

**SEISMIC PERFORMAMCE ASSESSMENT TOOL FOR CONCRETE
FRAMES WITH INFILLS (SPA-INFILL)**

VERSION 1.0

USER MANUAL

By

Henry V. Burton, PhD., S.E.

University of California, Los Angeles, U.S.A.

November 13, 2014

Table of Contents

1. Introduction	1
2. Building Information Assembly	2
2.1 Building Information and Model Parameters.....	2
2.1.1 Analysis Options	2
2.1.2 Analysis Parameters	3
2.1.3 Dynamic Properties	5
2.1.4 Geometry	5
2.1.5 Loads	6
2.1.6 Seismic Design Parameters	9
2.1.7 Structural Properties	9
2.1.8 Baseline .tcl Files for OpenSees Models	12
2.1.9 Building Model Data	12
2.2 MATLAB Executable for Assembling Building Model Data.....	12
3. OpenSees Model Construction	13
4. Seismic Performance Assessment	13
4.1 Eigen Value Analysis	14
4.2 Nonlinear Static Analysis	14
4.3 Nonlinear Dynamic (Single Intensity) Analysis.....	15
4.4 Parallel OpenSees (OpenSeesMP) for Nonlinear Dynamic Analyses.....	16
5. Visualization of Results	17
6. Example Application of SPA-INFILL	17
6.1 Example Building Description	17
6.2 Example Building Structural Modeling.....	19
6.3 Building Information and Model Parameter Entry for 4-Story Infill Frame Building	19
6.3.1 Analysis Options Entry	19
6.3.2 Analysis Parameters Entry	20
6.3.3 Dynamic Properties	23
6.3.4 Geometry	23
6.3.5 Loads	23
6.3.6 Seismic Design Parameters	25
6.3.7 Structural Properties	26
6.4 Seismic Performance Assessment Results	27
6.4.1 Pushover Analysis.....	27

6.4.2 Collapse Performance Assessment	30
---	----

1. INTRODUCTION

SPA-INFILL is an interactive MATLAB tool that enables performance-based assessment of concrete frame buildings with infills. It automates the construction of 2D and 3D numerical models in OpenSees as well as the seismic performance assessments through nonlinear static and dynamic analyses including incremental dynamic analyses to collapse. The tool was developed using the object oriented programming methodology. It consists of a series of MATLAB .m files, functions and classes that serve to assemble the necessary building information and model parameters, generate structural models and post process and visualize the analysis and performance assessment results. The main directory of the tool titled “SPA-INFILL” contains a series of folders where the required MATLAB, text and .tcl files are stored. The tool consists of the following four modules which are illustrated in Figure 1 and discussed in detail later in the manual.

1. Building Information Assembly (BIA)
2. OpenSees Model Construction (OMC)
3. Seismic Performance Assessment (SPA)
4. Visualization of Results (VOR)

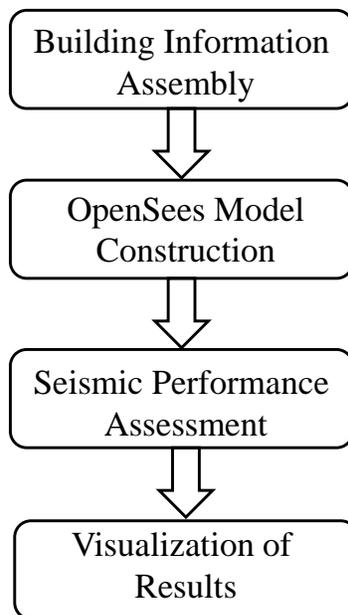


Figure 1 – Overview of SPA-INFILL Modules

2. BUILDING DATA ASSEMBLY

The BDA module is contained in the “BuildingData” and “BuildingDataAssembly” folders. The “BuildingData” folder serves as one of three user–interfaces where all relevant building information and model parameters are entered. The “BuildingDataAssembly” folder contains the MATLAB functions and classes associated with the BDA module. The user has access, but does not need to interface with these files.

2.1 Building Information and Model Parameter Entry

The *BuildingData* directory serves as the first point of entry for the user. It contains a separate folder for each building which contains a set of 9 sub-folders (Figure 2). All but one of the 9 sub-folders contain a set of text files into which the user enters various data and model parameters for a single building.



AnalysisOptions	10/22/2014 6:32 PM	File folder
AnalysisParameters	10/22/2014 6:32 PM	File folder
BaselineTclFiles	10/22/2014 6:35 PM	File folder
BuildingModelData	10/22/2014 6:36 PM	File folder
DynamicProperties	10/22/2014 6:36 PM	File folder
Geometry	11/6/2014 7:54 PM	File folder
Loads	10/22/2014 6:36 PM	File folder
SeismicDesignParameters	10/22/2014 6:36 PM	File folder
StructuralProperties	10/22/2014 6:37 PM	File folder

Figure 2 Sub–Folders with text files used to enter building data and model parameters

2.1.1 Analysis Options Entry

The tool permits the user to construct a 3D OpenSees model as well as 2D models for any or all of the frame lines in the two principal directions of the building. The orthogonal frames are identified as being associated with X- and Z-Direction grid lines, numbered sequentially starting from 1. The X-Direction corresponds to degree of freedom 1 in OpenSees and the Z-Direction corresponds to degree of freedom 3. “*XFrameLinesFor2DModels.txt*” and “*ZFrameLinesFor2DModels.txt*” are text files contained in the *AnalysisOptions* directory. These are used to define the grid numbers for which 2D models of the corresponding frame lines will be generated. The desired frame lines are entered in a single column vector within the text file. Figure 3 provides a snap shot of example entries for “*XFrameLinesFor2DModels.txt*” showing that the user would like to have 2D models generated for X-Direction frame lines 1, 2, 3 and 5.

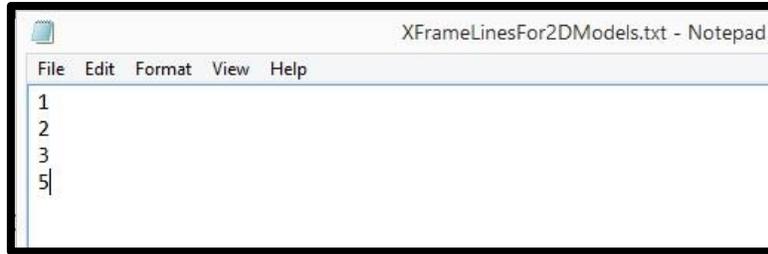


Figure 3 Format for entering frame lines for which 2D models will be generated

2.1.2 Analysis Parameters Entry

Nonlinear static and dynamic analyses are controlled using a number of user-defined parameters. These are entered into separate sub-folders in the *AnalysisParameters* directory. The following Static Analysis parameters (Figure 4) must be entered:

- Pushover analysis increment size – Pushover loading is applied in discrete displacement-based increments. The size of these increments as expressed by a fraction of the maximum displacement is entered as a scalar value into the “*PushoverIncrementSize.txt*” text file. The value is set to 0.01 in the example model and only needs to be changed (reduced) if there are issues with convergence.
- Maximum X-Direction Drift – The maximum roof drift for the pushover analysis in the X-Direction (expressed as a percentage i.e. a value of 1 means that the maximum pushover displacement applied will correspond to 1% roof drift or 0.01 roof drift ratio) is entered as a scalar value into the “*PushoverXDrift.txt*” text file.
- Maximum Z-Direction Drift – The maximum roof drift for the pushover analysis in the Z-Direction (expressed as a percentage i.e. a value of 1 means that the maximum pushover displacement applied will correspond to 1% roof drift or 0.01 roof drift ratio) is entered as a scalar value into the “*PushoverZDrift.txt*” text file.

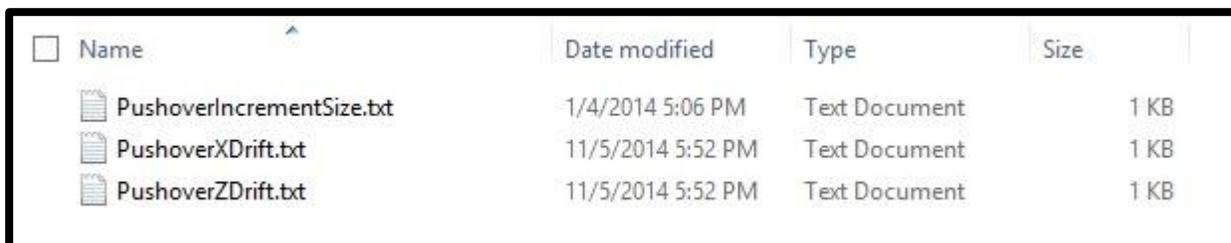


Figure 4 Image showing text files used to enter static analysis parameters

The following dynamic analysis parameters must be entered (Figure 5):

- MCE Spectral Intensity – The spectral acceleration corresponding to the 1st mode period is entered as a scalar value (in terms of the gravitational acceleration constant, g) into the “*SaMCE.txt*” text file.
- IDA Intensity Levels – The tool allows the user to perform incremental dynamic analyses at predefined spectral intensity levels, which are entered as a single column vector into the “*scalesForIDA.txt*” text file. The values are expressed as a percentage of the MCE spectral intensity i.e. a value of 5 refers to a spectral intensity corresponding to 5% of the MCE level.
- Single Scale Dynamic Analysis Spectral Intensity – The tool allows the user to perform nonlinear response history analysis at a single predefined spectral intensity, which is entered as a scalar value into the “*SingleScaleToRun.txt*” text file. This value is also expressed as a percentage of the MCE spectral intensity.
- MCE Scale Factor for Uni-Directional Inertial Loading – For Uni-Direction inertial loading (used for 2D models), a single scale factor is applied to all ground motions to anchor the median spectral intensity of the MCE level. This is entered as a scalar value into the “*UniDirectionMCEScaleFactor.txt*” text file.

Note that the FEMA P695 far field set is used for the example model. This suite consists of 22 pairs of ground motions (44 total). For the 2D models with uni-directional loading, the 44 individual ground motions are used. For the 3D model, the 22 pairs of bi-directional motions are used.

- MCE Scale Factors for Bi-Directional Inertial Loading – For Bi-Direction inertial loading (used for 3D models), a different scale factor is applied to each ground motion pair to anchor the geometric mean spectral intensity of that pair to the MCE level. This is entered as a single column vector into the “*BiDirectionMCEScaleFactors.txt*” text file each value corresponding to a ground motion pair.
- Drift Limit Used For Collapse – The tool allows the user to conduct incremental dynamic analysis to collapse. Collapse is detected based on the exceedance of a predefined drift limit. This drift limit is entered as a scalar decimal value into the “*CollapseDriftLimit.txt*” text file i.e. 0.1 is entered for a 10% drift threshold for collapse
- Initial Intensity Increment for Collapse Assessment – For IDA to collapse, the user defines an initial increment which is used up until collapse is first detected. After which, the intensity is scaled back to the level immediately prior to the one corresponding to collapse. The intensity is then increased using a smaller increments to obtain a more refined collapse estimate. The initial increment is entered as a scalar value into the “*InitialGroundMotionIncrementScaleForCollapse.txt*” text file i.e. 0.1 is entered for a 10% drift threshold for collapse expressed as a percentage of the MCE spectral intensity.

- Reduced Intensity Increment for Collapse Assessment –The reduced increment used after collapse is initially detected (see previous bullet point) is entered as a scalar value into the “*ReducedGroundMotionIncrementScaleForCollapse.txt*” text file.
- Number of Bi-Directional Ground Motion Pairs – The number of ground motion pairs used for bi-directional loading (for 3D model) is entered as a scalar value into the “*NumberOfBiDirectionGroundMotions.txt*” text file.
- Number of Uni-Directional Ground Motions – The number of ground motions used for uni-directional loading (for 2D model) is entered as a scalar value into the “*NumberOfUniDirectionGroundMotions.txt*” text file.

Name	Date modified	Type	Size
BiDirectionMCEscaleFactors.txt	5/15/2014 5:34 PM	Text Document	1 KB
CollapseDriftLimit.txt	10/17/2014 7:14 PM	Text Document	1 KB
InitialGroundMotionIncrementScaleForCollapse.txt	1/11/2014 12:19 PM	Text Document	1 KB
NumberOfBiDirectionGroundMotions.txt	10/17/2014 8:19 PM	Text Document	1 KB
NumberOfUniDirectionGroundMotions.txt	3/2/2014 8:21 PM	Text Document	1 KB
ReducedGroundMotionIncrementScaleForCollapse.txt	1/11/2014 4:27 PM	Text Document	1 KB
SaMCE.txt	5/15/2014 3:34 PM	Text Document	1 KB
scalesForIDA.txt	6/14/2014 3:13 PM	Text Document	1 KB
SingleScaleToRun.txt	2/9/2014 4:56 PM	Text Document	1 KB
UniDirectionMCEscaleFactor.txt	5/15/2014 5:31 PM	Text Document	1 KB

Figure 5 Image showing text files used to enter dynamic analysis parameters

2.1.3 Dynamic Properties

Rayleigh damping is applied to the OpenSees models based on the anchoring to the 1st and 3rd mode periods. The damping ratio is enter as a decimal (for example a value of 0.02 is entered for 2% damping) scalar quantity in the “*dampingRatio.txt*” text file, which is located in the “*DynamicProperties*” sub-folder. 5% damping is assumed in the example model that is provided with the tool.

2.1.4 Geometry

The text files used to enter the building geometry are located in the “*Geometry*” sub-folder. The following user-defined geometry parameters included:

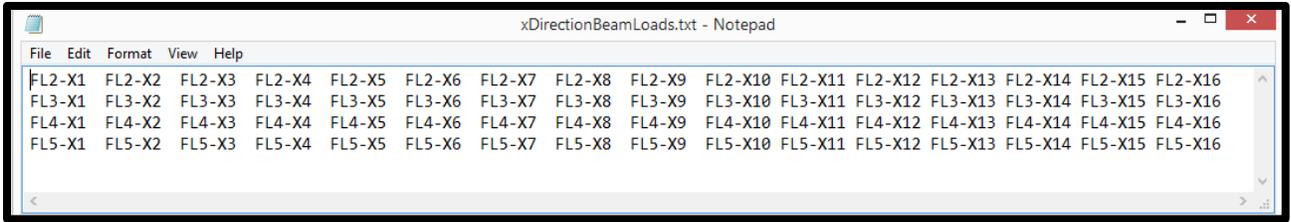
- The number of stories is entered as an integer, scalar quantity in the “*numberOfStories.txt*” text file.
- The number of bays in the X-Direction is entered as an integer, scalar quantity in the “*numberOfXBays.txt*” text file. Likewise, the number of bays in the Z-Direction is entered in the “*numberOfZBays.txt*” text file.

- The height of each story is entered as a single column vector into the “*storyHeights.txt*” text file. The height of story n in inches is entered into the n^{th} row.
- The X-Direction bay widths are entered as a single column vector with into the “*XBayWidths.txt*” text file. The width of story m in inches is entered into the m^{th} row. Likewise, the Z-Direction bay widths are entered into the “*ZBayWidths.txt*” text file.
- The user has the option to consider rocking/uplift of shallow foundations in the simulation model. This has been shown to influence the collapse performance of infill frame buildings (Burton and Deierlein, 2014). The option to consider shallow foundation uplift is entered as a binary value (0 or 1) into the “*foundationUplift.txt*” text file. If a value ‘1’ is entered, column bases are modeled as compression only springs and for a value of ‘0’, the column bases are modeled as pinned.

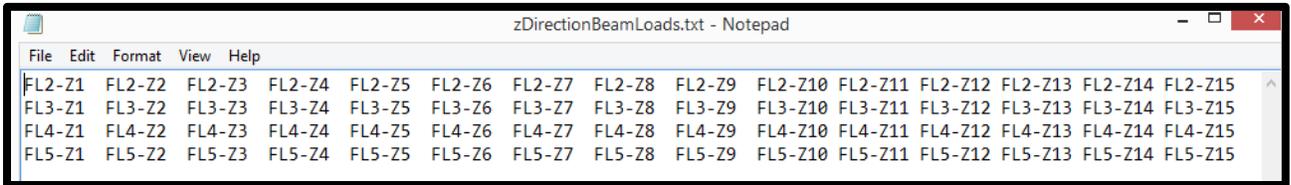
2.1.5 Loads

Floor loads and seismic weights are defined in the text files located in the “*Loads*” sub-folder. Gravity loads (unless defined separately as dead or live) should be based on expected value i.e. $1.05D + 0.25L$ using negative values to represent downward loads. The following user-defined load parameters are included.

- The floor weights used to compute seismic masses are defined as a single column vector in the “*floorWeights.txt*” text file. The values are entered in kips with the weight of the floor level n entered into the $(n - i)^{\text{th}}$ row with the first above ground floor taken as level 2.
- The area load for above ground floors is defined as a single column vector in the “*floorLiveLoad.txt*” and “*floorDeadLoad.txt*” text files. The values are entered in kips/in² with the area load on floor level n entered into the $(n - i)^{\text{th}}$ row with the first above ground floor taken as level 2.
- Uniform expected gravity loads for beams are incorporated in both the 2D and 3D models. These loads are defined using an $m \times n$ array in the “*xDirectionBeamLoads.txt*” and “*zDirectionBeamLoads.txt*” text files. Each entry represents the uniform expected ($1.05D + 0.25L$) gravity load. For the X-Direction beams, m is the number of suspended floor levels in the building and n is the number of beams. The same applies to the Z-Direction beams. The values are entered in kips/in with the beam loads on floor level n entered into the $(n - i)^{\text{th}}$ row with the first above ground floor taken as level 2. Figure 6 shows a level 2 floor plan for the example structure provided with the tool with a unique labels for each uniform beam load (FL2X1, FL2X2 etc.). It also shows the text file entries for the same loads.
- For the 3D models, all gravity load are defined based on the uniform beam loads discussed in the previous bullet point. However, for the 2D models, column point loads are needed to account for the loads transferred from orthogonal beams not included in the model. These concentrated loads are defined in the “*columnLoads.txt*” text file. The loads are defined in kips using an $m \times n$ array where m is the number of stories and n is the number of columns per story. Figure 7 shows a 1st

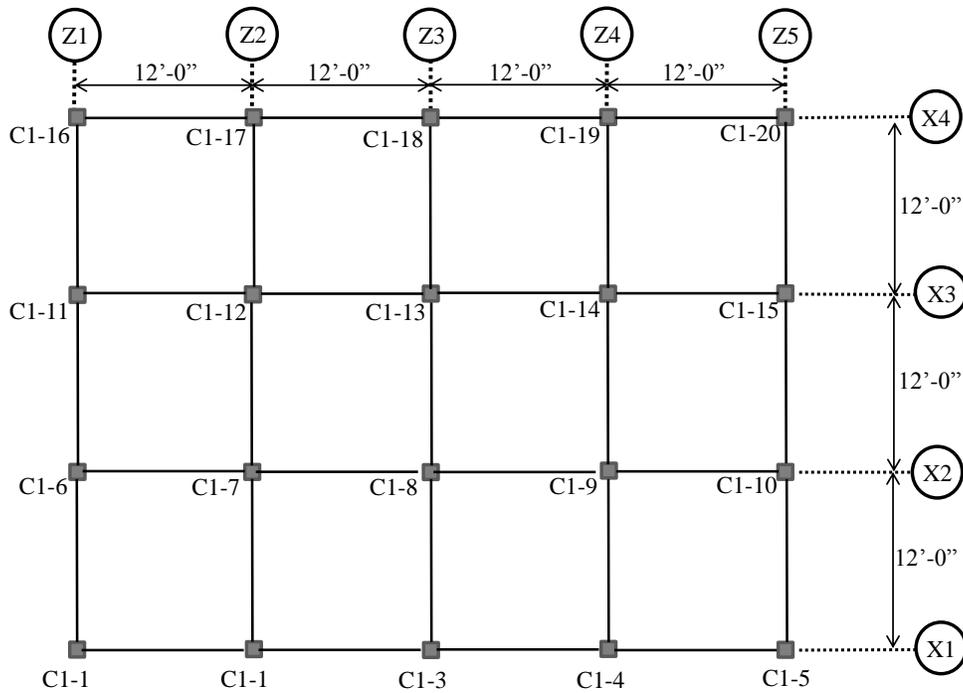


(b)

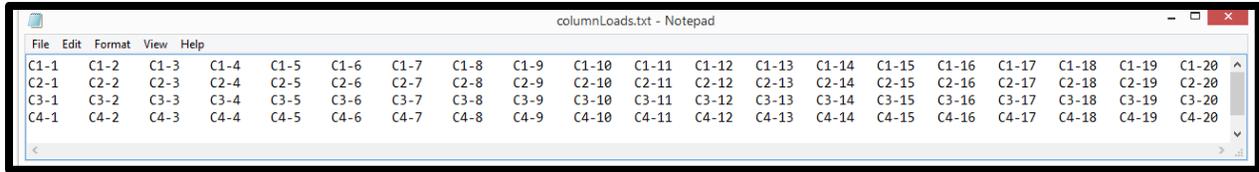


(c)

Figure 6 Defining beam uniform gravity loads: (a) Floor level 2 framing plan for example building showing location and identifiers for gravity beams, (b) Placeholders for uniform gravity load entries into “*xDirectionBeamLoads.txt*” based on identifiers shown in Figure 6a, and (c) Placeholders for uniform gravity load entries into “*zDirectionBeamLoads.txt*” based on identifiers shown in Figure 6a



(a)



(b)

Figure 7 Defining column concentrated gravity loads for 2D models: (a) 1st Story framing plan for example building showing location and labels for columns, (b) Placeholders for concentrated gravity load entries into “*columnLoads.txt*” based on labels shown in Figure 7a

2.1.6 Seismic Design Parameters

The *SeismicDesignParameters* sub-folder is used to define the ASCE 7-10 parameters needed to compute seismic forces. All parameters are entered using scalar values and the text files used to define them carry the same name as the parameters themselves. The following parameters are included:

- Parameters used to compute approximate code period per ACE 7-10 equation 12.8-7, C_b , C_u and x
- Site coefficients F_a and F_v
- The Importance (I), seismic response modification (R) and redundancy (ρ) factors used to compute the design base shear
- The 0.2 and 1.0 second spectral ordinates S_S and S_I
- The long period transition period T_L

2.1.7 Structural Properties

The structural properties used in the OpenSees models are defined in text files located in a series of folders embedded in the *StructuralProperties* sub-folder.

- *Column Hinge Properties* – Columns are modeled as elastic elements with concentrated flexural plastic hinges with the Ibarra-Krawinkler (2005) material to capture nonlinear behavior. The user also has the option to incorporate a zero-length shear spring at the column ends to simulate shear failure due to load transfer from the infill panels. Structural properties associated with the column hinges are defined in a set of text files located in the *ColumnHinges* subfolder. All properties are defined within their respective text files using an $m \times n$ array where m is the number of stories and n is the number of columns per story i.e. each entry represents the property value for a single column. The location of the array entries relative to the column plan location is the same as was defined for the *columnLoads* (Figure 7). The theory behind the column hinge flexural properties can be found in Haselton and Deierlein (2007) and the column shear spring properties can be found in Burton and Deierlein (2014).

- *Moment Parameters* – The moment-rotation response for the column hinge is defined by three moment parameters. The yield moment for bending about the local X and Z axes, M_{yXX} and M_{yZZ} in kip.inches, the ratio of capping moment to yield moment $M_{cOverMy}$ (M_c/M_y). The parameters are defined in text files of the same names.
- *Rotation Parameters* – The moment-rotation response for the column hinge is defined by two rotation parameters. The capping rotation θ_{cap} and the post capping rotation θ_{PC} defined in radians. The parameters are defined in text files of the same names.
- *Cyclic Deterioration Parameter* – Cyclic deterioration for the column hinge moment-rotation response is captured using a single parameter λ which is defined in a text file with the same name.
- *Consideration of Column Shear Failure* – The user has the option to consider column shear failure induced by load transfer from the infill panels. This has been shown to have a significant impact on the collapse performance of infill frame buildings (Burton and Deierlein, 2014). The option to consider column shear failure is entered as a binary value (0 or 1) into the “*considerColumnShearFailure.txt*” text file. If a value ‘1’ is entered, column shear failure is considered and for a value of ‘0’, column shear failure is not incorporated.
- *Column Shear Spring Force-Deformation Parameters* – The force-deformation response for the column shear springs is defined by three parameters: the shear stiffness for loading in the global x- and z-directions, $K_{ShearXX}$ and $K_{ShearZZ}$, in kip/in, the shear strength for loading in the global x- and z-directions, V_{nX} and V_{nZ} in kips and the post-capping shear deformation $\delta_{PCShear}$ in inches. The parameters are defined in text files of the same names.
- *Column Elastic Properties* – Structural properties associated with the elastic column elements are defined in a set of text files located in the *Columns* sub-folder. All except two of the properties are defined within their respective text files using the same format described earlier for the column hinge properties. The three exceptions are the concrete elastic and the product of the shear modulus and torsional moment of inertia. They are defined as scalar values since it is assumed that these values are the same for all columns.
- *Column Section Properties* – The cross sectional area, A , in square inches and the moment of inertia for bending about the local X and Z axes, I_{grXX} and I_{grZZ} in in^4 , are to be defined for each column. The parameters are defined in text files of the same names.
- *Concrete Modulus* – The concrete modulus, E , is defined in kips/ in^2 in the text file of the same names.

- *Shear Modulus times Torsional Moment of Inertia* – For the 3D models, torsional stiffness is assumed to be infinite, therefore the product of shear modulus times the torsional moment of inertia, GJ , is assigned a large number (1e6). While there is a text file that allows the user to change the value of this parameter, the default value should be adequate for all models.
- *Flexural Stiffness* – The elastic flexural stiffness of columns is defined by EI_{eff}/EI_g (EI_{eff}/EI_g) in a text file with the same name.
- *Beam Hinge Properties* – Beams are modeled the same as columns (elastic with plastic hinges) with the exception that there is no option to include shear springs. X- and Z-Beam hinge properties are defined in separate sub-folders ($XBeamHinges$ and $ZBeamHinges$). The relevant parameters and text file names are the same as described above for the column hinges. All properties are defined using an $m \times n$ array where m is the number of suspended floor levels and n is the number of beams (for the relevant direction) per level i.e. each entry represents the property value for a single beam. The location of the array entries relative to the column plan location is the same as was defined for the $xDirectionBeamLoads$ and $zDirectionBeamLoads$ (Figure 6).
- *Beam Elastic Properties* – The elastic properties for the columns described earlier also apply to the beams. X- and Z-Beam properties are defined in separate sub-folders ($XBeams$ and $ZBeams$). The relevant parameters and text file names are the same as described above for the columns. One exception is that the moment of inertia, I_{gtr} , is only defined for uni-axial bending. All except two of the properties (E and GJ) are defined within their respective text files using the same format described earlier for the beam hinge properties.
- *Infill Strut Properties* – The presence and behavior of infill panels are simulated by using two opposing pairs of diagonal compression-only struts, in which their inelastic degrading response is modeled by using a peak-oriented hysteretic material model. In each direction, a diagonal strut is placed between the nodes, representing the beam–column joints, and an off-diagonal strut is used to capture the interaction between the infill and the columns. More details on the strut modeling approach is provided in Burton and Deierlein (2014). X- and Z-Direction infill panel properties are defined in separate sub-folders ($XDirectionInfill$ and $ZDirectionInfill$). The parameters that can vary from panel to panel are defined within their respective text files using an $m \times n$ array where m is the number of stories and n is the number of bays per story for a given direction of interest (including those bays without panels). For example, the building shown in Figures 6 and 7 is a 4-story building with 4 bays in each of the four X-Direction frame lines, therefore the relevant properties will be entered into a 4×16 array. In the Z-Direction, there are 3 bays in each of the five frame lines, therefore the relevant properties will be entered into a 4×15 array. The location of the array entries relative to the plan location is the same as was defined for the $xDirectionBeamLoads$ and $zDirectionBeamLoads$ (Figure 6). The parameters that are the same for every panel are entered as scalar values in their respective text files.
- *Infill Panel Locations* – The location of infill panels within the building is defined using a binary designation in each array entry corresponding to a unique bay. ‘1’ is entered to indicate

the presence of a panel and '0' is entered to indicate the absence of a panel in a particular bay. These values are entered into the "*xInfillLocation.txt*" and "*zInfillLocation.txt*" text files.

- *Infill Strut Yield Force, Initial Stiffness and Length* – The user must define the yield force, *FyStrut*, in kips, the initial stiffness, *KStrut*, in kips/inch and the strut length, *length*, in inches. The values are entered as an array in text files of the same names.
- *Inelastic Parameters* – The parameters that determine the inelastic behavior of the infill strut include the ratio of capping to yield displacement, *deltaCOverDeltaY* (Δ_c/Δ_y), the strain hardening ratio, *strainHardeningRatio*, and the strain softening ratio, *strainSofteningRatio*. These parameters are the same for all struts so they are entered as scalar values into the respective text files.

2.1.8 Baseline .tcl Files for OpenSees Models

The "*BaselineTclFiles*" sub-folder contains a set of *.tcl* files that are used to generate the OpenSees models. These files are accessible to the user but do not require modification.

2.1.9 Building Model Data

The building model parameters which are entered into the text files are placed in MATLAB data structures (structs and objects) and stored in the "*BuildingModelData*" sub-folder (after running the "*AssembleBuildingModelInformation.m*" which is discussed later in the manual), before which the folder only contains the "*saveToStruct.m*" and "*saveToStruct2.m*" MATLAB functions. These functions should not be deleted. The files in the "*BuildingModelData*" sub-folder are accessible to the user but do not require modification.

2.2 MATLAB Executable (*AssembleBuildingModelInformation.m*) for Assembling Building Model Data

Once the relevant building information and model parameters have been entered via text files into the "*BuildingData*" sub-folder, the next step is for the user to run the "*AssembleBuildingModelInformation.m*" MATLAB file. This file is located in the "*UserExecutedFiles*" folder, and is the first of three MATLAB files that is to be run by the user. When this file is executed, the information that was entered into text files is packaged into a set of MATLAB data structures (structs and objects) to be accessed later in the process. Before running it, two lines of code (Lines 31 and 34) must be edited in the "*AssembleBuildingModelInformation.m*" MATLAB file.

- On Line 31, the user must specify the name of the building (Building ID) for which the information is to be assembled. The building name must match the Building ID used in the "*BuildingData*" and "*BuildingModels*" sub-folders. "*Building1*" is used as the ID for the example model provided with the tool.

- On Line 34, the user must define the path to the directory where the main sub-folders (*BuildingData*, *BuildingDataAssembly*, *BuildingModels*, *Documentation*, *OpenSeesModelAssembly* and *UserExecutedFiles*) are located.

Once the modifications have been made, “*AssembleBuildingModelInformation.m*” is ready to be executed, the result of which is the generation of a set of MATLAB structs and objects in the following location: *BuildingData/BuildingID/BuildingModelData*.

3. OPENSEES MODEL CONSTRUCTION (OMC)

The OMC module is contained in the “*BuildingModels*” and “*BuildingModelAssembly*” folders. The “*BuildingModelAssembly*” folder contains the MATLAB functions and classes associated with the OMC module. The user has access, but does not need to interface with these files. The .tcl files that are generated for the OpenSees models are stored in the “*BuildingModels*” folder. Note that an empty folder with the same name as the Building ID entered in Line 31 of “*AssembleBuildingModelInformation.m*” needs to be created before generating the OpenSees models.

The OpenSees models are generated when the user runs the “*ConstructOpenSeesModels.m*” MATLAB file located in the “*UserExecutedFiles*” sub-folder. The same modifications (Lines 31 and 34) that were done to “*AssembleBuildingModelInformation.m*” must be done for “*ConstructOpenSeesModels.m*” prior to running it.

At this point, the “*BuildingModels*” folder should contain both 3D and 2D (for the frame lines specified) models.

4. SEISMIC PERFORMANCE ASSESSMENT

The seismic performance of infill frame buildings are assessed through nonlinear static and dynamic analysis. Three options are included for the dynamic analysis: (1) nonlinear response history analyses where the ground motions are anchored to a single spectral intensity, (2) incremental dynamic analysis where the ground motions are anchored to incrementally increasing spectral intensities and (3) incremental dynamic analysis to collapse, where the ground motion spectral intensities are incrementally increased until collapse occurs. For the nonlinear static analysis, the building is pushed to a pre-defined roof drift (specified in the Building Information Assembly phase).

Executing the OMC module generates a set of .tcl files that represent the OpenSees model for the building. The user has the option of running any of the nonlinear static and dynamic analysis types discussed earlier for the 2D and 3D models. The “*OpenSees3DModels*” sub-folder located in the “*BuildingModels*” folder is also divided into three sub-folders: (1) *EigenValueAnalysis*, (2) *PushoverAnalysis* and (3) *DynamicAnalysis*. The “*OpenSees2DModels*” sub-folder located in the “*BuildingModels*” folder contains the same three sub-folders. However, two additional sets of directories are included to distinguish between the X- and Z-Direction frame line models. The X-Direction frame line models are located in the “*XDirectionFrameLines*” sub-folder and the Z-Direction frame line models are located in the “*ZDirectionFrameLines*” sub-folder. The models associated with each frame line is located in sub-folders labeled “*Line_#*” where # is the frame line number i.e. *Line_1*, *Line_2* etc.

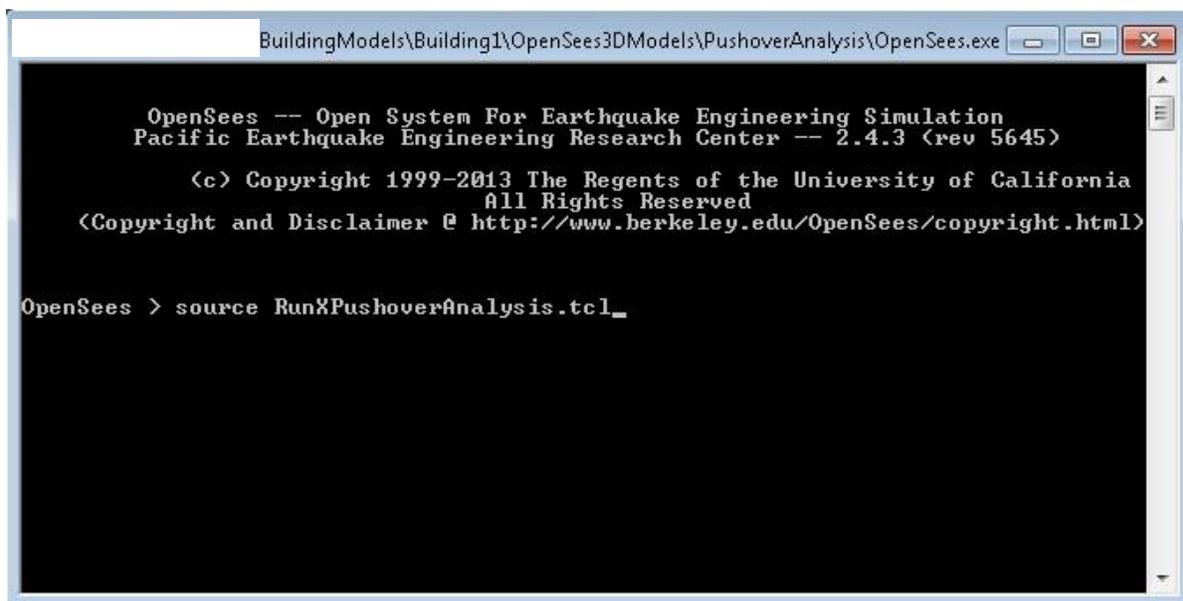
4.1 Eigen Value Analysis

The *EigenValueAnalysis* sub-folder contains the set of *.tcl* files that are concerned with computing the modal periods for the building models. Note that the eigen value analysis is automatically executed within OpenSees during the model construction phase. Therefore, the results are available immediately after the OMC module is executed. The modal periods are stored in an *.out* file that can be accessed in the following location: *EigenValueAnalysis\Analysis_Results\Modes\periods.out*. These periods are also used to define the Rayleigh damping in the dynamic analyses.

4.2 Nonlinear Static (Pushover) Analysis

The *PushoverAnalysis* sub-folder contains the set of *.tcl* files that are concerned with running a nonlinear static analysis in OpenSees. The user has access to these *.tcl* files, however, no modifications are necessary. To run the pushover analysis, the user has to open the OpenSees executable “*OpenSees.exe*”.

For the 3D models, the following command must be entered into the OpenSees executable window (See Figure 8): “*source RunXPushoverAnalysis.tcl*” to run a pushover analysis in the X-Direction and “*source RunZPushoverAnalysis.tcl*” to run a pushover analysis in the Z-Direction. When the analysis is complete, the following message will be displayed in the command window: “Pushover analysis completed SUCCESSFULLY”. The analysis run time will also be displayed in seconds. A set of output files will be generated in the following sub-folder: “*PushoverAnalysis\Static-Pushover-Output-Model3D-XPushoverDirection*” for the X-Direction pushover and “*PushoverAnalysis\Static-Pushover-Output-Model3D-ZPushoverDirection*” for the Z-Direction pushover. The output files include base node reactions, element force and deformation histories, node displacements and story drifts. The “Visualization of Results” module allows the user to automatically generate pushover curves from the result files. However, other types of plots (element force-deformation histories etc.) must be manually generated by the user.



```
BuildingModels\Building1\OpenSees3DModels\PushoverAnalysis\OpenSees.exe
OpenSees -- Open System For Earthquake Engineering Simulation
Pacific Earthquake Engineering Research Center -- 2.4.3 (rev 5645)

(c) Copyright 1999-2013 The Regents of the University of California
All Rights Reserved
(Copyright and Disclaimer @ http://www.berkeley.edu/OpenSees/copyright.html)

OpenSees > source RunXPushoverAnalysis.tcl_
```

Figure 8 OpenSees window showing command needed to run a pushover analysis in the X-Direction

The steps needed to run a pushover analysis for the 2D model is the same as that for the 3D model with the exception that the OpenSees command is “*source RunPushoverAnalysis.tcl*” for both the X- and Z-Direction frame line models. The analysis results are stored in the “*Static-Pushover-Output-Model2D*” sub-folder.

4.3 Nonlinear Dynamic (Single Intensity, IDA and IDA to Collapse) Analysis

The *DynamicAnalysis* sub-folder contains the set of *.tcl* files that are concerned with running a nonlinear dynamic analysis in OpenSees. The user has access to these *.tcl* files, however, no modifications are necessary. To run the dynamic analysis, the user has to open the OpenSees executable “*OpenSees.exe*”.

For the 3D models, the following commands must be entered into the OpenSees executable window (See Figure 9): (1) “*source RunSingleScale3DModelFinal.tcl*” to run the single intensity dynamic analyses, (2) “*source RunIDA3DModelFinal.tcl*” to run incremental dynamic analysis for predefined spectral intensities and (3) “*source RunIDAToCollapse3DModelFinal.tcl*” to run incremental dynamic analysis to collapse.

When the analysis is complete, the following message will be displayed in the command window: “Analysis Completed”. The analysis run time will also be displayed in minutes. A set of output files will be generated in the following sub-folder: “*ModelSingleScaleOutputBiDirection*” for the single intensity dynamic analyses, “*ModelIDAOutputBiDirection*” for incremental dynamic analysis for predefined spectral intensities and “*RunIDAToCollapse3DModelFinal*” for incremental dynamic analysis to collapse. As with the pushover analysis, the output files include base node reactions, element force and deformation histories, node displacements and story drifts. Node accelerations are also included in the dynamic analysis results. There are sub-folders (EQ_1, EQ_2 etc.) for analysis results associated with each ground motion. For the incremental dynamic analysis to collapse, two additional sub-folders are included, “*IDAGMScales*” and “*IDAMaxDrifts*”. The “*IDAGMScales*” folder contains the ground motion spectral intensities that were run (defined as a percentage of the MCE spectral intensity) for each ground motion in individual text files (EQ_1.txt, EQ_2.txt etc.). The “*IDAMaxDrifts*” folder contains the maximum story drift ratio corresponding to each spectral intensity for the ground motion spectral intensities that were run (defined as a percentage of the MCE spectral intensity) for each ground motion in individual text files (EQ_1.txt, EQ_2.txt etc.). The “Visualization of Results” module allows the user to automatically generate collapse fragility plots from the result files. However, other types of plots (story drift profiles for single intensity analysis, IDA plots etc.) must be manually generated by the user.

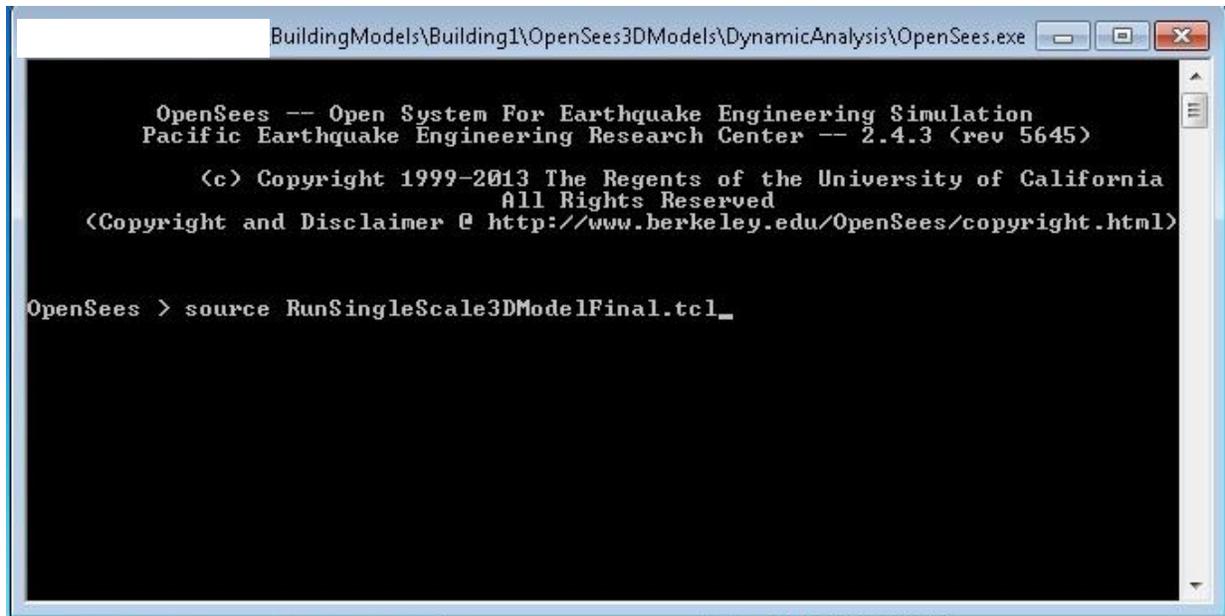


Figure 9 OpenSees window showing command needed to run single intensity dynamic analyses for 3D model

The steps needed to run a dynamic analysis for the 2D model is the same as that for the 3D model with the exception that the OpenSees command is (1) “*source RunSingleScale2DModelFinal.tcl*” to run the single intensity dynamic analyses, (2) “*source RunIDA2DModelFinal.tcl*” to run incremental dynamic analysis for predefined spectral intensities and (3) “*source RunIDAToCollapse2DModelFinal.tcl*” to run incremental dynamic analysis to collapse. The analysis results are stored in the “*ModelSingleScaleOutput*”, “*ModelIDAOutput*” and “*ModelCollapseOutput*” folders respectively.

4.4 Parallel OpenSees (OpenSeesMP) for Nonlinear Dynamic Analyses

SPA-INFILL supports the use of OpenSeesMP to incorporate parallel computing in performing dynamic analyses. This requires the user to have access to a desktop or remote workstation with multiple cores. Nonlinear response history analyses are parallelized by assigning runs for multiple ground motions to individual cores. Take for example a case where the user wants to run a single intensity analysis using 12 ground motions and has 12 available cores. Using OpenSeesMP, the analysis for each ground motion is assigned to each of the 12 cores allowing for a reduced run time. More details on parallel OpenSees including instructions for downloading and using OpenSeesMP is available on the OpenSees website (<http://opensees.berkeley.edu/OpenSees/parallel/parallel.php>).

To run a nonlinear response history analyses using OpenSeesMP, the following commands must be entered into the Command Prompt window (See Figure 10): “*mpiexec -np numProc OpenSeesMP RunSingleScale3DModelFinal.tcl*” to run the single intensity dynamic analyses. “*mpiexec*” is an MPICH2 or OpenMPI application that need to be installed on the user’s machine (see OpenSees website for more information). “*numProc*” is the number of cores on the user’s machine. Note that the directory should point to the location of the OpenSees model.

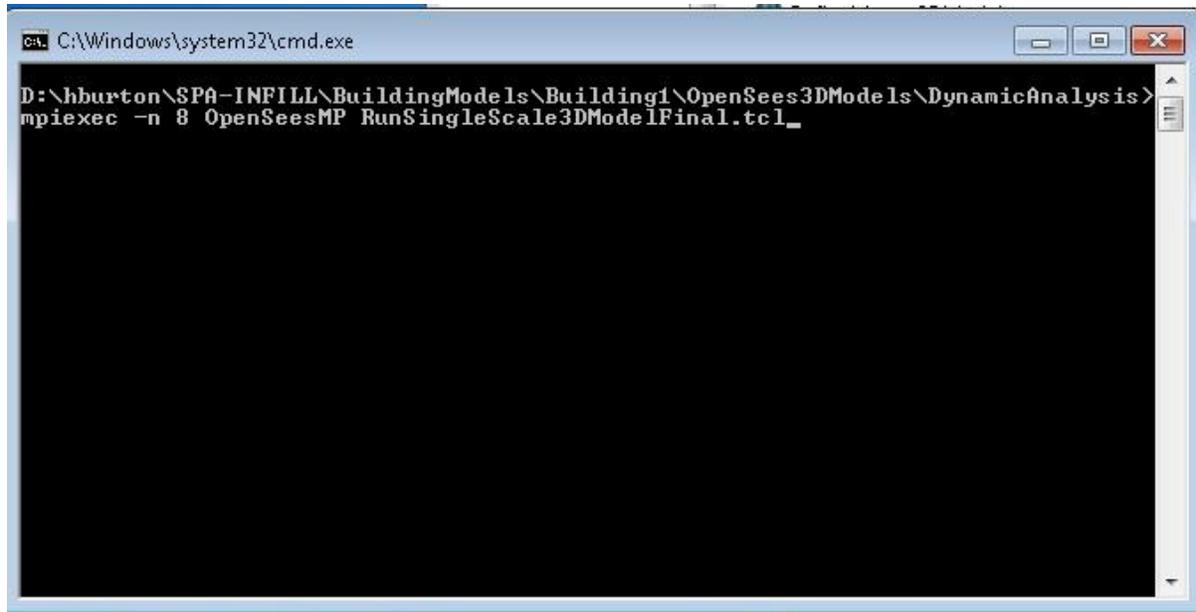


Figure 10 Command prompt window showing command needed to run single intensity dynamic analyses for 3D model using OpenSeesMP utilizing 8 cores

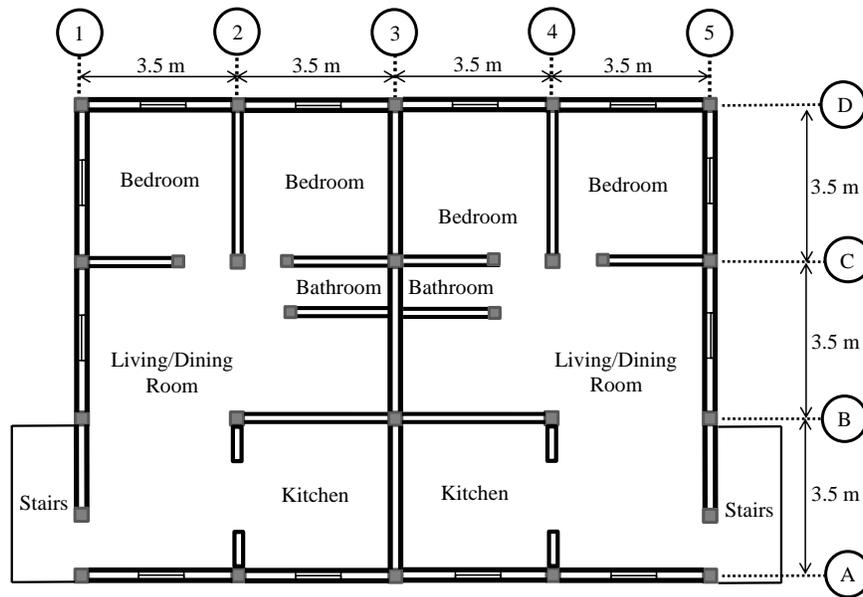
5. VISUALIZATION OF RESULTS

The current version of SPA-INFILL allows the user to automatically generate pushover plots from nonlinear static analyses and collapse fragility curves from incremental dynamic analyses. Other types of plots will need to be generated manually by the user. The VOR module is contained in the “*AnalysisPlots*” and “*GenerateAnalysisPlots*” folders. The “*GenerateAnalysisPlots*” folder contains the MATLAB functions and classes associated with the VOR module. The user has access, but does not need to interface with these files. The analysis plots that are generated by executing the VOR module are stored in the “*AnalysisPlots*” folder. Note that an empty folder with the same name as the Building ID entered in Line 31 of “*AssembleBuildingModelInformation.m*” needs to be created before generating the plots. The collapse fragility curves and pushover plots are generated when the user runs the “*GenerateAnalysisPlots.m*” MATLAB file located in the “*UserExecutedFiles*” sub-folder. The same modifications (Lines 31 and 34) that were done to “*AssembleBuildingModelInformation.m*” must be done for “*UserExecutedFiles.m*” prior to running it. In addition to this, the user must indicate what types of plots they would like to be generated (pushover and/or collapse fragility) and for which models (2D and/or 3D models). Note that for the pushover analysis plots, the base shear is normalized by the seismic weight. For the 2D models, the seismic weight is computed based on the defined “*xFrameLinesTributarySeismicMassRatios*” and “*zFrameLinesTributarySeismicMassRatios*”.

6. EXAMPLE USING SPA-INFILL: SEISMIC PERFORMANCE ASSESSMENT OF A 4-STORY INFILL FRAME BUILDING

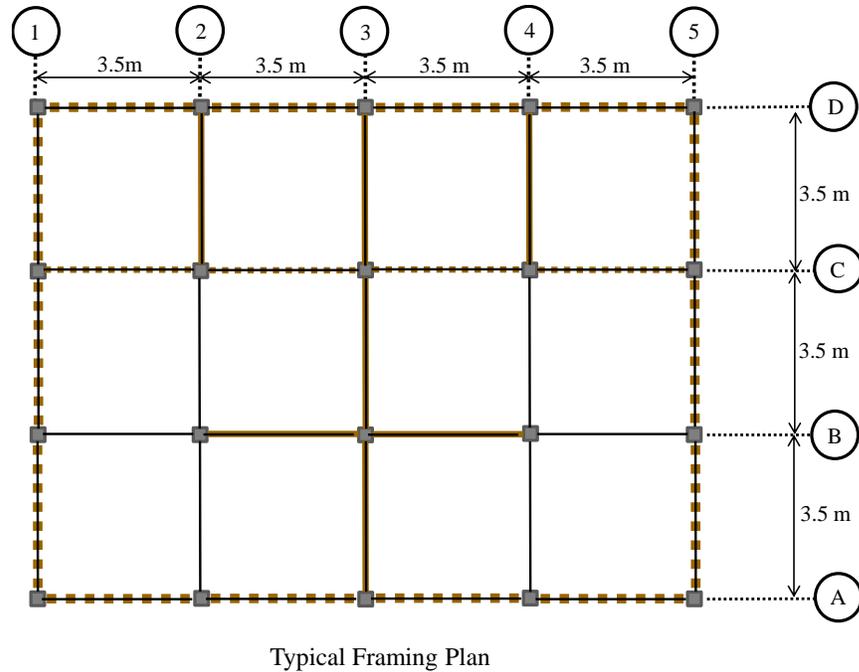
6.1 Example Building Description

The capabilities of SPA-INFILL are illustrated through a comprehensive step-by-step example that involves the seismic performance of a 4-story residential building with infill frames. The building was designed for gravity loads only. The architectural layout and structural plan is shown in Figure 11. There are two 2-bedroom units on each of the four stories, each with 73.5 m² of space and 2 to 4 occupants. The overall floor plan has four bays in the longitudinal direction and three in the transverse direction. The building has a uniform grid spacing of 3.5 m and all stories are 3.5 m in height. Floor weights are typically 1480 kN, with the exception of the roof level, which is 1170 kN. All beams are 305 mm wide and 355 mm deep with 3-#4 longitudinal bars top and bottom and #3 stirrups spaced at 305 mm. All columns are 355 mm square with 8-#4 longitudinal bars and #3 stirrups spaced at 305 mm. • Infill panels are constructed using manually fabricated bricks with dimensions 230 mm long x 110 mm wide x 70 mm thick. Interior panels are constructed using a single-wythe of infill and exterior panels are constructed as double-wythe walls. The prism strength of infill panels is 3.5 MPa and the modulus is 17.5 MPa.



Typical Floor Plan

(a)



(b)

Figure 11 4-story residential building (a) architectural layout and (b) structural floor plan

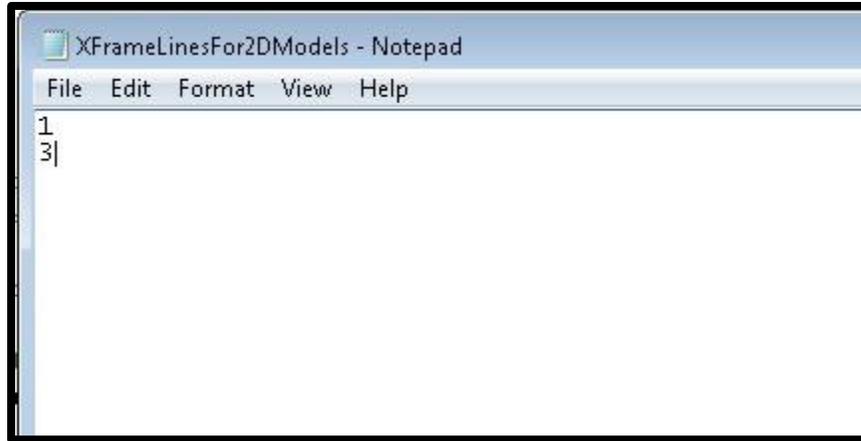
6.2 Example Building Structural Modeling

2D and 3D models of the example building are constructed in OpenSees using SPA-INFILL. Beams and columns are idealized using elastic elements with concentrated flexural plastic hinges with the Ibarra-Krawinkler (2005) material to capture nonlinear behavior. The parameters for the plastic hinges of the reinforced beams and concrete columns are obtained using the predictive equations developed by Haselton et al. (2008). Infill panels are modeled using a pair of inelastic compression-only struts that incorporate strength and stiffness deterioration. The guidelines developed by Burton and Deierlein (2014) are used to obtain the model parameters including the capping strength, F_c , the initial stiffness, K_e , the ratio of capping strength to yield strength, F_c/F_y , the ratio of capping displacement to yield displacement, Δ_c/Δ_y , and the ratio of post-capping stiffness to yield stiffness, K_c/K_e . A leaning column is incorporated into the 2D models to consider P-Delta effects.

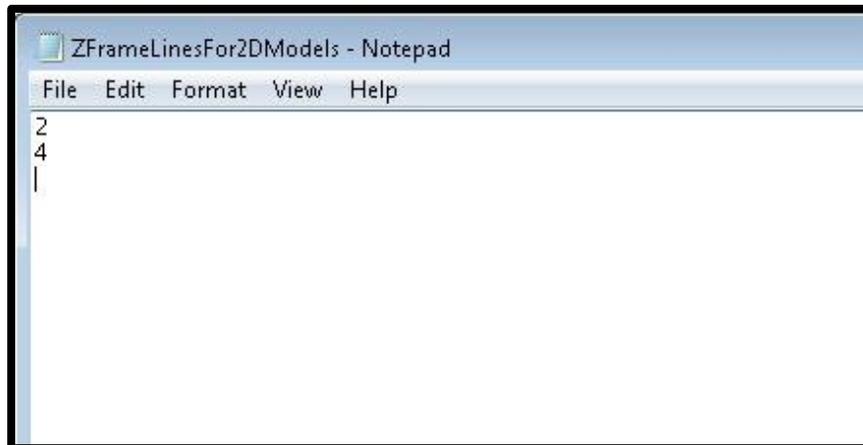
6.3 Building Information and Model Parameter Entry for 4-Story Infill Frame Building

6.3.1 Analysis Options Entry

A 3D model will be constructed for the example building as well as 2D models for both the X- and Z-Direction frame lines. For the X-Direction, models will be constructed for frame lines 1 and 3, and in the Z-Direction, models will be constructed for lines 2 and 4. The entries for the “XFrameLinesFor2DModels.txt” and “ZFrameLinesFor2DModels.txt” text files are shown in Figure 12.



(a)



(b)

Figure 12 Frame lines for (a) X- and (b) Z-Direction 2D models

6.3.2 Analysis Parameters Entry

The following static analysis parameters are used for the example building

- Pushover analysis increment size – A default value of 0.01 is entered into the “*PushoverIncrementSize.txt*” text file.
- Maximum Drift – A maximum roof drift of 1% is entered (entered as 1) into the “*PushoverXDrift.txt*” and “*PushoverZDrift.txt*” text files.
- An MCE Spectral Intensity of 0.6 g (entered into the “*SamCE.txt*” text file) is assumed.
- IDA Intensity Levels ranging from 20 to 100% MCE using 20% increments (Figure 13) is entered into the “*scalesForIDA.txt*” text file.

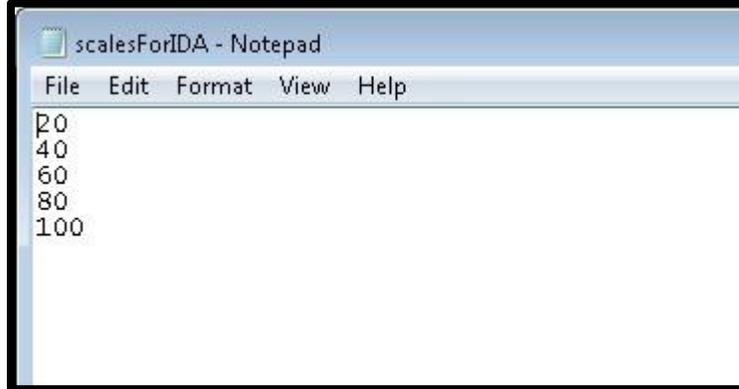


Figure 13 Spectral intensities (entered as a percentage of the MCE level) used for IDA

- The DBE intensity level (entered as 67 into the “*SingleScaleToRun.txt*” text file) is used for the Single Scale Dynamic Analysis.
- A scale factor of 0.83 is used (into the “*UniDirectionMCEScaleFactor.txt*” text file) is used to anchor the median spectral intensity of the ground motion set to the MCE hazard level.

Note that the FEMA P695 far field set is used for the example model. This suite consists of 22 pairs of ground motions (44 total). For the 2D models with uni-directional loading, the 44 individual ground motions are used. For the 3D model, the 22 pairs of bi-directional motions are used.

- The scale factors needed to anchor the ground motion pairs to the MCE hazard level (for Bi-Directional Inertial Loading) is shown in Figure 14 (entered as a single column vector into the “*BiDirectionMCEScaleFactors.txt*” text file).

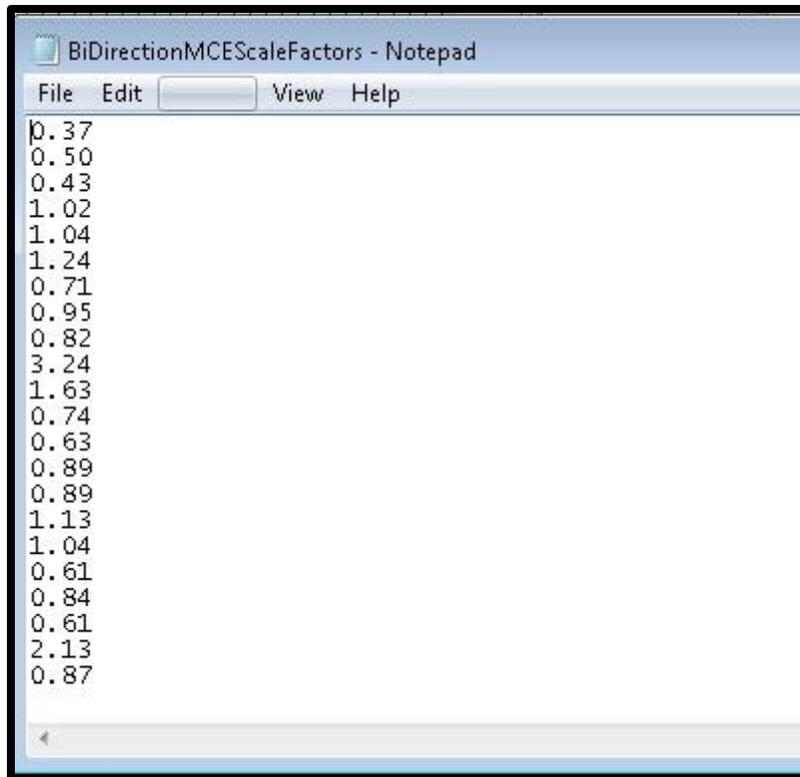


Figure 14 – Scale factors used for bi-directional loading in 3D model

- A 10% drift threshold (entered as 0.1 in the “*CollapseDriftLimit.txt*” text file) is assumed for the collapse criteria.
- The initial spectral intensity increment for the collapse performance assessment is set to 40% of the MCE intensity (entered as 40 in the “*InitialGroundMotionIncrementScaleForCollapse.txt*” text file).
- The reduced spectral intensity increment after initial collapse is set to 5% of the MCE intensity (entered as 5 in the “*ReducedGroundMotionIncrementScaleForCollapse.txt*” text file).
- 22 pairs of ground motions is used for bi-directional inertial loading for the 3D model (entered into the “*NumberOfBiDirectionGroundMotions.txt*” text file) and 44 ground motions are used for uni-directional loading for the 2D model (entered into the “*NumberOfUniDirectionGroundMotions.txt*” text file).

6.3.3 Dynamic Properties

5% damping is assumed in the example model.

6.3.4 Geometry

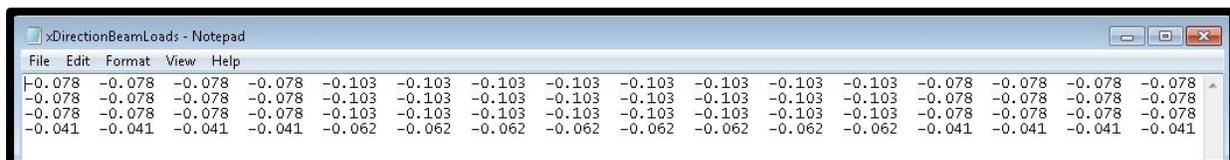
The following geometry parameters are used:

- The number of stories (4) is entered in the “*numberOfStories.txt*” text file.
- The example building has 4 bays in the X-Direction (entered in the “*numberOfXBays.txt*” text file) and 3 in the Z-Direction (entered in the “*numberOfZBays.txt*” text file).
- All stories are 12'-0” in height (entered as a single column vector in the “*storyHeights.txt*” text file).
- 12'-0” bays are used in the X- (entered as a single column vector into the “*XBayWidths.txt*” text file) and Z-Direction bays (entered into the “*ZBayWidths.txt*” text file).
- Rocking of shallow foundations is not considered (0 is entered in the “*foundationUplift.txt*” text file).

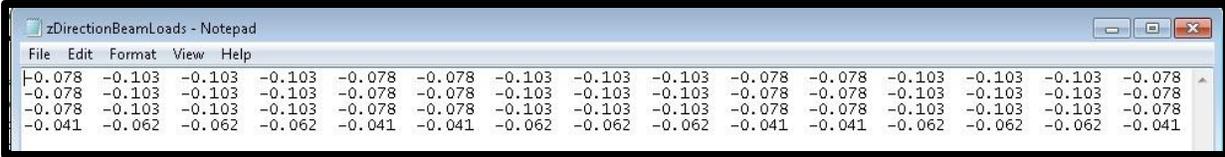
6.3.5 Loads

The following load parameters are used for the example building model.

- All floor weights are 333 kips with the exception of the roof level which is 263 kips (entered as a column vector in the “*floorWeights.txt*” text file).
- The area dead load for all above ground floors is 0.0014 kips per square inch (200 psf) with the exception of the roof level which has a dead load of 0.0012 kips per square inch (160 psf) (entered as a column vector in the “*floorDeadLoad.txt*” text file). The area live load for all above ground floors is 0.000347 kips per square inch (50 psf) with the exception of the roof level which has a live load of 0.000139 kips per square inch (20 psf) (entered as a column vector in the “*floorLiveLoad.txt*” text file).
- The uniform expected gravity loads used for beams (in kip/in) is showed in Figure 15 (entered in the “*xDirectionBeamLoads.txt*” and “*zDirectionBeamLoads.txt*” text files).



(a)



(b)

Figure 15 Uniform expected gravity loads used for beams in example building

- The concentrated gravity loads used for the 2D model (kips) is shown in Figure 16 (entered in the “columnLoads.txt” text file).

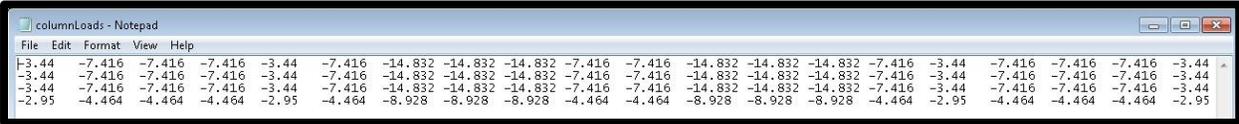
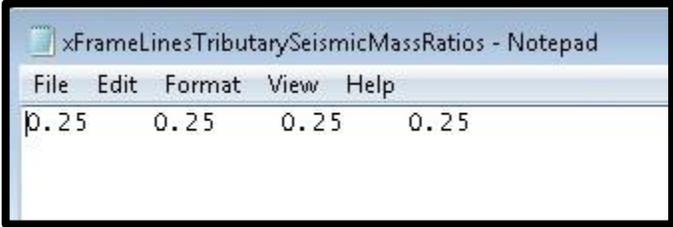
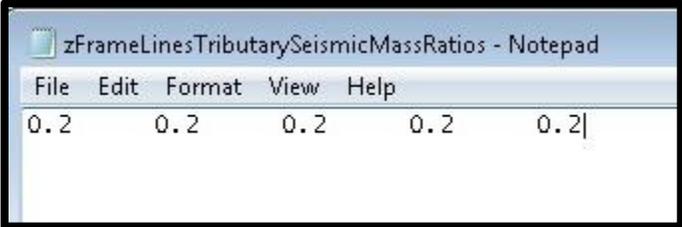


Figure 16 Expected concentrated gravity loads used for columns in 2D model

- 25% of the floor mass will be assumed to be tributary to each frame line in the X-Direction (entered in the “xFrameLinesTributarySeismicMassRatios.txt” text file) and 20% is assumed for the Z-Direction frame lines (entered in the “zFrameLinesTributarySeismicMassRatios.txt” text file). The entries are shown in Figure 17.



(a)

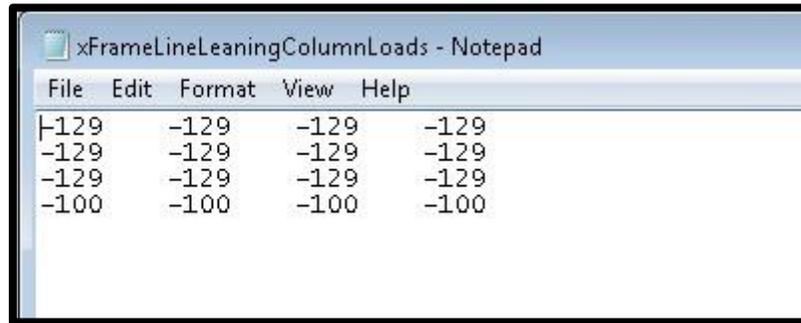


(b)

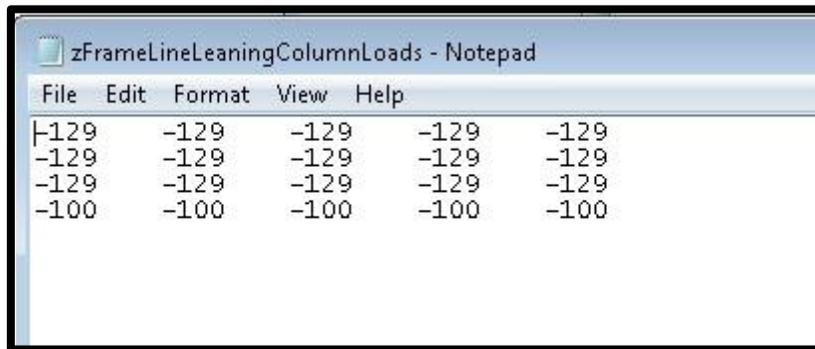
Figure 17 Fraction of tributary floor mass assigned to frame line in 2D models

- The leaning column loads for the X-Direction 2D models (in kips) defined in the “xFrameLineLeaningColumnLoads.txt” text file is shown in Figure 18a. The loads for the Z-

Direction models is defined in the “zFrameLineLeaningColumnLoads.txt” text file and shown in Figure 18b.



(a)



(b)

Figure 18 Leaning column loads for 2D models

6.3.6 Seismic Design Parameters

The following seismic design parameters are used for the example building model:

- $C_t = 0.02$, $C_u = 1.4$ and $x = 0.75$ (used to compute the ASCE 7-10 code period)
- The site coefficients $F_a = 1.0$ and $F_v = 1.5$ are used.
- The importance factor $I = 1.0$, seismic response modification $R = 5$ and redundancy factor $\rho = 1.0$
- The spectral ordinates $S_S = 0.6$ and $S_I = 0.2$ are used
- The long period transition period $T_L = 8$

6.3.7 Structural Properties

The following column hinge properties are used in the OpenSees models for the example building:

- The moment-rotation response parameters for the column hinge (My_{XX} , My_{ZZ} , $McOverMy$, $thetaCap$ and $thetaCap$) are defined in text files of the same names using the array layout shown in Figure 16.
- The cyclic deterioration parameter for the column hinge moment-rotation response ($lambda$) is defined in a text file with the same name using the array layout shown in Figure 16.
- Column shear failure is considered (a value of 1 is entered into the “*considerColumnShearFailure.txt*” text file).
- The force-deformation response for the column shear springs ($KShear_{XX}$, $KShear_{ZZ}$, VnX , VnZ and $deltaPCShear$) are defined in text files of the same names using the array layout shown in Figure 16.

The following elastic column properties are used in the OpenSees models for the example building:

- The elastic column parameters (A , I_{grXX} , I_{grZZ} , and $EI_{effOverEIg}$) are defined in text files of the same names using the array layout shown in Figure 16.
- An elastic modulus, E of 3122 ksi is defined in a text file of the same name
- The product of shear modulus times the torsional moment of inertia, GJ , is assigned a large number (1e6) a text file of the same name

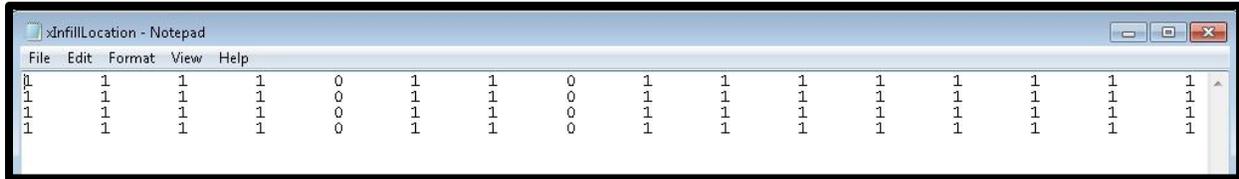
The beam hinge moment-rotation response and cyclic deterioration parameters (My_{XX} , My_{ZZ} , $McOverMy$, $thetaCap$, $thetaCap$ and $lambda$) are defined in text files of the same names using the array layout shown in Figure 15.

The following elastic beam parameters are used in the OpenSees models for the example building:

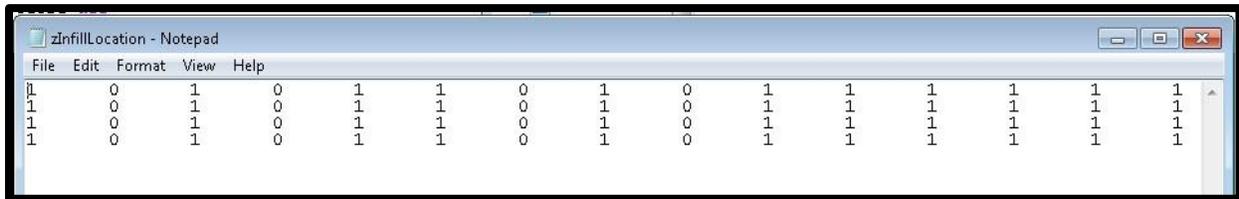
- The elastic column parameters (A , I_{grXX} , I_{grZZ} , and $EI_{effOverEIg}$) are defined in text files of the same names using the array layout shown in Figure 15.
- An elastic modulus, E of 3122 ksi is defined in a text file of the same name
- The product of shear modulus times the torsional moment of inertia, GJ , is assigned a large number (1e6) a text file of the same name.

The following infill strut parameters are used in the OpenSees models for the example building:

- The location of the X- and Z-Direction infill panels in the example building is shown in Figure 19 as defined in the “*xInfillLocation.txt*” and “*zInfillLocation.txt*” text files.



(a)



(b)

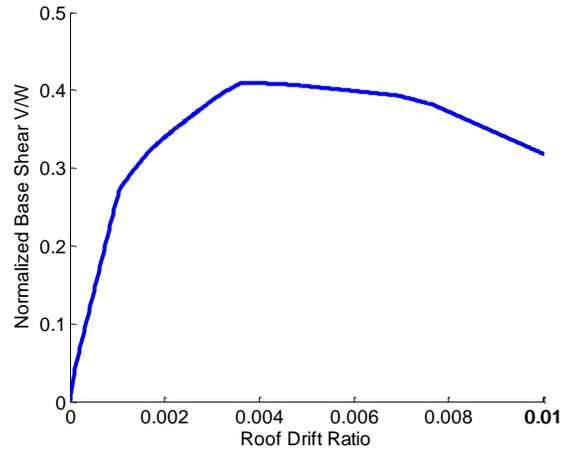
Figure 19 Infill panel locations in example building

- The force-deformation parameters for the X- and Z-Direction infill panels (*FyStrut*, *KStrut* and *length*) are defined in text files with the same name using the array layouts shown in Figure 19.
- A ratio of capping to yield displacement of 5.2 (defined as a scalar value in the “*deltaCOverDeltaY.txt*” text file) is used for the example building model.
- A strain hardening ratio of 0.0952 (defined as a scalar value in the “*strainHardeningRatio.txt*” text file) is used for the example building model.
- A strain softening ratio of -0.035 (defined as a scalar value in the “*strainSofteningRatio.txt*” text file) is used for the example building model.

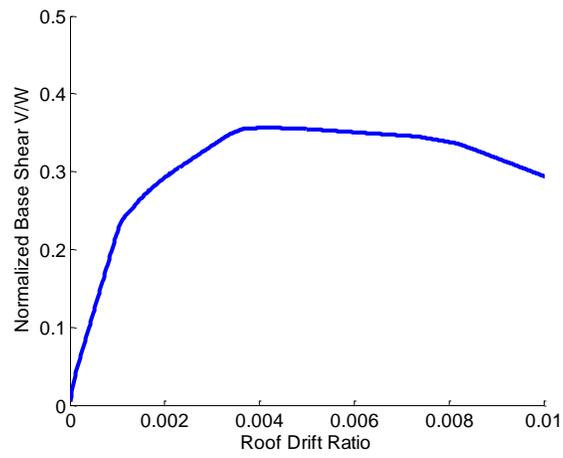
6.4 Seismic Performance Assessment Results

6.4.1 Pushover Analysis

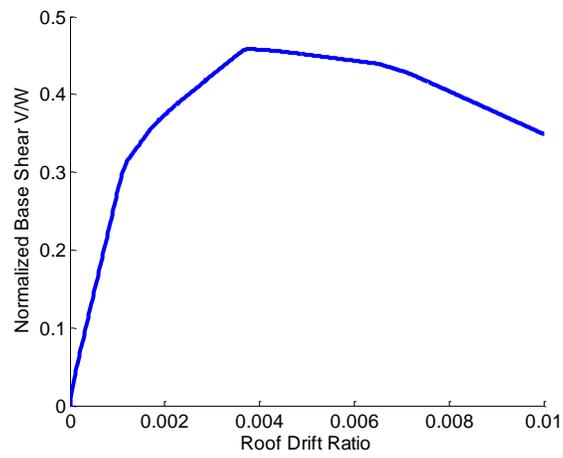
Figure 20 shows the pushover response plots for the example building models including the 3D model for loading in the X- and Z-Direction, 2D models for Lines 1 and 3 in the X-Direction and Lines 2 and 4 in the Z-Direction. All models were pushed to a maximum roof drift ratio of 0.01. The base shear in the pushover curve was normalized by the seismic weight. For the 2D models, the seismic weight used to normalize the base shear is based on the tributary seismic mass used in the dynamic analysis.



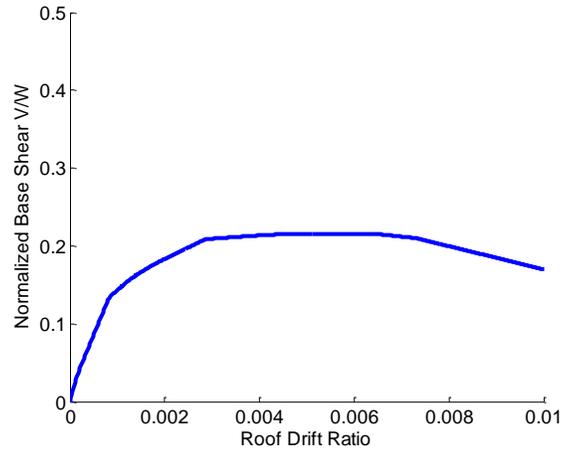
(a)



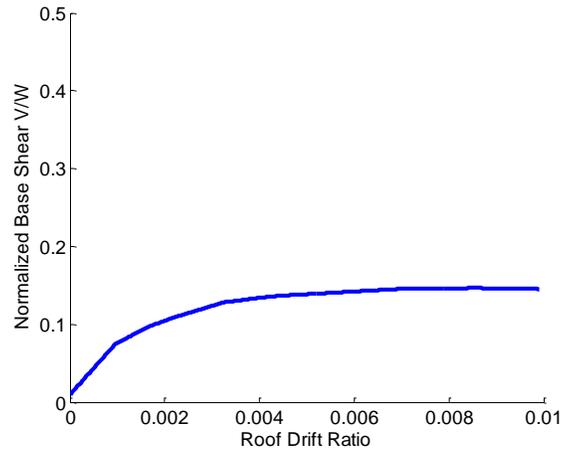
(b)



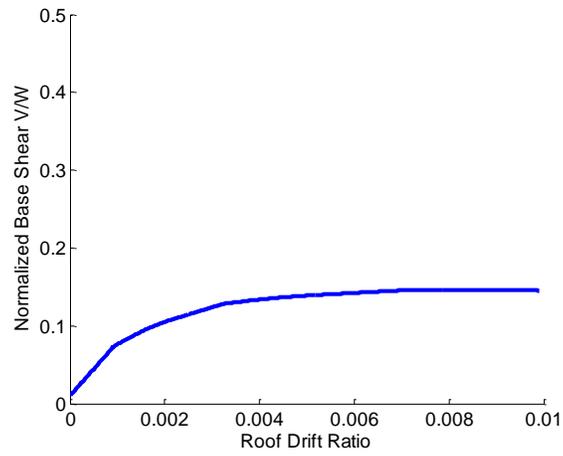
(c)



(d)



(e)

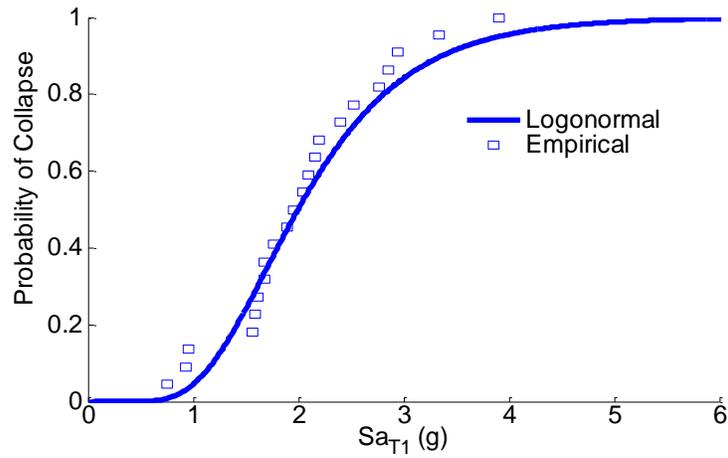


(f)

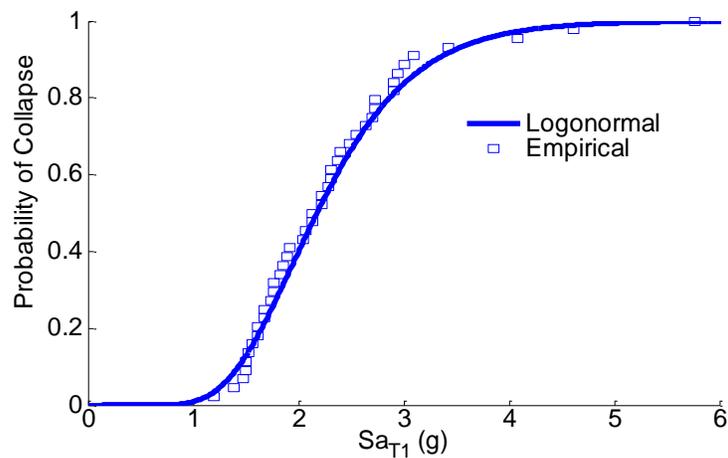
Figure 20 Pushover response for example building model: (a) 3D model X-Direction, (b) 3D model Z-Direction, (c) 2D model X-Direction Frame Line 1, (d) 2D model X-Direction Frame Line 3, (e) 2D model Z-Direction Frame Line 2 and (f) 2D model Z-Direction Frame Line 4

6.4.2 Collapse Performance Assessment

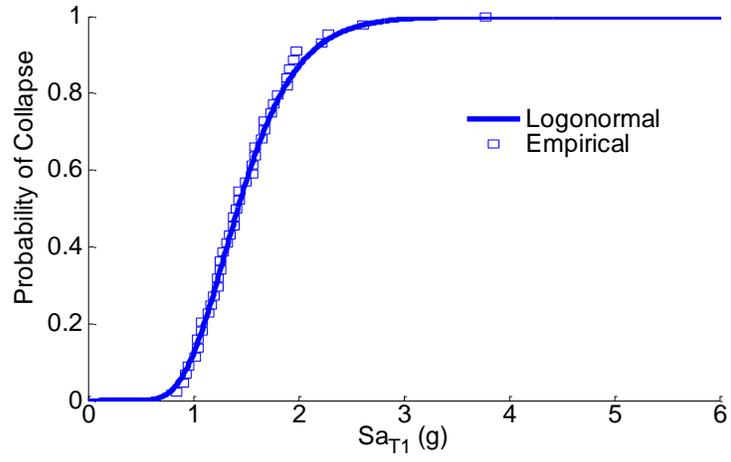
The collapse fragility curves for the example building models is shown in Figure 21. This includes the 3D model with bi-directional loading and the 2D models for Lines 1 and 3 in the X-Direction and Lines 2 and 4 in the Z-Direction with uni-directional loading. The 1st mode spectral intensity is used as the intensity measure.



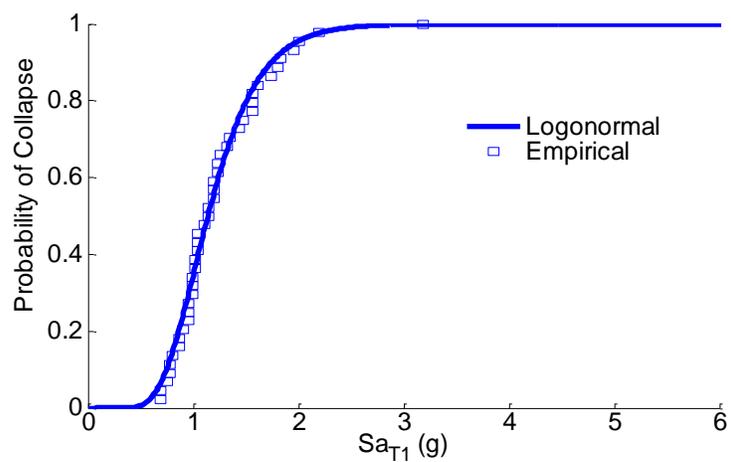
(a)



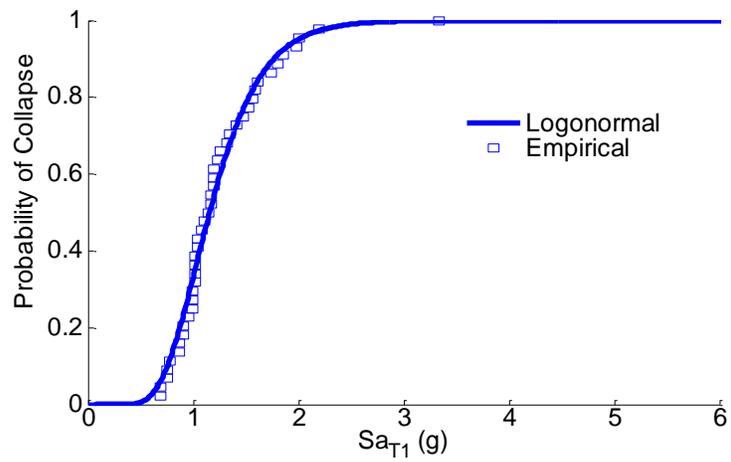
(b)



(c)



(d)



(e)

Figure 21 Collapse fragility curves for example building model: (a) 3D model with bi-directional loading, (b) 2D model X-Direction Frame Line 1, (c) 2D model X-Direction Frame Line 3, (d) 2D model Z-Direction Frame Line 2 and (e) 2D model Z-Direction Frame Line 4