NEESit

Enabling the Network for Earthquake Engineering Simulation

TR-2009-[ID]

OpenFresco Framework for Hybrid Simulation: SimDomain Experimental Control 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: 2009-08-03 Version: 2.6

Acknowledgment: This work was supported by the George E. Brown, Jr. Network for Earthquake Engineering Simulation (NEES) Program of the National Science Foundation under Award Number CMS-0402490. Visit http://it.nees.org/ for more information.

Table of Contents

Table of Figures

1 Introduction: Distributed Hybrid Simulation Example Using SimDomain Experimental Control with OpenSees

This example explains how to use the Simulation Domain experimental control in OpenFresco to run a hybrid simulation. The Simulation Domain experimental control makes the entire OpenSees domain, including the material, section and element libraries, available in OpenFresco. This makes it possible to simulate a physical experimental specimen using OpenSees (http://opensees.berkeley.edu). This feature provides a wide variety of applications. For example, the Simulation Domain can be used to test the computational model, experimental setup, and network communication to ensure that all non-experimental aspects of a hybrid simulation are functioning properly before conducting an actual experiment.

The Simulation Domain example uses a simple L-Shaped Column model with the Alpha-OS time integration scheme. OpenSees is used as the computational driver for the simulation. This example is a fully simulated test, meaning that no physical specimen is required. The simulations of two different experimental setup configurations are demonstrated. The first one uses the `NoTransformation` experimental setup, and the second one uses the `ThreeActuators` experimental setup. The response results from the simulations are provided for comparison.

2 Required Files

For the Simulation Domain example, the following files are required. These are located in:

```
User's Directory\OpenFresco\trunk\EXAMPLES\LShapedColumn
```

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:

- `LShapedColumn_Client1.tcl`
- `LShapedColumn_Server1a.tcl`
- `LShapedColumn_Server1b.tcl`
- `elcentro.txt`

The 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). Follow the directions carefully on this website.

3 Structural Model

The L-Shaped Column is modeled in OpenSees using the Tcl file, `LShapedColumn_Client1.tcl`. It consists of two beam-column elements (Figure 1). The vertical one is connected to nodes 1 and 2 and the horizontal one to nodes 2 and 3. Lumped masses and mass moments of inertia are defined for nodes 2 and 3. The base support of the vertical beam-column is fixed. The following Tcl script from `LShapedColumn_Client1.tcl` defines the geometry of the model:

```
# Define geometry for model
# -------------------------
set mass2 0.04
set mass3 0.02
# node $tag $xCrd $yCrd $mass
node  1     0.0    0.0
node  2     0.0   54.0  -mass $mass2 $mass2 0.4
node  3    36.0   54.0  -mass $mass3 $mass3 0.1

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

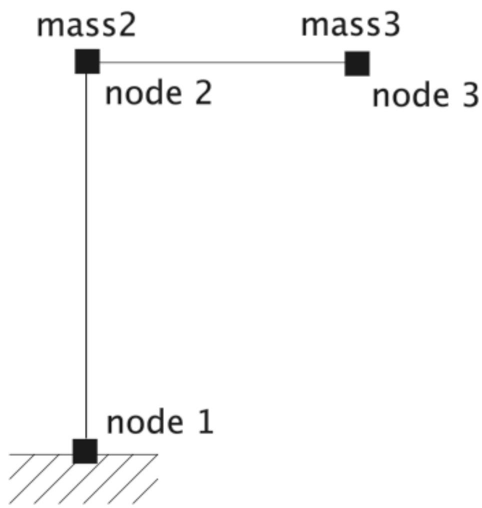

Figure 1: OpenSees L-Shaped Column Model.

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).

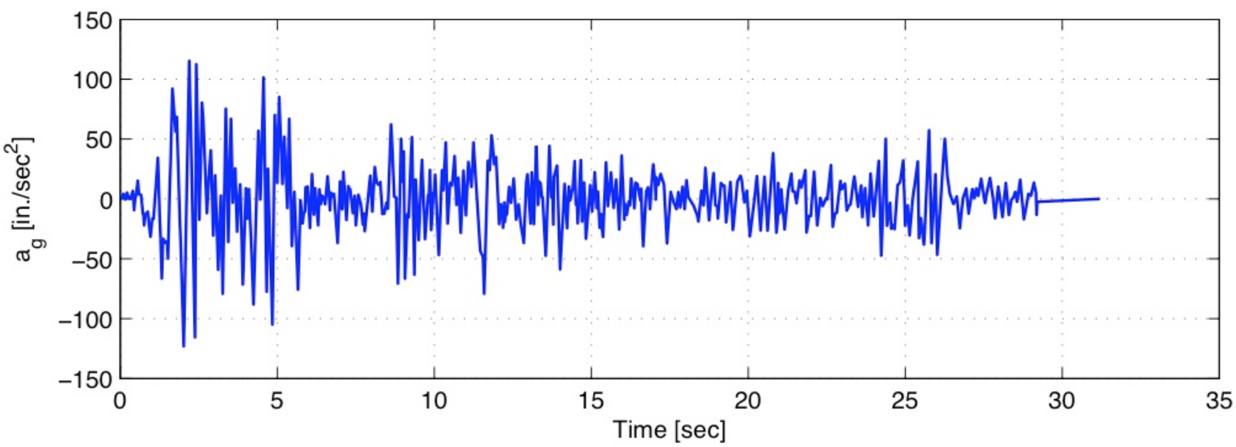

Figure 2: 1940 El Centro Ground Motion.

5 OpenFresco Tcl Commands

This section contains explanations of the OpenFresco Tcl commands used in this example. `LShapedColumn_Server1a.tcl` uses the `NoTransformation` experimental setup. `LShapedColumn_Server1b.tcl` uses the `ThreeActuators` experimental setup. The OpenSees analysis uses the script file `LShapedColumn_Client1.tcl`. 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 above script is an excerpt from `LShapedColumn_Client1.tcl`. The `loadPackage OpenFresco` command is necessary for the examples to execute properly.

5.1 Experimental Control

The experimental control is set to `SimDomain`. The `SimDomain` command uses the previously defined experimental control points. The experimental control points are defined using the `expControlPoint` command. In this example, the `SimDomain` command is used to simulate a nonlinear beam-column element in OpenSees.

The following excerpts are from `LShapedColumn_Server1a.tcl`. The first defines the nonlinear force-beam-column element in Opensees. The OpenSees website (http://opensees.berkeley.edu) contains more information about defining elements in OpenSees.

```
# Define element
# --------------
# element nonlinearBeamColumn $eleTag $iNode $jNode $numIntgrPts $secTag $transfTag
element nonlinearBeamColumn 1 1 2 5 1 1
```

The following script uses the experimental control points with tag 1 and 2 as the trial control point and the output control point, respectively, to define the `SimDomain` experimental control:

```
# Define experimental control
# ---------------------------
# expControl SimDomain $tag -trialCP cpTags -outCP cpTags
expControl SimDomain  1  -trialCP 1  -outCP 2
```

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

- `$tag` is the unique control tag.
- `$cptags` are the tags of previously defined experimental control points.

The following script shows how to define experimental control points. Both control points are defined on node 2, which is the top of the vertical beam-column element.

```
# Define control points
# ---------------------
# expControlPoint tag nodeTag dir resp <-fact f> <-lim l u> ...
expControlPoint 1 2  ux disp uy disp rz disp
```

```
expControlPoint 2 2  ux disp ux force uy disp uy force rz disp rz force
```

The `expControlPoint` command parameters are:

- `$tag` is the unique control point tag.
- `$nodeTag` is the unique node tag.
- `dir` is the direction of the response quantity. The input parameters are:
  - `ux` = along the X-axis
  - `uy` = along the Y-axis
  - `uz` = along the Z-axis
  - `rx` = around the X-axis
  - `ry` = around the Y-axis
  - `rz` = around the Z-axis
- `resp` is the response quantity. The input parameters are:
  - `disp` = displacement
  - `vel` = velocity
  - `accel` = acceleration
  - `force` = force
  - `time` = time
- `$f` is the factor applied to the response quantity. It is optional.
- `$l` is the lower limit. It is optional. Whenever the lower or the upper limit is exceeded, the program will prompt the user to stop the experiment or to use the limit as the trial value.
- `$u` is the upper limit. It is optional.

When using the `ThreeActuators` setup, the experimental control and the experiment control points are defined differently. The following script excerpts are from `LShapedColumn_Server1b.tcl`:

```
# Define element
# --------------
# first story column and rigid loading beam
# element nonlinearBeamColumn $eleTag $iNode $jNode $numIntgrPts $secTag $transfTag
element nonlinearBeamColumn 1 1 2 5 1 1
element elasticBeamColumn 2 3 2 2260 29000 5068 1
element elasticBeamColumn 3 2 4 2260 29000 5068 1
```

This defines the nonlinear beam column element and the rigid loading beams in OpenSees.

The parameters for `expControl SimDomain` and `expControlPoint` commands are the same as before.

```
# Define experimental control
# --------------------------
# expControl SimDomain $tag -trialCP cpTags -outCP cpTags
expControl SimDomain  1  -trialCP 1 2  -outCP 3 4
```

The next script uses two more experimental control points than before. Control points 1 and 2 are for the trial control points and Control points 3 and 4 for the output control points. Experimental control points 1 and 3 are defined on node 3. Experimental control points 2 and 4 are defined on node 4. Nodes 3 and 4 are the end points of the rigid loading beam, where the actuators are attached.

```
# Define control points
# ---------------------
# expControlPoint tag nodeTag dir resp <-fact f> <-lim l u> ...
expControlPoint 1 3  ux disp uy disp
expControlPoint 2 4  uy disp
expControlPoint 3 3  ux disp ux force uy disp uy force
expControlPoint 4 4  uy disp uy force
```

5.2 Experimental Setup

5.2.1 NoTransformation Experimental Setup

The `NoTransformation` experimental setup is being used in `LShapedColumn_Server1a.tcl`. This experimental setup consists of up to six (6) actuators (in 3D), which are set to control any of the basic degrees of freedom of a specimen (Figure 3). The OpenFresco Command Language Manual has a detailed explanation of how to utilize this experimental setup. The following script executes the `NoTransformation` experimental control Tcl command:

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

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

- `$tag` is the unique setup tag.
- `$ctrlTag` is the tag of a previously defined control object. In this case, it is `SimDomain`.
- `$dirs` are the directions of the imposed displacements in the element basic reference coordinate system, which are set to 2 1 3.
- `$sizeTrial` and `$sizeOut` are the sizes of the element trial and output data vectors, which are both set to 3.
- `$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. Here they are all set to 1.0 -1.0 1.0 to reverse the direction of element degree of freedom 2.

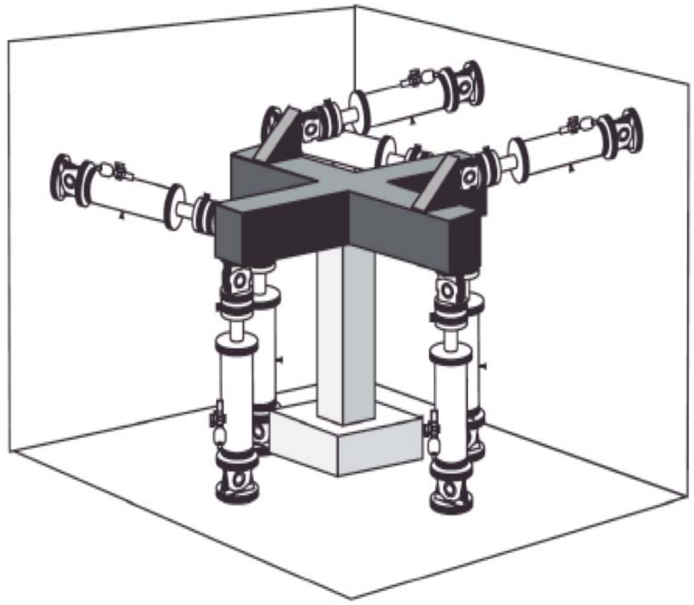

Figure 3: NoTransformation Experimental Setup.

5.2.2 ThreeActuators Experimental Setup

The `LShapedColumn_Server1b.tcl` script uses the `ThreeActuators` experimental setup, as illustrated in Figure 4. This experimental setup consists of three actuators to control the two translational and the rotational degrees of freedom of the specimen. The following script implements the command for the experimental setup:

```
# Define experimental setup
# ------------------------
# expSetup ThreeActuators $tag <-control $ctrlTag> $La1 $La2 $La3 $L1 $L2 <-nlGeom>
<-posAct1 $pos> <-phiLocX $phi> <-trialDispFact $f> ...
expSetup ThreeActuators 1 -control 1 54.0 54.0 54.0 36.0 36.0 -phiLocX 90.0
```

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

- `$tag` is the unique setup tag.
- `$ctrlTag` is the tag of a previously defined control object. In this case, it is `SimDomain`.
- `$La1` is the length of actuator 1.
- `$La2` is the length of actuator 2.
- `$La3` is the length of actuator 3.
- `$L1` is the length of rigid link 1.
- `$L2` is the length of rigid link 2.
- `-nlGeom` is the nonlinear geometry option. The default is off.
- `$pos` is the position of actuator 1. The options are left (l) or right (r). If not specified, the default is left.
- `$phi` is the angle from the rigid loading beam to the local x-axis in degrees. If not specified, the default is 0 degree.
- `$f` are optional factors applied to trial and measured response quantities. The default values are 1.0.

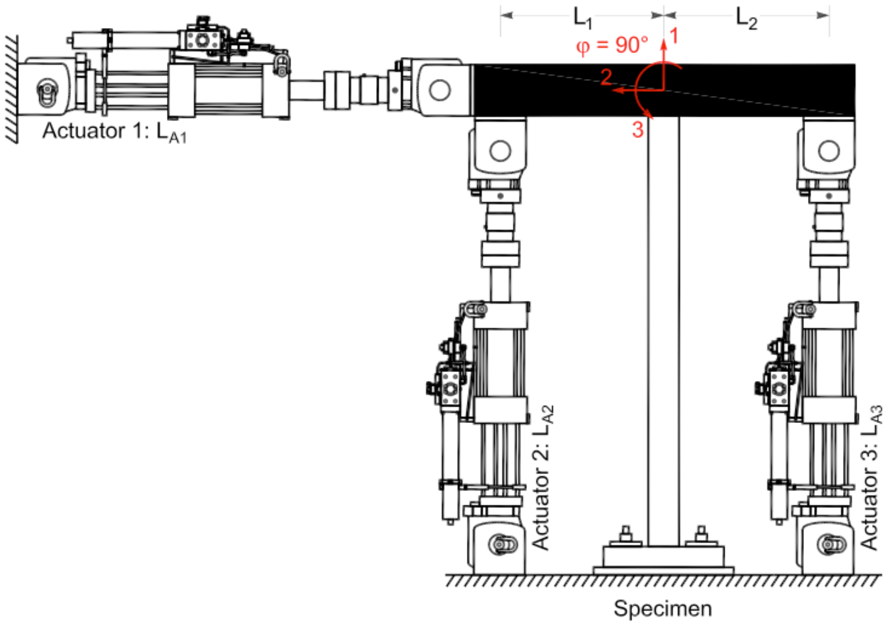

Figure 4: ThreeActuators Experimental Setup.

5.3 Experimental Element

The experimental element is set to be a `beamColumn` experimental element (Figure 5), and it is defined by two nodes. The following script is in `LShapedColumn_Client1.tcl`:

```
# Define experimental element
# ---------------------------
# expElement beamColumn $eleTag $iNode $jNode $transTag -site $siteTag -initStif $Kij ...
<-iMod> <-rho $rho>
expElement beamColumn 1 1 2 1 -site 1 -initStif 1213 0 0 0 11.2 -302.4 0 -302.4 10886.4
```

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 `Linear`.
- `$siteTag` is the tag of a previously defined site object. In this example, it is set to `RemoteSite`.
- `$Kij` is the initial stiffness matrix entered row-wise. For this example,

$$K_{ij} = \begin{bmatrix} 1213 & 0 & 0 \\ 0 & 11.2 & -302.4 \\ 0 & -302.4 & 10886.4 \end{bmatrix}$$

- `-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. The default is 0.

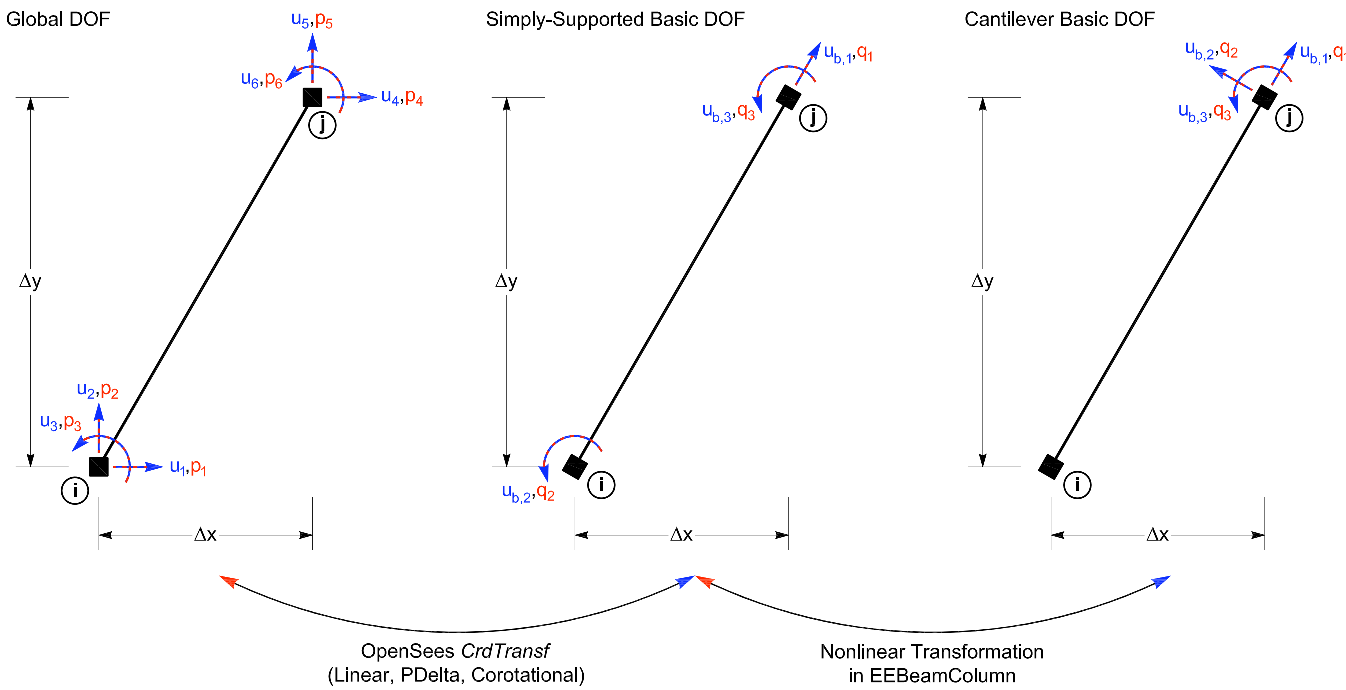

Figure 5: beamColumn Experimental Element.

6 Running Distributed Hybrid Simulation with Setup on Server Side

When using OpenSees as the computational driver, the distributed test may be run with the client-server architecture shown in Figure 6. The experimental setup may be defined on either the client side or the server side. In this example, the experimental setup is located on the server side.

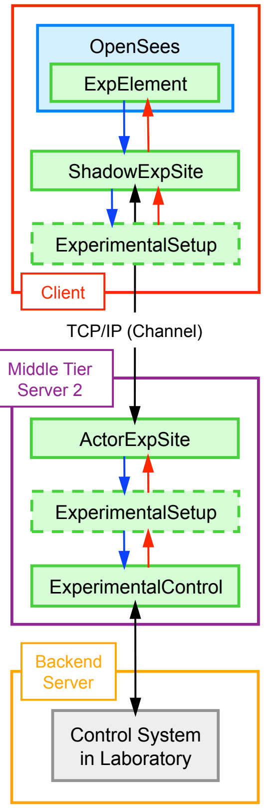

Figure 6: Distributed Hybrid Simulation using OpenSees.

The script excerpt below, from `LShapedColumn_Client1.tcl`, shows that the `expSite` on the client is set to `ShadowSite`:

```
# Define experimental site
# ------------------------
# expSite ShadowSite $tag <-setup $setupTag> $ipAddr $ipPort <-ssl> <-dataSize $size>
expSite ShadowSite 1 "127.0.0.1" 8090
```

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

- `$tag` is the unique site tag.
- `$setupTag` is the optional tag of a previously defined experimental setup object.
- `$ipAddr` is the IP address of the corresponding `ActorSite`.
- `$ipPort` is the IP port number of the corresponding `ActorSite`.
- `-ssl` is an option that uses OpenSSL. The default is off.
- `$size` is the optional data size being sent.

The `expSite` is set to `ActorSite` on the server side using the following script from `LShapedColumn_Server1a.tcl` or `LShapedColumn_Server1b.tcl`:

```
# Define experimental site
# ------------------------
# expSite ActorSite $tag -setup $setupTag $ipPort <-ssl>
expSite ActorSite 1 -setup 1 8090
```

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

- `$tag` is the unique site tag.
- `$setupTag` is the tag of a previously defined experimental setup object.
- `$ipPort` is the IP port number of the `ActorSite`.
- `-ssl` is an option that uses OpenSSL. The default is off.

To run this simulation perform the following steps:

- Start the OpenFresco executable file (OpenFresco.exe) from the directory where `LShapedColumn_Server1a.tcl` and `LShapedColumn_Server1b.tcl` are located.
- At the prompt, type **`source LShapedColumn_Server1a.tcl`** or **`source LShapedColumn_Server1b.tcl`** depending on which experimental setup configuration to run.
- Hit **enter** (Figure 7).

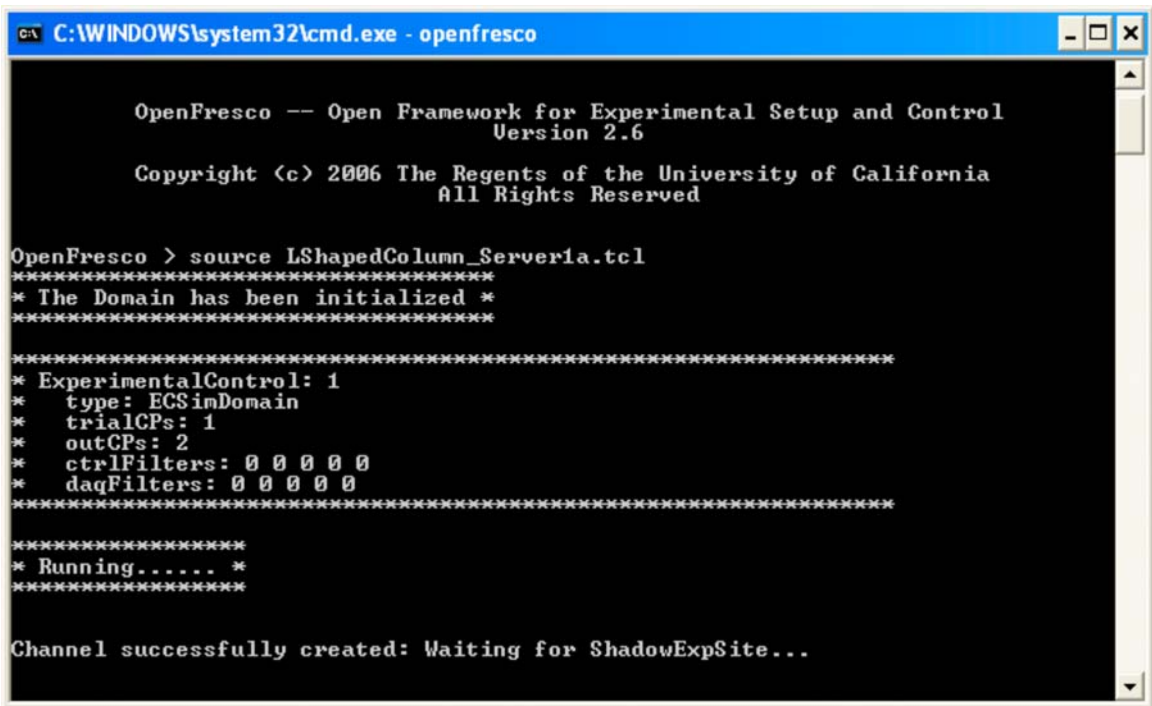

Figure 7: OpenFresco Server Command Window for Distributed Test.

- Start the OpenSees executable file (openSees.exe) from the directory where `LShapedColumn_Client1.tcl` resides.

- At the command window prompt, type **`source LShapedColumn_Client1.tcl`** and hit **enter** (Figure 8). This runs the simulation.

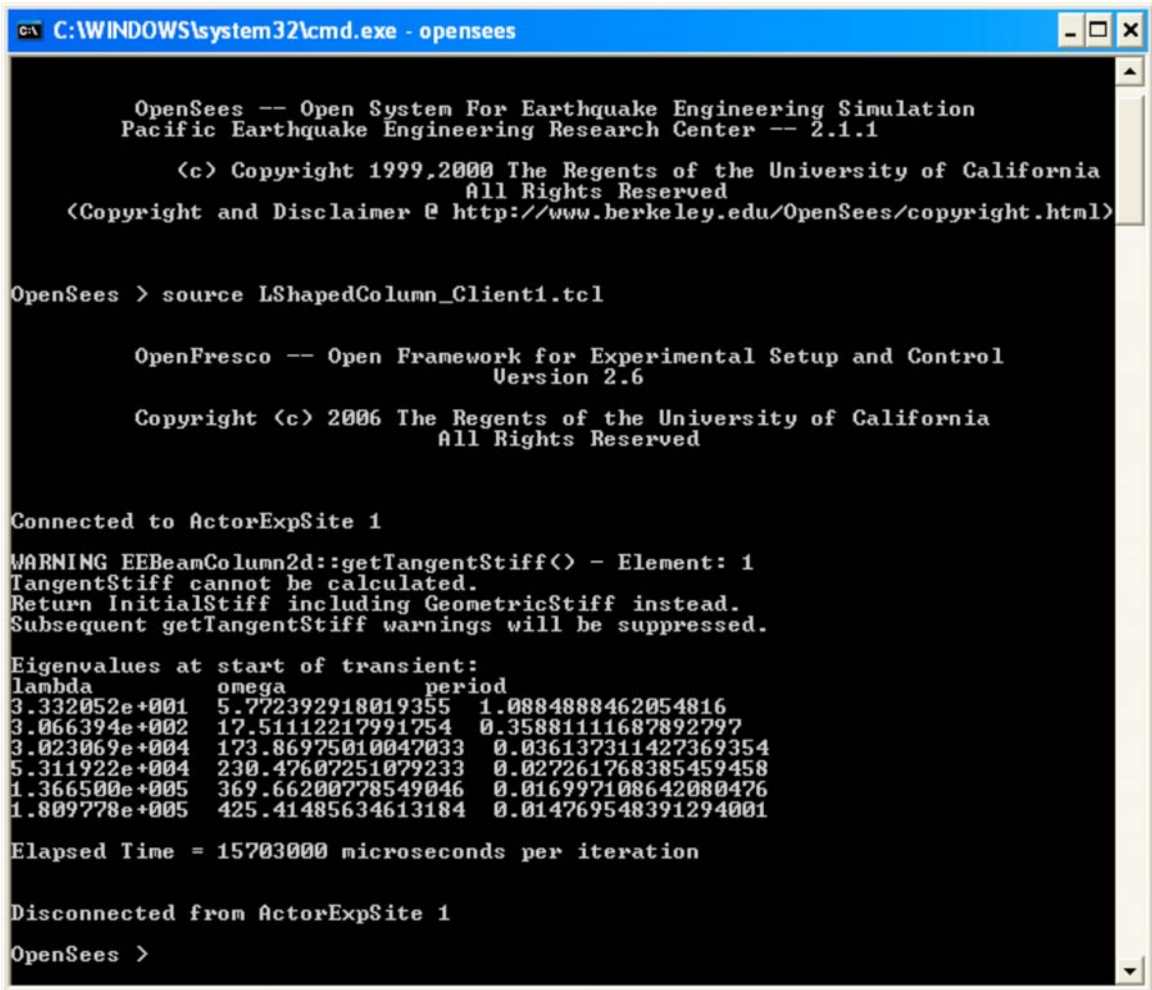

Figure 8: OpenSees Client Command Window for Distributed Test after Simulation.

- After the simulation has completed, the OpenFresco server command window should look like Figure 9.

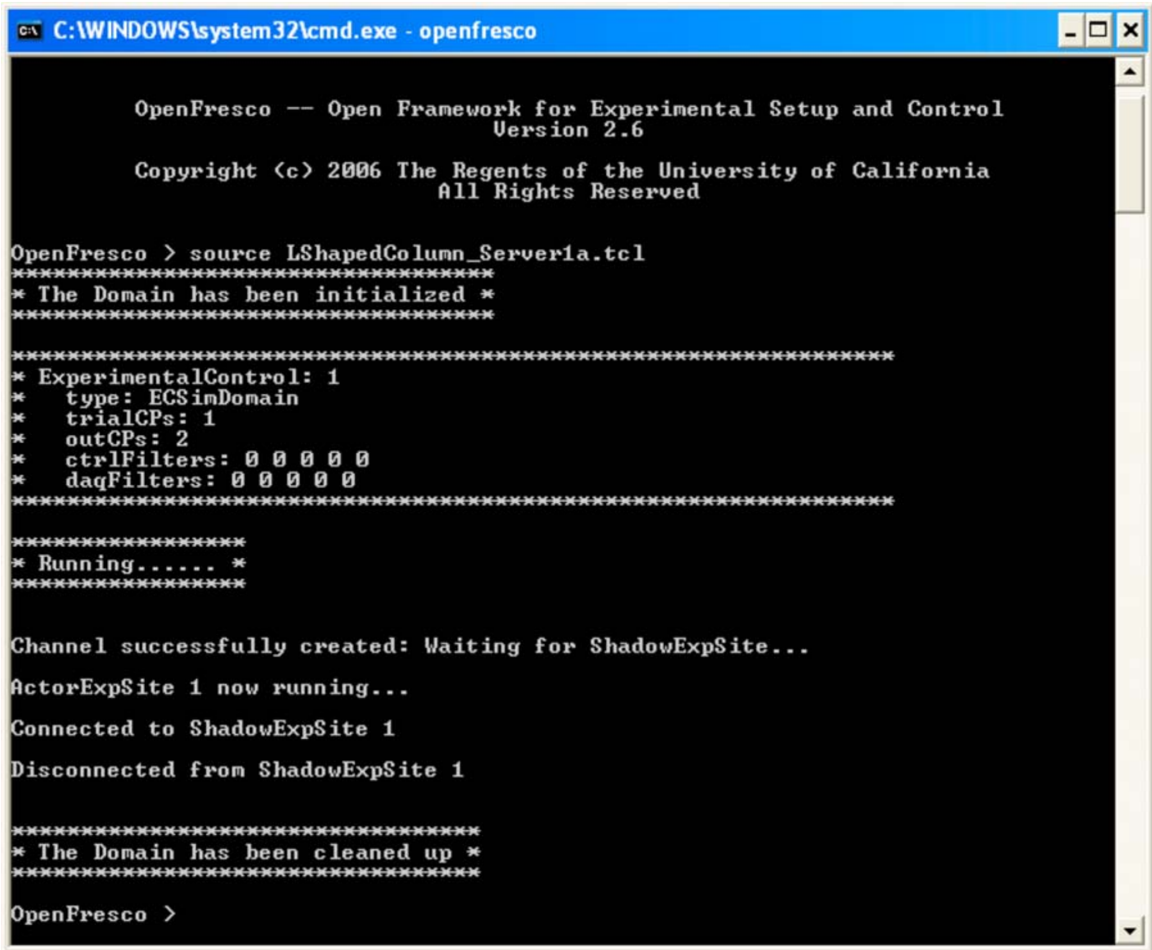

Figure 9: OpenFresco Server Command Window for Distributed Test after Simulation.

7 Results

The OpenSees command window should display the following results:

```
Eigenvalues at start of transient:
lambda              omega             period
3.332052e+001       5.77239291802     1.08848884621
3.066394e+002       17.5111221799     0.358811116879
3.023069e+004       173.7697501       0.0361373114274
5.311922e+004       230.476072511     0.0272617683855
1.366500e+005       369.662007785     0.0169971086241
```

The response quantities for the Simulation Domain example are plotted in Figures 10 to 15.

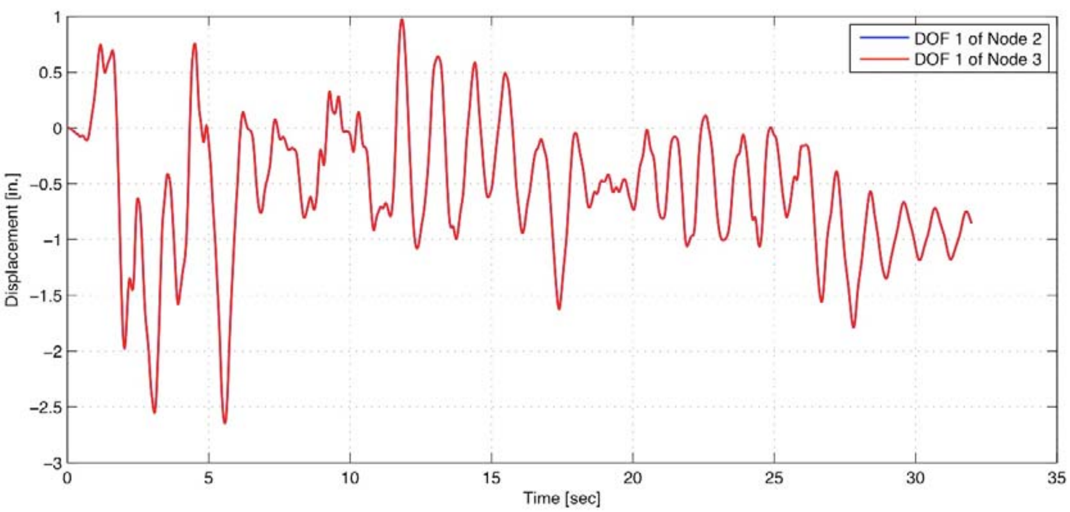

Figure 10: Displacements of DOF 1 vs. Time for Simulation Domain Example.

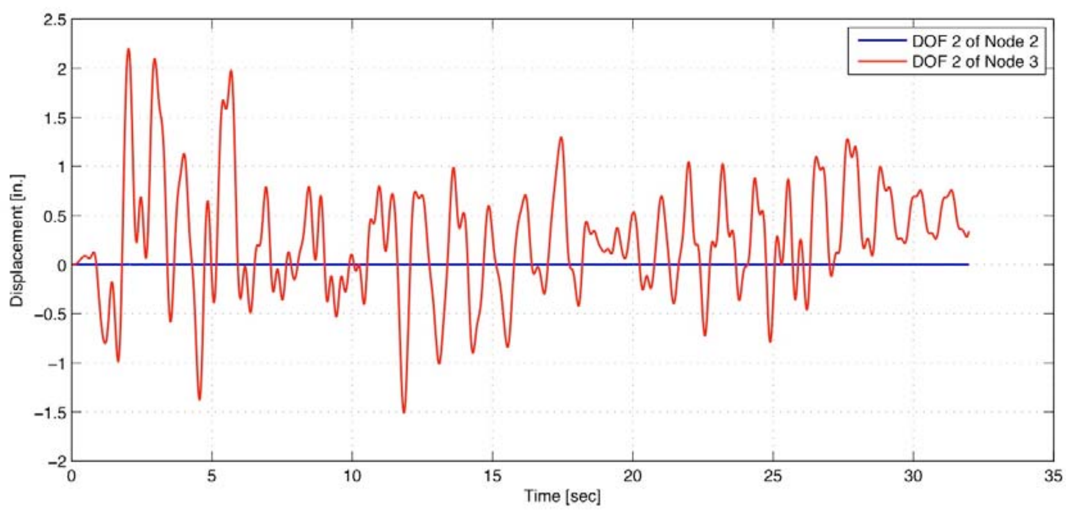

Figure 11: Displacements of DOF 2 vs. Time for Simulation Domain Example.

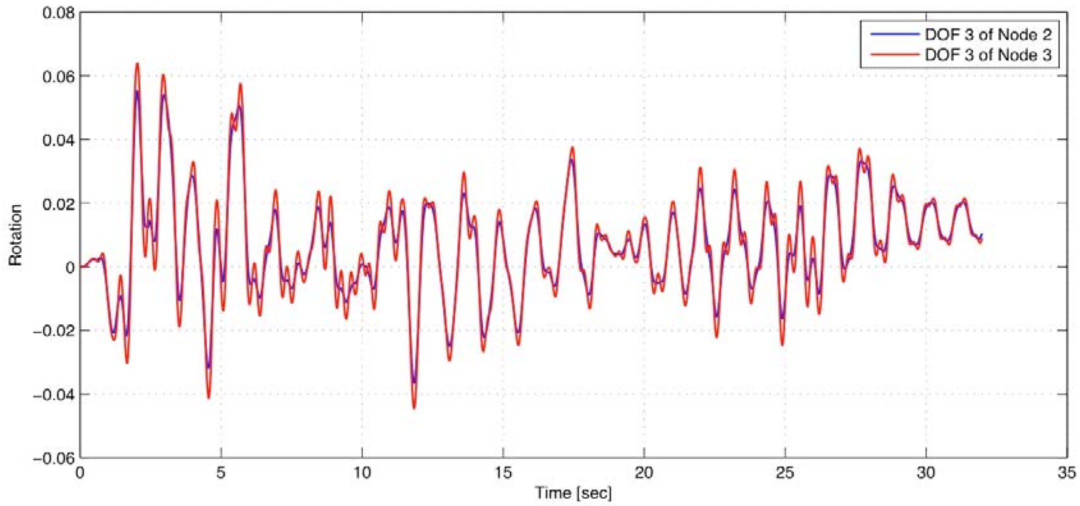

Figure 12: Displacements of DOF 3 vs. Time for Simulation Domain Example.

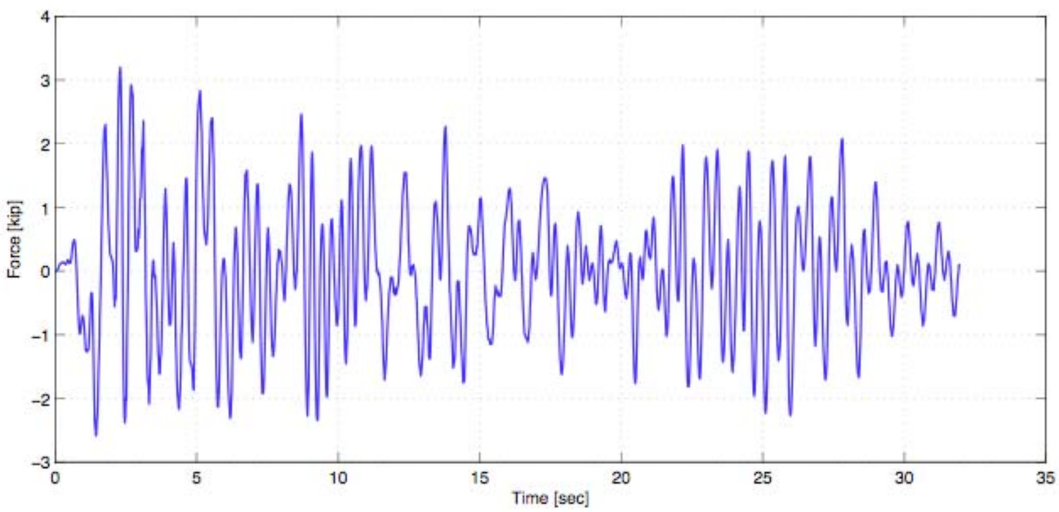

Figure 13: Axial Force (Force 1 of Element 1) vs. Time for Simulation Domain Example.

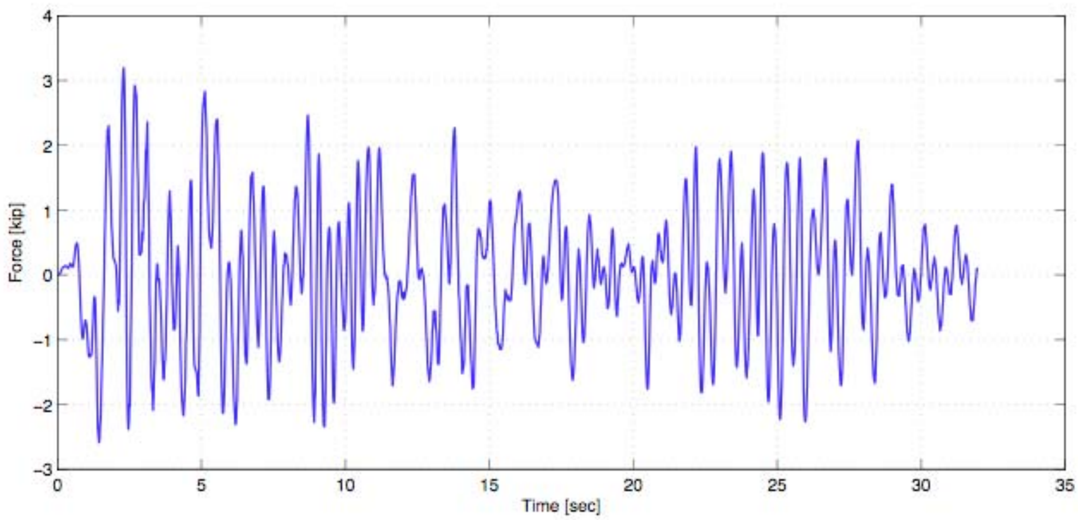

Figure 14: Base Shear (Force 2 of Element 1) vs. Time for Simulation Domain Example.

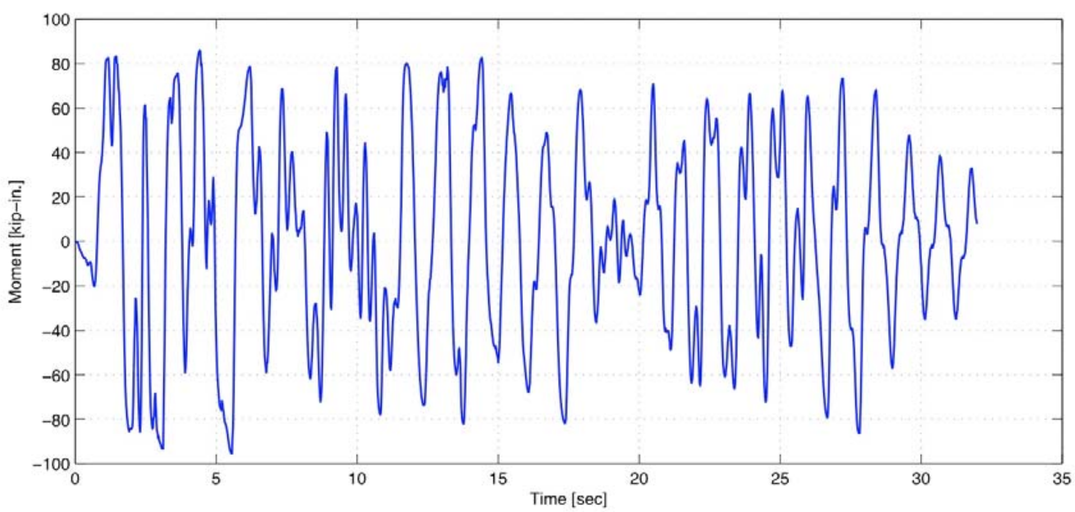

Figure 15: Overturning Moment (Force 3 of Element 1) vs. Time for Simulation Domain Example.

8 References

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