



*Enabling the Network for
Earthquake Engineering Simulation*



TR-2009-[ID]

OpenFresco Framework for Hybrid Simulation: Portal Frame with Gravity Load Example

Andreas Schellenberg, Hong K. Kim, Yoshikazu
Takahashi, Gregory L. Fenves, and Stephen A. Mahin

Department of Civil and Environmental Engineering,
University of California, Berkeley

Last Modified: 2009-08-12 Version: 2.6

Table of Contents

1	Introduction: Portal Frame with Gravity Load Example Using OpenSees	3
2	Required Files.....	3
3	Structural Model.....	3
4	Ground Motion	4
5	OpenFresco Tcl Commands	5
5.1	Experimental Control.....	5
5.2	Experimental Setup	6
5.3	Experimental Site.....	7
5.4	Experimental Element.....	7
6	Running the Local Hybrid Simulation	8
7	Results	11
8	References	14

Table of Figures

Figure 1:	OpenSees Portal Frame Model.....	4
Figure 2:	1978 Tabas Ground Motion.	5
Figure 3:	OneActuator Setup.....	6
Figure 4:	beamColumn Experimental Element.	8
Figure 5:	Local Hybrid Simulation using OpenSees.	8
Figure 6:	OpenSees Command Window for Local Test.....	9
Figure 7:	OpenSees Command Window for Local Test after Simulation.....	10
Figure 8:	Displacements vs. Time for Portal Frame Example.....	12
Figure 9:	Velocities vs. Time for Portal Frame Example.....	12
Figure 10:	Accelerations vs. Time for Portal Frame Example.....	13
Figure 11:	Forces vs. Time for Portal Frame Example without Gravity Loads.	13
Figure 12:	Forces vs. Time for Portal Frame Example with Gravity Loads.	13
Figure 12:	Column Hysteresis Loops for Portal Frame Example.....	14
Figure 14:	Actuator Hysteresis Loops for Portal Frame Example.	14



1 Introduction: Portal Frame with Gravity Load Example Using OpenSees

One of the advantages in hybrid simulation is that geometric nonlinearities, three-dimensional effects, multiple support excitation and soil-structure interaction can be investigated by incorporating them into the analytical portion of the hybrid model. To demonstrate how geometric nonlinearities can be accounted for in the numerical part of the model, a hybrid simulation, wherein a portal frame is tested by consistently accounting for second-order effects due to gravity loads, is carried out using OpenSees and OpenFresco. This example explains how to run this fairly complex local hybrid simulation of such portal frame model. The two portal frame columns are axially loaded and simulated experimentally through OpenFresco. The axial forces and the large P-Delta ($P-\Delta$) effects are accounted for in the numerical model. Thus, the experimental setups impose no axial forces on the specimens, just displacements. The Portal Frame example uses the `NewmarkHSTFixedNumIter` time integration scheme with a fixed number of 5 sub-steps. This example is a fully simulated test, meaning that no physical specimens are required. The response results from the simulation are provided for comparison.

2 Required Files

For the Portal Frame example, the following files are required. These are located in:

User's Directory\OpenFresco\trunk\EXAMPLES\PortalFrame

if OpenFresco was installed in the default location, the User's Directory is `C:\Program Files`.

The following Tcl files should be in this directory:

- `PortalFrame_Local.tcl`
- `SACNF01.txt`

The OpenSees executable and Tcl/Tk 8.5.x are required to run this example. If not done so already, they can freely be downloaded from the [OpenSees website](http://opensees.berkeley.edu/OpenSees/user/download.php) (<http://opensees.berkeley.edu/OpenSees/user/download.php>). Follow the directions carefully on this website.

3 Structural Model

The portal frame is modeled in OpenSees using the Tcl file, `PortalFrame_Local.tcl`. It consists of two vertical experimental beam-column elements and one horizontal analytical beam-column element (Figure 1). The two experimental beam-columns are axially loaded. Lumped masses are placed at nodes 3 and 4. The gravity loads can be turned off or on by setting the parameter `withGravity` to either 0 or 1. The bases of the experimental beam-columns are pinned. The following Tcl script from `PortalFrame_Local.tcl` defines the geometry of the model:

```
# -----
# Start of model generation
# -----
# create ModelBuilder (with two-dimensions and 3 DOF/node)
model BasicBuilder -ndm 2 -ndf 3
```



```

# Load OpenFresco package
# -----
# (make sure all dlls are in the same folder as openSees.exe)
loadPackage OpenFresco

# Define geometry for model
# -----
set withGravity 1;
set Pc 10.638;
set P [expr 0.5*$Pc];
set mass3 [expr $P/386.1];
set mass4 [expr $P/386.1];
# node $tag $xCrd $yCrd $mass
node 1 0.0 0.0
node 2 100.0 0.0
node 3 0.0 50.0 -mass $mass3 $mass3 0.0
node 4 100.0 50.0 -mass $mass4 $mass4 0.0

# set the boundary conditions
# fix $tag $DX $DY $RZ
fix 1 1 1 0
fix 2 1 1 0

```

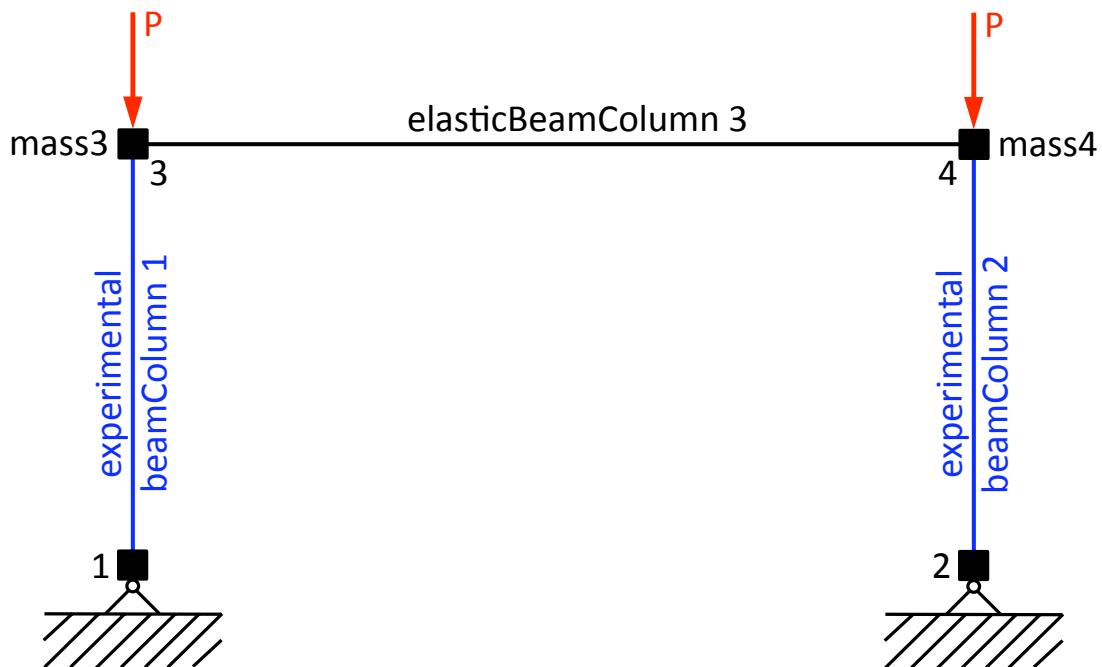


Figure 1: OpenSees Portal Frame Model.

4 Ground Motion

The structure is subjected horizontally to the 1978 Tabas earthquake scaled to a peak ground acceleration of 0.906g. The file, SACNF01.txt, contains the acceleration data recorded at every 0.01 seconds (Figure 2).



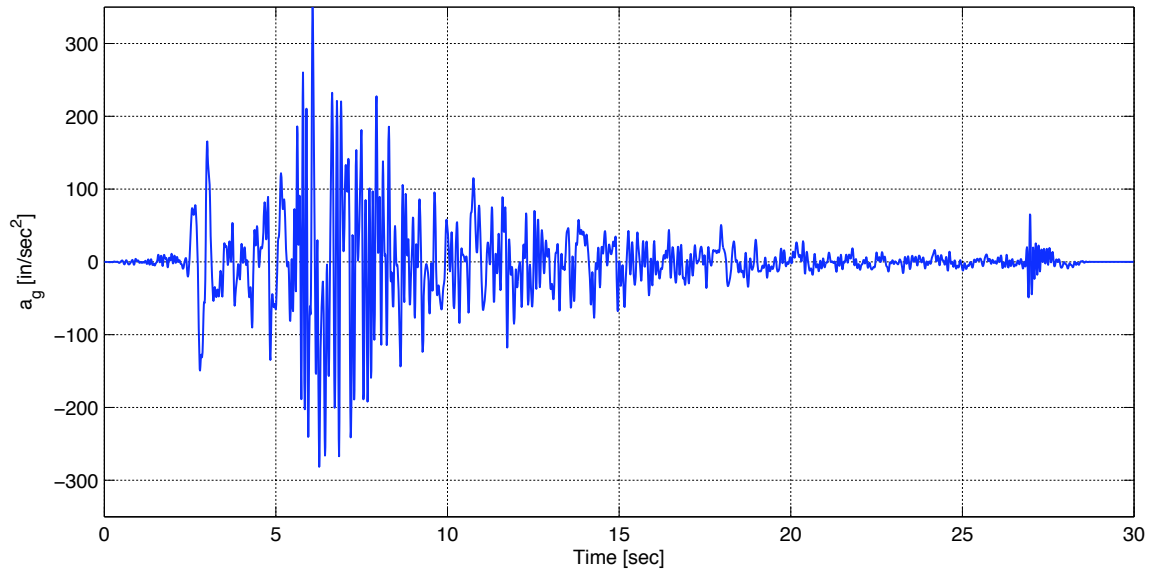


Figure 2: 1978 Tabas Ground Motion.

5 OpenFresco Tcl Commands

This section contains explanations of the OpenFresco Tcl commands used in this example. Each subsection highlights an OpenFresco Tcl command and the script that contains the command. All Tcl script excerpts are from `PortalFrame_Local.tcl`.

```
# Load OpenFresco package
# -----
# (make sure all dlls are in the same folder as openSees.exe)
loadPackage OpenFresco
```

The `loadPackage OpenFresco` command is necessary for the examples to execute properly.

5.1 Experimental Control

Both experimental controls are set to `SimUniaxialMaterials` for this example. The `SimUniaxialMaterials` controls use a previously defined `Steel02` material with a `matTag` 1, to simulate the responses of the experimental elements.

```
# Define materials
# -----
# uniaxialMaterial Steel02 $matTag $Fy $E $b $R0 $cR1 $cR2 $a1 $a2 $a3 $a4
uniaxialMaterial Steel02 1 1.5 2.8 0.01 18.5 0.925 0.15 0.0 1.0 0.0 1.0
#uniaxialMaterial Elastic 1 2.8

# Define experimental control
# -----
# expControl SimUniaxialMaterials $tag $matTags
expControl SimUniaxialMaterials 1 1
#expControl xPCTarget 1 1 "192.168.2.20" 22222 HybridControllerD3D3_1Act
"D:/PredictorCorrector/RTActualTestModels/cmAPI-xPCTarget-STS"
expControl SimUniaxialMaterials 2 1
```



The `expControl` command parameters for `SimUniaxialMaterials` are:

- `$tag` is the unique control tag.
- `$matTags` are the tags of previously defined uniaxial material objects.

5.2 Experimental Setup

The `OneActuator` setup is being used for both experimental setups (Figure 3). The sizes of the trial and output vectors that interface with the experimental elements are both 3 because the two-dimensional experimental beam-column element has 3 basic degrees of freedom.

```
# Define experimental setup
# -----
# expSetup OneActuator $tag <-control $ctrlTag> $dir -sizeTrialOut $sizeTrial $sizeOut
<-trialDispFact $f> ...
expSetup OneActuator 1 -control 1 2 -sizeTrialOut 3 3
expSetup OneActuator 2 -control 2 2 -sizeTrialOut 3 3
```

The `expSetup` command parameters for `OneActuator` are:

- `$tag` is the unique setup tag.
- `$ctrlTag` is the tag of a previously defined control object. In this case, it is `SimUniaxialMaterials`.
- `$dir` is the direction of the imposed displacement in the element basic reference coordinate system.
- `$sizeTrial` and `$sizeOut` are the sizes of the element trial and output data vectors, respectively.
- `$f` are trial displacement factor, output displacement factor, and output force factor, respectively. These optional fields are used to factor the imposed and the measured data. The default values are 1.0.

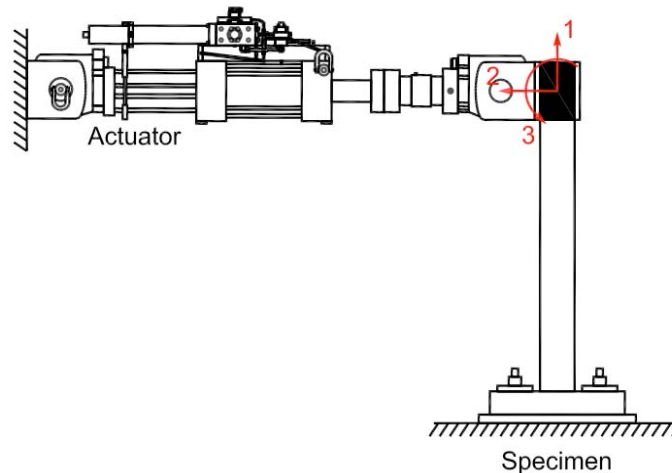


Figure 3: OneActuator Setup.



5.3 Experimental Site

The two experimental sites are set to `LocalSite` since a local hybrid simulation is being performed.

```
# Define experimental site
# -----
# expSite LocalSite $tag $setupTag
expSite LocalSite 1 1
expSite LocalSite 2 2
```

The `expSite` command parameters for `LocalSite` are:

- `$tag` is the unique site tag.
- `$setupTag` is the tag of a previously defined experimental setup object.

5.4 Experimental Element

Both experimental elements are set to `beamColumn` experimental elements (Figure 4). Two nodes define these elements. The experimental elements use the `PDelta` or `Corotational` geometric transformation.

```
# Define geometric transformation
# -----
#geomTransf PDelta 1
geomTransf Corotational 1

# Define experimental elements
# -----
# left and right columns
# expElement beamColumn $eleTag $iNode $jNode $transTag -site $siteTag -initStif $Kij ...
<-iMod> <-rho $rho>
expElement beamColumn 1 3 1 1 -site 1 -initStif 1310.8 0 0 0 11.2 -280.0 0 -280.0
9333.3333
expElement beamColumn 2 4 2 1 -site 2 -initStif 1310.8 0 0 0 11.2 -280.0 0 -280.0
9333.3333
```

The `expElement` command parameters for `beamColumn` are:

- `$eleTag` is the unique element tag.
- `$iNode` and `$jNode` are the end nodes that connect the beam-column element.
- `$transTag` is the previously defined coordinate transformation object. Here it is set to `PDelta` or `Corotational`.
- `$siteTag` is the tag of a previously defined site object. In this example, it is set to `LocalSite`.
- `$Kij` is the initial stiffness matrix entered row-wise. For this example, the initial stiffness matrices of `beamColumns` 1 and 2 are identical:

$$K_1 = \begin{bmatrix} 1310.8 & 0 & 0 \\ 0 & 11.2 & -280.0 \\ 0 & -280.0 & 9333.3333 \end{bmatrix} \quad \text{and} \quad K_2 = \begin{bmatrix} 1310.8 & 0 & 0 \\ 0 & 11.2 & -280.0 \\ 0 & -280.0 & 9333.3333 \end{bmatrix}$$

- `-iMod` allows for error correction using Nakashima's initial stiffness modification. It is optional. The default is false.
- `$rho` is the mass per unit length. Its default is 0.



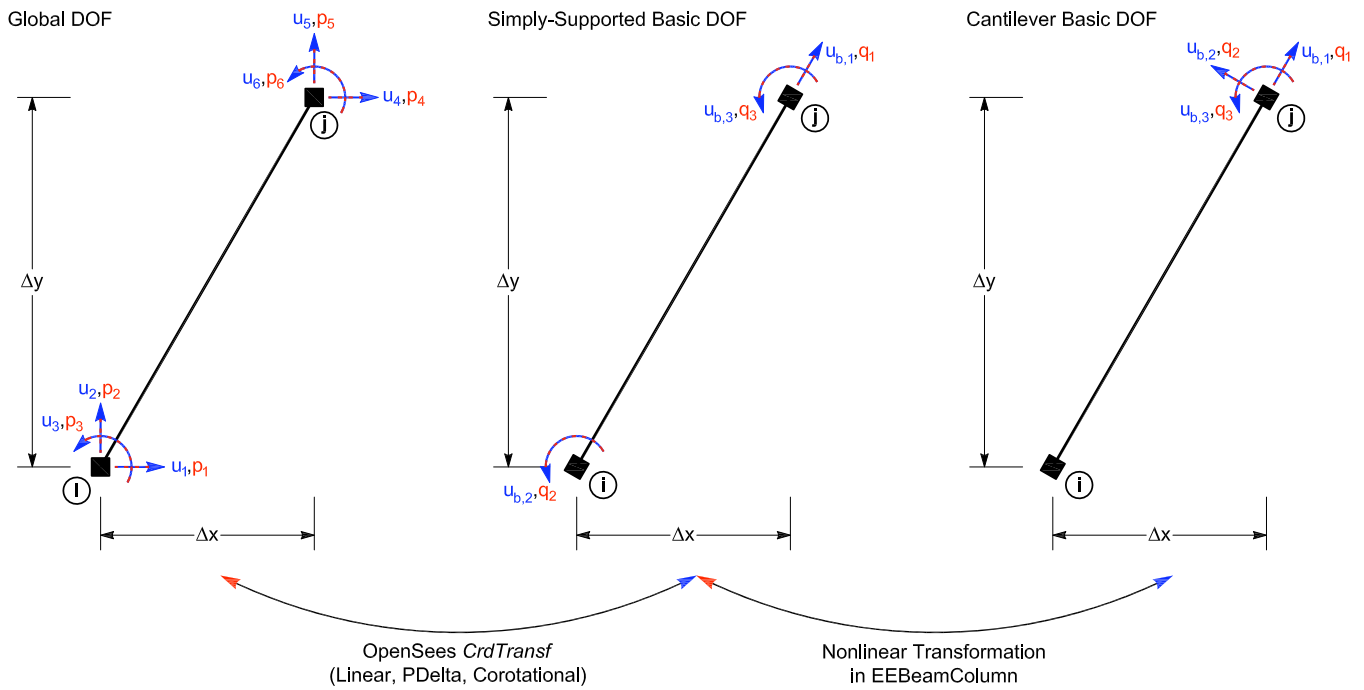


Figure 4: beamColumn Experimental Element.

6 Running the Local Hybrid Simulation

This section contains step-by-step instructions for running a local hybrid simulation for the Portal Frame example. Since OpenSees is used as the computational driver for a local simulation, there is no client-server communication (Figure 5).

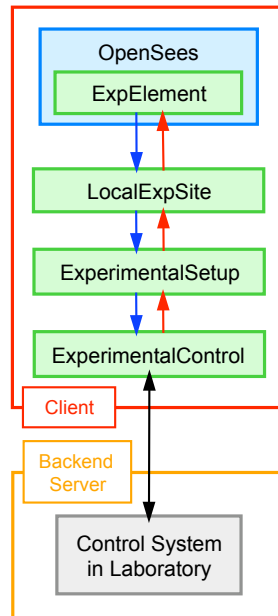
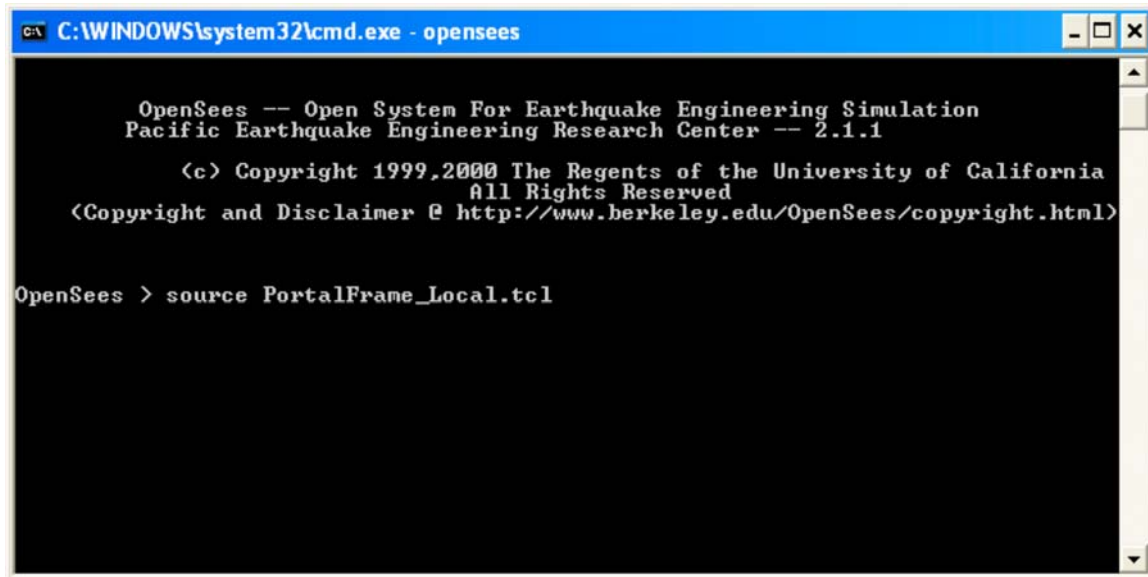


Figure 5: Local Hybrid Simulation using OpenSees.



To run the simulation perform the following steps:

- Start the OpenSees executable file (openSees.exe) from the directory where PortalFrame_Local.tcl resides.
- At the prompt, type **source PortalFrame_Local.tcl** and press **enter**. This runs the simulation (Figure 6).



```
C:\WINDOWS\system32\cmd.exe - opensees

OpenSees -- Open System For Earthquake Engineering Simulation
Pacific Earthquake Engineering Research Center -- 2.1.1

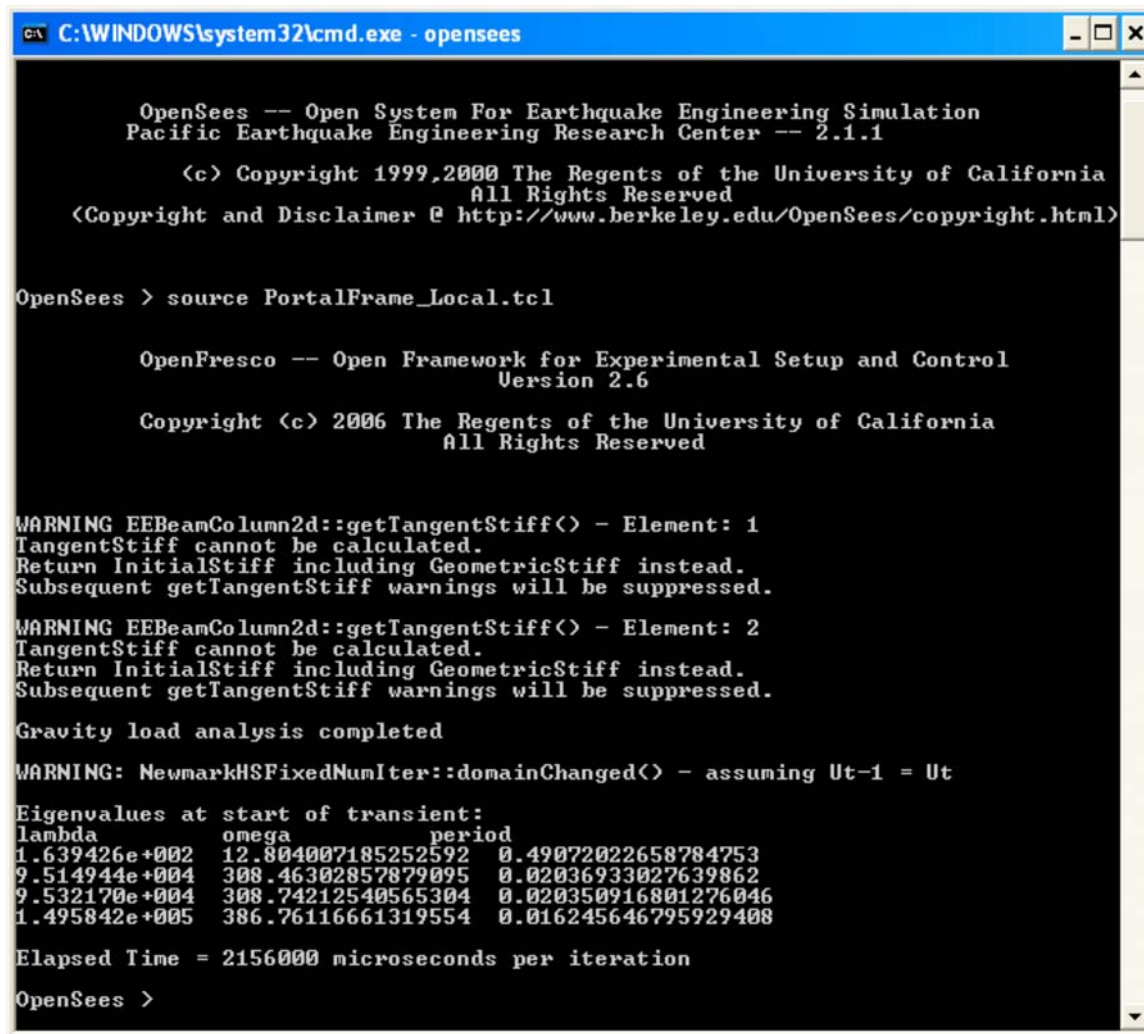
(c) Copyright 1999,2000 The Regents of the University of California
All Rights Reserved
(Copyright and Disclaimer @ http://www.berkeley.edu/OpenSees/copyright.html)

OpenSees > source PortalFrame_Local.tcl
```

Figure 6: OpenSees Command Window for Local Test.



- After the simulation has ended, the OpenSees command window should look like Figure 7.

The image shows a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe - opensees". The window contains the following text:

```
OpenSees -- Open System For Earthquake Engineering Simulation
Pacific Earthquake Engineering Research Center -- 2.1.1

(c) Copyright 1999,2000 The Regents of the University of California
All Rights Reserved
<Copyright and Disclaimer @ http://www.berkeley.edu/OpenSees/copyright.html>

OpenSees > source PortalFrame_Local.tcl

OpenFresco -- Open Framework for Experimental Setup and Control
Version 2.6

Copyright (c) 2006 The Regents of the University of California
All Rights Reserved

WARNING EEBeamColumn2d::getTangentStiff() - Element: 1
TangentStiff cannot be calculated.
Return InitialStiff including GeometricStiff instead.
Subsequent getTangentStiff warnings will be suppressed.

WARNING EEBeamColumn2d::getTangentStiff() - Element: 2
TangentStiff cannot be calculated.
Return InitialStiff including GeometricStiff instead.
Subsequent getTangentStiff warnings will be suppressed.

Gravity load analysis completed

WARNING: NewmarkHSFixedNumIter::domainChanged() - assuming Ut-1 = Ut

Eigenvalues at start of transient:
lambda      omega      period
1.639426e+002  12.804007185252592  0.490720222658784753
9.514944e+004  308.46302857879095  0.02036933027639862
9.532170e+004  308.74212540565304  0.020350916801276046
1.495842e+005  386.76116661319554  0.016245646795929408

Elapsed Time = 2156000 microseconds per iteration

OpenSees >
```

Figure 7: OpenSees Command Window for Local Test after Simulation.



7 Results

The OpenSees command window should display the following results:

Eigenvalues at start of transient:

lambda	omega	period
1.639426e+002	12.804007185252592	0.49072022658784753
9.514944e+004	308.46302857879095	0.02036933027639862
9.532170e+004	308.74212540565304	0.020350916801276046
1.495842e+005	386.76116661319554	0.016245646795929408

There are now output files from the simulation in the directory:

User's Directory\OpenFresco\trunk\EXAMPLES\PortalFrame

The following are the output files:

- Control_ctrlDsp.out
- Control_daqDsp.out
- Control_daqFrc.out
- Elmt_glbFrc.out
- Node_Acc.out
- Node_Dsp.out
- Node_Rxn.out
- Node_Vel.out

If the withGravity parameter is set to 1, two extra output files are created:

- Gravity_Dsp.out
- Gravity_Frc.out

These files are created using the recorder command. Below is the part of the script that executes this command. The OpenFresco Command Language Manual contains more information about the element recorder commands for all the experimental elements. For the node recorder commands refer to the OpenSees Command Language Manual on the [OpenSees website](http://opensees.berkeley.edu) (<http://opensees.berkeley.edu>).

```
# -----
# Start of recorder generation
# -----
# create the recorder objects
recorder Node -file Node_Dsp.out -time -node 3 4 -dof 1 2 3 disp
recorder Node -file Node_Vel.out -time -node 3 4 -dof 1 2 3 vel
recorder Node -file Node_Acc.out -time -node 3 4 -dof 1 2 3 accel
recorder Node -file Node_Rxn.out -time -node 1 2 3 4 -dof 1 2 3 reactionIncludingInertia

recorder Element -file Elmt_glbFrc.out -time -ele 1 2 3 forces
expRecorder Control -file Control_ctrlDsp.out -time -control 1 2 ctrlDisp
expRecorder Control -file Control_daqDsp.out -time -control 1 2 daqDisp
expRecorder Control -file Control_daqFrc.out -time -control 1 2 daqForce
# -----
# End of recorder generation
# -----
```



In Figure 8 to Figure 14, the response quantities for the Portal Frame example without gravity loads (the withGravity parameter set to 0) are compared against the ones with gravity loads (the withGravity parameter set to 1).

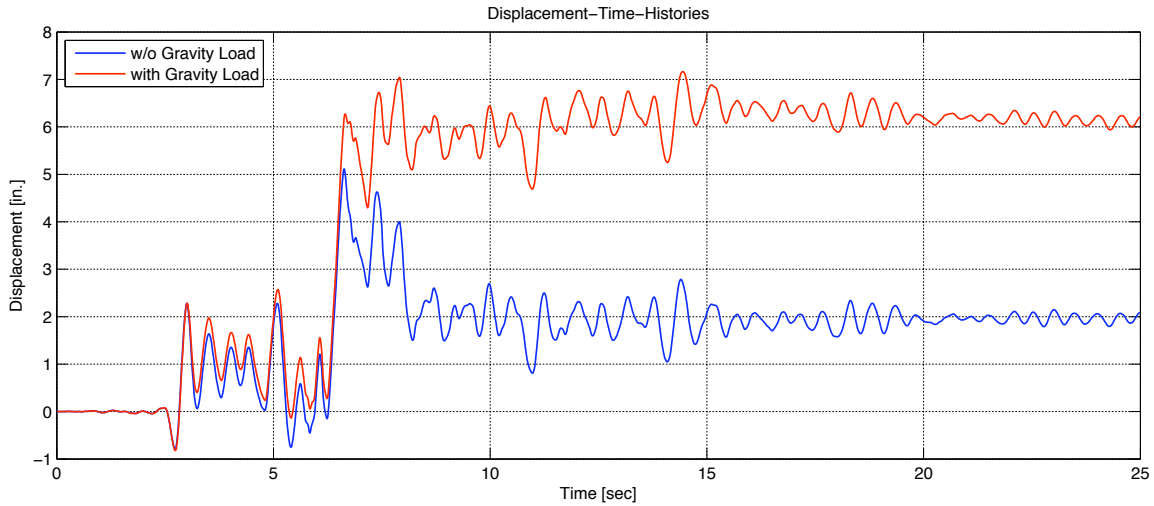


Figure 8: Displacements vs. Time for Portal Frame Example.

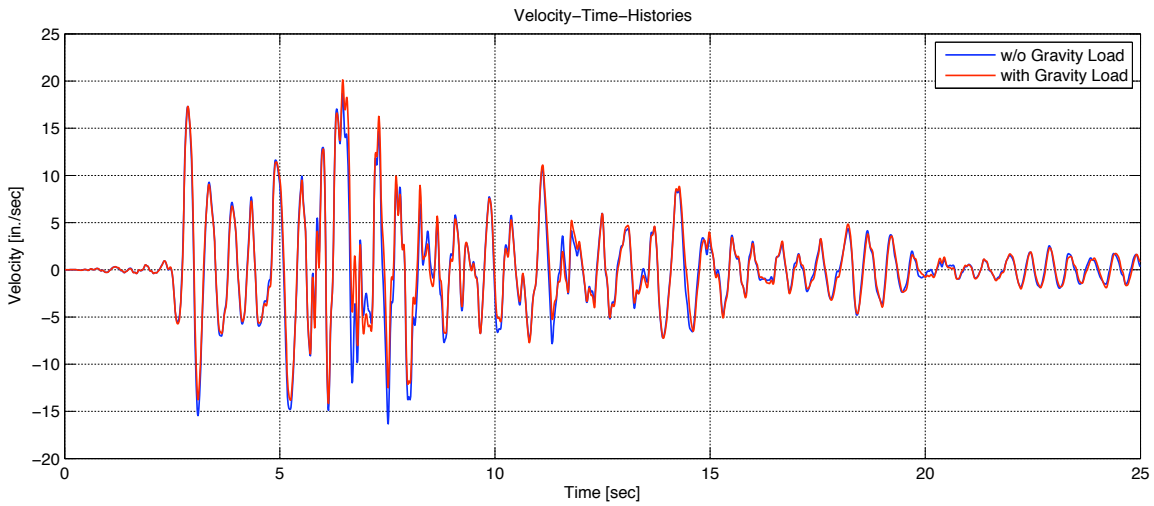


Figure 9: Velocities vs. Time for Portal Frame Example.



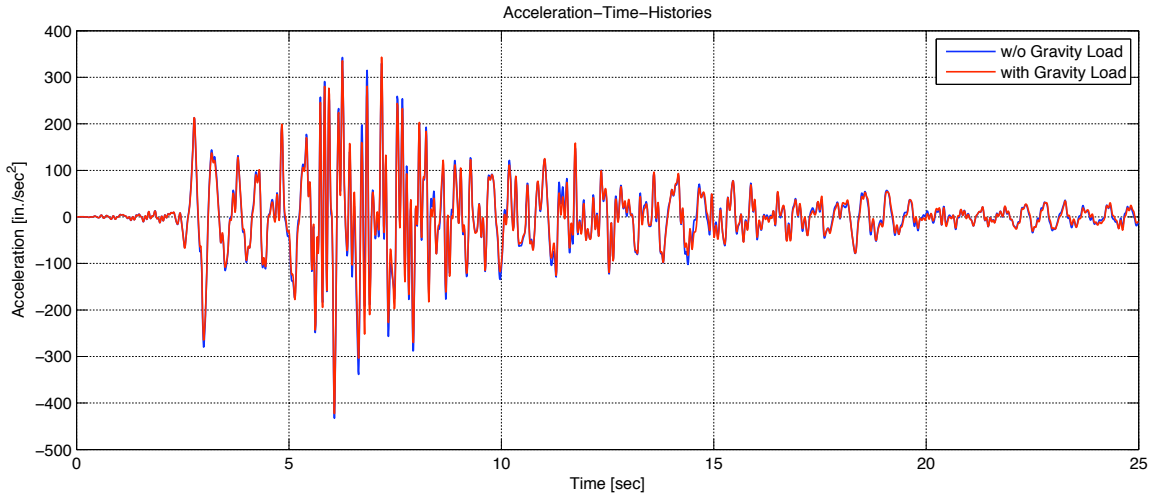


Figure 10: Accelerations vs. Time for Portal Frame Example.

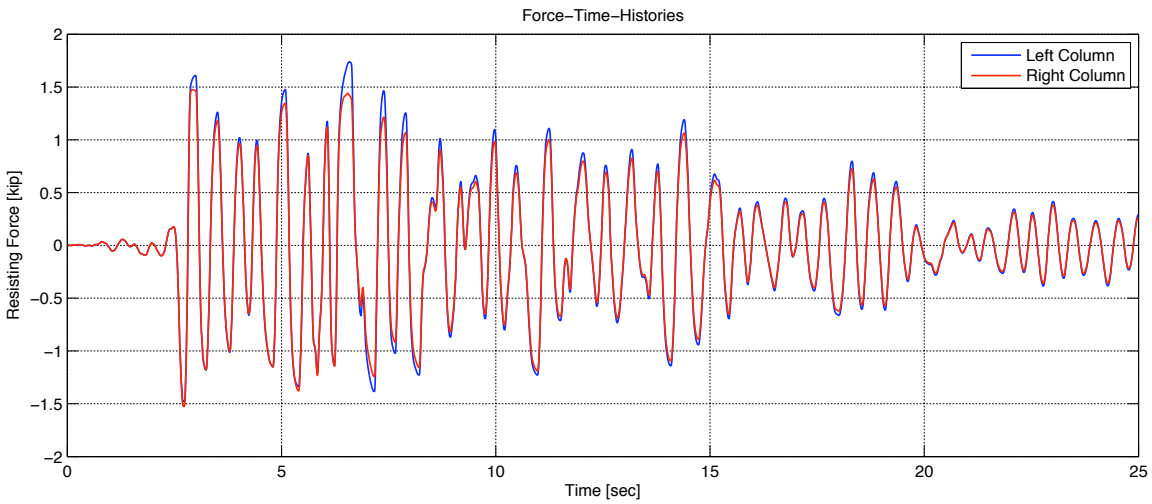


Figure 11: Forces vs. Time for Portal Frame Example without Gravity Loads.

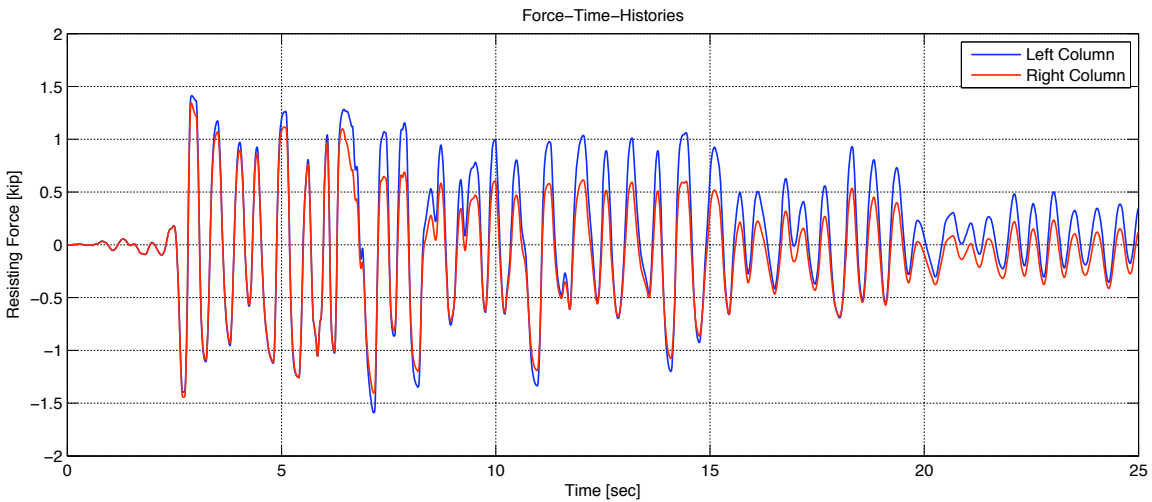


Figure 12: Forces vs. Time for Portal Frame Example with Gravity Loads.



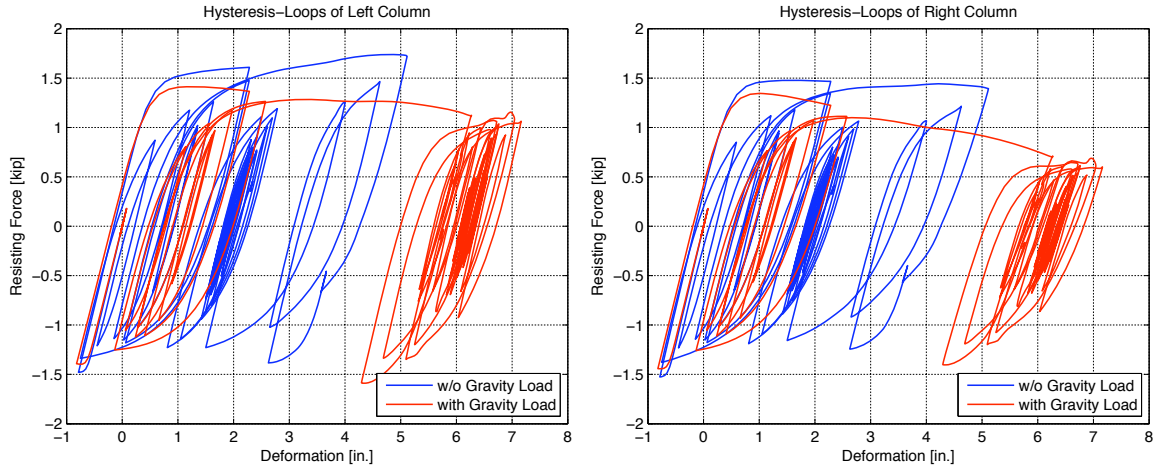


Figure 13: Column Hysteresis Loops for Portal Frame Example.

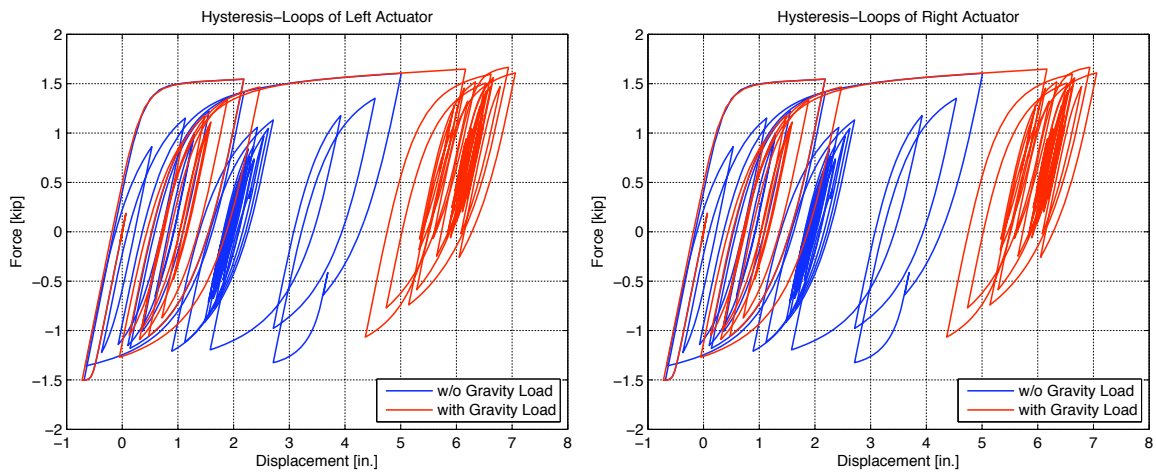


Figure 14: Actuator Hysteresis Loops for Portal Frame Example.

8 References

Schellenberg, A., “Advanced Implementation of Hybrid Simulation”, Dissertation, University of California, Berkeley, 2008, p. 282.

