NEESit
Enabling the Network for
Earthquake Engineering Simulation

TR-2009-[ID]

OpenFresco Framework for Hybrid Simulation: Simulation Finite Element Adapter 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: Example Using Simulation Finite Element Adapter Experimental Control with OpenSees

This example explains how to use the Simulation Finite Element Adapter experimental control in OpenFresco to run a simulation between two OpenSees processes. The adapter element (Schellenberg et al., 2008) is used in one of the OpenSees processes in conjunction with the Simulation Finite Element Adapter experimental control in the other process. This example uses a simple Cantilever Column model with the Implicit Newmark time integration scheme. This document includes explanation of the OpenFresco Tcl commands used in the input files, step-by-step instructions on how to run the example, and results.

2 Required Files

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

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

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:

- `Cantilever_Master.tcl`
- `Cantilever_Slave.tcl`
- `elcentro.txt`

The latest OpenSees executable and Tcl/Tk 8.5.x are required to run this example. If not done so already, OpenSees and Tcl/Tk 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 cantilever column is modeled in OpenSees using the Tcl file, `Cantilever_Master.tcl`. Figure 1 shows the model. The two end-nodes of the cantilever column are nodes 1 and 2. Lumped masses in global X- and Y-direction are defined for node 2. The base support of the column is fixed. The following Tcl script from `Cantilever_Master.tcl` defines the geometry of the model:

```
# Define geometry for model
# -------------------------
# node $tag $xCrd $yCrd $mass
node  1     0.0    0.0
node  2     0.0   54.0   -mass 0.08 0.08 0.0

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

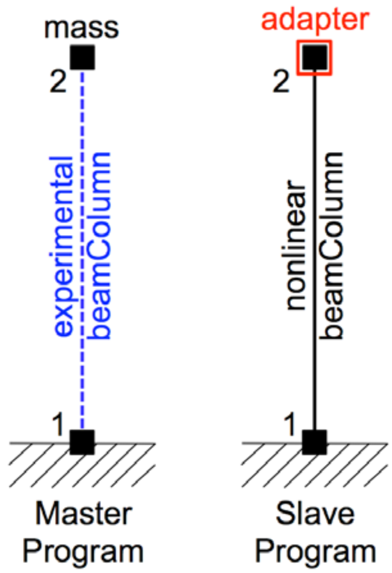

Figure 1: OpenSees Cantilever Column Model.

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).

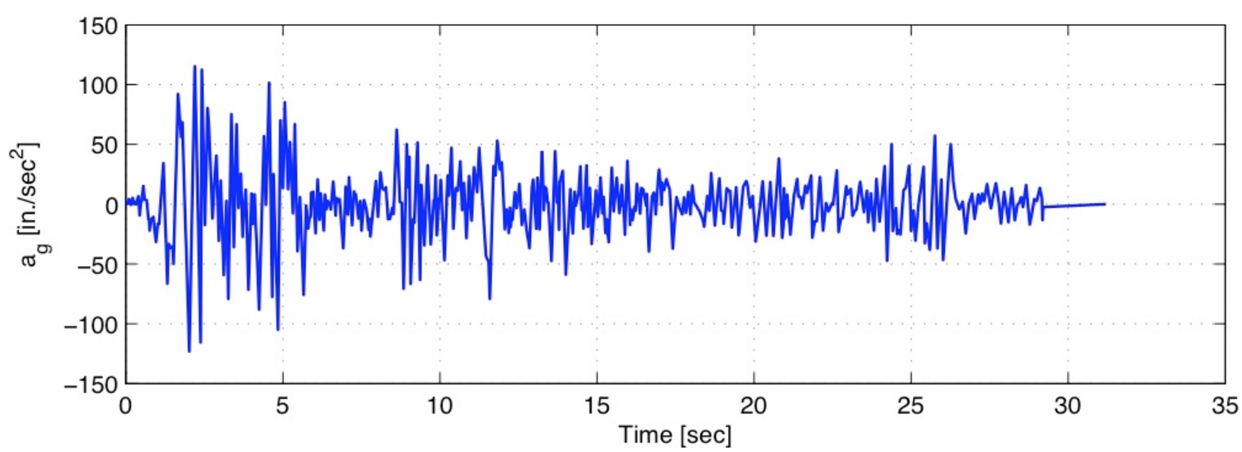

Figure 2: 1940 El Centro Ground Motion.

5 OpenFresco Tcl Commands

This section contains explanations of the OpenFresco Tcl commands from `Cantilever_Master.tcl` and some OpenSees Tcl commands from `Cantilever_Slave.tcl`. More detailed information about the OpenFresco Tcl commands are found in the OpenFresco Command Language Manual. The OpenSees website (http://opensees.berkeley.edu). has more information about the OpenSees Tcl commands used in the two Tcl files.

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

The above script is excerpted from `Cantilever_Master.tcl`. The `loadPackage OpenFresco` command is necessary for the examples to execute properly.

5.1 Experimental Control

The experimental control is set to `SimFEAdapter`. The `SimFEAdapter` command enables the user to couple two finite element programs via the adapter element. The adapter element is an element to be programmed by the user. The adapter element for this example is defined in `Cantilever_Slave.tcl` while the `SimFEAdapter` experimental control is defined in `Cantilever_Master.tcl`.

The following excerpt is from `Cantilever_Master.tcl`. It shows what parameters are being used for this example.

```
# Define experimental control
# ---------------------------
# expControl SimFEAdapter $tag ipAddr $ipPort
expControl SimFEAdapter 1 "127.0.0.1" 44000
```

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

- `$tag` is the unique control tag.
- `ipAddr` is the IP Address of the slave process, which is “127.0.0.1”.
- `$ipPort` is the IP Port of the adapter element running on the slave process, which is 44000.

The `SimFEAdapter` experimental control is used in conjunction with the adapter element in OpenSees. The Tcl script below is from `Cantilever_Slave.tcl` and shows how the adapter element is defined in OpenSees.

```
# element adapter eleTag -node Ndi Ndj ... -dof dofNdi -dof dofNdj ... -stif Kij ipPort
<-mass Mij>
element adapter 2 -node 2 -dof 1 -stif 1E5 44000
```

The `element` command parameters for `adapter` are:

- `$eleTag` is the element tag.
- `$Ndi, Ndj, …` are the node numbers that define the element. The example uses node 2.
- `$dofNdi` are the global degrees of freedom at each node, which is 1 (global X) at node 2.
- `$Kij` are the stiffness matrix components (row-wise) of the element, which is set to 1E5.
- `$ipPort` is the IP port number of the master process. Hence, it must be equal to the IP port number used in the `SimFEAdapter` experimental control definition, which is 44000.

5.2 Experimental Setup

The `OneActuator` setup is used for the experimental setup in Figure 3. The script below is located in `Cantilever_Master.tcl`. The Tcl command for the experimental setup is:

```
# Define experimental setup
# -------------------------
# expSetup OneActuator $tag <-control $ctrlTag> $dir -sizeTrialOut $sizeTrial $sizeOut
<-trialDispFact $f> ...
expSetup OneActuator 1 -control 1 2 -sizeTrialOut 3 3 -ctrlDispFact -1.0 -daqDispFact -
1.0 -daqForceFact -1.0
```

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 SimFEAdpater experimental control.
- `$dir` is the direction of the imposed displacement in the element basic reference coordinate system, which is set to 2.
- `$sizeTrial` and `$sizeOut` are the sizes of the element trial and output data vectors, which are both set to 3.
- `$f` are control displacement factor, daq displacement factor, and daq force factor, respectively. These optional fields are used to factor the imposed and the measured data. Here they are all set to -1.0 to reverse directions.

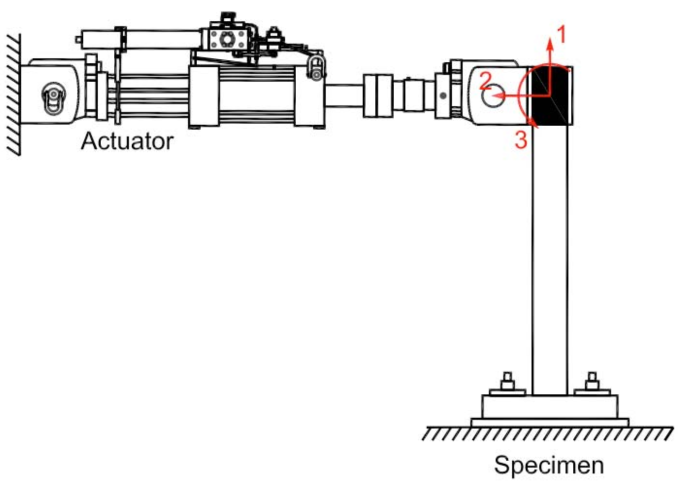

Figure 3: OneActuator Experimental Setup.

5.3 Experimental Element

The experimental element is set to be a `beamColumn` experimental element (Figure 4), which is defined by two nodes. The following script is in `Cantilever_Master.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`.
- `$siteTag` is the tag of a previously defined site object. In this example, it is set to `LocalSite`.
- `$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.

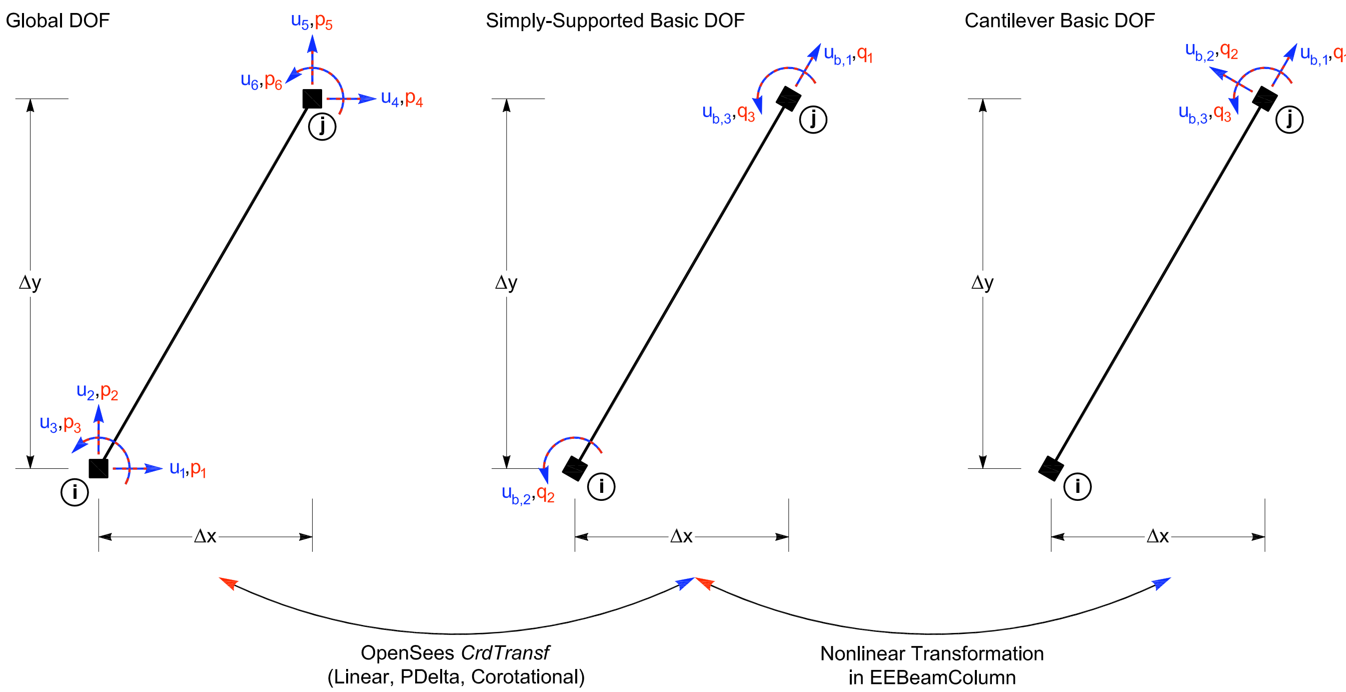

Figure 4: beamColumn Experimental Element.

6 Running Local Simulation with Two OpenSees Processes

When using OpenSees as the computational driver, the local test may be run with the architecture shown in Figure 5.

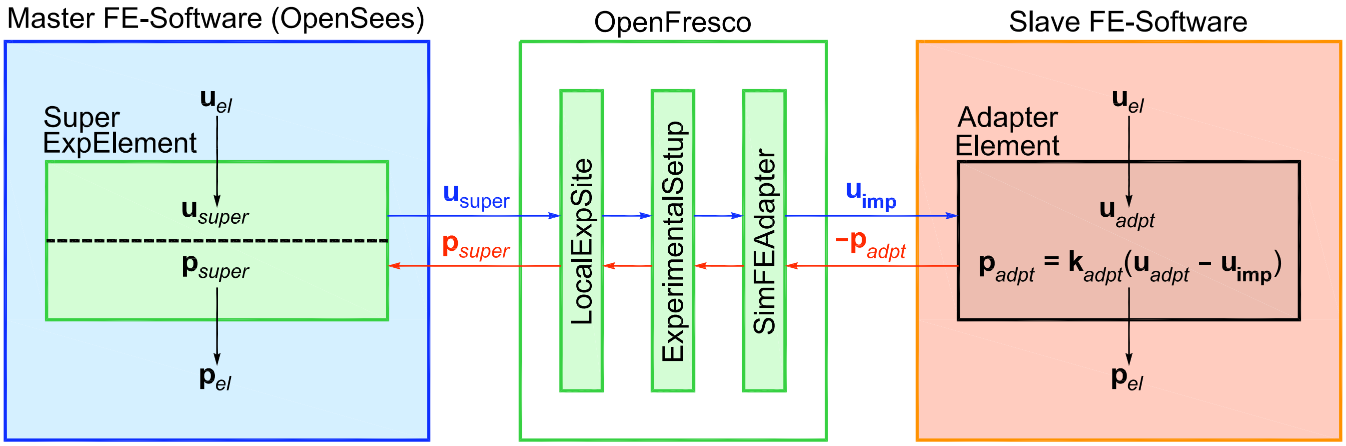

Figure 5: Local Hybrid Simulation using OpenSees.

The script excerpt below, from `Cantilever_Master.tcl`, 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.

To run this simulation perform the following steps:

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

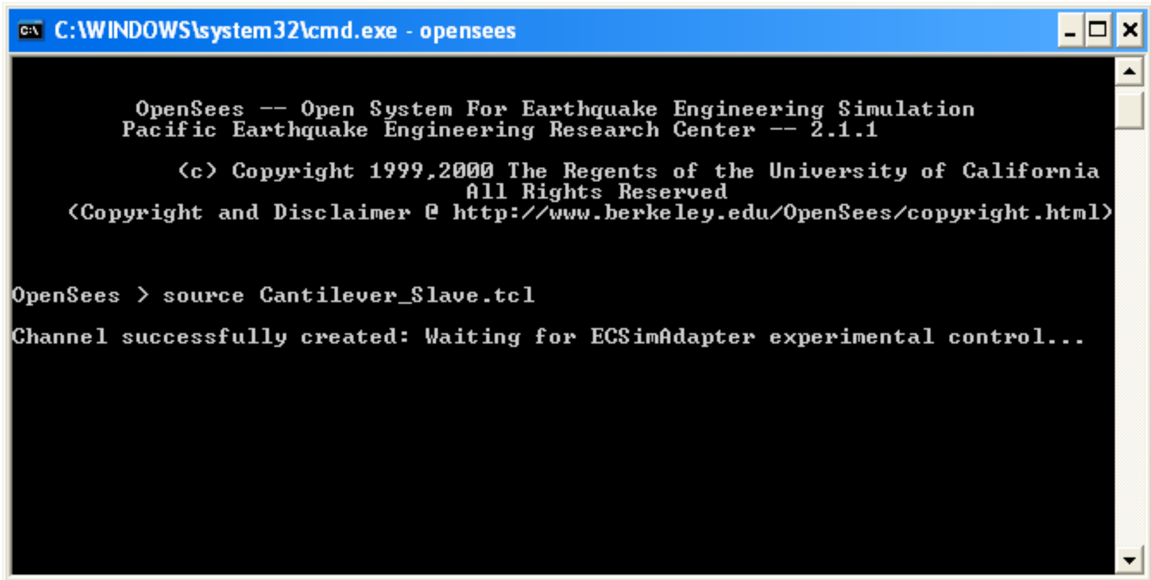

Figure 6: OpenSees Slave Command Window for Simulation Finite Element Adapter Example.

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

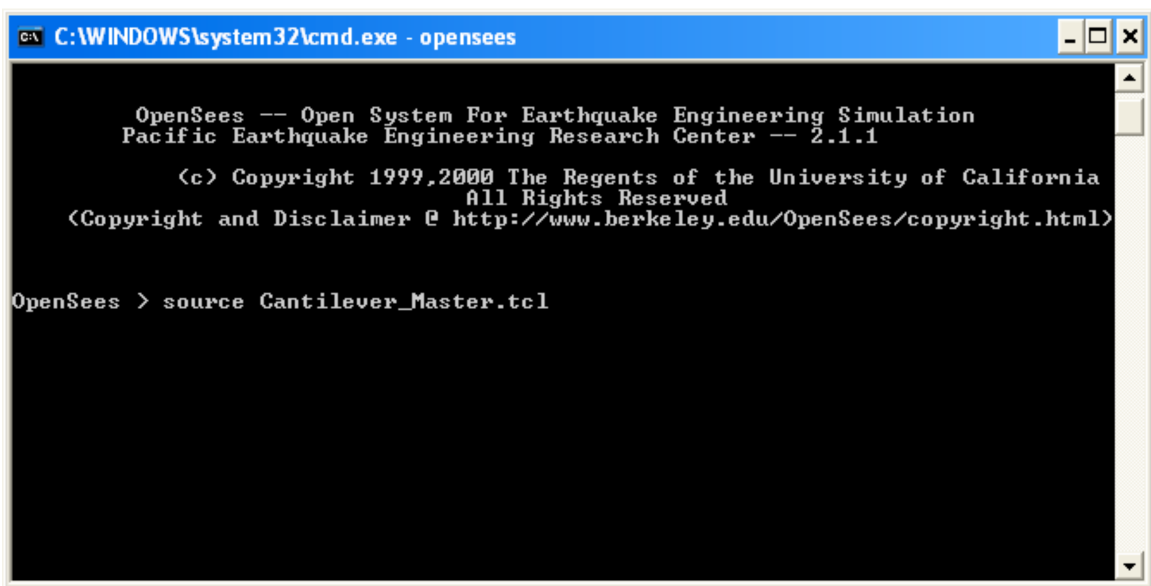

Figure 7: OpenSees Master Command Window for Simulation Finite Element Adapter Example.

- After the simulation has completed, the OpenSees Slave and Master processes command windows should look like Figure 8 and 9 respectively.

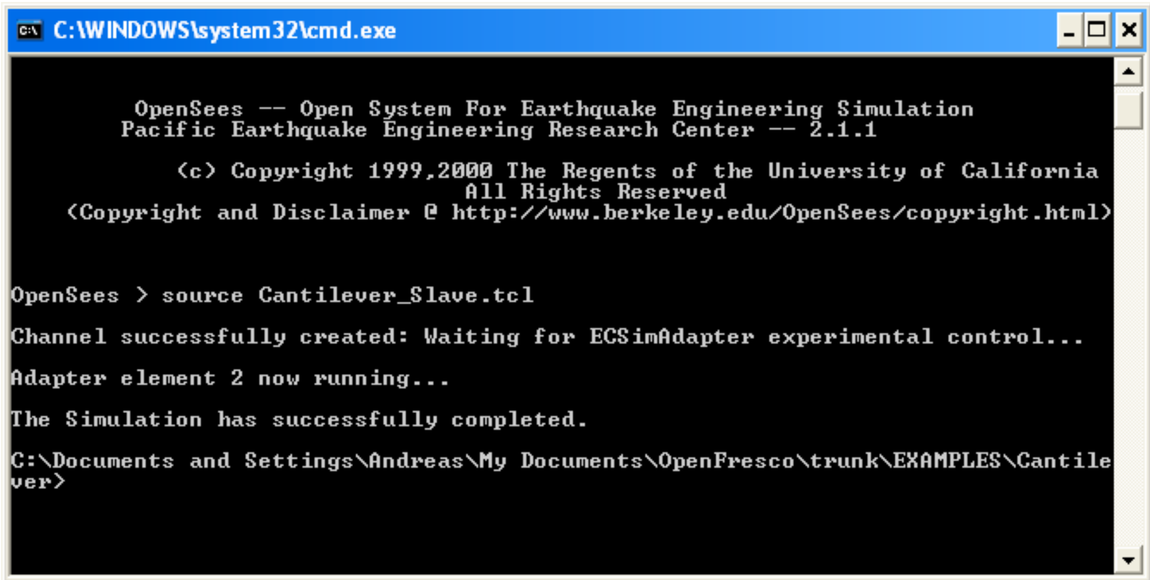

Figure 8: OpenSees Slave Command Window after Simulation.

Figure 9: OpenSees Master Command Window after Simulation.

7 Results

The OpenSees command window should display the following results:

```
Eigenvalues at start of transient:
lambda            omega           period
3.500000e+001     5.9160797831    1.06205215912
1.516250e+004     123.13610356    0.0510263450403
```

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

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

The following are the output files:

- `Master_Elmt_ctrlDsp.out`
- `Master_Elmt_daqDsp.out`
- `Master_Elmt_Frc.out`
- `Master_Node_Acc.out`
- `Master_Node_Dsp.out`
- `Master_Node_Vel.out`
- `Slave_Elmt_ctrlDsp.out`
- `Slave_Elmt_daqDsp.out`
- `Slave_Elmt_Frc.out`
- `Slave_Node_Acc.out`
- `Slave_Node_Dsp.out`
- `Slave_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](opensees.berkeley.edu) (opensees.berkeley.edu). The following excerpt is from `Cantilever_Master.tcl`.

```
# ------------------------------
# Start of recorder generation
# ------------------------------
# create the recorder objects
recorder Node -file Master_Node_Dsp.out -time -node 2 -dof 1 2 3 disp
recorder Node -file Master_Node_Vel.out -time -node 2 -dof 1 2 3 vel
recorder Node -file Master_Node_Acc.out -time -node 2 -dof 1 2 3 accel

recorder Element -file Master_Elmt_Frc.out     -time -ele 1 forces
recorder Element -file Master_Elmt_ctrlDsp.out -time -ele 1 ctrlDisp
recorder Element -file Master_Elmt_daqDsp.out  -time -ele 1 daqDisp
# --------------------------------
# End of recorder generation
# --------------------------------
```

The following excerpt is from `Cantilever_Slave.tcl`.

```
# ------------------------------
# Start of recorder generation
# ------------------------------
# create the recorder objects
recorder Node -file Slave_Node_Dsp.out -time -node 2 -dof 1 2 3 disp
recorder Node -file Slave_Node_Vel.out -time -node 2 -dof 1 2 3 vel
recorder Node -file Slave_Node_Acc.out -time -node 2 -dof 1 2 3 accel

recorder Element -file Slave_Elmt_Frc.out     -time -ele 1 2 forces
recorder Element -file Slave_Elmt_ctrlDsp.out -time -ele   2 ctrlDisp
recorder Element -file Slave_Elmt_daqDsp.out  -time -ele   2 daqDisp
# --------------------------------
# End of recorder generation
# --------------------------------
```

The response quantities for the Simulation Finite Element Adapter example are plotted in Figures 10 to 14.

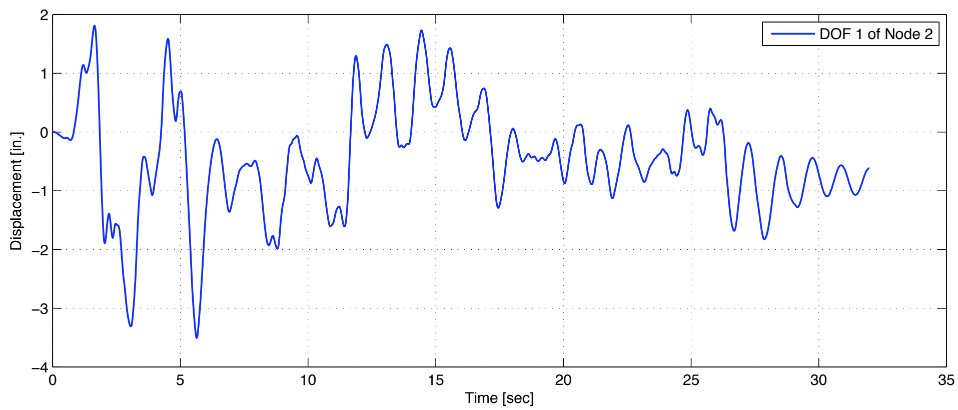

Figure 10: Displacement of DOF 1 of Node 2 vs. Time for Master Process.

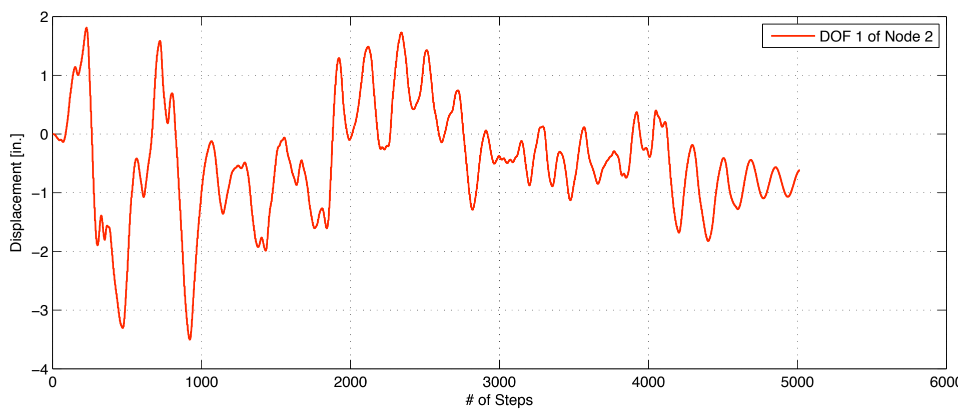

Figure 11: Displacement of DOF 1 of Node 2 vs. Time for Slave Process.

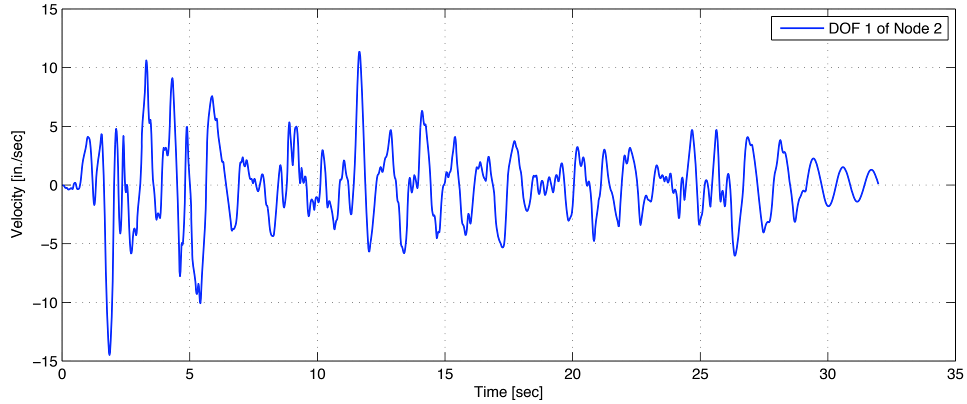

Figure 12: Velocity of DOF 1 of Node 2 vs. Time for Master Process.

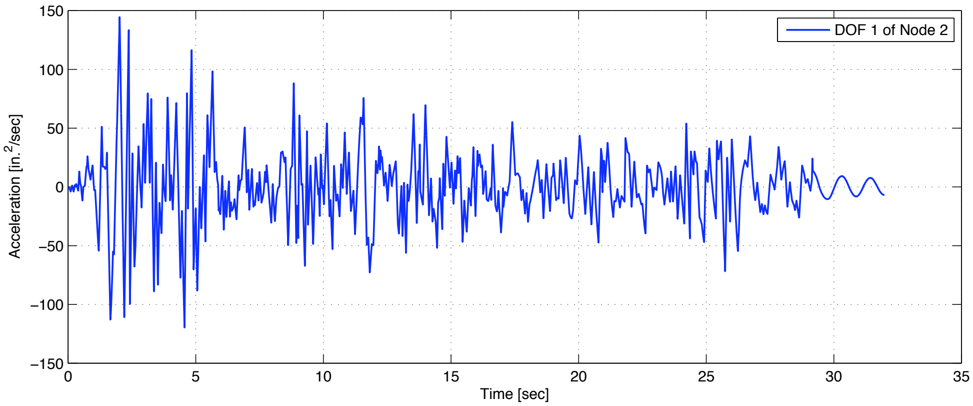

Figure 13: Acceleration of DOF 1 of Node 2 vs. Time for Master Process.

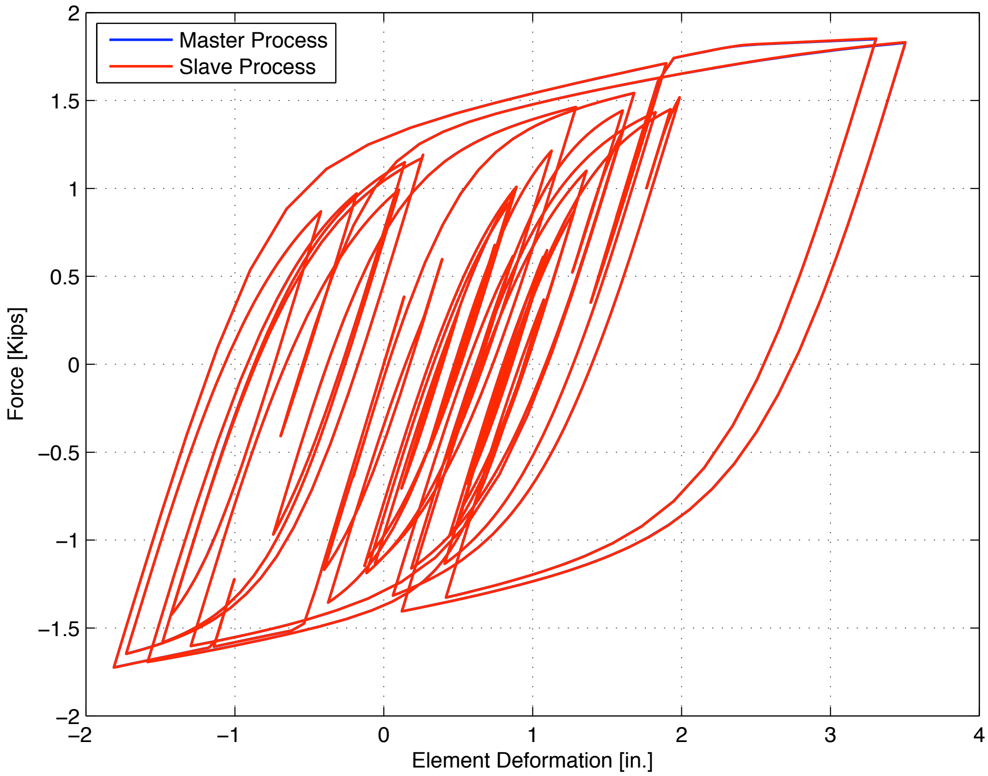

Figure 14: Element Hysteresis Loops for Simulation Finite Element Adapter Example.

8 References

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

Schellenberg, A., Huang, Y. and Mahin, S. A., "Structural FE-Software Coupling through the Experimental Software Framework, OpenFresco", Proceedings 14th World Conference on Earthquake Engineering, Beijing, China, 2008