



*Enabling the Network for
Earthquake Engineering Simulation*



TR-2009-[ID]

OpenFresco Framework for Hybrid Simulation: UI-SimCor v2.6 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: 2009-08-14 Version: 2.6

Table of Contents

1	Introduction: UI-SimCor Example Using One-Bay Frame Model	3
2	Required Files.....	3
3	Structural Model.....	3
4	Ground Motion	4
5	The Input Scripts	4
5.1	Coordinator	5
5.2	OpenFresco Modules	6
5.3	OpenSees Module	7
6	Running the Example	8
7	Results	17
8	References	19

Table of Figures

Figure 1: One-Bay Frame Model.....	4
Figure 2: 1940 El Centro Ground Motion.	4
Figure 3: Local Hybrid Simulation using The Experimental Element.	8
Figure 4: OpenFresco Command Window for Element 1.....	9
Figure 5: OpenFresco Command Window for Element 2.....	9
Figure 6: NEES-SAM Open Window for Element 3.	10
Figure 7: NEES-SAM Window.....	10
Figure 8: MATLAB [®] Command Window starting UI-SimCor.	11
Figure 9: UI-SimCor GUI.....	11
Figure 10: Simulation Monitor of Element 1.	12
Figure 11: Simulation Monitor of Element 2.	12
Figure 12: Simulation Monitor of Element 3.	13
Figure 13: OpenFresco Command Window for Element 1.....	13
Figure 14: OpenFresco Command Window for Element 2.....	14
Figure 15: NEES-SAM Window after Connection is established.....	14
Figure 16: UI-SimCor GUI after Simulation.....	15
Figure 17: Simulation Monitor of Element 1 after Simulation.....	15
Figure 18: Simulation Monitor of Element 2 after Simulation.....	16
Figure 19: Simulation Monitor of Element 3 after Simulation.....	16
Figure 20: NEES-SAM Window after Simulation.	17
Figure 21: Displacements vs. Time for UI-SimCor Example.	18
Figure 22: Element Hysteresis Loops for UI-SimCor Example.....	18



1 Introduction: UI-SimCor Example Using One-Bay Frame Model

This example shows how UI-SimCor version 2.6¹ can be used as the coordinator in a hybrid simulation with two OpenFresco modules and one OpenSees module. UI-SimCor or the simulation coordinator is a MATLAB[®] based software that runs hybrid simulations (Kwon et al. 2007). This example uses a simple One-Bay Frame model with the Alpha-OS time integration scheme. The One-Bay Frame 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. The response results from the simulation are provided for comparison.

2 Required Files

For the UI-SimCor example, the following files are necessary. They are located in:

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

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

The following files are in the \00_Coordinator directory:

- SimConfig.m
- elcentro.txt
- Query_OpenFresco1D.m

The following files are in the \01_LeftCol_OpenFresco directory:

- ColumnLeft.tcl

The following files are in the \02_Beam_OpenSees directory:

- Beam.tcl
- Module.cfg
- StaticAnalysisEnv.tcl

The following files are in the \03_RightCol_OpenFresco directory:

- ColumnRight.tcl

3 Structural Model

The model consists of two columns, Element 1 and 2, connected by a spring, Element 3. A lumped mass is placed at the top of each column. The two column bases are fixed. The columns are axially rigid, and the tops are free to rotate. Imperial units are used [in., kips, and sec].

¹ This document does not explain how to install and run UI-SimCor. Refer to the [UI-SimCor neesforge website](http://simcor.neesforge.nees.org) (<http://simcor.neesforge.nees.org>) for detailed information on UI-SimCor and NEES-SAM. Note that UI-SimCor requires the Instrument Control Toolbox from Mathworks to run.



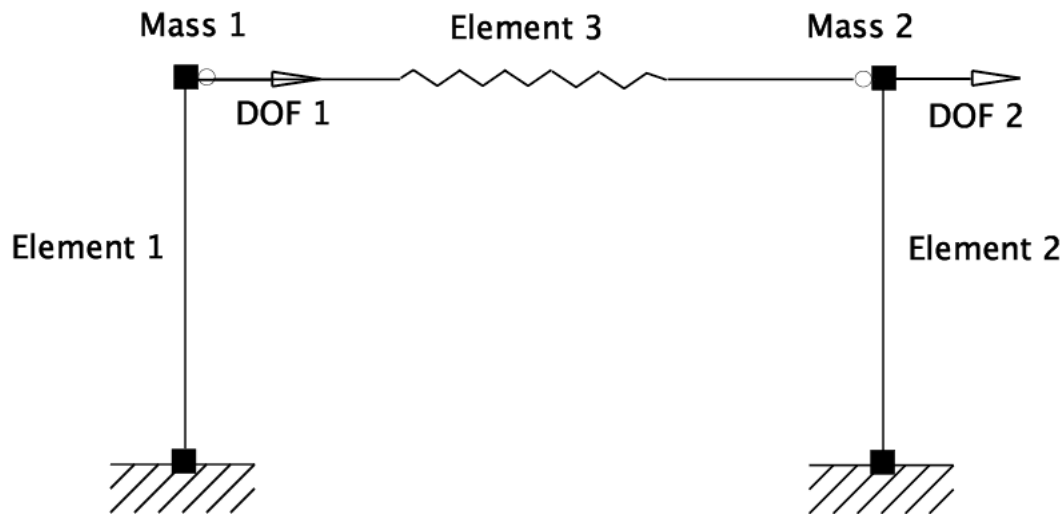


Figure 1: One-Bay Frame Model.

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).

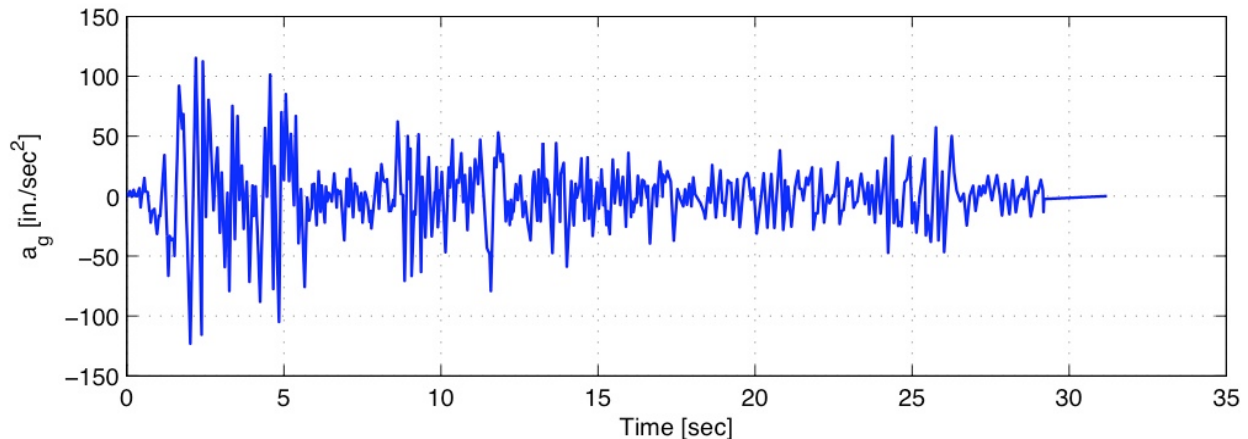


Figure 2: 1940 El Centro Ground Motion.

5 The Input Scripts

This example uses one coordinator and three restoring force modules. The first and third modules simulate Element 1 and Element 3 in OpenFresco, respectively, while the second module simulates Element 2 using OpenSees. This section explains the function of each module along with its input scripts. Section 2 shows the location of all the input scripts.

5.1 Coordinator

As the name suggests, the coordinator coordinates the processes of all the modules during a hybrid simulation. Among other actions, the coordinator is responsible for advancing the Alpha OS time integration scheme. The input script for this example is `SimConfig.m`. `SimConfig.m` is a MATLAB® script. Only the important parts of the script are highlighted. Refer to the UI-SimCor user manual for more information.

```
% Integration parameter related to the alpha-OS method.
% Alpha = (0 ~ 1/3). In most cases, SC.Alpha = 0.05 worked.
Sys.Alpha = 0.05;
Sys.Beta = 1/4*(1+Sys.Alpha)^2;
Sys.Gamm = 1/2 + Sys.Alpha;
```

The above portion of `SimConfig.m` sets the parameters for the Alpha OS time integration scheme. It is worth noting that currently Alpha OS is the only time integration scheme available in UI-SimCor v2.6. α is set to 0.05 and β and γ are subsequently calculated from α . Here β is 0.28, and γ is 0.55.

```
% Number of dynamic analysis steps
Sys.Num_Dynamic_Step = 500;

% Dynamic analysis time steps
Sys.dt = 0.02;
```

The number of dynamic steps or the number of time steps is set to 500. The coordinator will stop the simulation on the number of time steps is reached regardless of how many time steps the recorded ground motions contains. The duration of each time step is set to 0.02 sec.

```
% Number of restoring force modules.
Sys.Num_RF_Module = 3;

% Number of auxiliary modules.
Sys.Num_AUX_Module = 0;

% Total number of effective nodes. Effective nodes are interface nodes between
% modules and nodes where lumped masses are defined.
Sys.Num_Node = 2;

% Lumped mass assigned for each DOF for each node.
% Node number = x, y, z, rx, ry, rz directional mass
Sys.Node_Mass{1} = [0.04, 0.04, 0, 0, 0, 0];
Sys.Node_Mass{2} = [0.02, 0.02, 0, 0, 0, 0];
```

It is important that the number of restoring force modules and the number of nodes are correct. In this case, there are three restoring force modules and two nodes. Each node is assigned directional mass as shown above.

```
% URL of each module
% Format - IP address:port number
% ex) 'http://cee-nsp4.cee.uiuc.edu:11997'
% for local machine - '127.0.0.1:11997'
MDL(1).URL = '127.0.0.1:8090';
MDL(2).URL = '127.0.0.1:8091';
MDL(3).URL = '127.0.0.1:8092';
```



As seen from the IP addresses of each module, a local hybrid simulation is performed in this example.

```
% Communication protocol for each module.
%      NTCP      : communicate through NEESPOP server
%      TCPIP     : binary communication using TCPIP
%      LabView1   : ASCII communication with LabView plugin format (Propose-Query-
Execute-Query)
%      LabView2   : same as LabView1 but Propose-Query
%      OpenFresco1D : OpenFresco, only 1 DOF is implemented now.
%      NHCP      : NHCP, linear 1 DOF simulation mode, Mini MOST 1 and 2 at UIUC or
SDSC
MDL(1).protocol = 'OpenFresco1D';
MDL(2).protocol = 'LabView1';
MDL(3).protocol = 'OpenFresco1D';
```

The coordinator offers several different communication protocols. The OpenFresco1D communication protocol is used to communicate with the OpenFresco restoring force modules. The OpenFresco 1D communication utilizes the `TCPSocket.mex2` file in OpenFresco for communications. The LabView1 communication protocol is used to communicate with the NEES Static Analysis Module (NEES-SAM), which is running the OpenSees module.

```
% Module 1: ColumnLeft -----
MDL(1).node   = [1];           % Control point node number
MDL(1).EFF_DOF = [1 0 0 0 0 0]; % Effective DOF for CP 1

% Module 2: Beam -----
MDL(2).node   = [1 2];         % Control point node number
MDL(2).EFF_DOF = [1 0 0 0 0 0 % Effective DOF for CP 1
                  1 0 0 0 0 0]; % Effective DOF for CP 2

% Module 3: ColumnRight -----
MDL(3).node   = [2];           % Control point node number
MDL(3).EFF_DOF = [1 0 0 0 0 0]; % Effective DOF for CP 1
```

Each module is assigned one or more control point nodes with the effective degrees-of-freedom associated with each node. The One-Bay Frame model has two nodes with one horizontal degree-of-freedom at each node.

5.2 OpenFresco Modules

Two OpenFresco restoring force modules are employed for this example. Module 1 simulates the left column (Element 1) of the structure, while Module 3 simulates the right column (Element 2). The Tcl scripts used to run Module 1 and Module 3 are `ColumnLeft.tcl` and `ColumnRight.tcl` respectively. These modules receive the target displacements at each time step from the coordinator and return the resisting forces after the displacements are imposed.

`ColumnLeft.tcl` and `ColumnRight.tcl` are very similar to `OneBayFrame_Local_SimAppServer.tcl` script from the One-Bay Frame Example using OpenSees. Identical experimental control, setup, and element commands are used in all three input files.

² For more information on compiling the `TCPSocket.mex` file, refer to the OpenFresco MATLAB[®] Example Manual.



For more information on the Tcl commands used in the scripts, refer to Section 5.5 of the OpenFresco Installation and Getting Started Guide.

5.3 OpenSees Module

UI-SimCor works in tandem with NEES-SAM to run the OpenSees module. NEES-SAM is included in the UI-SimCor v2.6 release. There are several different NEES-SAM executable files. This example uses NEESSAM_LabView1.exe since the coordinator is set to use the LabView1 communication protocol with the OpenSees module.

The beam.tcl³ Tcl input file contains the structural properties of Element 3. It defines the dimensions, boundary conditions, material properties, and type of the element. The element is 100 inches long where the vertical translations of the nodes are fixed. It is using an OpenSees elastic uniaxial material with a modulus of 200 ksi and an OpenSees truss element with a stiffness of 2 kips/in. The StaticAnalysisEnv.tcl file contains the analysis information for the module. For this example, the Modified Newton method under load control is used for the analysis.

The Module.cfg file defines the configuration parameters for the OpenSees Module. The following script is an excerpt from Module.cfg.

```
# Connection port to controller
Port = 8091

# Module application, 1 for Zeus-NL, 2 for OpenSees
MDL_Type = 2

# Effective node numbers in Simulation coordinator.
# The order of node number should be identical to that specified in the simulation
# coordinator configuration file.
SC_Node = [1 2]

# Corresponding effective node numbers in the model.
MDL_Node = [1 2]

# Effective DOFs in control point.
EFF_DOF: Use one line per each control point
1 0 0 0 0 0
1 0 0 0 0 0

# model file name without extension
MODEL_FILE = 'Beam'

# Model dimension
MDL_Dim = 2

# Time history monitoring point, Node number, direction (x, y, z, rx, ry, rz), D for disp
or F for force
# This should be defined after SC_Node and MDL_Node
TH_MONITOR = 1, x, D

# Disp-Force monitoring point
DF_MONITOR = 1, x
```

³ Refer to the [OpenSees website](http://opensees.berkeley.edu) (<http://opensees.berkeley.edu>) for more information on OpenSees.



The parameters in this file should be consistent with the parameters defined in the `SimConfig.m` file. The port number should match the port number defined in the `SimConfig.m` file for the OpenSees module, Module 2. `MDL_Node` should also be identical to `MDL(2).node` in `SimCor.m`. The rest of the parameters are clearly explained by the comment lines.

6 Running the Example

This example uses the three-tier architecture for running a local hybrid simulation with an experimental element and a simulation application site server (Figure 3).

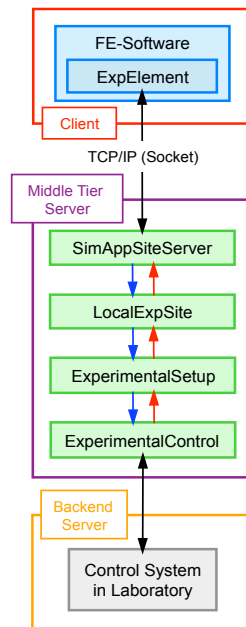


Figure 3: Local Hybrid Simulation using The Experimental Element.

The experimental site is set to `LocalSite`. There is a client to middle-tier-server communication in this example. The code segment below is in the `ColumnLeft.tcl` and `ColumnRight.tcl` scripts.

```
# 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.

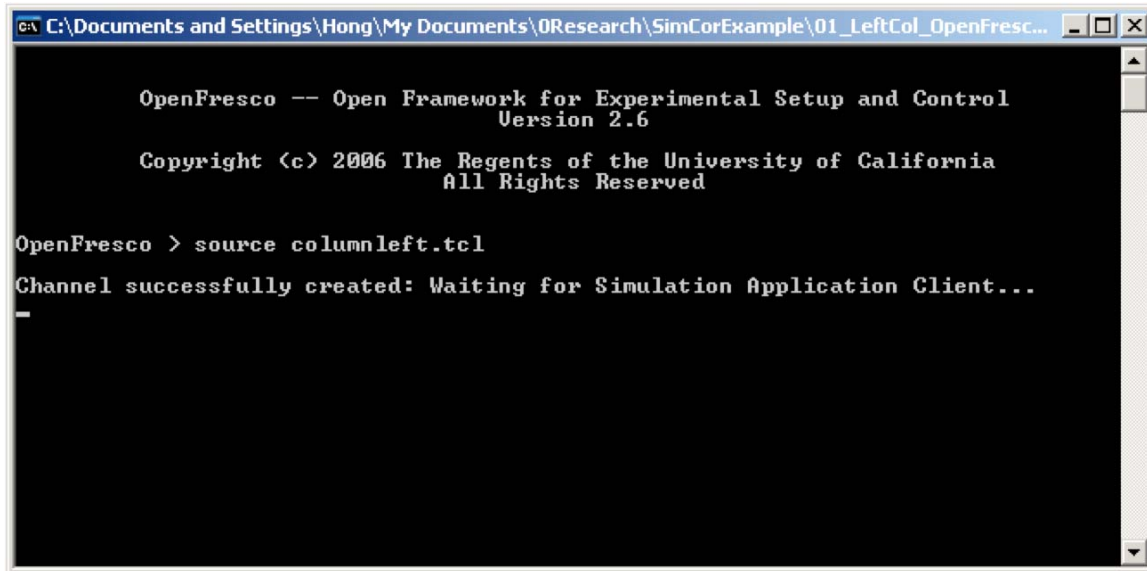
To run this simulation perform the following steps:

- Replace `Query_OpenFresco1D.m` in `C:\SIMCOR\01_SIMCOR\@MDL_RF\private` with `Query_OpenFresco1D.m` from the `\00_Coordinator` directory.
- Replace `TCPSocket.mex32` in `C:\SIMCOR\01_SIMCOR\@MDL_RF\private` with `TCPSocket.mex32` from `User's Directory\OpenFresco\trunk\src\`



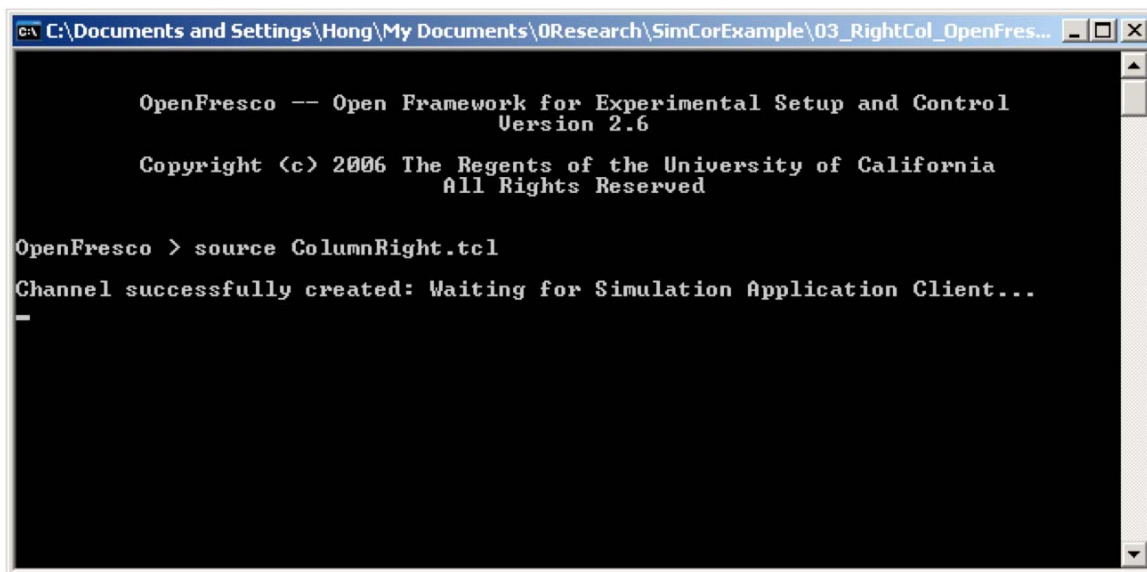
simApplicationClient\matlab. If OpenFresco was installed in the default location, the User's Directory is C:\Program Files.

- The previous two steps are necessary because of the updates made to OpenFresco since the release of UI-SimCor v2.6.
- Start the OpenFresco executable file (OpenFresco.exe) from the directory where the ColumnLeft.tcl resides.
- At the prompt, type **source ColumnLeft.tcl** and hit **enter**.
- Repeat the previous two steps using ColumnRight.tcl instead of ColumnLeft.tcl.
- There are now two OpenFresco executables running (Figure 4 and 5).



```
C:\Documents and Settings\Hong\My Documents\OResearch\SimCorExample\01_LeftCol_OpenFresc...  
  
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 columnleft.tcl  
Channel successfully created: Waiting for Simulation Application Client...  
-
```

Figure 4: OpenFresco Command Window for Element 1.



```
C:\Documents and Settings\Hong\My Documents\OResearch\SimCorExample\03_RightCol_OpenFresc...  
  
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 ColumnRight.tcl  
Channel successfully created: Waiting for Simulation Application Client...  
-
```

Figure 5: OpenFresco Command Window for Element 2.



- Start `NEESSAM_LabView1.exe`. This executable file is located in `c:\SIMCOR\02NEES-SAM`. Open `Module.cfg` file from the window shown in Figure 6. NEES-SAM window looks like Figure 7.

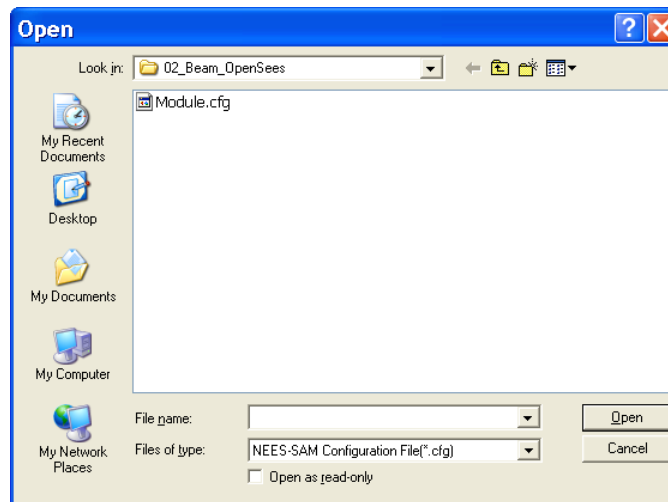


Figure 6: NEES-SAM Open Window for Element 3.

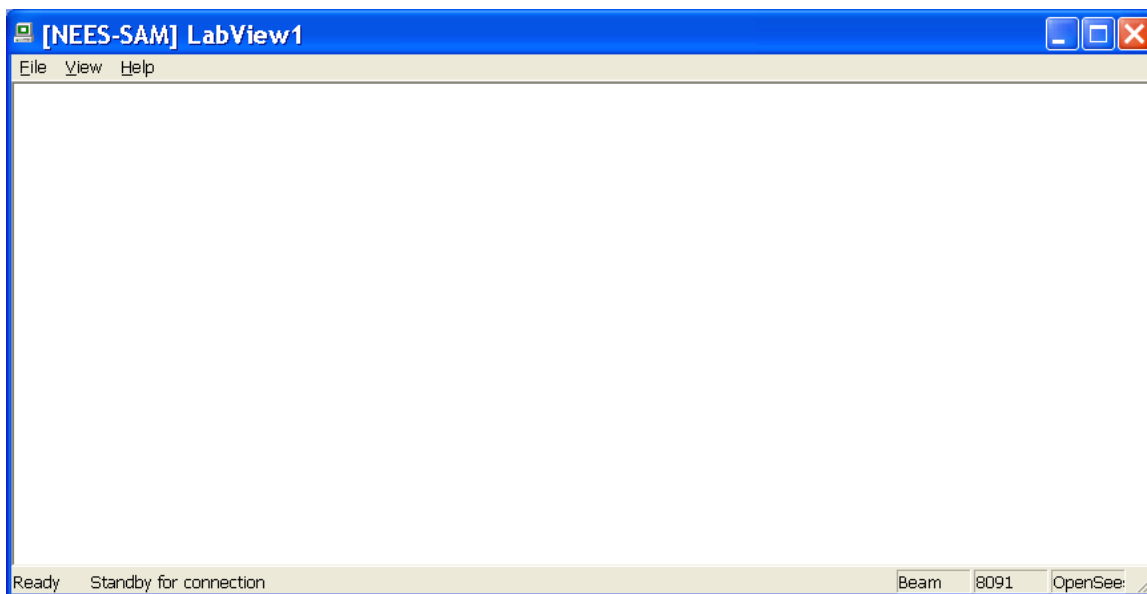


Figure 7: NEES-SAM Window.

- Start MATLAB[®]. Change the current directory to the folder where `SimConfig.m` and the ground motion file is located. Run UI-SIMCOR by inputting the following command `UI_SimCor` at the prompt and hitting **enter** (Figure 8). This launches UI-SimCor and its GUI (Figure 9) and three module monitoring windows (Figure 10-12). These monitors show the current status of the simulation. `HSF` option shows the results after scaling whereas `Remote site` shows the results before scaling.



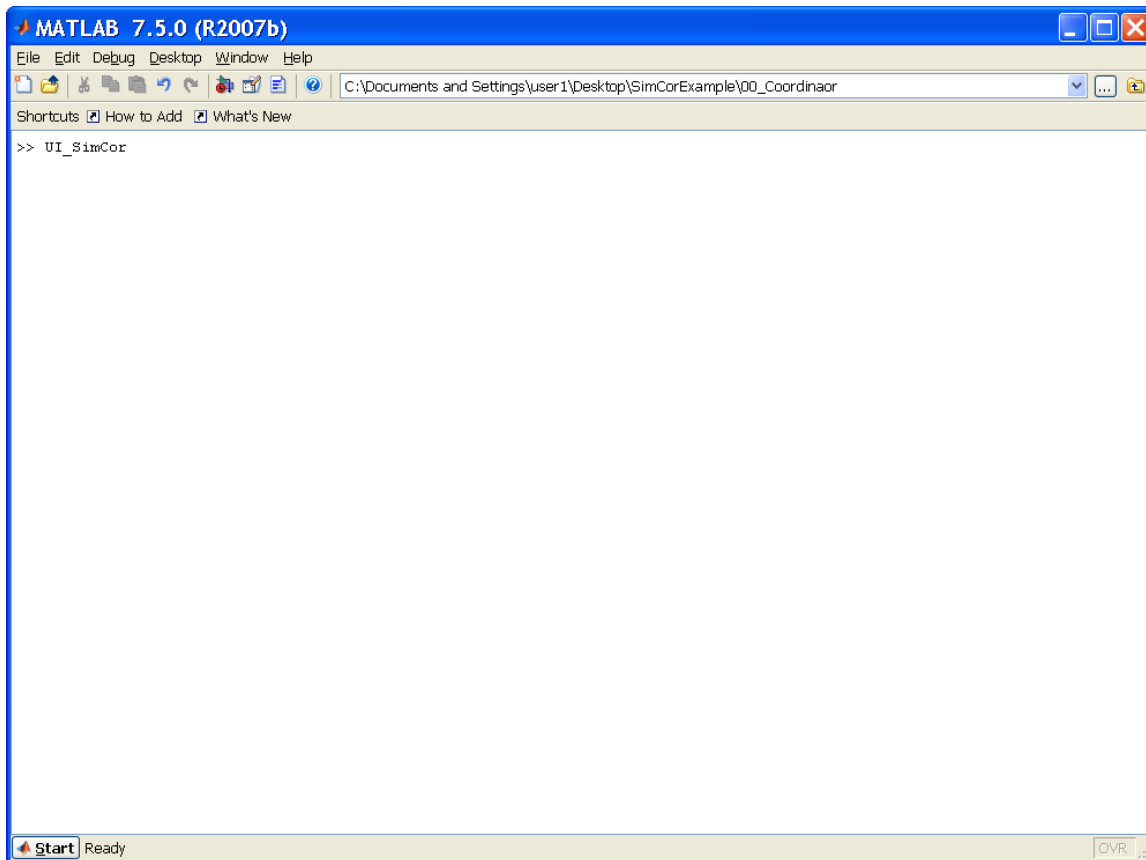


Figure 8: MATLAB® Command Window starting UI-SimCor.

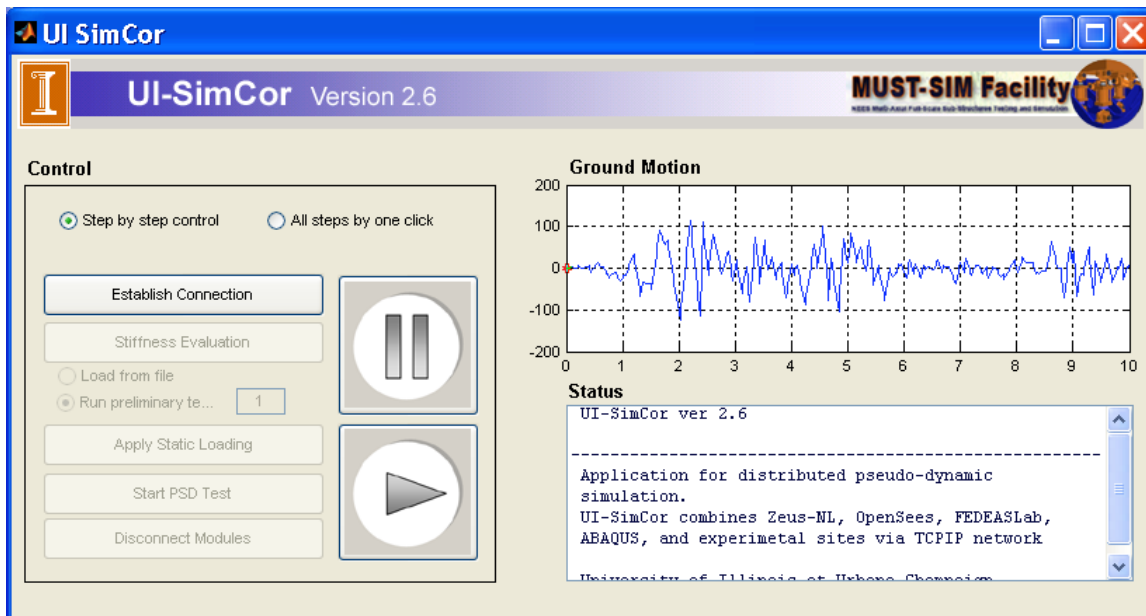


Figure 9: UI-SimCor GUI.



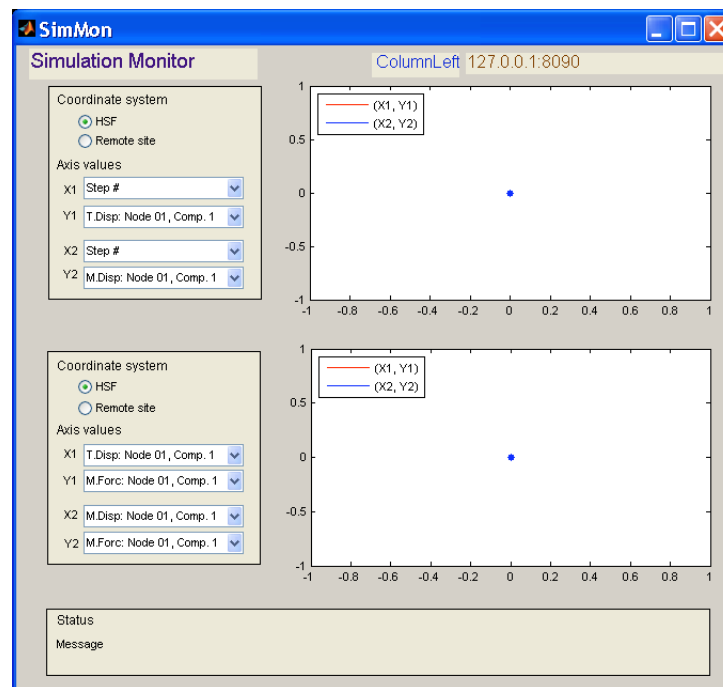


Figure 10: Simulation Monitor of Element 1.

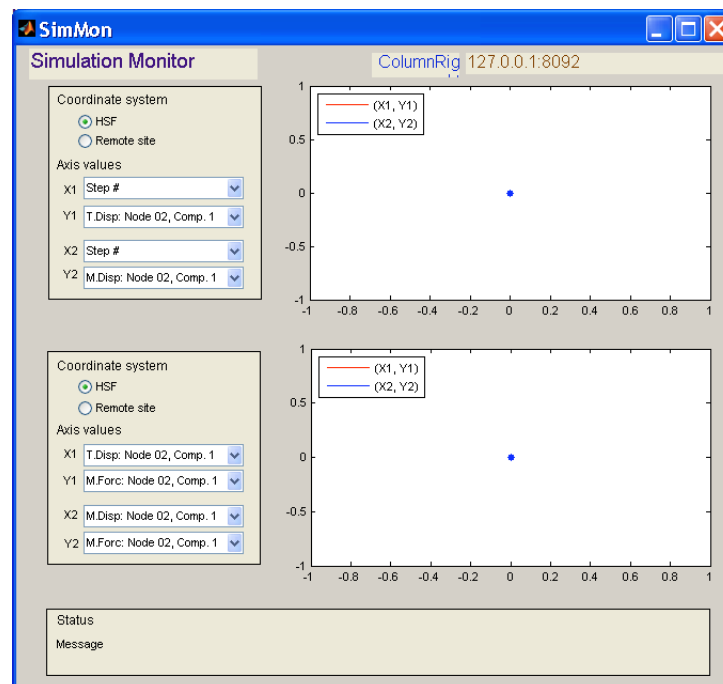


Figure 11: Simulation Monitor of Element 2.



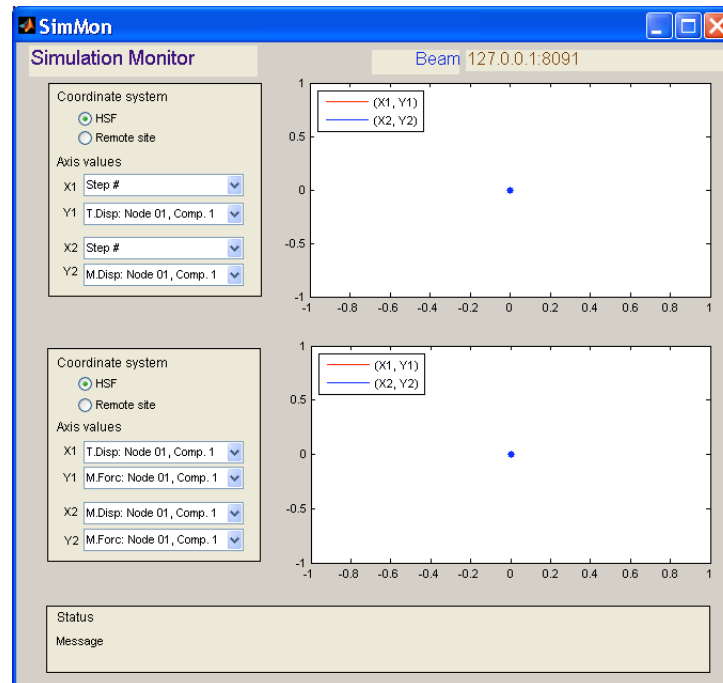


Figure 12: Simulation Monitor of Element 3.

- There are two options in the UI-SIMCOR GUI. The Step by step option runs each of the following actions by the click of the according buttons: Establish Connection, Stiffness Evaluation, Apply Static Loading, Start PSD Test, and Disconnect Modules. The All steps by one click option runs all steps automatically by clicking the start button. Select the Step by step option for this example.
- Click **Establish Connection** button on the GUI. Following windows result (Figure 13-15).

```

C:\Documents and Settings\user1\Desktop\SimCorExample\Op...
OpenFresco > source columnleft.tcl
Channel successfully created: Waiting for Simulation Application Client...
SimAppSiteServer with ExpSite 1 now running...

```

Figure 13: OpenFresco Command Window for Element 1.



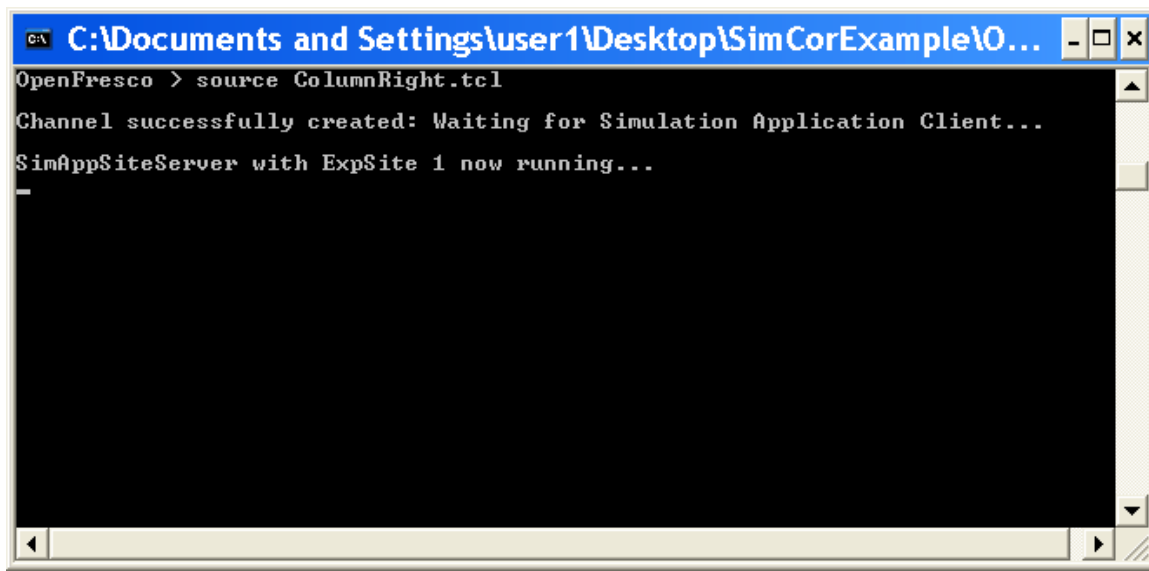


Figure 14: OpenFresco Command Window for Element 2.

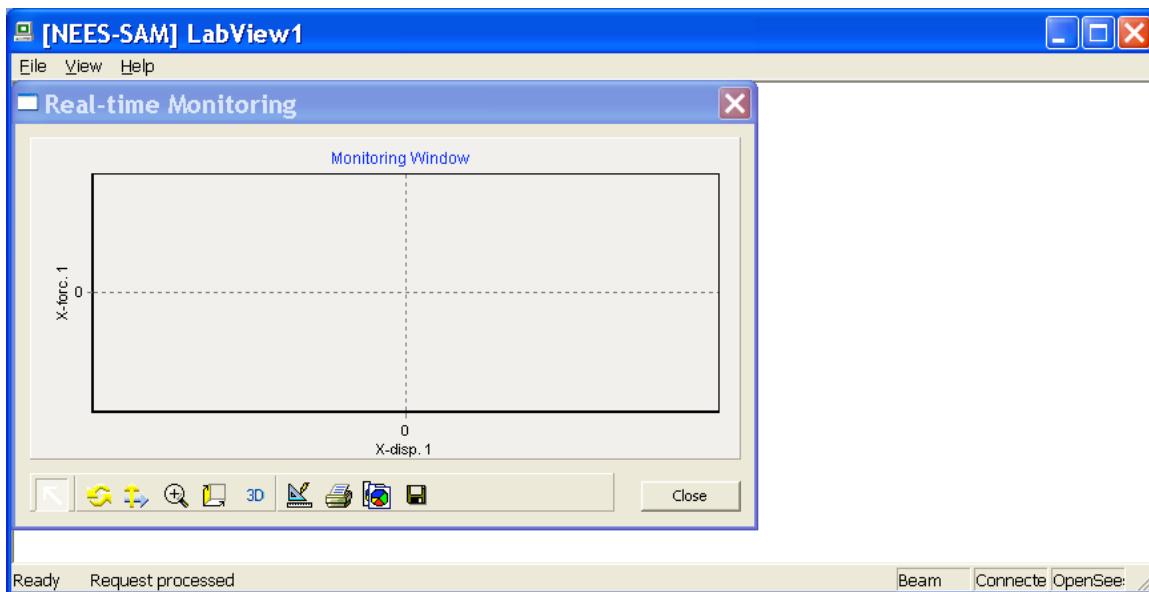


Figure 15: NEES-SAM Window after Connection is established.

- Click **Stiffness Evaluation** button and then **Start PSD Test** button on the GUI. This runs the simulation. While the simulation is in progression, the monitoring windows display information about the modules. After the simulation is complete, the follow figures result (Figures 16-20).



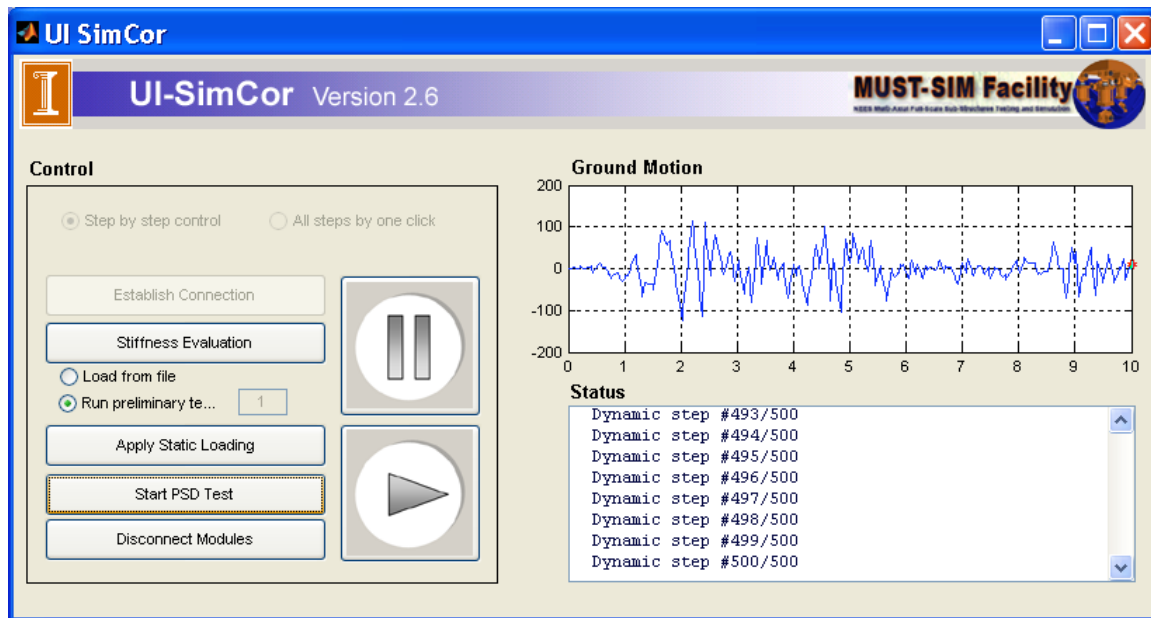


Figure 16: UI-SimCor GUI after Simulation.

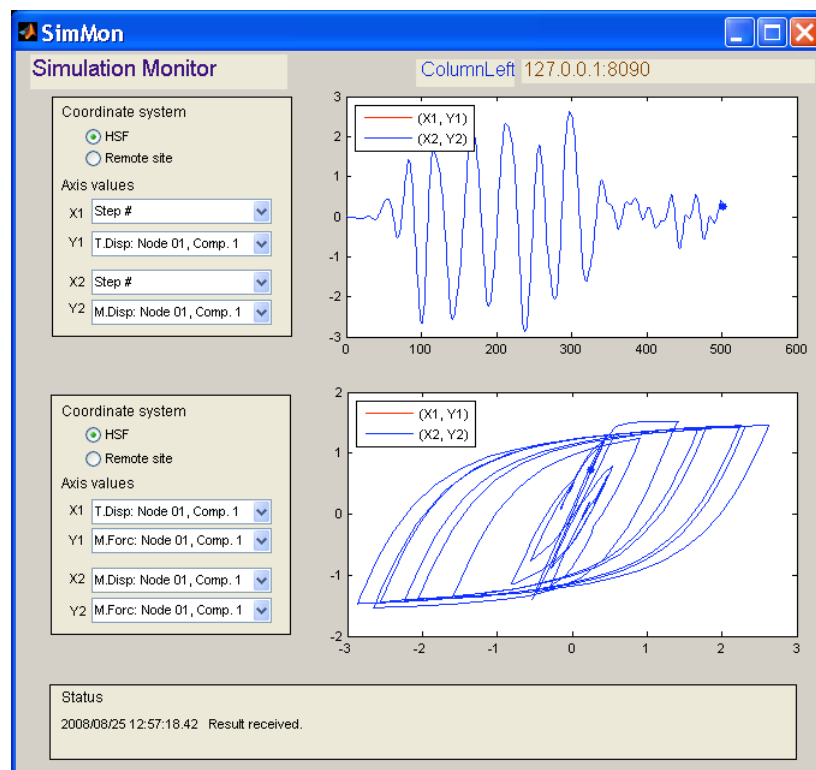


Figure 17: Simulation Monitor of Element 1 after Simulation.



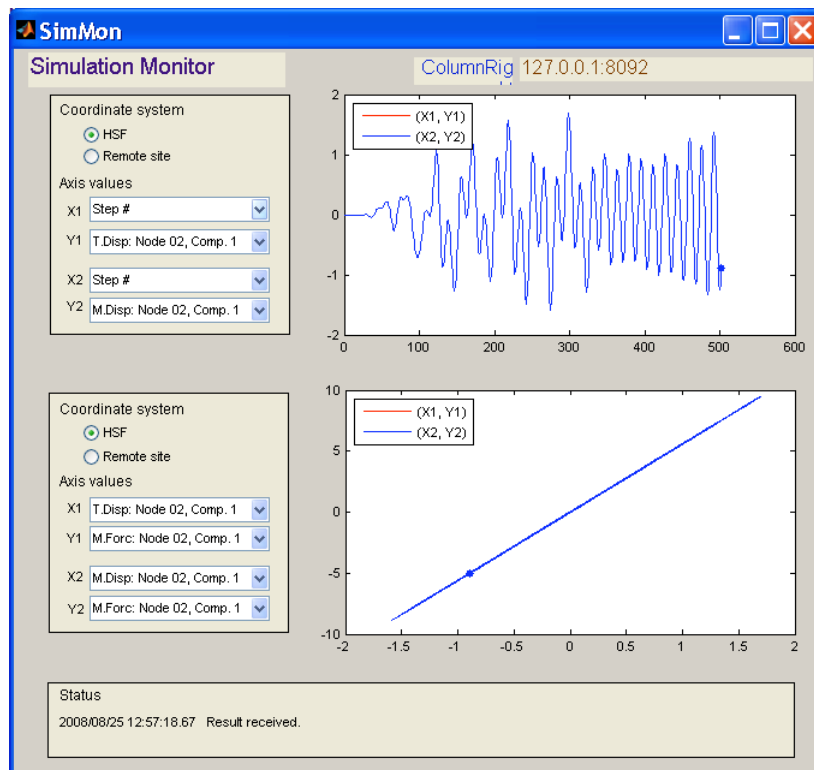


Figure 18: Simulation Monitor of Element 2 after Simulation.

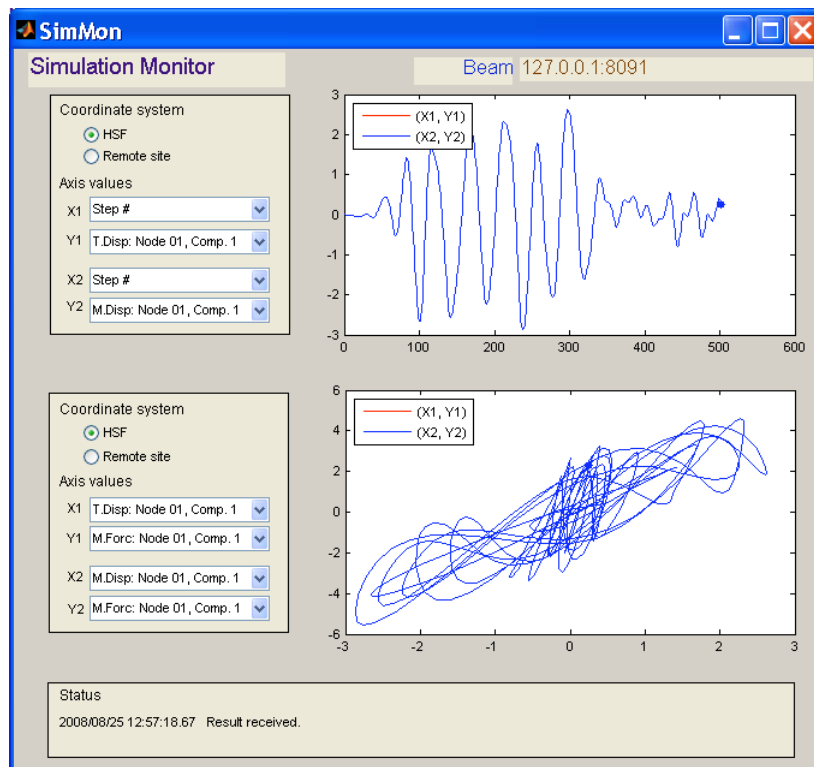


Figure 19: Simulation Monitor of Element 3 after Simulation.



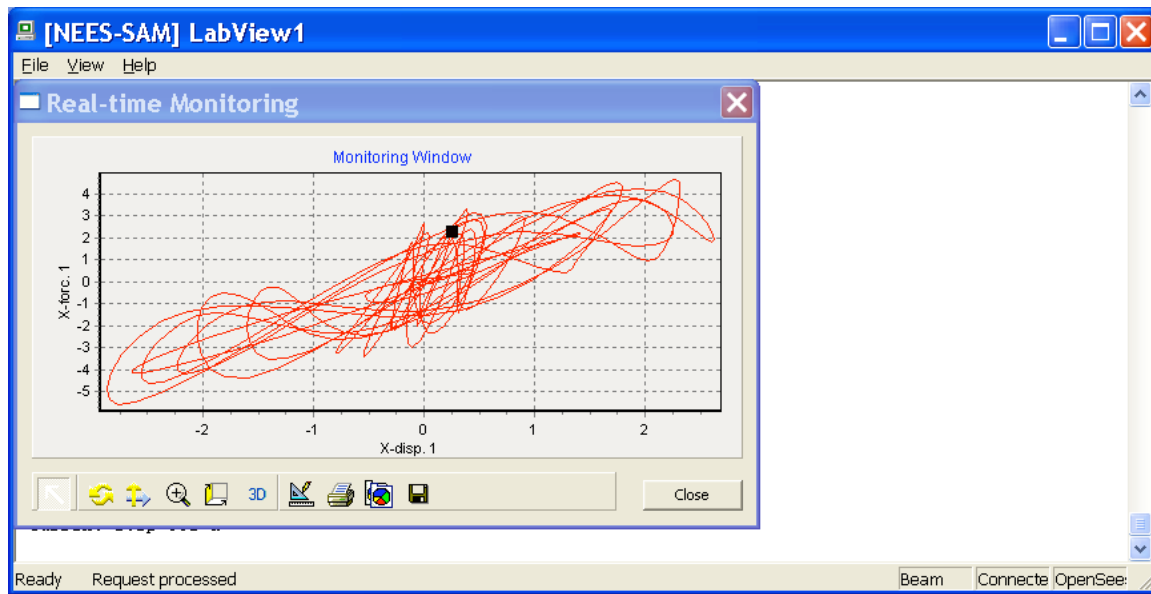


Figure 20: NEES-SAM Window after Simulation.

7 Results

After the simulation ends, output files are created in the coordinator directory (\00_Coordinator) and the OpenSees module directory (\02_Beam_OpenSees). The figures in this section are plotted from these output files.

The following output files are created in the coordinator directory:

- Global_K.txt
- MDL01_K.txt
- MDL01_recv.txt
- MDL02_K.txt
- MDL02_recv.txt
- MDL03_K.txt
- MDL03_recv.txt
- NetwkLog.txt
- NodeDisp.txt

The following output files are created in the coordinator directory:

- Cur_Displ.txt
- Cur_Forc.txt
- NetLog.txt



The response quantities for this example are plotted in Figures 21 and 22.

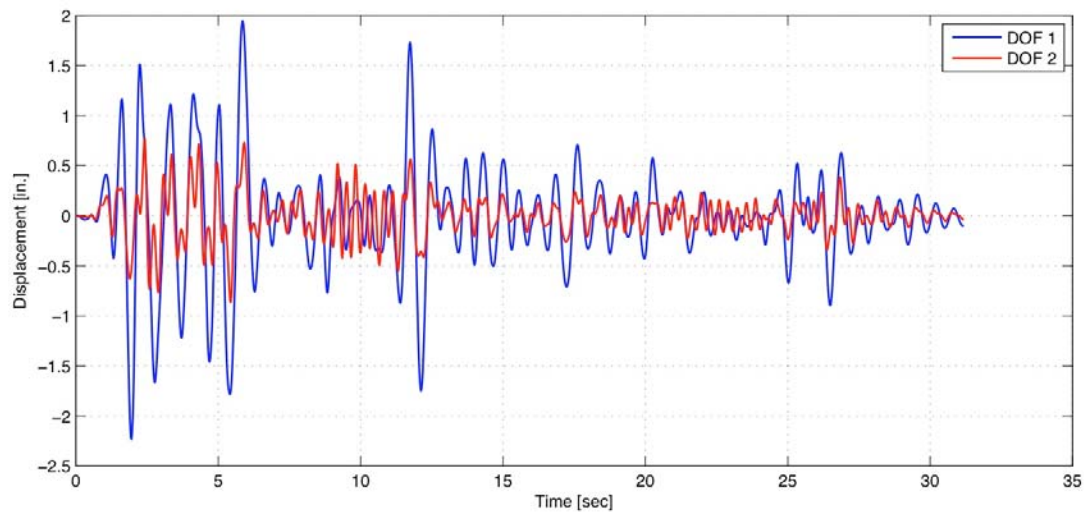


Figure 21: Displacements vs. Time for UI-SimCor Example.

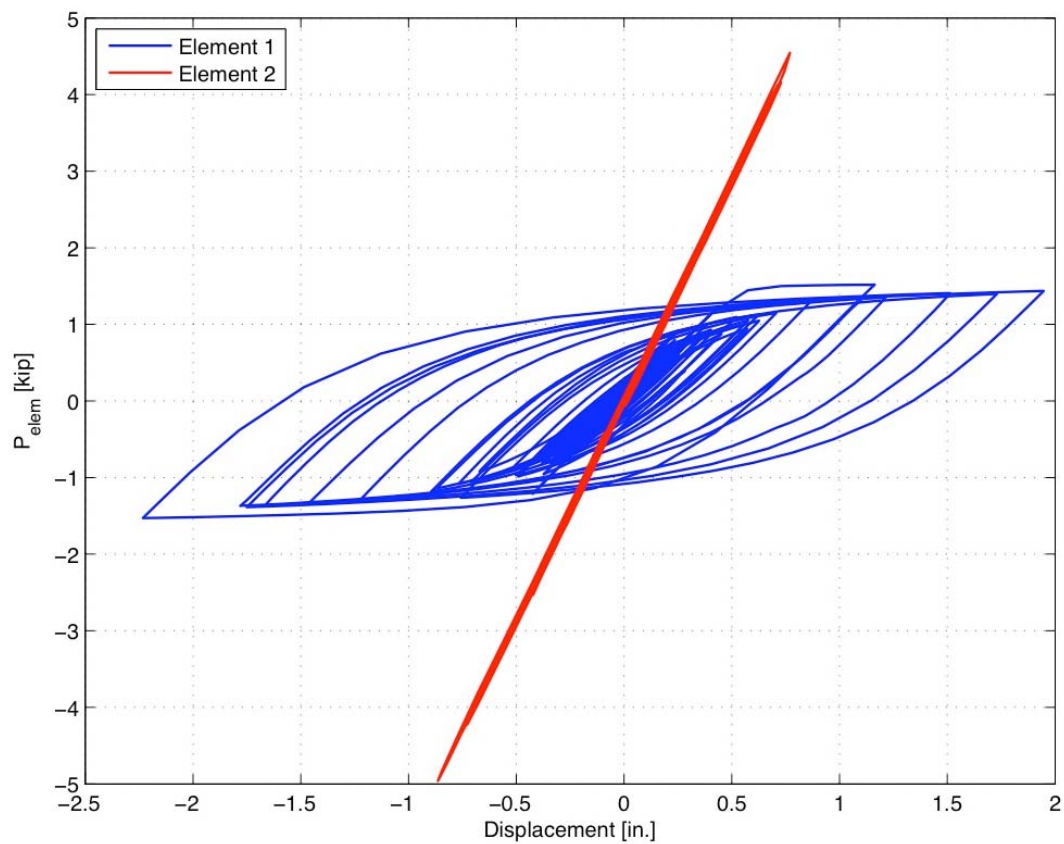


Figure 22: Element Hysteresis Loops for UI-SimCor Example.



8 References

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

Kwon, O.-S., Nakata, N., Park, K.-S., Elnashai, A., and Spencer, B. (2007). User Manual and Examples for UI-SIMCOR and NEES-SAM, University Of Illinois, Urbana-Champaign, IL, United States.

