Enabling the Network for Earthquake Engineering Simulation

TR-2009-[ID]

OpenFresco Framework for Hybrid Simulation: xPC-Target Experimental Control Example

Andreas Schellenberg, Catherine Whyte, Yoshikazu Takahashi, Gregory L. Fenves, and Stephen A. Mahin

Department of Civil and Environmental Engineering,
University of California, Berkeley

Last Modified: 2009-08-15 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

1 Introduction: Local Hybrid Simulation Example Using xPC-Target Experimental Control 3
2 Required Software and Files 4
2.1 xPC-Host Computer 4
2.2 xPC-Target Computer 4
2.3 MTS 493 Controller and Host Computers 5
3 Setting up the Simulink Predictor-Corrector 5
4 Test with Simulated Specimen 9
5 Hybrid Simulation Using the µNEES or Similar Setup 9
6 Results 13
7 References 17

Figures

Figure 1: Three Loop Architecture. 3
Figure 2: Hybrid Controller Parameters in initializeSimulation.m. 5
Figure 3: Signal Counts and SCRAMNet Setup in initializeSimulation.m. 6
Figure 4: HybridControllerD3D3_1Act Simulink Predictor-Corrector Model. 7
Figure 5: Configuration Parameters -> Simulation Target -> Custom Code. 8
Figure 6: Configuration Parameters -> Real-Time Workshop -> Custom Code. 8
Figure 7: µNEES Experimental Setup with Two Independent Specimens. 9
Figure 8: STS Main Window. 10
Figure 9: STS Setpoints and Limit Detectors Windows. 10
Figure 10: Excerpt from PortalFrame_Local.tcl. 11
Figure 2: Elastic Ground Motion, No Gravity, Time Histories. 13
Figure 3: Elastic Ground Motion, No Gravity, Hysteresis Loops. 13
Figure 4: Inelastic Ground Motion, No Gravity, Time Histories. 14
Figure 5: Inelastic Ground Motion, No Gravity, Hysteresis Loops. 14
Figure 6: Elastic Ground Motion, With Gravity, Time Histories. 15
Figure 7: Elastic Ground Motion, With Gravity, Hysteresis Loops. 15
Figure 8: Inelastic Ground Motion, With Gravity, Time Histories. 16
Figure 9: Inelastic Ground Motion, With Gravity, Hysteresis Loops. 16

Tables

Table 1: Test Matrix. 4

1 Introduction: Local Hybrid Simulation Example Using xPC-Target Experimental Control

This example describes how to operate the hybrid simulation laboratory at UC Berkeley. The xPC-Target Control is a three-loop architecture as shown in the figure below. The innermost Servo-Control Loop contains the MTS STS controller, which was customized for the nees@berkeley laboratory by MTS. It sends command displacements to the specimen and reads back measured displacements and forces. The middle Predictor-Corrector Loop contains the xPC-Target real-time digital signal processor (DSP), sending commands to the STS controller over the shared memory SCRAMNet. The xPC-Target runs Predictor-Corrector Simulink models in real-time. Finally, the outer Integrator Loop contains the xPC-Host PC, which runs OpenSees and OpenFresco and communicates with the xPC-Target over a TCP/IP connection.

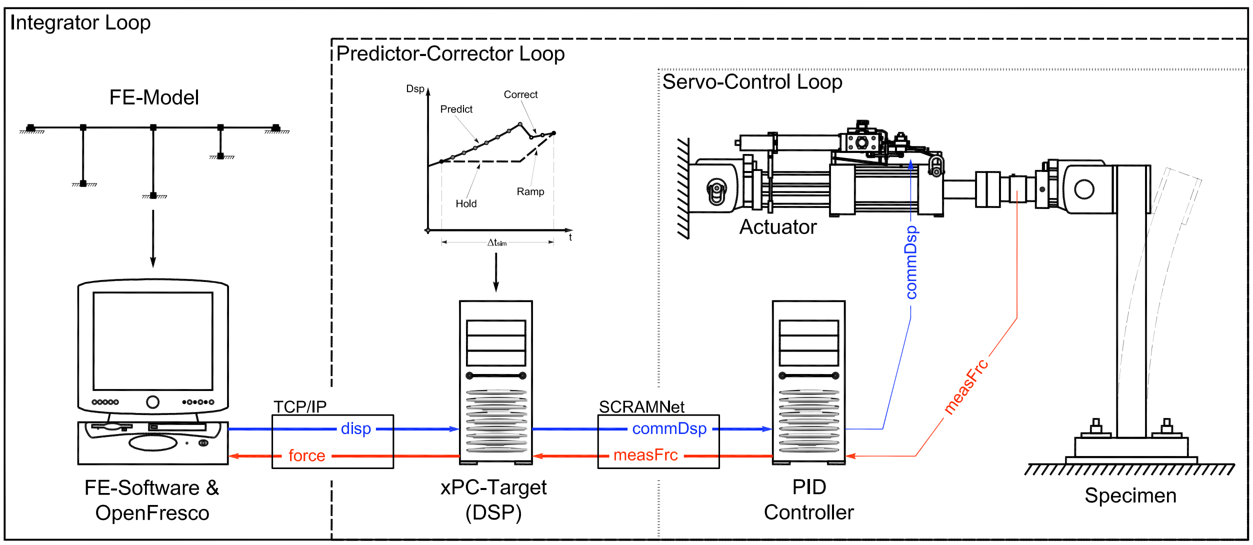

Figure 1: Three Loop Architecture.

The xPC-Host's OpenSees model converges to the next command displacement at large and somewhat variable time intervals. On the other hand, the STS controller is required to send commands to the specimen at a fast rate of 1024 Hz. In order to bridge these two time scales, the xPC-Target's Predictor-Corrector model predicts target displacements for the STS controller until the next displacement arrives from OpenSees. Then the Predictor-Corrector model corrects to the new target displacement.

This example uses the Portal Frame model located in the examples directory (`User's Directory \OpenFresco\trunk\EXAMPLES\PortalFrame`). The computational driver is OpenSees (http://opensees.berkeley.edu). The portal frame model can be modified for a distributed test, but only the local configuration is described here. Response results for four test cases are provided. These are summarized in the following matrix.

Table 1: Test Matrix.

	Without Gravity	With Gravity
Elastic level ground motion (0.15 X ground motion)	X	X
Inelastic level ground motion (1.3 X ground motion)	X	X

2 Required Software and Files

2.1 xPC-Host Computer

Microsoft Visual Studio 2008
Tcl/Tk 8.5.x (http://opensees.berkeley.edu/OpenSees/user/download.php)
OpenSees 2.1.1 (http://opensees.berkeley.edu/OpenSees/developer/download.php)
OpenFresco 2.6 (http://neesforge.nees.org/frs/?group_id=36)
Matlab/Simulink R2009a
Matlab Toolboxes: Real-Time Workshop, Stateflow, Stateflow Coder, xPC Target

First, install Microsoft Visual Studio 2008. Then Tcl/Tk can be downloaded from the Active State Tcl or the OpenSees website. Be careful to install it in the location specified on the OpenSees website (in the Program Files folder). Next, install the OpenSees source code from the link specified above. The website provides instructions for compiling. The OpenFresco Installer can be downloaded from the link above. Finally, you must have Matlab and the toolboxes listed above.

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

The `PortalFrame_Local.tcl` and ground motion file `SACNF01.txt` are located in:

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

The `initializeSimulation.m` and the Simulink model `HybridControllerD3D3_1Act.mdl` are located in:

```
User's Directory\OpenFresco\trunk\SRC\experimentalControl\Simulink\
RTActualTestModels\cmAPI-xPCTarget-STS
```

2.2 xPC-Target Computer

To setup the xPC-Target machine, follow the Mathworks documentation for the xPC Target product line. The xPC Target software does not require Microsoft DOS, Microsoft Windows®, Linux®, or any another operating system on the target PC. Instead, you boot the target PC with a boot disk that includes the xPC Target kernel. Make sure that the versions of the boot disk and the dynamic link library (xpcapi.dll), used with OpenFresco, are identical (currently Version 4.1).

2.3 MTS 493 Controller and Host Computers

The MTS 493 is a real-time digital controller that provides closed loop PID, differential Feedforward and Delta-P control capabilities. The accompanying STS (Structural Test System) software, which runs on a host PC, is a graphical user interface that allows the operator to access the controller for calibration, configuration, test setup and data display.

3 Setting up the Simulink Predictor-Corrector

On the xPC Host computer, start Matlab and open the `initializeSimulation.m` file, located in the folder specified above. Set the integration time step dtInt = 0.01 s, corresponding to the ground motion time step. Also set the simulation time step dtSim = 0.01 s (the simulation time step size can be adjusted to run a hybrid simulation at any desired testing rate). Save and close the file.

```
%%%%%%%%%% HYBRID CONTROLLER PARAMETERS %%%%%%%%%%

% set time steps
HybridCtrlParameters.dtInt = 0.01;          % integration time step (sec)
HybridCtrlParameters.dtSim = 0.01;          % simulation time step (sec)
HybridCtrlParameters.dtCon = 1/1024;        % controller time step (sec)
HybridCtrlParameters.delay = 0.0;           % delay due to undershoot (sec)

% calculate max number of substeps
HybridCtrlParameters.N = round(HybridCtrlParameters.dtSim/HybridCtrlParameters.dtCon);
% update simulation time step
HybridCtrlParameters.dtSim = HybridCtrlParameters.N*HybridCtrlParameters.dtCon;

% calculate number of delay steps
if (HybridCtrlParameters.delay == 0.0)
   HybridCtrlParameters.iDelay = HybridCtrlParameters.N;
else
   HybridCtrlParameters.iDelay = round(HybridCtrlParameters.delay./HybridCtrlParameters.dtCon);
   % check that finite state machine does not deadlock
   delayRatio = HybridCtrlParameters.iDelay/HybridCtrlParameters.N;
   if (delayRatio>0.6 && delayRatio<0.8)
      warndlg(['The delay compensation exceeds 60% of the simulation time step.', ...
         'Please consider increasing the simulation time step in order to avoid oscillations.'], ...
         'WARNING');
   elseif (delayRatio>=0.8)
      errordlg(['The delay compensation exceeds 80% of the simulation time step.', ...
         'The simulation time step must be increased in order to avoid deadlock.'], ...
         'ERROR');
      return
   end
   % update delay time
   HybridCtrlParameters.delay = HybridCtrlParameters.iDelay*HybridCtrlParameters.dtCon;
end

% calculate testing rate
HybridCtrlParameters.Rate = HybridCtrlParameters.dtSim/HybridCtrlParameters.dtInt;

disp('Model Properties:');
disp('=================');
disp(HybridCtrlParameters);
```

Figure 2: Hybrid Controller Parameters in initializeSimulation.m.

Schellenberg et al.
Updated: 2009-08-15
Web it.nees.org
Email it-support@nees.org

The second part in the `initializeSimulation.m` file sets up and initializes the signal counts and the SCRAMNet partitions for the Simulink Predictor-Corrector model to interact with the MTS 493 controller across SCARAMNet. This section needs to be adjusted according to the MTS controller and the number of actuators utilized at a specific testing facility.

```
%%%%%%%%%% SIGNAL COUNTS %%%%%%%%%%

nAct    = 8;                               % number of actuators
nAdcU   = 12;                              % number of user a/d channels
nDucU   = 8;                               % number of user ducs
nEncU   = 2;                               % number of user encoders
nDinp   = 4;                               % no. of digital inputs written to scramnet
nDout   = 4;                               % no. of digital outputs driven by scramnet
nUDPOut = 1+7*nAct+nAdcU+nDucU+nEncU+nDinp;  % no. of outputs from simulink bridge
nUDPInp = 1+6*nAct+nAdcU+nDucU+nEncU+nDinp;  % no. of inputs to simulink bridge

%%%%%%%%%% SAMPLE PERIOD %%%%%%%%%%

samplePeriod = 1/1024;

%%%%%%%%%% SCRAMNET PARTITIONS %%%%%%%%%%

%%%%%%%%%% outputs to scramnet %%%%%%%%%%

% master span
baseAddress           = 0;
partition(1).Address = ['0x', dec2hex(baseAddress*4)];
partition(1).Type    = 'single';
partition(1).Size    = '1';

% control modes
partition(2).Type = 'uint32';
partition(2).Size = num2str(nAct);

% displ commands
partition(3).Type = 'single';
partition(3).Size = num2str(nAct);

% force commands
partition(4).Type = 'single';
partition(4).Size = num2str(nAct);

% displ feedbacks (used only in realtime simulation mode)
partition(5).Type = 'single';
partition(5).Size = num2str(nAct);

% force feedbacks (used only in realtime simulation mode)
partition(6).Type = 'single';
partition(6).Size = num2str(nAct);

...
```

Figure 3: Signal Counts and SCRAMNet Setup in initializeSimulation.m.

Next open the Simulink model `HybridControllerD3D3_1Act.mdl`.

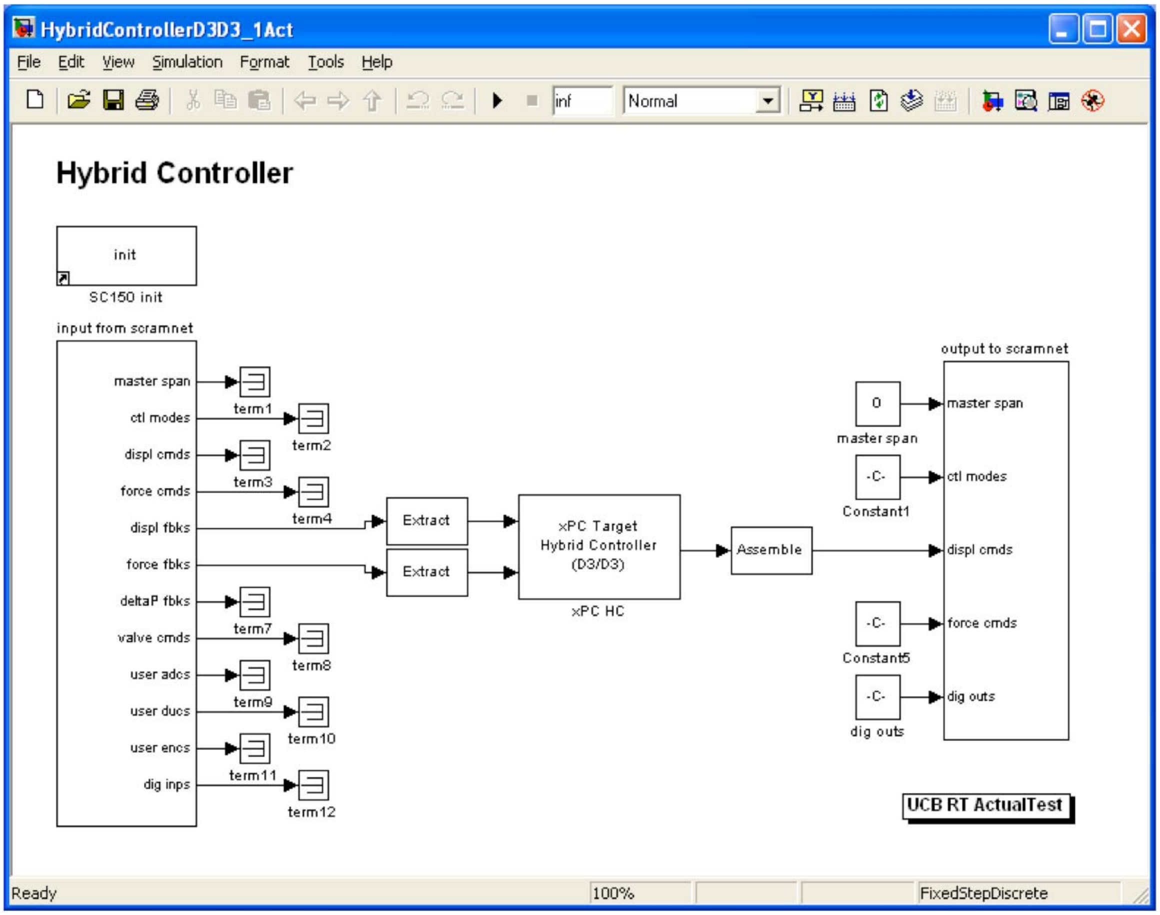

Figure 4: HybridControllerD3D3_1Act Simulink Predictor-Corrector Model.

"ctrl-B" will build the model onto the xPC Target. However, before building the model make sure that the following directories with the `PredictorCorrector.h` and `PredictorCorrector.c` files are added to the Matlab path:

```
...\OpenFresco\trunk\SRC\experimentalControl\Simulink\HybridSimToolbox
...\OpenFresco\trunk\SRC\experimentalControl\Simulink\GeneralUtilities
```

In addition, it is advised to check that the Simulink model points to the correct include directories and to modify them if necessary. To do so, select **Simulation -> Configuration Parameters**. First, navigate to **Simulation Target -> Custom Code** (see Figure 5) and modify the specified path for the HybridSimToolbox directory to the location where it was installed (see path above). Next, navigate to **Real-Time Workshop -> Custom Code** (see Figure 6) and also modify the specified path for the HybridSimToolbox directory to the location where it was installed (see path above).

Finally, save the model and hit **ctrl-B** to build the model. When the build is finished, type **`close(xpc)`** at the Matlab command prompt and hit **enter** to disconnect Matlab from the target PC and free the connection for OpenFresco. Then you can close the Simulink model.

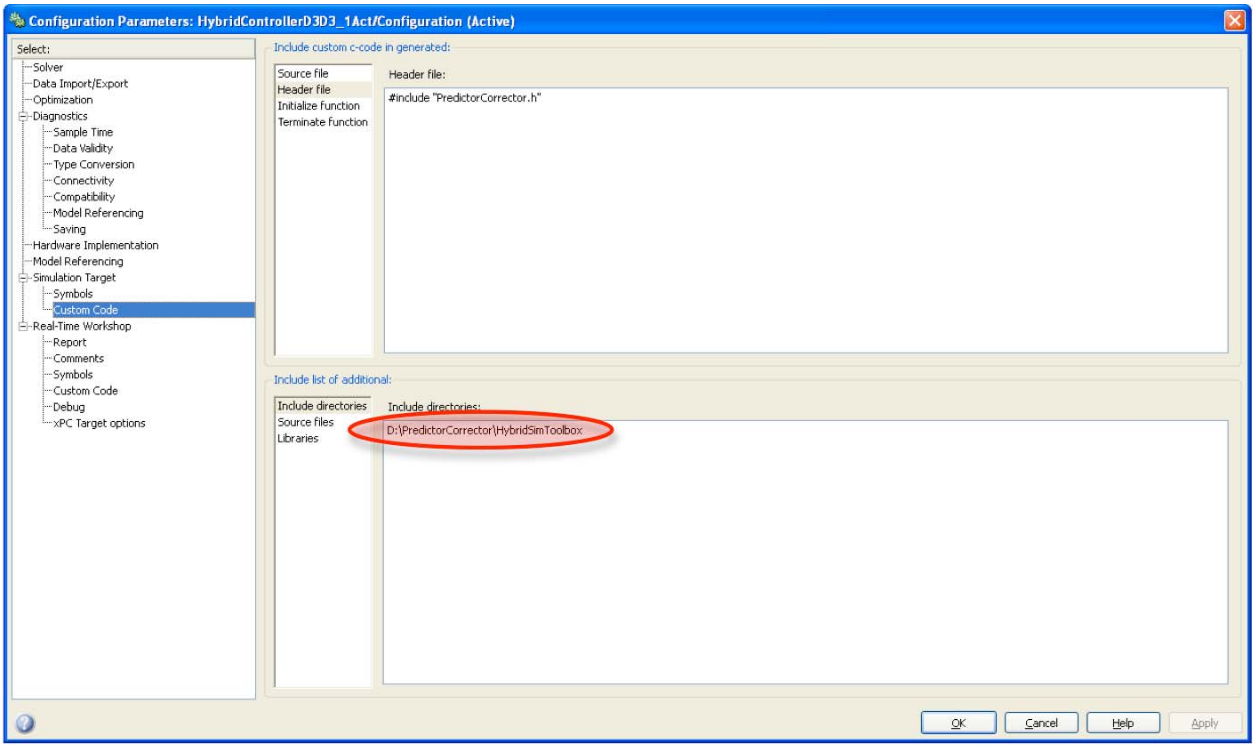

Figure 5: Configuration Parameters -> Simulation Target -> Custom Code.

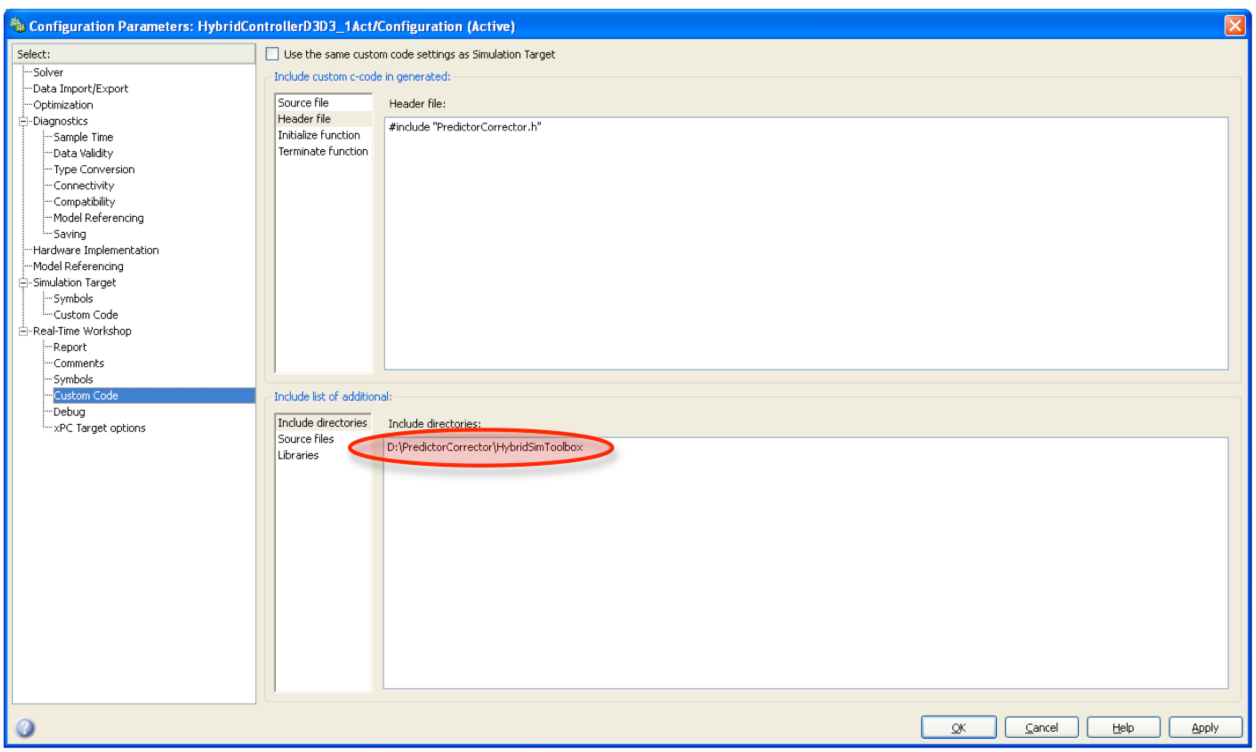

Figure 6: Configuration Parameters -> Real-Time Workshop -> Custom Code.

4 Test with Simulated Specimen

Copy the `PortalFrame_Local.tcl` file and the `SACNF01.txt` ground motion file to a new folder. See the PortalFrame example manual for detailed information about the model and the necessary steps to run the hybrid simulation. Open the Tcl file and under "`Define geometry for model`," check that "`set withGravity 0`" for excluding gravity effects (or 1 for including gravity effects). Check that "`expControl SimUniaxialMaterials 1 1`" is active and the xPCTarget control is commented out. The first test will use a simulated experimental specimen to check that everything is working properly. The Portal Frame uses an implicit integration method. Toward the bottom of the file, below "`Define dynamic loads`," set the magnitude of the ground motion. For a linear simulation, type "`set scale 0.15`." For a nonlinear simulation, set the scaling factor to 1.3. In the "`Start of analysis generation`" section, set "`test FixedNumIter 5`." This sets a fixed 5 iterations per step. The simulation time step was set to be equal to the integration time step so the fixed iterations per step will make the test 5 times slower than real-time.

To run the initial simulation, open a command window and map to the location of the folder with the Tcl file and ground motion file. Type **`OpenSees`** and hit **enter** then type **`source PortalFrame_Local.tcl`** and hit **enter** to run the simulation. Refer the PortalFrame example manual for a discussion of the simulation results.

5 Hybrid Simulation Using the μNEES or Similar Setup

Prior to the simulation, remove the coupons from the μNEES clevis and warm up the actuator using a simple sine motion from the STS function generator.

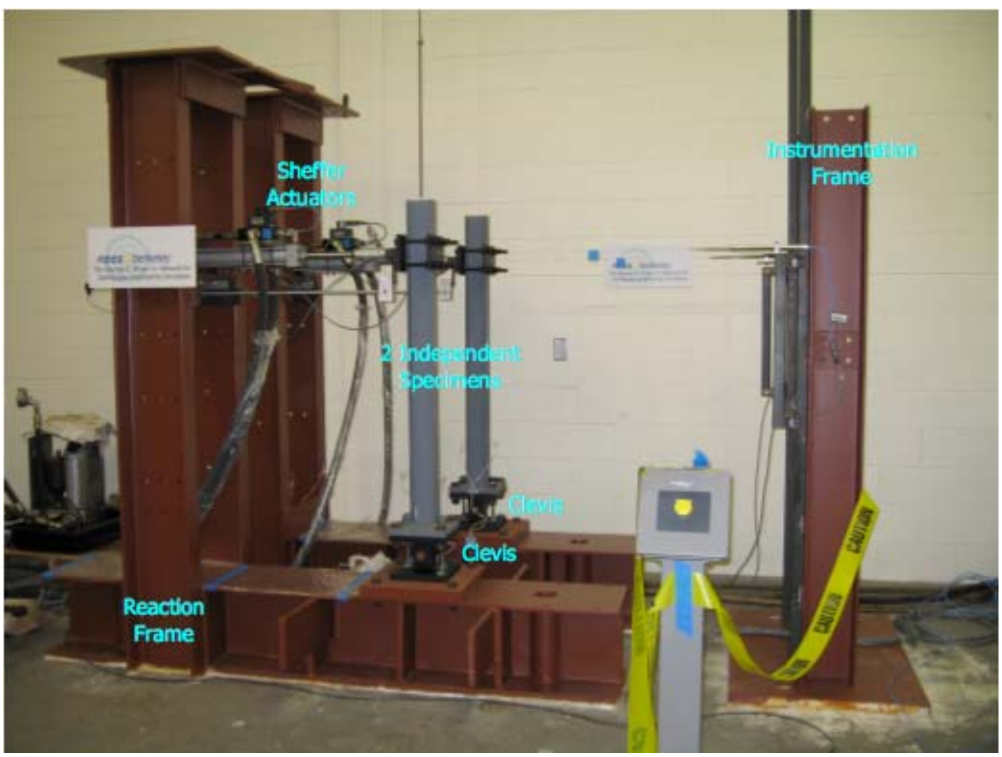

Figure 7: μNEES Experimental Setup with Two Independent Specimens.

Then replace the coupons and make sure that the STS controller is set to receive a signal from SCRAMNet (see Figure 8). Adjust the displacement so the force feedback is 0 (see Figure 9). Set the limit detectors just outside of the expected motion range (see Figure 9).

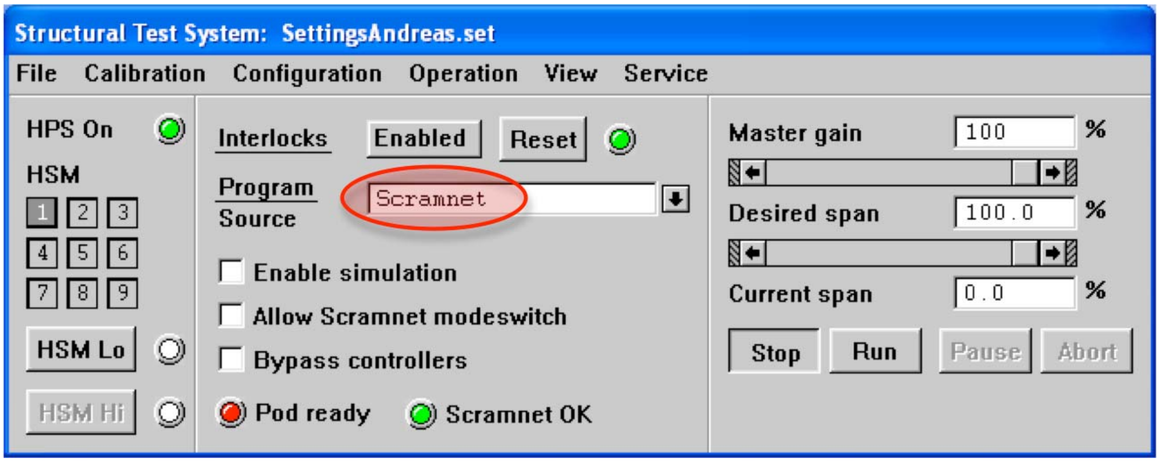

Figure 8: STS Main Window.

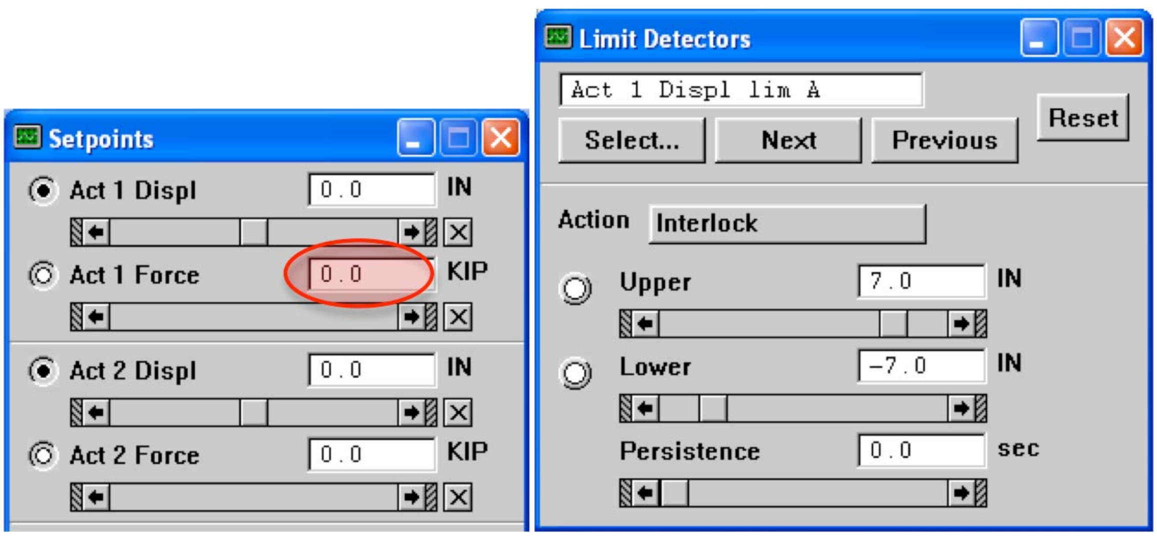

Figure 9: STS Setpoints and Limit Detectors Windows.

Since the Simulink model has already been built onto the xPC-Target machine, you do not have to do anything. If you choose to try a different simulation time step or use a different predictor-corrector model, you will have to modify and run the initializeSimulation.m file and then rebuild the Simulink model.

Copy the `PortalFrame_Local.tcl` file and the `SACNF01.txt` ground motion file to a new folder. Otherwise, the OpenSees/OpenFresco output files will be overwritten during the next test. Open the Tcl file and under "`Define geometry for model`," check that "`set withGravity 0`" for excluding gravity effects (or 1 for including gravity effects). Change the experimental control to "`expControl xPCtarget 1 1 "192.168.2.20" 22222 HybridControllerD3D3_1Act "D:/PredictorCorrector/RTActualTestModels/cmAPI-xPCTarget-STS"`." Check that the corresponding SimUniaxialMaterials experimental control is commented out instead. Toward the bottom of the file, below “`Define dynamic loads`,” set the magnitude of the ground motion. For a linear simulation, type “`set scale 0.15`.” For a nonlinear simulation, set the scaling factor to 1.3. In the “`Start of analysis generation`” section, set “`test FixedNumIter 5`.” This sets a fixed 5 iterations per step.

```
# Define experimental control
# ---------------------------
# expControl SimUniaxialMaterials $tag $matTags
expControl SimUniaxialMaterials 1 1
expControl xPCtarget 1 1 "192.168.2.20" 22222 HybridControllerD3D3_1Act "D:/PredictorCorrector/RTActualTestModels/cmAPI-xPCTarget-STS"
#expControl SimUniaxialMaterials 2 1

# Define experimental setup
# -------------------------
# expSetup OneActuator $tag <-control $ctrlTag> $dir -sizeTrialOut $t $o <-trialDispFact $f> ...
expSetup OneActuator 1 -control 1 2 -sizeTrialOut 3 3
expSetup OneActuator 2 -control 2 2 -sizeTrialOut 3 3

# Define experimental site
# ------------------------
# expSite LocalSite $tag $setupTag
expSite LocalSite 1 1
expSite LocalSite 2 2

# Define geometric transformation
# -------------------------------
#geomTransf PDelta 1
geomTransf Corotational 1

# Define experimental elements
# ----------------------------
# left and right columns
# expElement beamColumn $eleTag $iNode $jNode $transTag -site $siteTag -initStif $Kij <-iMod> <-rho $rho>
expElement beamColumn 1 3 1 1 -site 1 -initStif 1310.8 0 0 0 11.2 -280.0 0 -280.0 9333.3333
expElement beamColumn 2 4 2 1 -site 2 -initStif 1310.8 0 0 0 11.2 -280.0 0 -280.0 9333.3333

# Define numerical elements
# -------------------------
# element elasticBeamColumn $eleTag $iNode $jNode $A $E $Iz $transfTag
element elasticBeamColumn 3 3 4 3.55 29000 22.1 1
```

Figure 10: Excerpt from PortalFrame_Local.tcl.

To run the actual hybrid simulation perform the following steps:

- Open a command window and map to the location of the folder with the Tcl file and ground motion file.
- Type **`OpenSees`** and press **enter** then type **`source PortalFrame_Local.tcl`** and press **enter**.
- You will be prompted to make sure that the offset values of the controller are set to zero. This allows the user to adjust the displacements until the actuator forces are zero. Once you are satisfied press **enter** to continue.
- The command window will then display the current displacement and force feedbacks. This allows the user to further adjust the actuators, reacquire the feedbacks or abort the test.
- On the STS computer, check that the SCRAMNet command is zero and then start the data recorder. Afterwards press **Run** and wait until the “Current span” reaches 100%.
- Now press **enter** again on the xPC-Host computer to start the hybrid simulation.

- When the test ends, stop the controller and recorder on the STS computer.
- On the xPC-Host computer, type **`exit`** and press **enter**.
- Go to Matlab and set the current path to the folder containing the Tcl file and ground motion file. You will download the data recorded on the xPC-Target machine to this location.
- Type **`endSimulationXPCtarget('filename.mat')`** and press **enter** to start downloading the data.
- When this finishes, plots of the xPC-Target data will be generated.

6 Results

Results for 15% of the SACNF01 ground motion without gravity effects are shown below.

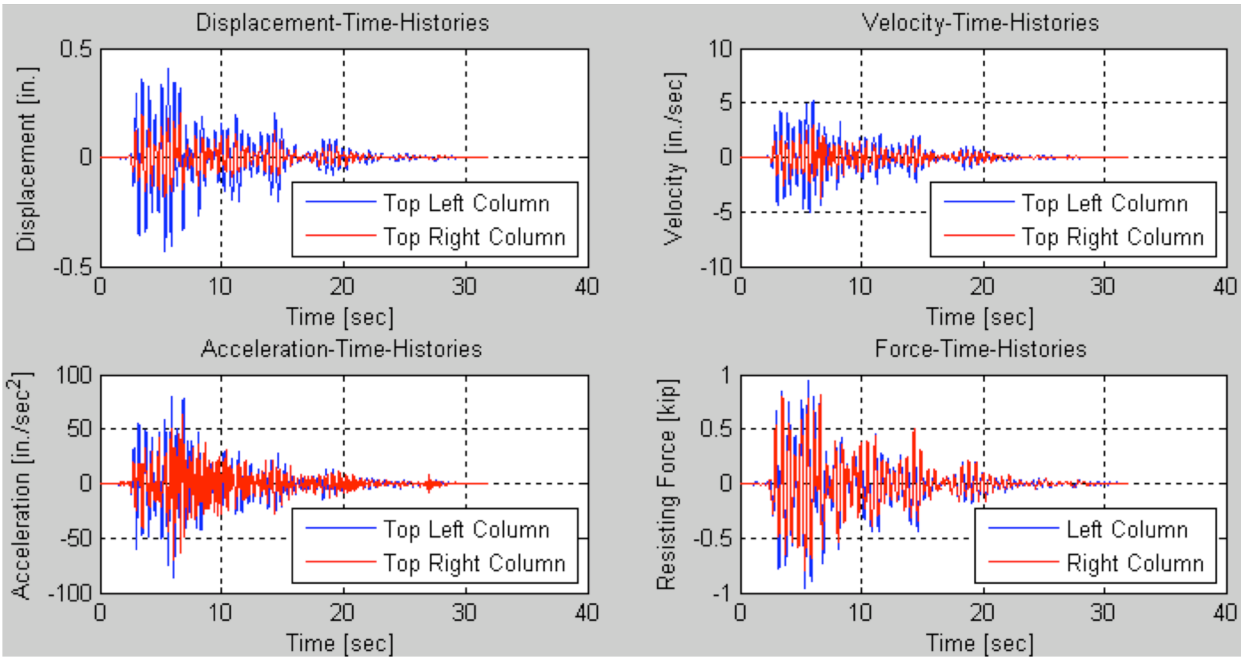

Figure 11: Elastic Ground Motion, No Gravity, Time Histories.

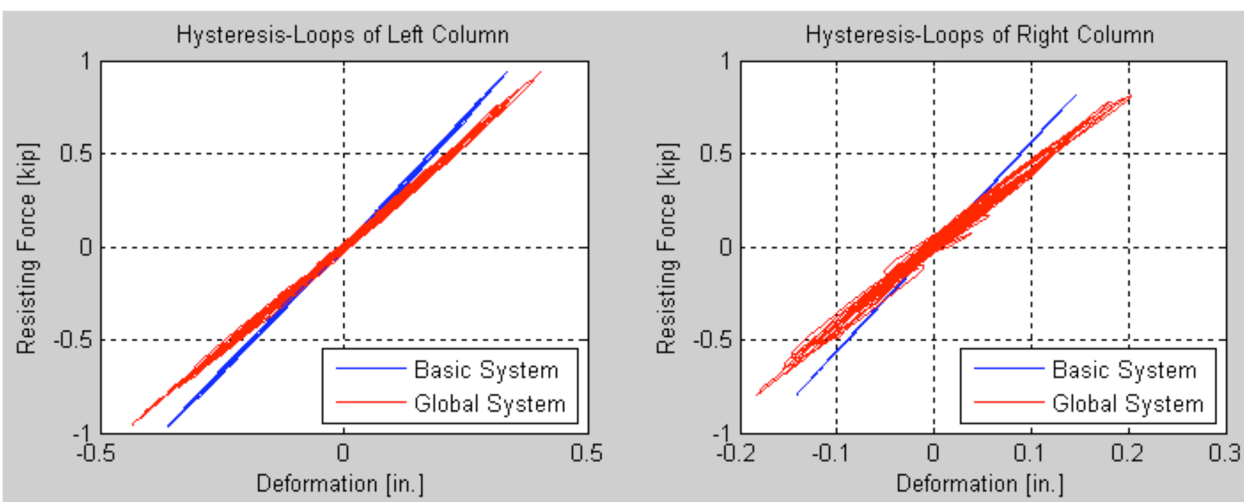

Figure 12: Elastic Ground Motion, No Gravity, Hysteresis Loops.

Results for 130% of the SACNF01 ground motion without gravity effects are shown below.

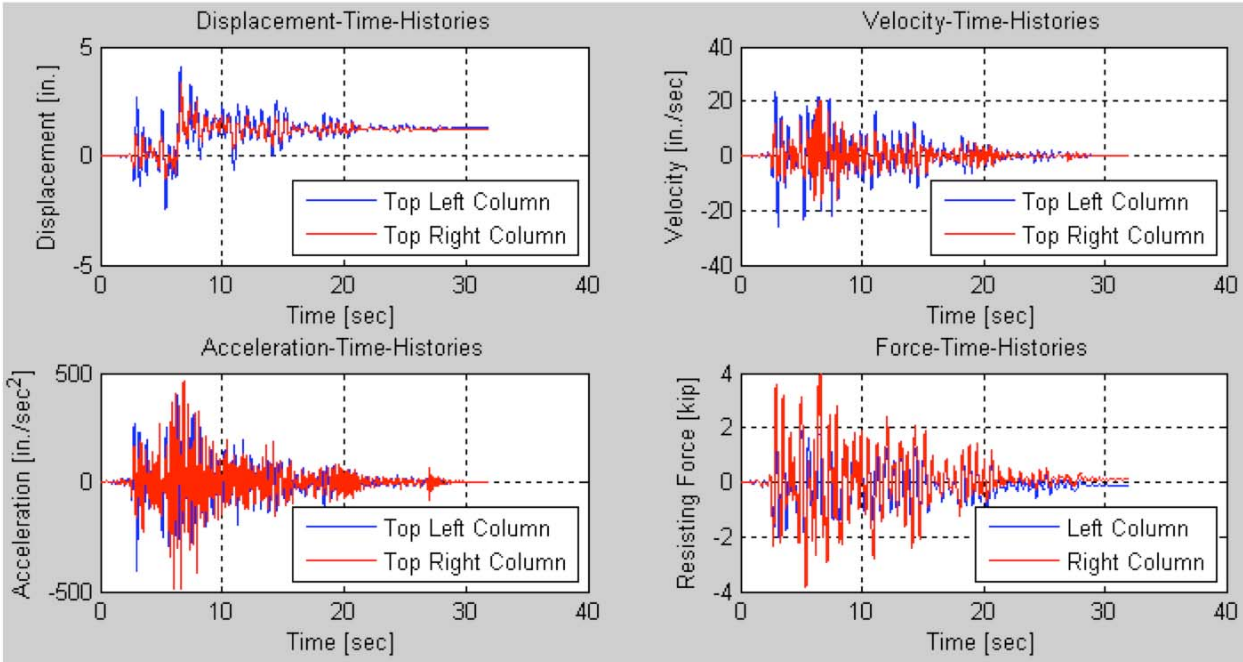

Figure 13: Inelastic Ground Motion, No Gravity, Time Histories.

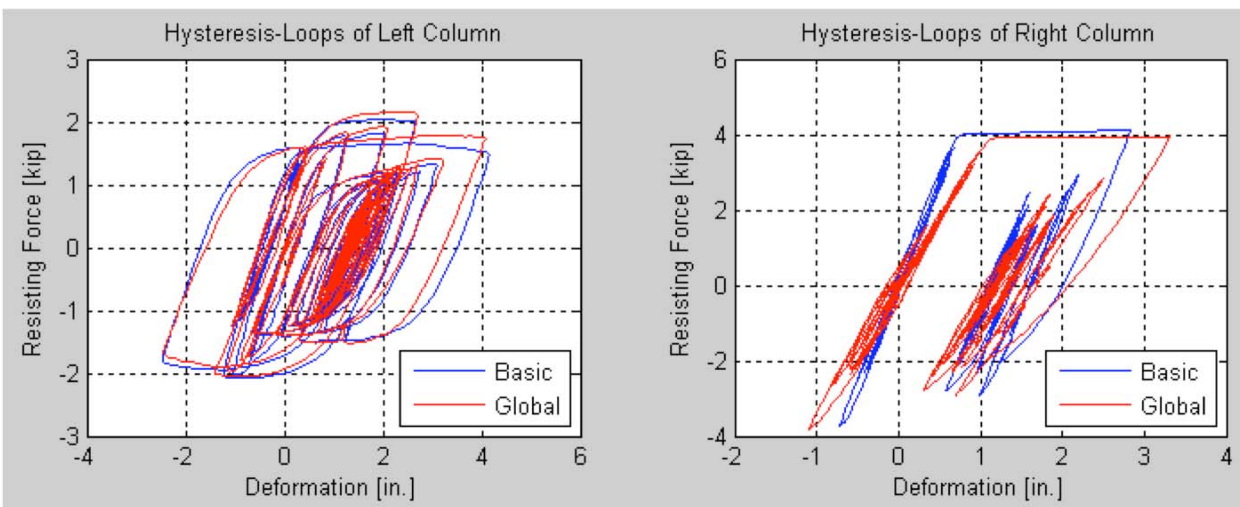

Figure 14: Inelastic Ground Motion, No Gravity, Hysteresis Loops.

Results for 15% of the SACNF01 ground motion with gravity effects are shown below.

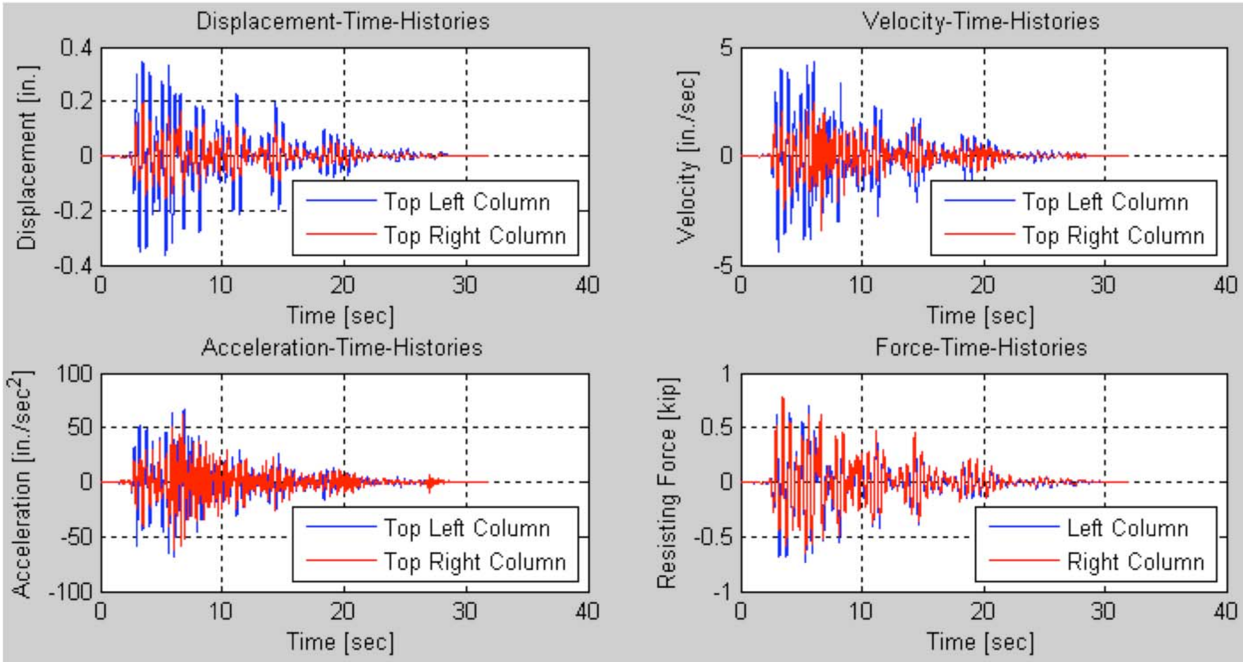

Figure 15: Elastic Ground Motion, With Gravity, Time Histories.

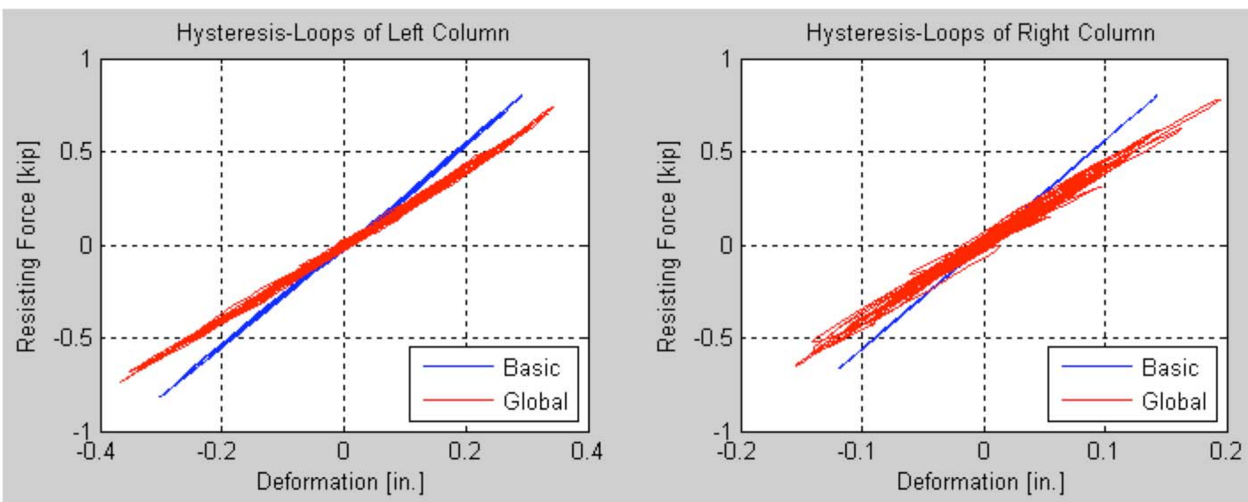

Figure 16: Elastic Ground Motion, With Gravity, Hysteresis Loops.

Results for 130% of the SACNF01 ground motion with gravity effects are shown below.

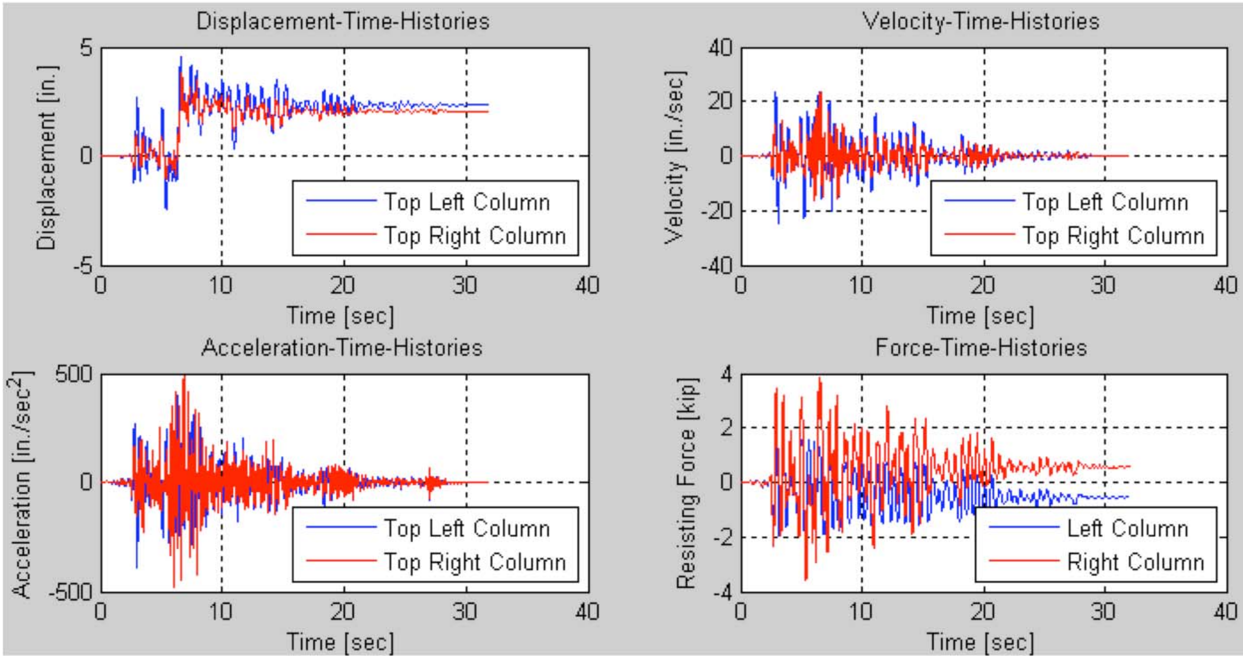

Figure 17: Inelastic Ground Motion, With Gravity, Time Histories.

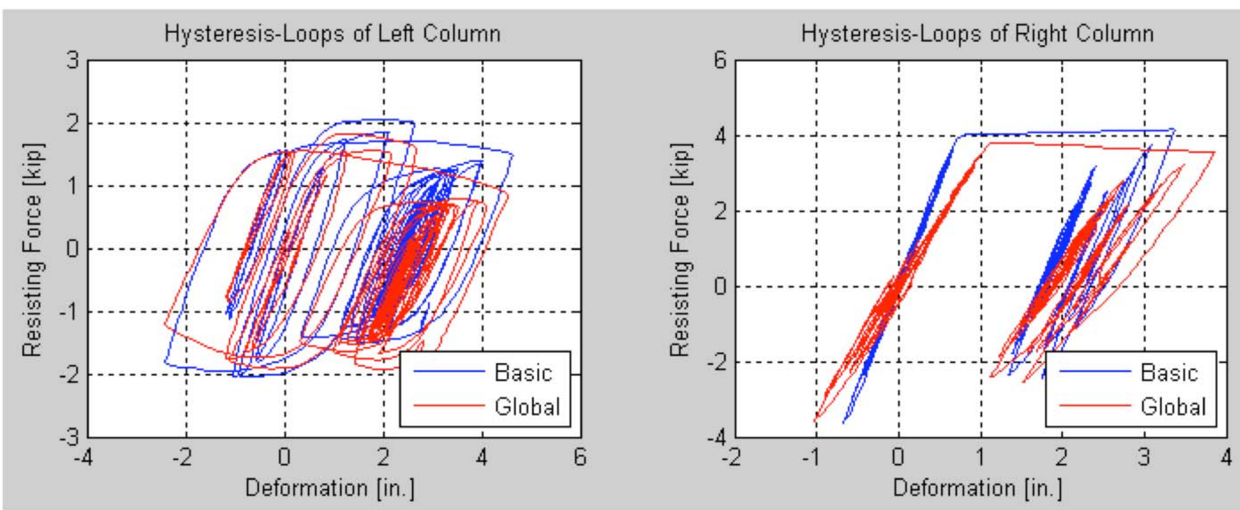

Figure 18: Inelastic Ground Motion, With Gravity, Hysteresis Loops.

7 References

Mathworks, 2009, “MATLAB, Simulink, Stateflow, xPC Target,” <http://www.mathworks.com>.

Mosqueda, G., 2003, “Continuous Hybrid Simulation with Geographically Distributed Substructures,” Dissertation, University of California, Berkeley.

OpenFresco, 2009, “Open Framework for Experimental Setup and Control” <http://openfresco.neesforge.nees.org>.

OpenSees, 2009, “Open System for Earthquake Engineering Simulation,” <http://opensees.berkeley.edu>.

Schellenberg, A., 2008, “Advanced Implementation of Hybrid Simulation,” Dissertation, University of California, Berkeley.