



*Enabling the Network for  
Earthquake Engineering Simulation*



TR-2009-[ID]

# OpenFresco Framework for Hybrid Simulation: Experimental Signal Filter 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: Hybrid Simulation Example Using Experimental Signal Filters.....	3
2	Required Files.....	3
3	Structural Model.....	3
4	Ground Motion .....	4
5	OpenFresco Tcl Commands .....	4
5.1	Experimental Signal Filter .....	5
5.2	Experimental Control .....	5
5.3	Experimental Setup .....	6
5.4	Experimental Site .....	7
5.5	Experimental Element.....	7
6	Running Local Hybrid Simulation using OpenSees.....	9
7	Results .....	10
8	References .....	16

## Table of Figures

Figure 1:	OpenSees One-Bay Frame Model.....	4
Figure 2:	1940 El Centro Ground Motion. ....	4
Figure 3:	OneActuator Experimental Setup. ....	6
Figure 4:	Local Hybrid Simulation using OpenSees. ....	7
Figure 5:	twoNodeLink Experimental Element.....	8
Figure 6:	OpenSees Command Window for Experimental Signal Filter Example. ....	9
Figure 7:	OpenSees Command Window after Simulation. ....	10
Figure 8:	Displacements vs. Time for Simulations Without and With Errors.....	12
Figure 9:	Velocities vs. Time for Simulations Without and With Errors. ....	13
Figure 10:	Accelerations vs. Time for Simulations Without and With Errors. ....	14
Figure 11:	Resisting Forces vs. Time for Simulations Without and With Errors.....	15
Figure 12:	Element Hysteresis Loops for Simulations Without and With Errors. ....	16



# 1 Introduction: Hybrid Simulation Example Using Experimental Signal Filters

This example demonstrates how to run a hybrid simulation using OpenFresco's experimental signal filters. The Experimental Signal Filter example is a fully simulated test, meaning that the experimental control is set to simulation mode. It does not require a physical specimen to run. OpenSees is used as the computational driver for the hybrid simulation. The Newmark explicit time integration scheme is used for this example. The response plots from the simulation are provided for comparison.

## 2 Required Files

For the Signal Filter example, the following files are required. These are located in:

User's Directory\OpenFresco\trunk\EXAMPLES\OneBayFrame\TestSignalFilter

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:

- OneBayFrame\_Local.tcl      and      elcentro.txt

The latest OpenSees executable and Tcl/Tk 8.5.x are required to run this example. If not done so already, they can 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.

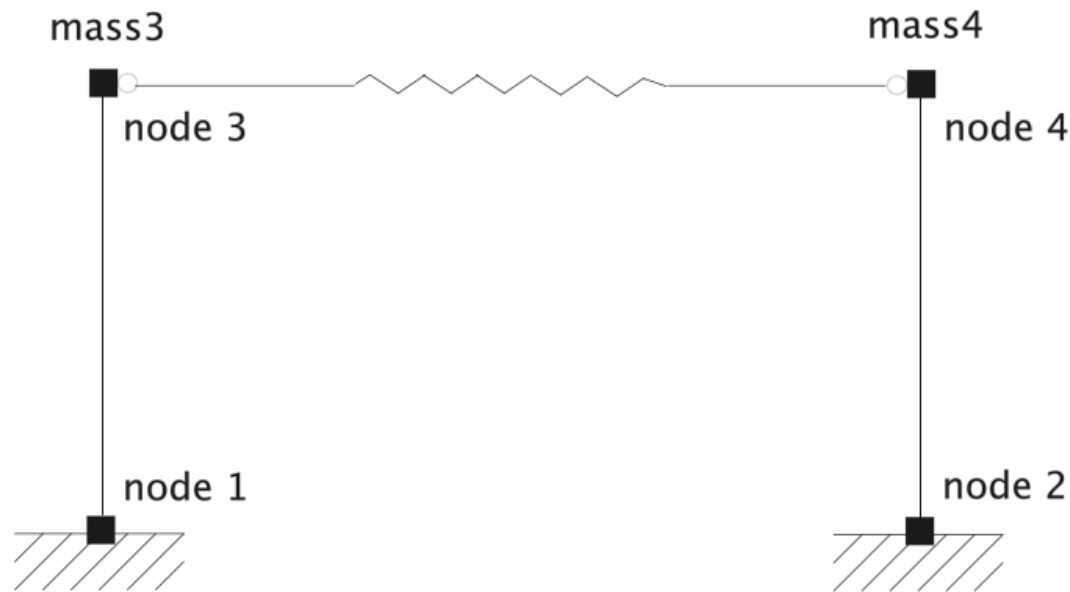
## 3 Structural Model

The model consists of two columns and a spring. The columns are element 1 connected by nodes 1 and 3 and element 2 connected by nodes 2 and 4. The spring is element 3 connected by nodes 3 and 4. A lumped mass is placed at the top of each column. The bases of the two columns are fixed, and the columns are considered axially rigid. Imperial units are used [in., kips, and sec]. Both columns are simulated experimentally. Element 1 is simulated locally during the local test and is simulated on the simulation application server side during the distributed test. However, element 2 is simulated locally during both the local and distributed tests. The following Tcl script from OneBayFrame\_Local.tcl defines the geometry of the model:

```
# Define geometry for model
# -----
set mass3 0.04
set mass4 0.02
# node $tag $xCrd $yCrd $mass
node 1      0.0   0.00
node 2    100.0   0.00
node 3      0.0  54.00  -mass $mass3 $mass3
node 4    100.0  54.00  -mass $mass4 $mass4

# set the boundary conditions
# fix $tag $DX $DY
fix 1 1 1
fix 2 1 1
fix 3 0 1
fix 4 0 1
```

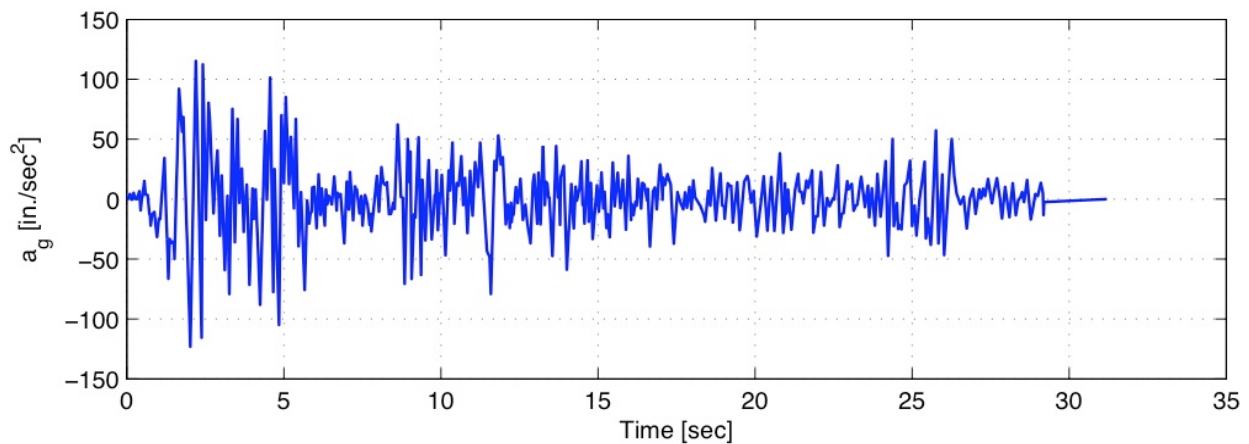




**Figure 1: OpenSees One-Bay Frame Model.**

## 4 Ground Motion

The structure is subjected horizontally to the north-south component of the ground motion recorded at a site in El Centro, California during the Imperial Valley earthquake of May 18, 1940 (Chopra 2006). The file, `elcentro.txt`, contains the acceleration data recorded at every 0.02 seconds (Figure 2).



**Figure 2: 1940 El Centro Ground Motion.**

## 5 OpenFresco Tcl Commands

This section contains explanations of the OpenFresco Tcl commands from `OneBayFrame_Local.tcl`. All excerpts in this document are from this file. More detailed information about the OpenFresco Tcl commands are found in the OpenFresco Command Language Manual.

```
# 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 Signal Filter

The experimental signal filter being used in this example is `ErrorSimUndershoot`. `ErrorSimUndershoot` simulates an undershooting error.

```
# Define experimental signal filter
# -----
# expSignalFilter ErrorSimUndershoot $tag $error
expSignalFilter ErrorSimUndershoot 1 0.01
```

The `expSignalFilter` command parameters for `ErrorSimUndershoot` are:

- `$tag` is the unique signal filter tag.
- `$error` is the undershoot error.

## 5.2 Experimental Control

The experimental controls are set to be `SimUniaxialMaterials` for this example. `SimUniaxialMaterials` uses the `Elastic` and `Steel02` materials, which have a `matTag` of 1 and 2, respectively. The `Tcl` command is invoked by `expControl` as seen in the excerpts below.

```
# Define materials
# -----
# uniaxialMaterial Steel02 $matTag $Fy $E $b $R0 $cR1 $cR2 $a1 $a2 $a3 $a4
#uniaxialMaterial Elastic 1 2.8
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 2 5.6
```

```
# Define experimental control
# -----
# expControl SimUniaxialMaterials $tag $matTags <-ctrlFilters (5 $cfTag)> <-daqFilters (5 $dfTag)>
expControl SimUniaxialMaterials 1 1 -ctrlFilters 1 0 0 0 0
expControl SimUniaxialMaterials 2 2 -ctrlFilters 1 0 0 0 0
```

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

- `$tag` is the unique control tag.
- `$matTags` are the tags of previously defined uniaxial material objects.
- `$cfTags` are the tags of previously defined experimental signal filters. In this example, `ErrorSimUndershoot` described in the previous section is applied only to the displacement being sent to the control system while velocity, acceleration, force, and time have no filters applied to them for each experimental control.



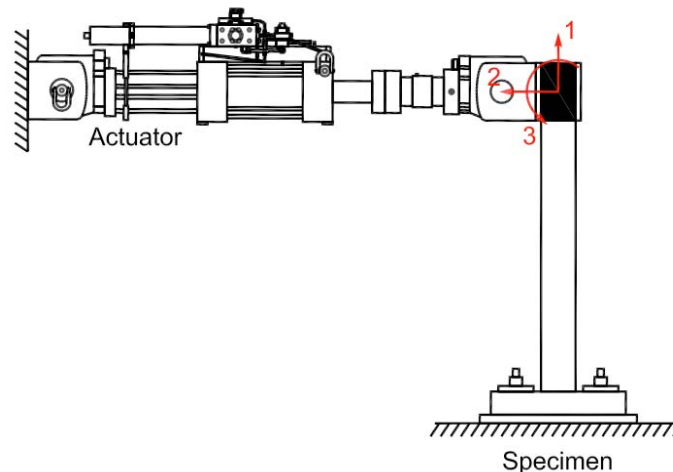
### 5.3 Experimental Setup

The OneActuator setup is used for the experimental setups (Figure 3). The Tcl command for this experimental setup is:

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

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 experimental control.
- \$dir is the direction of the imposed displacement in the element basic reference coordinate system, which is set to 1.
- \$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.

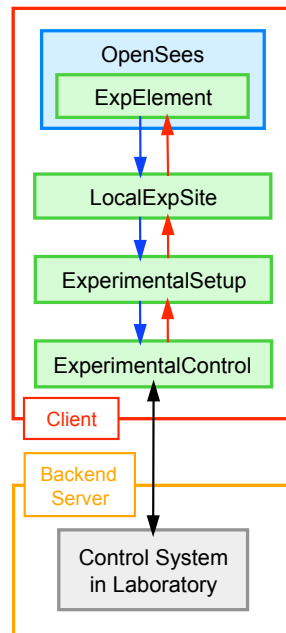


**Figure 3: OneActuator Experimental Setup.**



## 5.4 Experimental Site

Figure 4 shows how OpenFresco can use OpenSees for a local hybrid simulation.



**Figure 4: Local Hybrid Simulation using OpenSees.**

The script excerpt below shows that the `expSite`, experimental site, is set to `LocalSite`.

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

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

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

## 5.5 Experimental Element

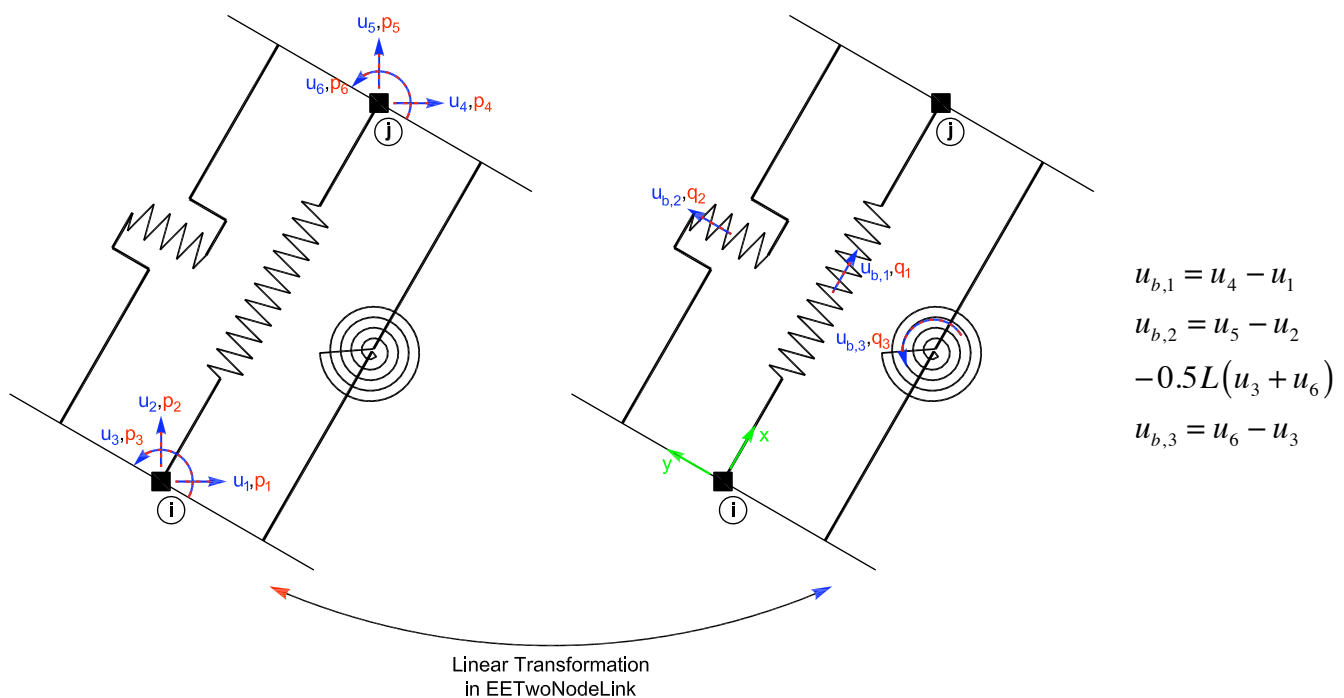
The experimental elements are set to be `twoNodeLink` elements (Figure 5). Note that an experimental element is defined by the Tcl command `expElement`.

```
# Define experimental elements
# -----
# left and right columns
# expElement twoNodeLink $eleTag $iNode $jNode -dir $dirs -site $siteTag -initStif $Kij
<-orient <$x1 $x2 $x3> $y1 $y2 $y3> <-iMod> <-mass $m>
expElement twoNodeLink 1 1 3 -dir 2 -site 1 -initStif 2.8 -orient -1 0 0
expElement twoNodeLink 2 2 4 -dir 2 -site 2 -initStif 5.6 -orient -1 0 0
```



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

- `$eleTag` is the unique element tag.
- `$iNode` and `$jNode` are the end nodes that the element connects to.
- `$siteTag` is the tag of a previously defined site object. In this example, it is the `LocalSite` for the local test.
- `$dirs` are the force-deformation directions in the element local reference coordinate system.
- `$Kij` are the (row wise) initial stiffness matrix components of the element.
- `$x1 $x2 $x3 $y1 $y2 $y3` set the orientation vectors for the element. `x1`, `x2`, and `x3` are vector components in the global coordinates defining the local x-axis. `y1`, `y2`, and `y3` are the same except that they define the y vector which lies in the local x-y plane for the element. `<-orient <$x1 $x2 $x3> $y1 $y2 $y3>` field is optional with default being the global X and Y.
- `$Mratio` are the optional P-Delta moment contribution ratios. The size of the ratio vector is 4 (entries: `[My-$iNode, My-$jNode, Mz-$iNode, Mz-$jNode]`) `My-$iNode + My-$jNode <= 1.0`, `Mz-$iNode + Mz-$jNode <= 1.0`. The remaining P-Delta moments are resisted by shear couples. (default = `[0.0 0.0 0.0 0.0]`)
- `-iMod` and `$m` are also optional fields. `-iMod` allows for error correction using Nakashima's initial stiffness modification. Default for `-iMod` is false. `$m` is used to assign mass to the element. Its default is zero.



**Figure 5: twoNodeLink Experimental Element.**



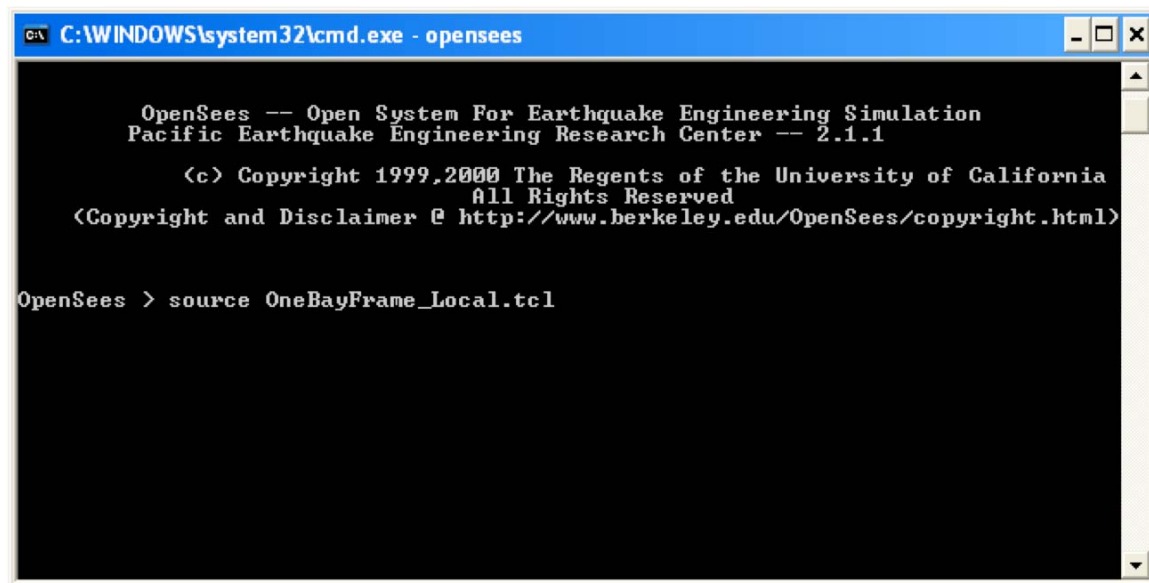


## 6 Running Local Hybrid Simulation using OpenSees

This section shows step-for-step how to run a local hybrid simulation using OpenSees.

To run this simulation perform the following steps:

- Start the OpenSees executable file (openSees.exe) from the directory where OneBayFrame\_Local.tcl is located.
- At the prompt, type **source OneBayFrame\_Local.tcl** and hit **enter** (Figure 6).



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

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

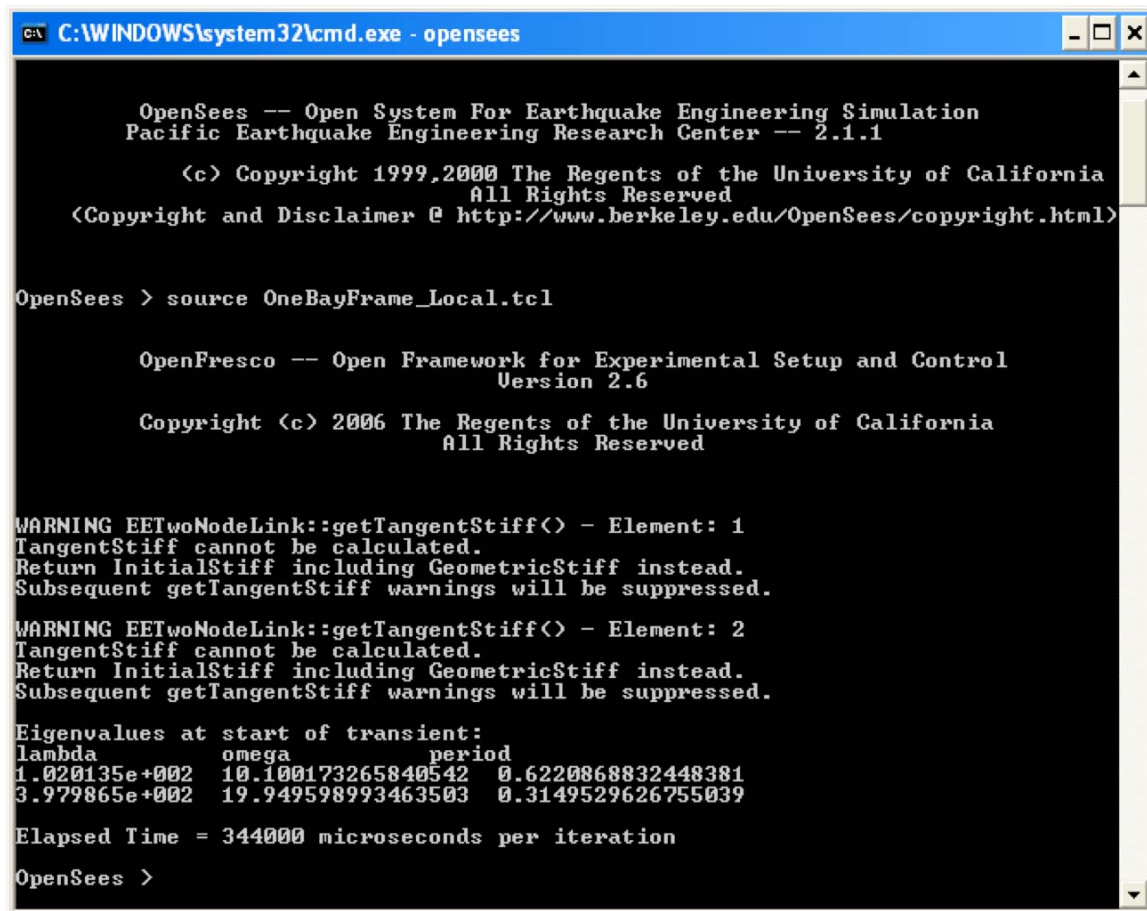
(c) Copyright 1999,2000 The Regents of the University of California
All Rights Reserved
(Copyright and Disclaimer @ http://www.berkeley.edu/OpenSees/copyright.html)

OpenSees > source OneBayFrame_Local.tcl
```

Figure 6: OpenSees Command Window for Experimental Signal Filter Example.

- After the simulation has completed, the OpenSees command windows should look like Figure 7.





```

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

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

Eigenvalues at start of transient:
lambda      omega      period
1.020135e+002  10.100173265840542  0.6220868832448381
3.979865e+002  19.949598993463503  0.3149529626755039

Elapsed Time = 344000 microseconds per iteration

OpenSees >

```

Figure 7: OpenSees Command Window after Simulation.

## 7 Results

The OpenSees command window should display the following results:

```

Eigenvalues at start of transient:
lambda      omega      period
1.020135e+002  10.1001732658  0.622086883245
3.979865e+002  19.9495989935  0.314952962676

```

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

User's Directory\OpenFresco\trunk\EXAMPLES\OneBayFrame\TestSignalFilter

The following are the output files:

- Elmt\_Frc.out
- Elmt\_ctrlDsp.out
- Elmt\_daqDsp.out
- Node\_Acc.out
- Node\_Dsp.out
- Node\_Vel.out



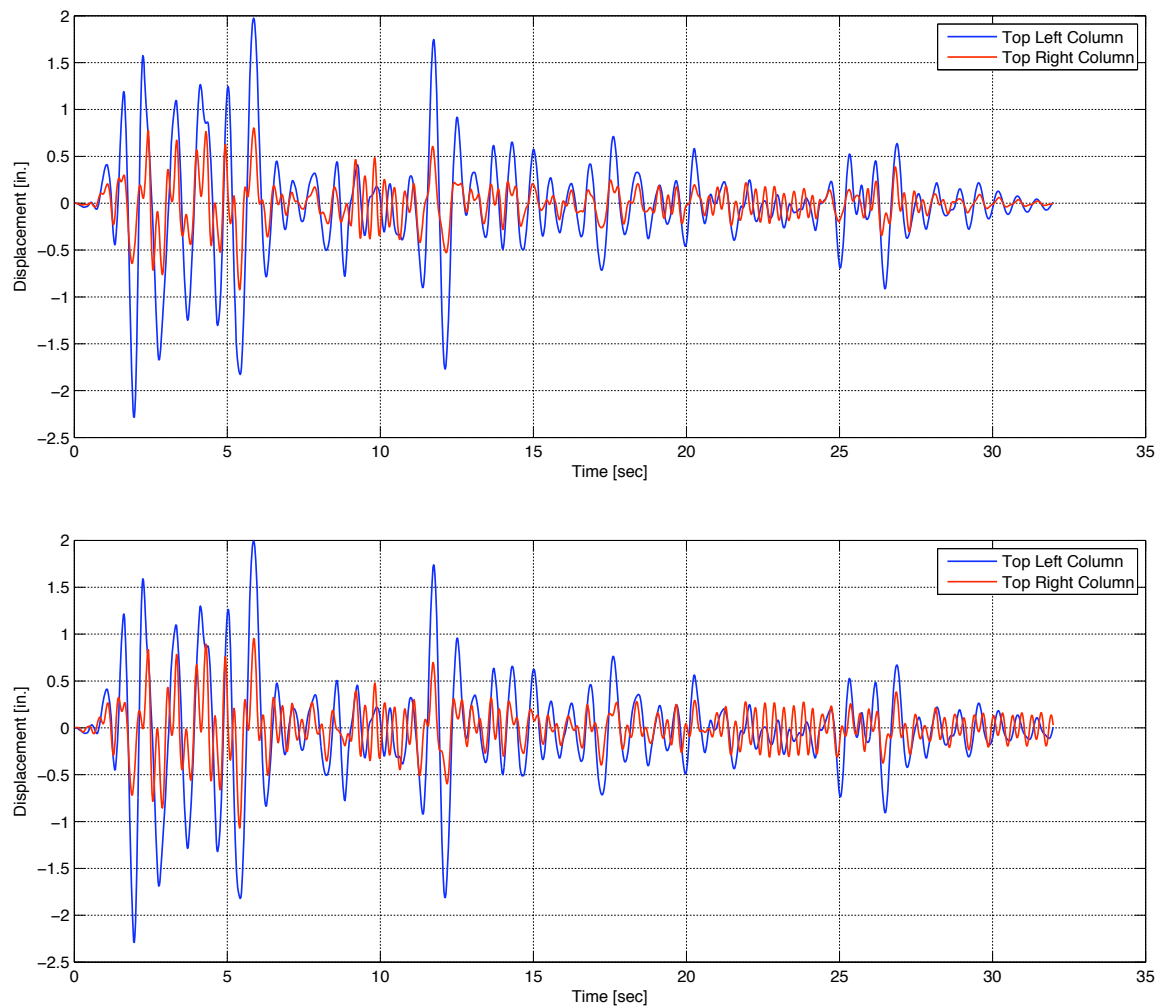
These files are created using the recorder command. Below are the parts of the scripts that execute this command. The OpenFresco Command Language Manual contains more information about the element recorder command. For the node recorder commands refer to the OpenSees Command Language Manual on the [OpenSees website](http://opensees.berkeley.edu) (opensees.berkeley.edu). The following excerpt is from OneBayFrame\_Local.tcl.

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

recorder Element -file Elmt_Frc.out -time -ele 1 2 3 forces
recorder Element -file Elmt_ctrlDisp.out -time -ele 1 2 ctrlDisp
recorder Element -file Elmt_daqDisp.out -time -ele 1 2 daqDisp
# -----
# End of recorder generation
# -----
```

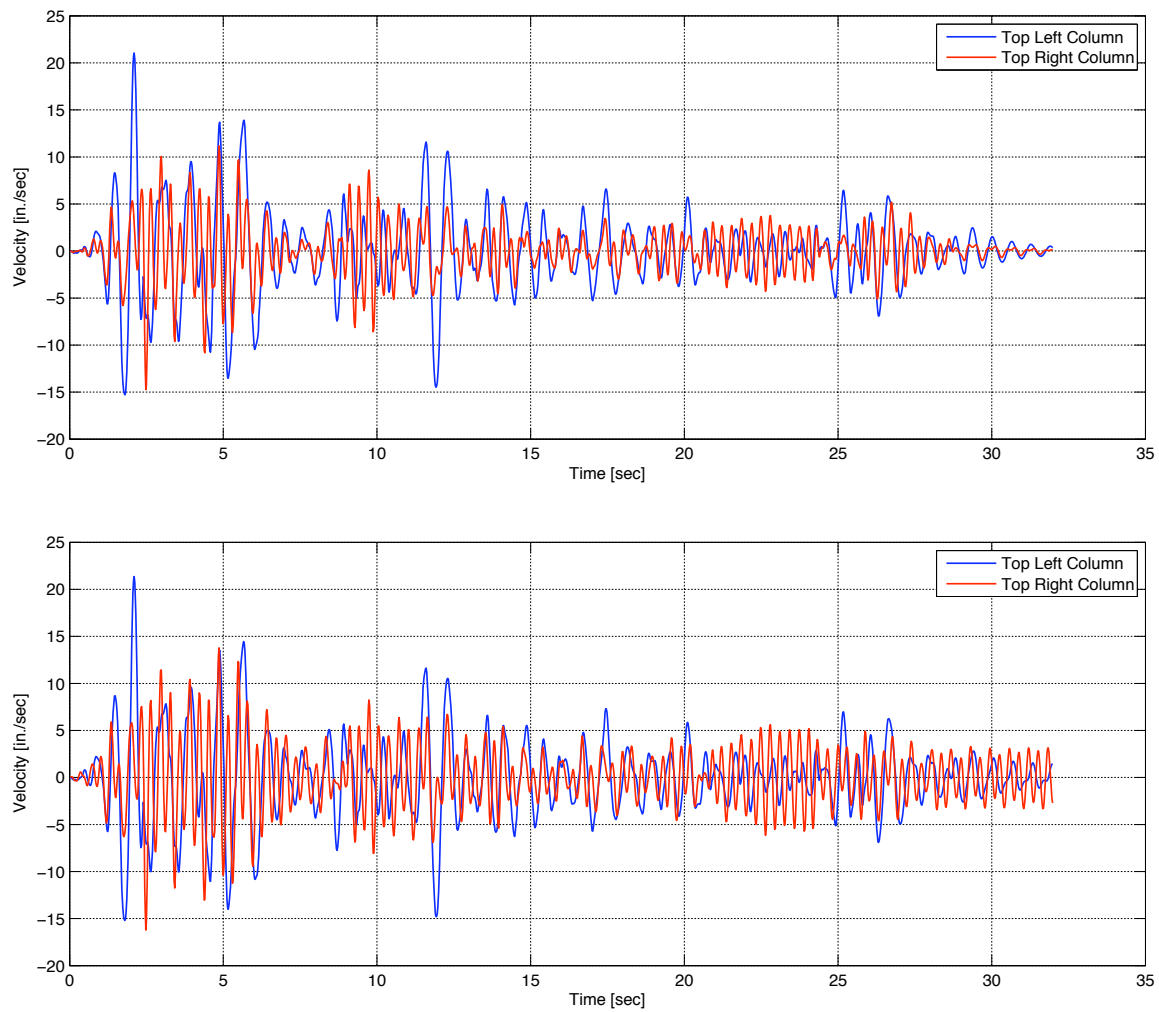
The response quantities for the Signal Filter example are plotted in Figures 8 to 12. All quantities are compared against the ones from an analysis without undershoot errors (e.g. undershoot error = 0.0). The comparisons clearly demonstrate how undershoot errors in an experimental test introduce energy into the system. This can lead to very inaccurate results and in the worst case such errors can even destabilize the hybrid simulation.





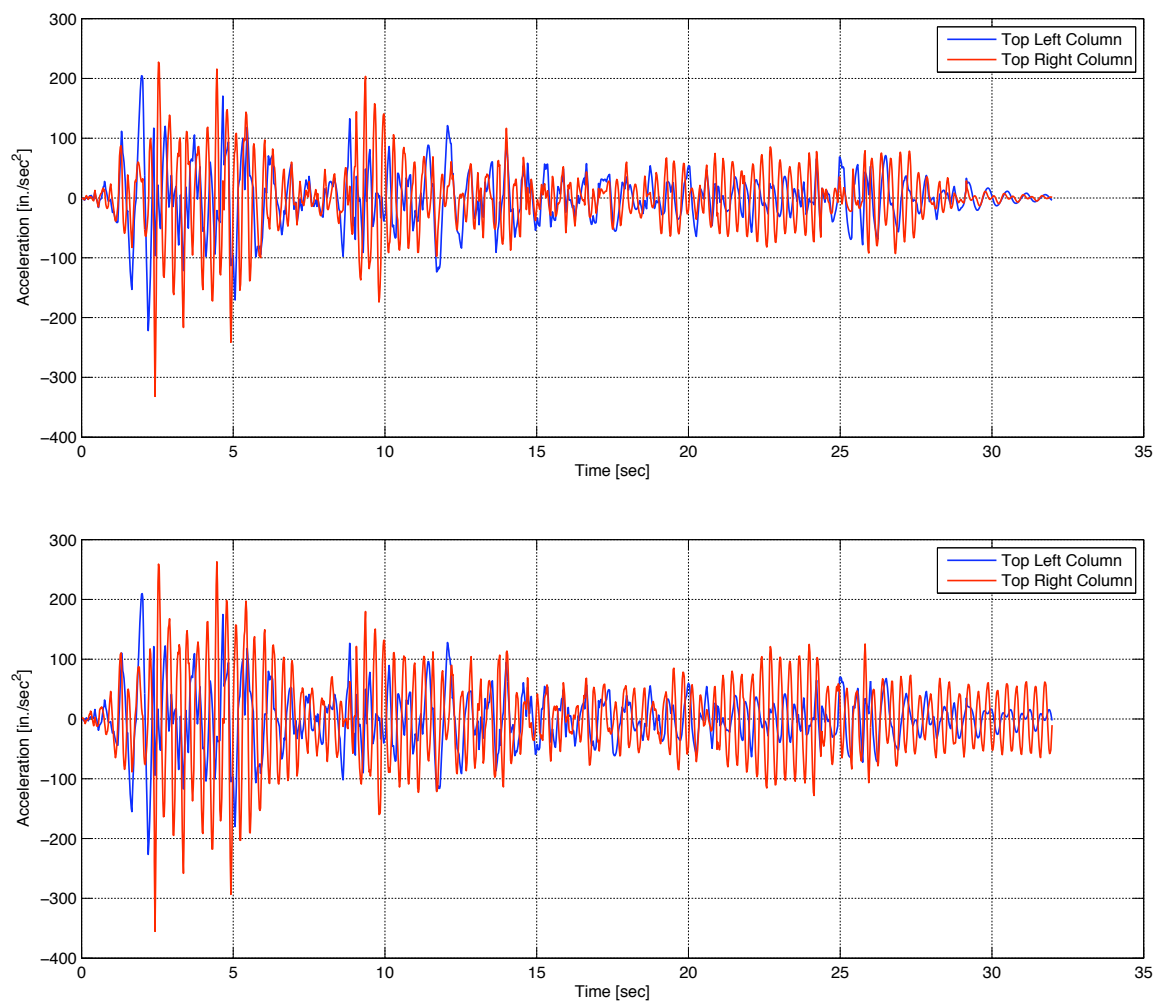
**Figure 8: Displacements vs. Time for Simulations Without and With Errors.**





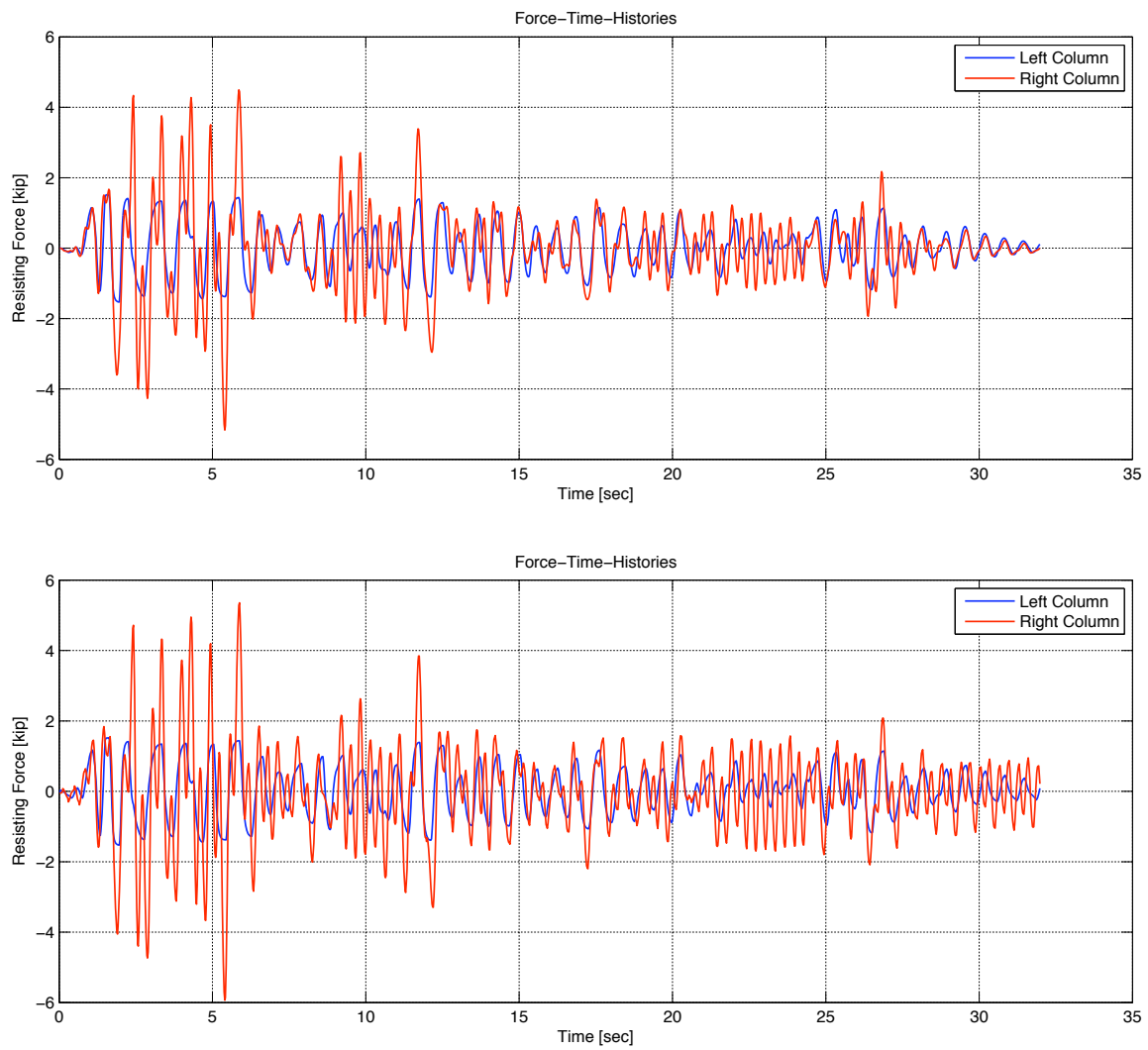
**Figure 9: Velocities vs. Time for Simulations Without and With Errors.**





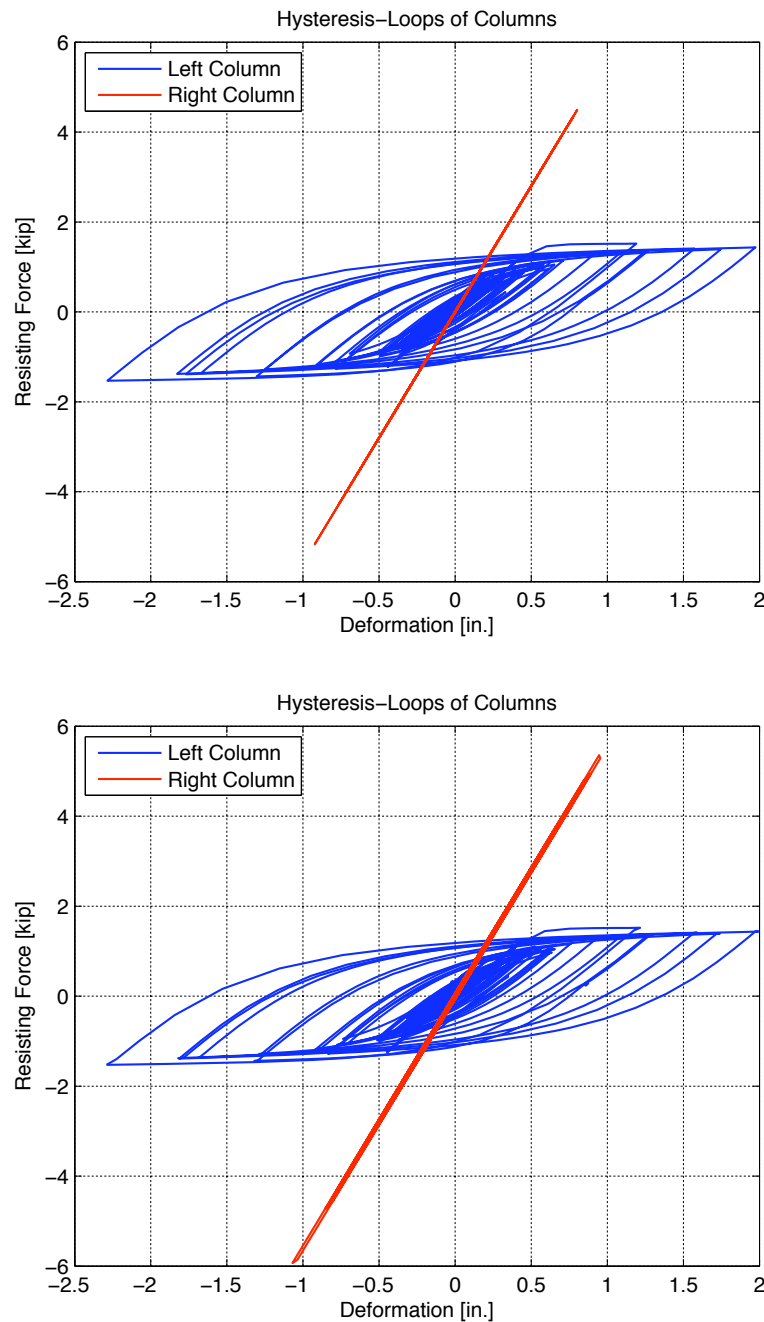
**Figure 10: Accelerations vs. Time for Simulations Without and With Errors.**





**Figure 11: Resisting Forces vs. Time for Simulations Without and With Errors.**





**Figure 12: Element Hysteresis Loops for Simulations Without and With Errors.**

## 8 References

Chopra, A.K., "Dynamics of Structures, Theory and Applications to Earthquake Engineering", 3rd edition, Prentice Hall, 2006, 912 pp.

