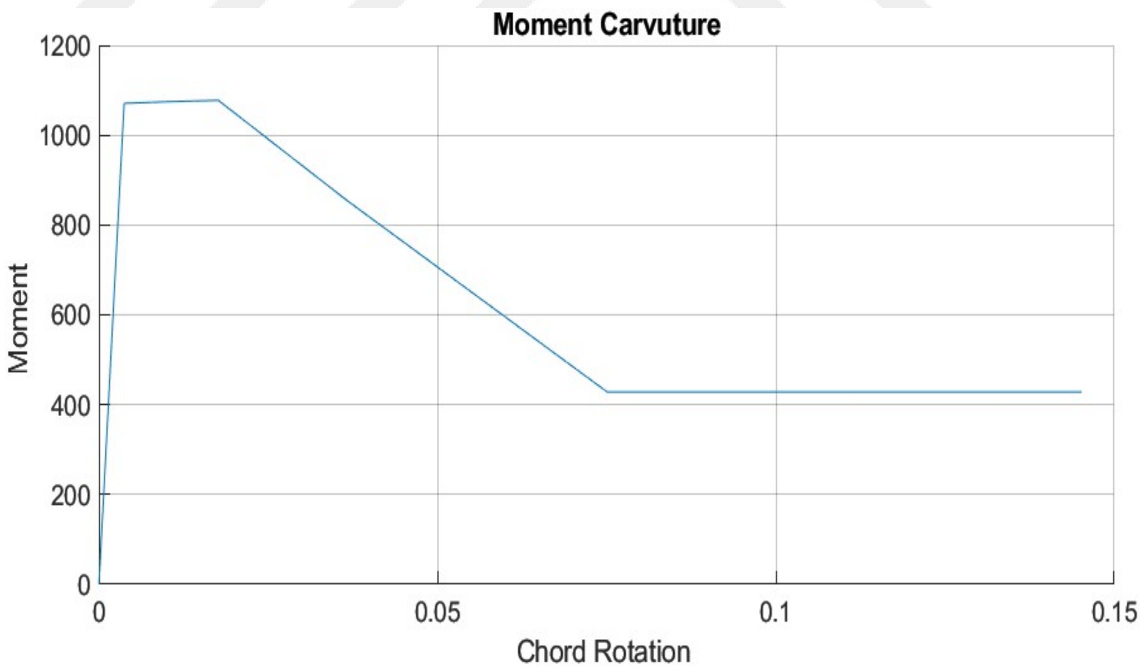

Figure B. 13: Sixth hinge M-C relationship