



*Enabling the Network for  
Earthquake Engineering Simulation*



TR-2009-[ID]

# OpenFresco Framework for Hybrid Simulation: OpenSSL 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

## Table of Contents

1	Introduction: Distributed Hybrid Simulation Example Using OpenSSL and OpenSees .....	3
2	Required Files.....	3
3	Structural Model.....	4
4	Ground Motion .....	5
5	OpenFresco Tcl Commands .....	5
5.1	Experimental Control.....	6
5.2	Experimental Setup.....	6
5.3	Experimental Element .....	7
5.4	Experimental Site.....	8
6	Running Distributed Hybrid Simulation with Setup on Server Side.....	9
7	Results .....	12
8	References .....	15

## Table of Figures

Figure 1:	OpenSees Two DOF Model. ....	5
Figure 2:	1940 El Centro Ground Motion. ....	5
Figure 3:	OneActuator Experimental Setup. ....	7
Figure 4:	twoNodeLink Experimental Element.....	8
Figure 5:	Client-Server Architecture Using OpenSees and OpenFresco.....	9
Figure 6:	OpenFresco Server Window for Distributed Test.....	10
Figure 7:	OpenSees Client Command Window for Distributed Test after Simulation. ....	11
Figure 8:	OpenFresco Server Window for Distributed Test after Simulation. ....	12
Figure 9:	Displacements vs. Time for OpenSSL Example.....	13
Figure 10:	Velocities vs. Time for OpenSSL Example. ....	13
Figure 11:	Accelerations vs. Time for OpenSSL Example. ....	14
Figure 12:	Element Hysteresis Loops for OpenSSL Example. ....	14



# 1 Introduction: Distributed Hybrid Simulation Example Using OpenSSL and OpenSees

This example shows how to run a distributed hybrid simulation using OpenFresco with OpenSSL (Open-source Secure Socket Layer). The OpenFresco Installation and Getting Started Manual contains instructions for setting up OpenSSL and generating security certificates. The example structure is a Two-DOF (Degrees Of Freedom) model and the explicit Newmark time integration scheme is employed to solve the equations of motion. The OpenSSL 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. The response results from the simulation are provided for comparison.

## 2 Required Files

For the OpenSSL example, the following files are necessary. These are located in:

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

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

The following files should be in this directory:

- elc.dat
- TwoDOF\_Client1.tcl
- TwoDOF\_Client2.tcl
- TwoDOF\_Local.tcl
- TwoDOF\_Server1.tcl
- TwoDOF\_Server2.tcl

If not done so already, follow the instruction in the OpenSSL How To Guide or the OpenFresco Installation and Getting Started Manual to set up OpenSSL and to generate the necessary security certificates. The certificates and keys that are created during the OpenSSL setup procedure should be in:

C:\Program Files\OpenSSL\bin

The files in this directory are:

- client\_ca.crt
- client.crt
- client.key
- server\_ca.crt
- server.crt
- server.key

The client files need to be placed in the same directory where the client Tcl files reside. The server files need to be placed in the same directory with the server Tcl files.



### 3 Structural Model

The model simulates a single-column bridge pier with two lead-rubber seismic isolation bearings supporting the girder. OpenSees is used to build the model as shown in Figure 1. The model consists of three nodes connected by an analytical `zeroLength` element and an experimental `twoNodeLink` element. The analytical element at the bottom is modeled utilizing a uniaxial elastic material. The experimental element is simulated using a uniaxial `Steel01` material. Two lumped masses, `mass1` and `mass2`, are placed at node 2 and node 3, respectively. SI units are used in the computational model [N, kg, m], and units of ton force and mm are used in the experimental setup. If desired, damping may be added by uncommenting the line under the heading `set the Rayleigh` damping in the input Tcl file. The following Tcl script from `TwoDOF_Client1.tcl` defines the geometry and the variables of the model:

```
# Define geometry for model
# -----
set mass1 130000.; # [kg]
set mass2 240000.; # [kg]
# node $tag $xCrd $yCrd $mass
node 1 0.0 0.0
node 2 0.0 0.0 -mass $mass1 $mass1 0.0
node 3 0.0 0.0 -mass $mass2 $mass2 0.0

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

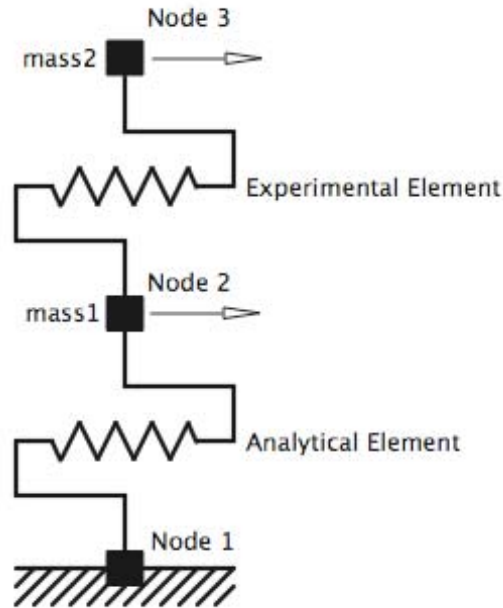
# Define materials
# -----
set E1 35.E6; # [N/m], Stiffness of the column
set E2 50.E6; # [N/m], Stiffness of two isolators
set Fy2 250.E3; # [N], Yield strength of two isolators

# Define similitude
set S 0.5; # ratio of length from prototype to specimen
set factNtoTonf [expr 1./9.8E3]; # from [N] to [tonf]
set factMtoMM 1000.; # from [m] to [mm]

set nIso 2; # number of isolators

set Ee [expr $E2*$factNtoTonf/$factMtoMM*$S/$nIso]; # [tonf/mm], Stiffness of one
isolator in test setup unit
set Fye [expr $Fy2*$factNtoTonf*$S*$S/$nIso]; # [tonf], Yield strength of one
isolator in test setup unit
```

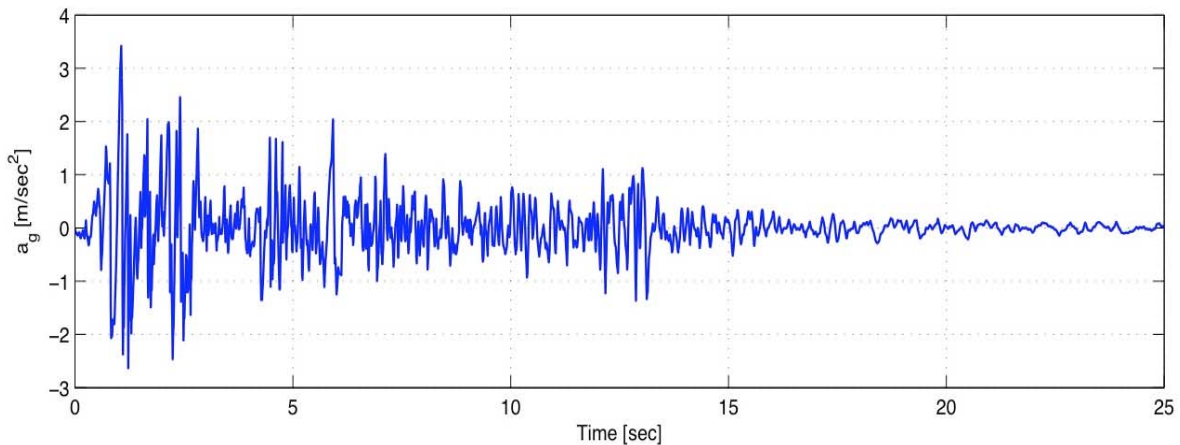




**Figure 1: OpenSees Two DOF Model.**

## 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, `elc.dat`, 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 common OpenFresco Tcl commands used in the input files. Each subsection highlights an OpenFresco Tcl command and the script that contains the command. The following excerpt is from `TwoDOF_Client1.tcl`, and `TwoDOF_Client2.tcl`:



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

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

## 5.1 Experimental Control

The experimental control is set to be `SimUniaxialMaterials` for this example. The `SimUniaxialMaterials` control uses the `Steel01` material, which has a `matTag 2`, to simulate the response of the experimental element. The following excerpt is from `TwoDOF_Server1.tcl`, and `TwoDOF_Server2.tcl`.

```
# uniaxialMaterial Steel01 $matTag $Fy $E $b
uniaxialMaterial Elastic 1 $E
uniaxialMaterial Steel01 2 $Fye $Ee 0.1

# Define experimental control
# -----
# expControl SimUniaxialMaterials $tag $matTags
expControl SimUniaxialMaterials 1 2
```

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

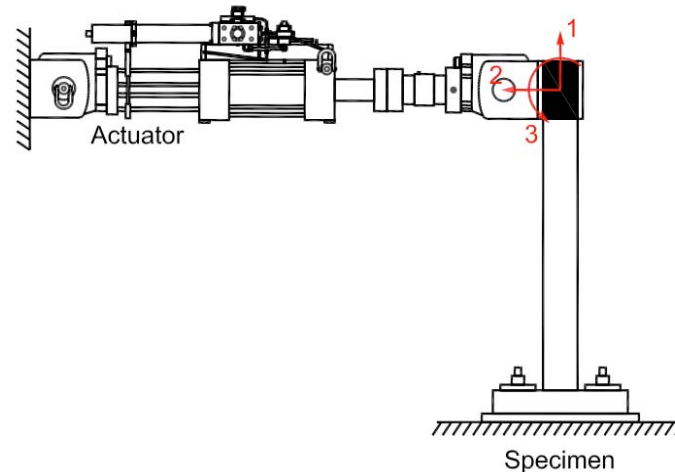
The `OneActuator` setup is being used for the experimental setup (Figure 3). The following script is from `TwoDOF_Server1.tcl` and `TwoDOF_Client2.tcl`. The experimental setup can be located on either the client side or the server side for a distributed test.

```
# Define experimental setup
# -----
# expSetup OneActuator $tag <-control $ctrlTag> $dir -sizeTrialOut $sizeTrial $sizeOut
<-trialDispFact $f> ...
expSetup OneActuator 1 -control 1 1 -sizeTrialOut 1 1 -trialDispFact [expr $S*$factMtoMM]
-outDispFact [expr 1.0/($S*$factMtoMM)]
-outForceFact [expr 1.0/($S*$S*$factNtoTonf/$nIso)]
```

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`. It is optional.
- `$dir` is the direction of the imposed displacement in the element basic reference coordinate system.
- `$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 Experimental Setup.**

### 5.3 Experimental Element

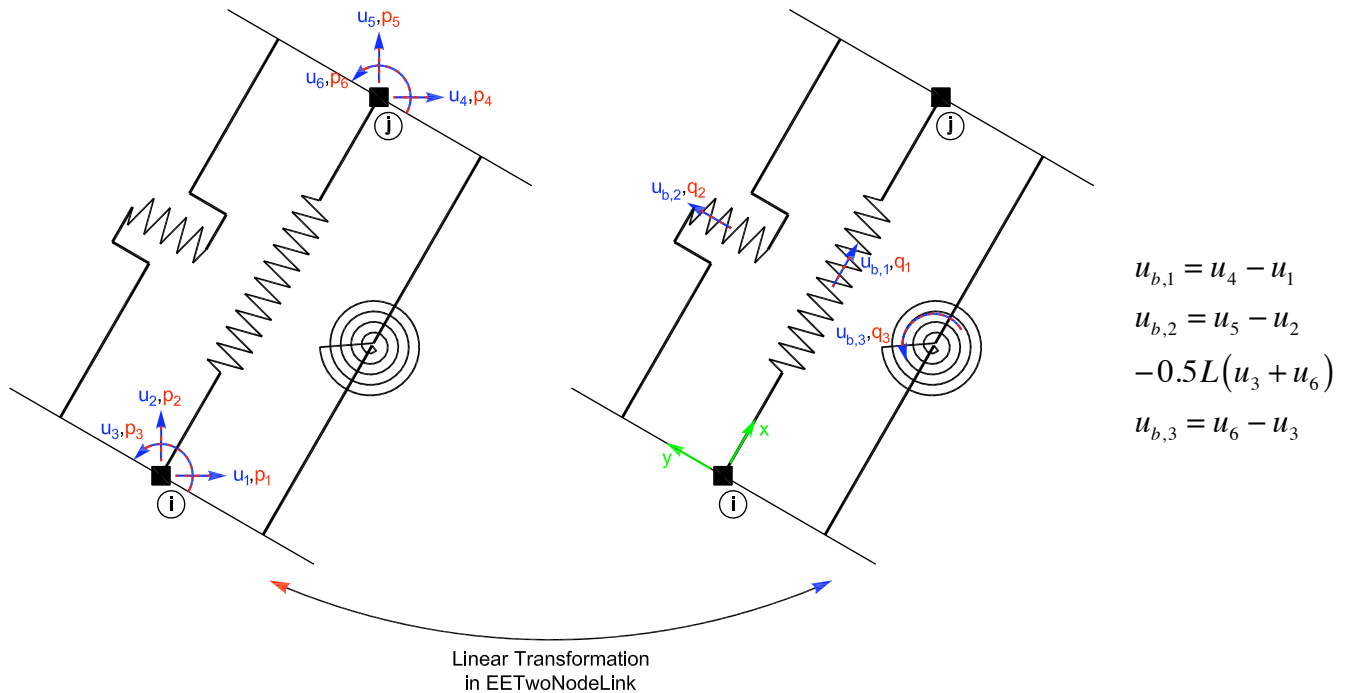
The experimental element is set to be a `twoNodeLink` element (Figure 4). The following script is located in `TwoDOF_Client1.tcl`, and `TwoDOF_Client2.tcl`. In this example, the experimental element is defined directly in the FE software, OpenSees.

```
# Define experimental elements
# -----
# expElement twoNodeLink $eleTag $iNode $jNode -dir $dirs -site $siteTag -initStif $Kij
<-orient <$x1 $x2 $x3> $y1 $y2 $y3> <-pDelta (4 $Mratio)> <-iMod> <-mass $m>
expElement twoNodeLink 2 2 3 -dir 1 -site 1 -initStif $E2 -orient 1 0 0 0 1 0
```

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

- `$eleTag` is the unique element tag.
- `$iNode` and `$jNode` are the end nodes that the `twoNodeLink` element connects to.
- `$siteTag` is the tag of a previously defined site object. In this example, it is the `ShadowSite` for the distributed 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 4: twoNodeLink Experimental Element.**

### 5.4 Experimental Site

The experimental site is defined as ShadowSite in TwoDOF\_Client1.tcl and TwoDOF\_Client2.tcl. The following excerpt is from TwoDOF\_Client1.tcl.

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

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.

In TwoDOF\_Server1.tcl and TwoDOF\_Server2.tcl, the experimental site is set to ActorSite. The following excerpt is from TwoDOF\_Server1.tcl.

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

The expSite command parameters for ActorSite are:



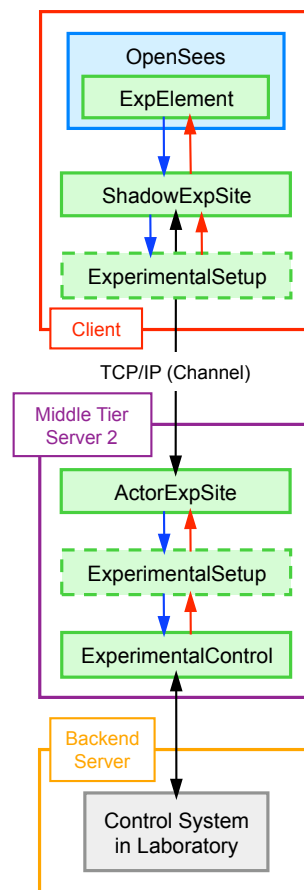


- `$tag` is the unique site tag.
- `$setupTag` is the tag of a previously defined experimental setup object.
- `$ipPort` is the IP port number of the `ActorSite`.
- `-ssl` is an option that uses OpenSSL. The default is off.

The `ActorSite` and `ShadowSite` commands are used during the distributed hybrid simulation and the `-ssl` option is turned on for this example.

## 6 Running Distributed Hybrid Simulation with Setup on Server Side

A distributed test is preformed using a client-server architecture as shown in Figure 5. While a multi-tier architecture is more commonly used for a distributed test in OpenFresco, there is a less complex client-server option that can be employed when using OpenSees as the computational driver. This option runs a bit faster than the general multi-tier option. In some ways, it is easier to setup and run. For this simulation, OpenSees is the client, and OpenFresco is the server. The following steps show how to run a distributed test with the experimental setup on the server side. To run a distributed test with the setup on the client side, follow the same directions but use `TwoDOF_Client2.tcl` instead of `TwoDOF_Client1.tcl` and `TwoDOF_Server2.tcl` instead of `TwoDOF_Server1.tcl`.

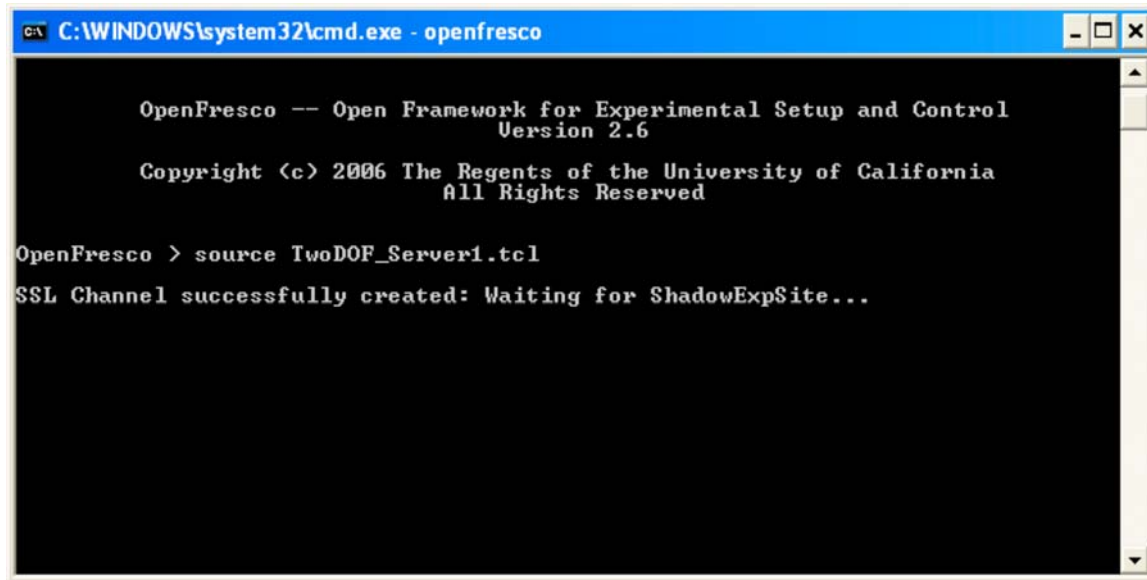


**Figure 5: Client-Server Architecture Using OpenSees and OpenFresco.**



To run this simulation perform the following steps:

- Start the OpenFresco executable file (OpenFresco.exe) from the directory where the TwoDOF\_Server1.tcl resides.
- At the prompt, type **source TwoDOF\_Server1.tcl** and hit **enter** (Figure 6).



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

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

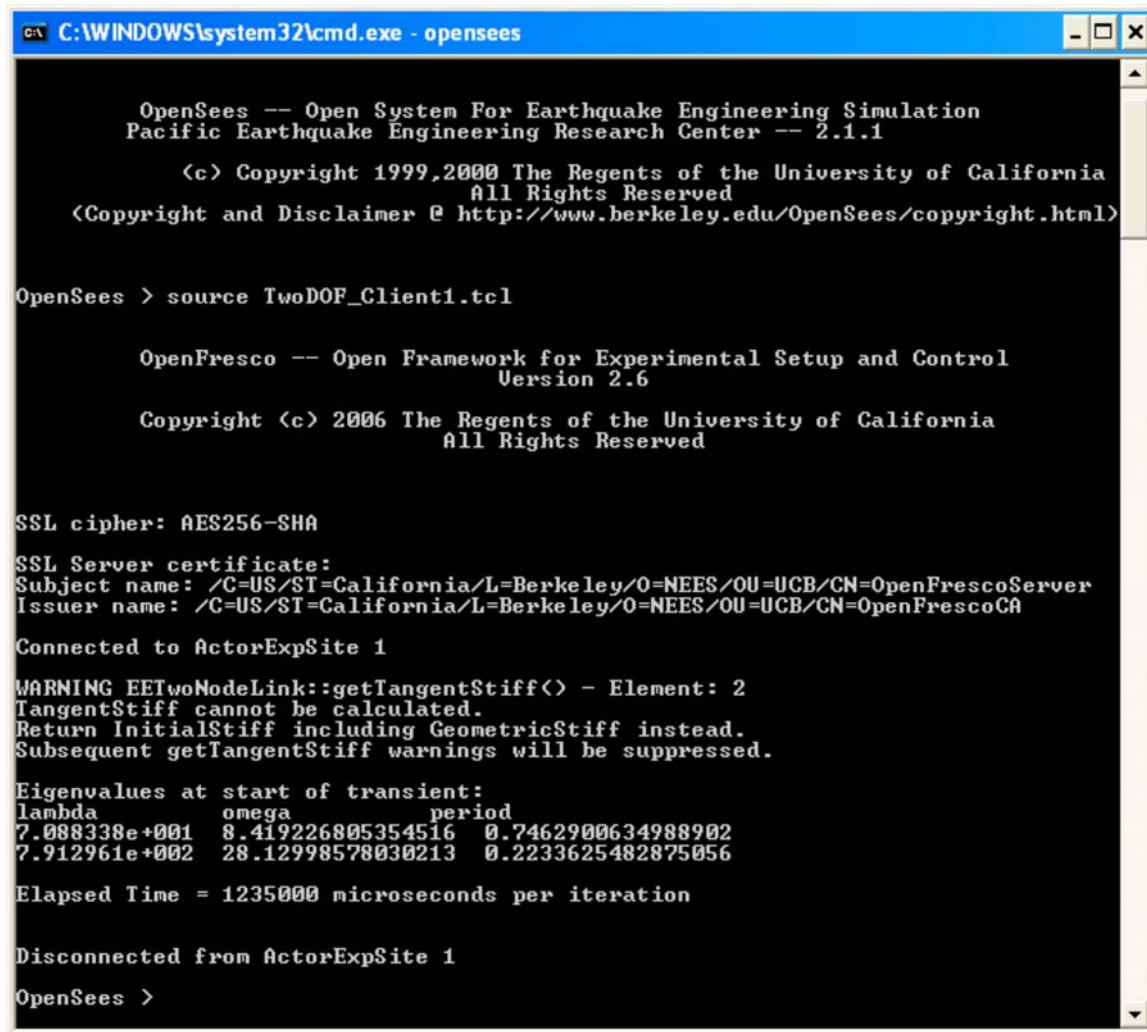
Copyright (c) 2006 The Regents of the University of California
All Rights Reserved

OpenFresco > source TwoDOF_Server1.tcl
SSL Channel successfully created: Waiting for ShadowExpSite...
```

**Figure 6: OpenFresco Server Window for Distributed Test.**



- Start the OpenSees executable file (openSees.exe) from the directory where the TwoDOF\_Client1.tcl resides.
- At the command window prompt, type **source TwoDOF\_Client1.tcl** and hit **enter** (Figure 7). This runs the simulation.



```

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 TwoDOF_Client1.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

SSL cipher: AES256-SHA
SSL Server certificate:
Subject name: /C=US/ST=California/L=Berkeley/O=NEES/OU=UCB/CN=OpenFrescoServer
Issuer name: /C=US/ST=California/L=Berkeley/O=NEES/OU=UCB/CN=OpenFrescoCA

Connected to ActorExpSite 1

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
7.088338e+001  8.419226805354516  0.7462900634988902
7.912961e+002  28.12998578030213  0.2233625482875056

Elapsed Time = 1235000 microseconds per iteration

Disconnected from ActorExpSite 1

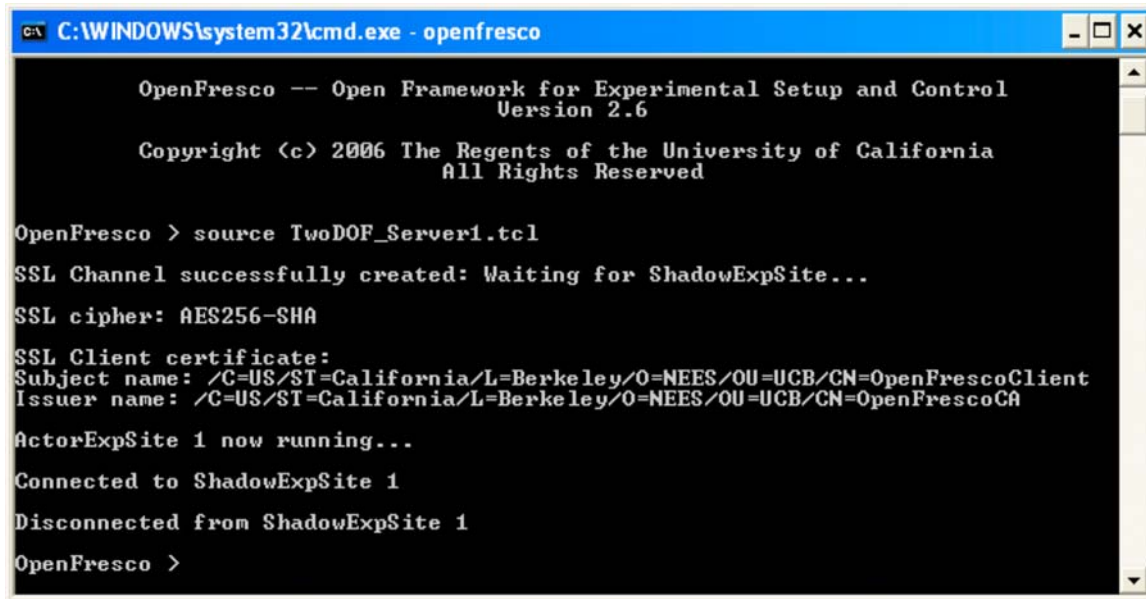
OpenSees >

```

Figure 7: OpenSees Client Command Window for Distributed Test after Simulation.



- After the simulation, the OpenFresco server command window looks like Figures 8.



```

C:\WINDOWS\system32\cmd.exe - openfresco

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

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

OpenFresco > source TwoDOF_Server1.tcl
SSL Channel successfully created: Waiting for ShadowExpSite...
SSL cipher: AES256-SHA
SSL Client certificate:
Subject name: /C=US/ST=California/L=Berkeley/O=NEES/OU=UCB/CN=OpenFrescoClient
Issuer name: /C=US/ST=California/L=Berkeley/O=NEES/OU=UCB/CN=OpenFrescoCA
ActorExpSite 1 now running...
Connected to ShadowExpSite 1
Disconnected from ShadowExpSite 1
OpenFresco >

```

Figure 8: OpenFresco Server 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
7.088338e+001   8.419226805354516  0.7462900634988902
7.912961e+002   28.12998578030213  0.2233625482875056

```

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

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

The following are the output files:

- Elmt\_Hys.out
- Node\_Acc.out
- Node\_Dsp.out
- Node\_Vel.out

These files are created using the recorder command. Below is 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>).



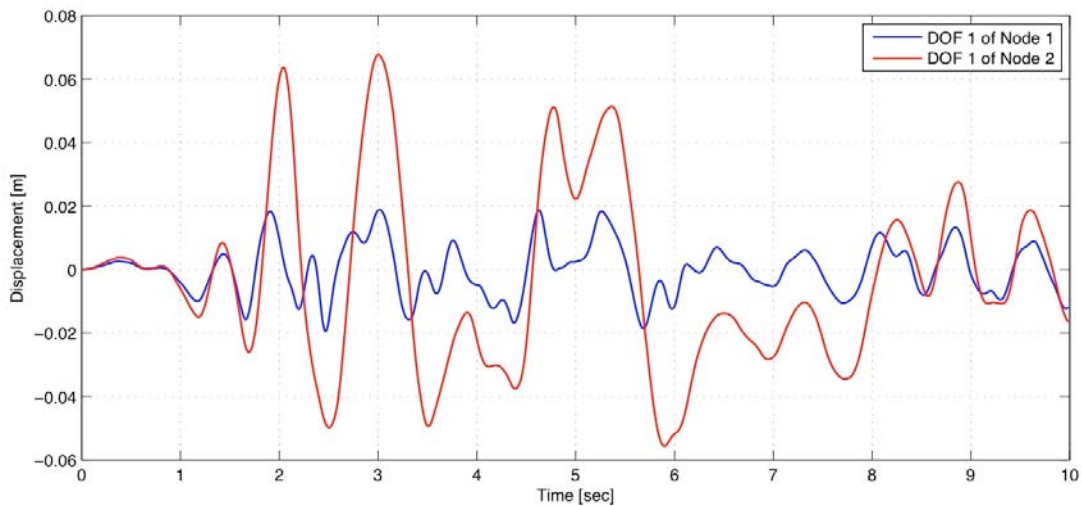
```

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

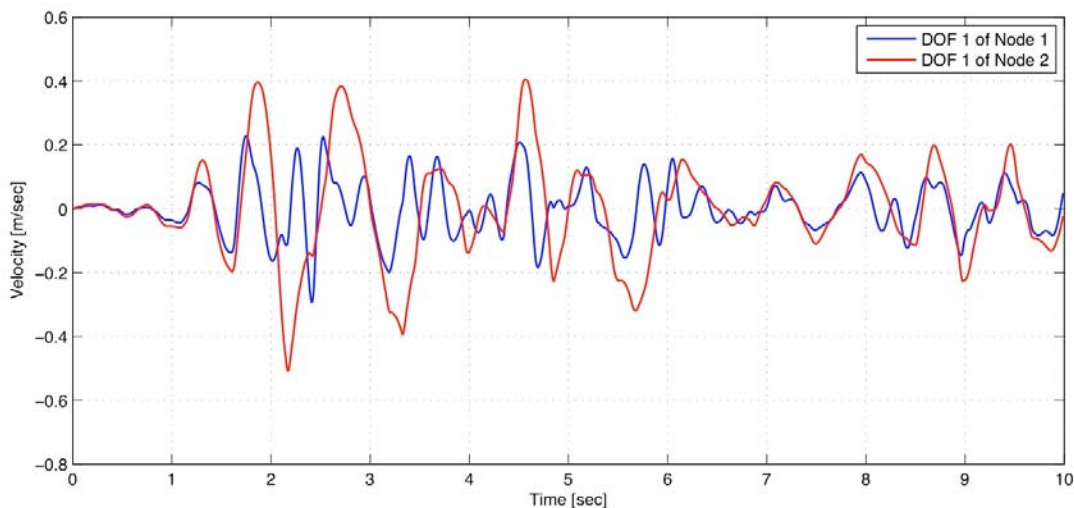
recorder Element -file Elmt_Hys.out -time -ele 1 2 deformationsANDforces

```

The response quantities are plotted in Figures 9 to 12.



**Figure 9: Displacements vs. Time for OpenSSL Example.**



**Figure 10: Velocities vs. Time for OpenSSL Example.**



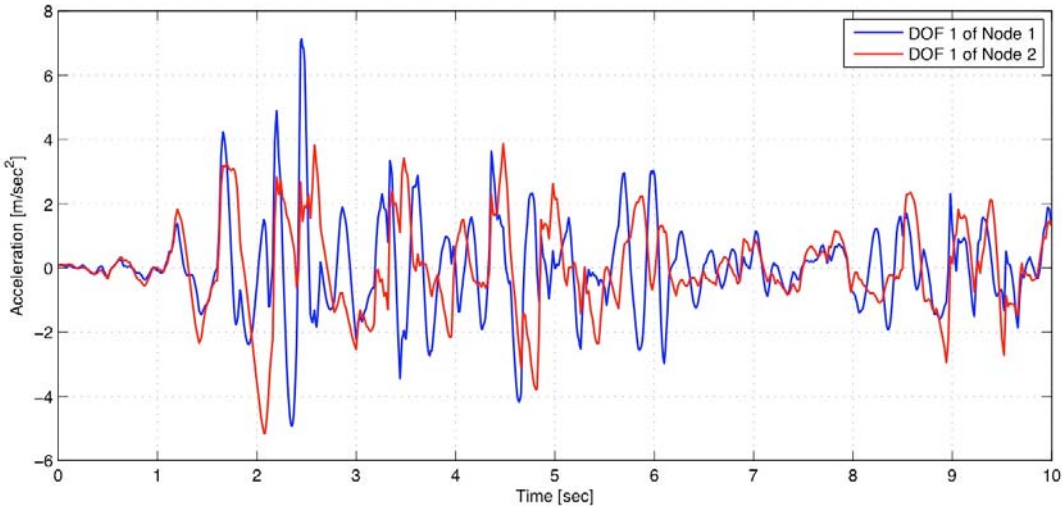


Figure 11: Accelerations vs. Time for OpenSSL Example.

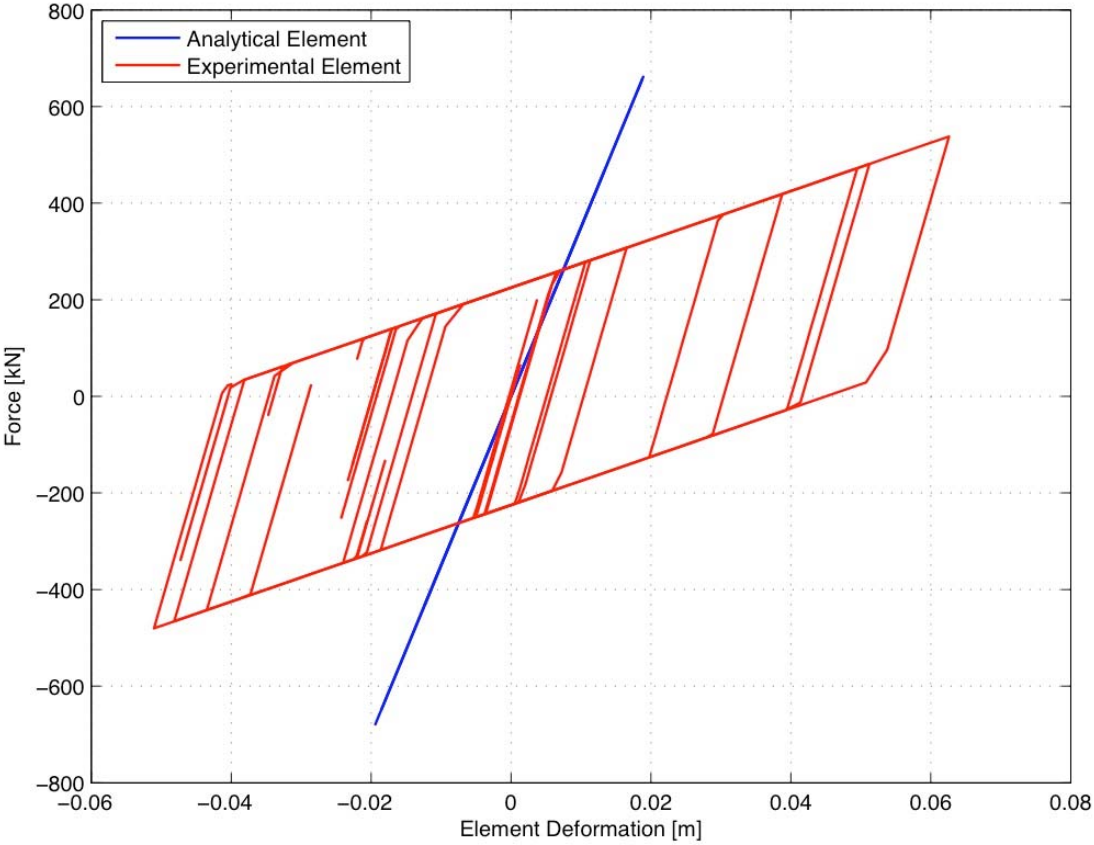


Figure 12: Element Hysteresis Loops for OpenSSL Example.



## 8 References

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

