

Discovering OpenSees: Conducting Hybrid Simulations with OpenSees/OpenFresco

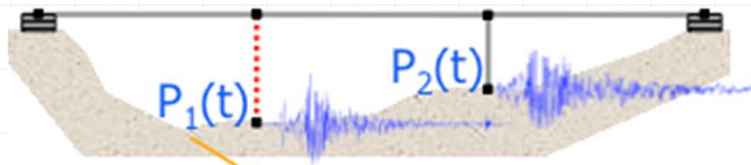
Andreas Schellenberg

Pacific Earthquake Engineering Research Center, University of California, Berkeley



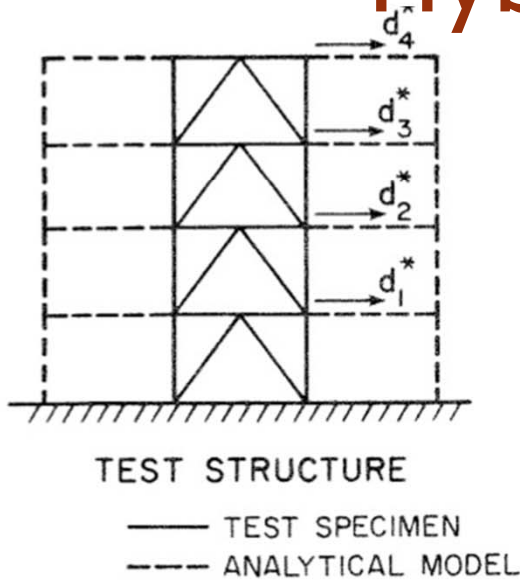
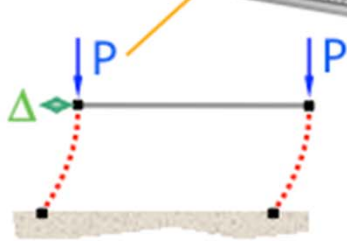
Outline of Presentation

1. Introduction to Hybrid Simulation
2. OpenFresco Architecture and TCL commands
3. Downloading and Installing OpenFresco
4. Building a Hybrid Model in OpenSees/OpenFresco
5. Simulated vs. Real Controllers
6. Using other Computational Drivers
7. Summary & Conclusions

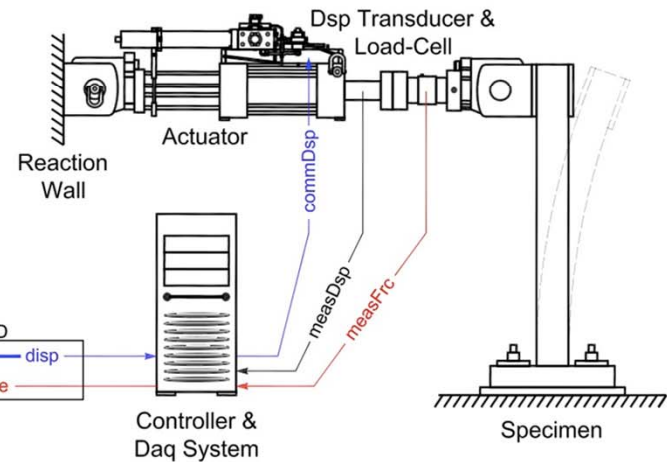
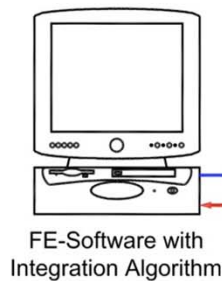


$$\cdot \ddot{\mathbf{u}} + \mathbf{C} \cdot \dot{\mathbf{u}} + \mathbf{P}_r(\mathbf{u}) = \mathbf{P}(t)$$

Introduction to Hybrid Simulation



Discrete FE-Model with static & dynamic loading



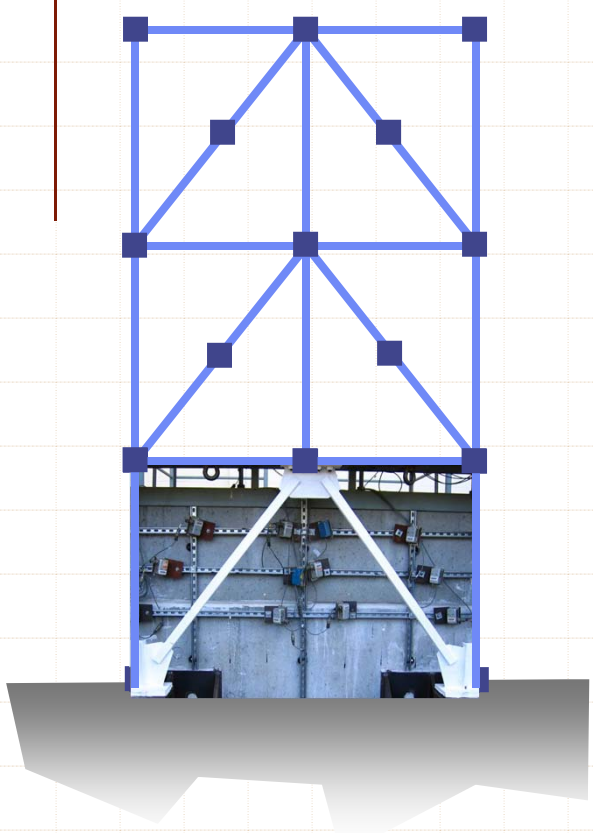
Comparison of Exp. Test Methods

	Quasi-Static	Shaking Table	Hybrid Simulation
Dynamics	NO	YES	YES
Strain Rate Effects	NO	YES	YES (if real-time test)
Large- or Full-Scale	YES	NO (limited by table)	YES

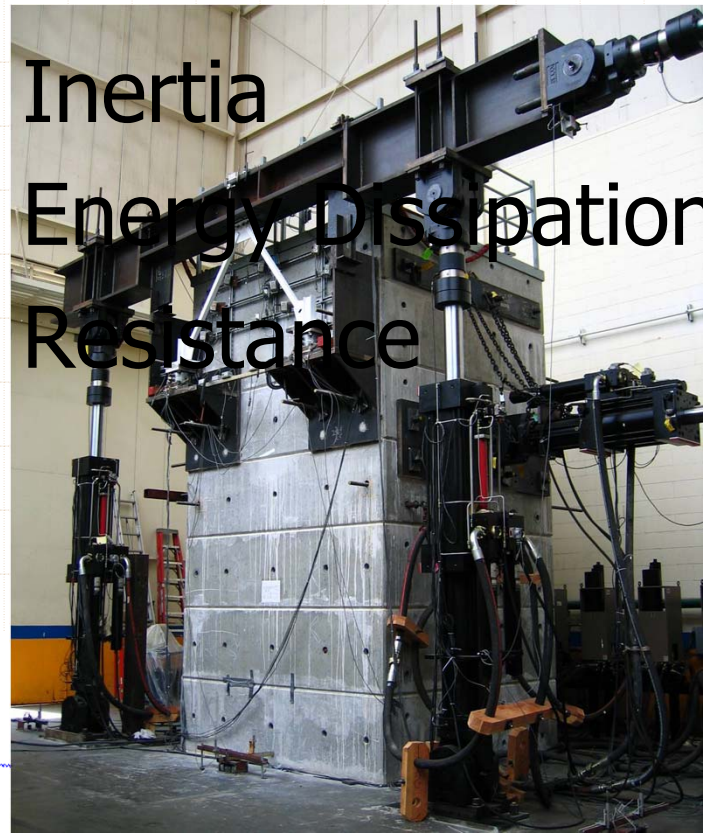
- ★ First Hybrid Simulation (Online Test) in 1975 by Takanashi et al.

Hybrid Simulation

$$\mathbf{M} \cdot \ddot{\mathbf{u}} + \mathbf{C} \cdot \dot{\mathbf{u}} + \mathbf{P}_r(\mathbf{u}) = \mathbf{P}(t)$$



- ✦ Inertia
- ✦ Energy Dissipation
- ✦ Resistance



Hybrid Simulation

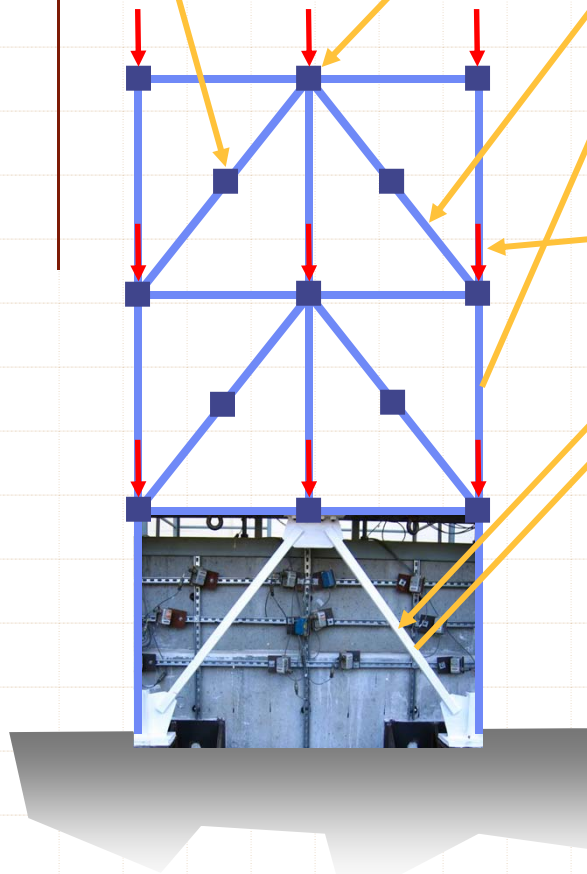
$$\mathbf{M} \cdot \ddot{\mathbf{u}} + \mathbf{C} \cdot \dot{\mathbf{u}} + \mathbf{P}_r(\mathbf{u}) = \mathbf{P}(t)$$

Dynamic Loading:

- Seismic
- Wind
- Blast/Impact
- Wave
- Traffic

Static Loading:

- Gravity
- Prestress



analytically add nonlinear geometric effects to measured resisting forces

 analytical model of structural energy dissipation and inertia

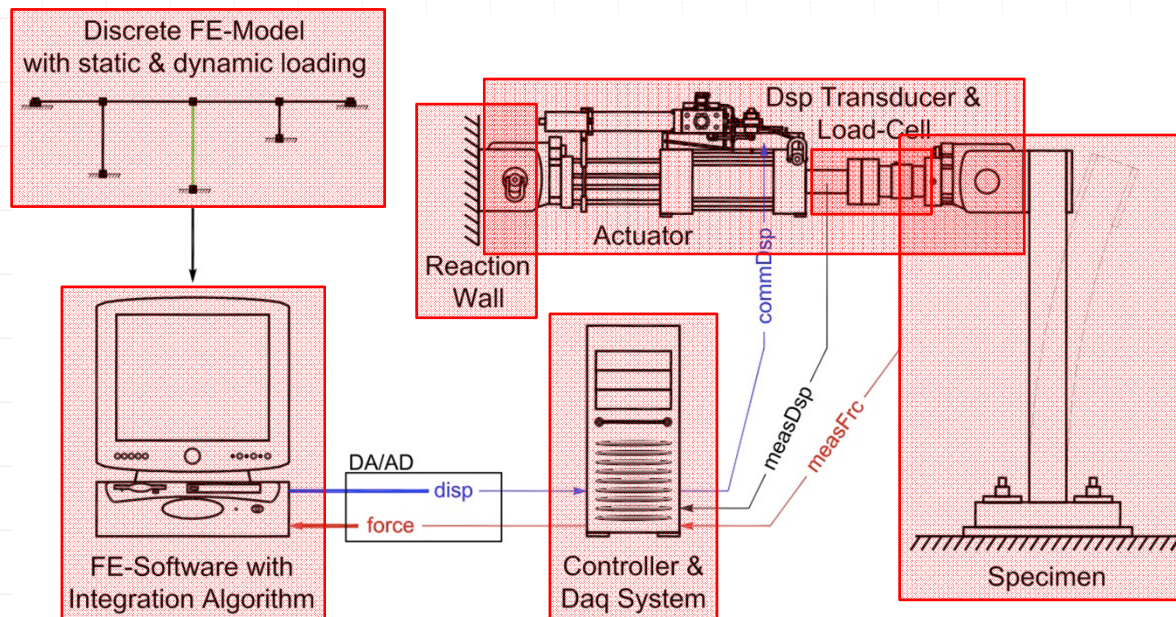
 physical model of structural resistance

Hybrid Simulation

- ★ Model the well understood parts of a structure in a finite element program on one or more computers
- ★ Leave the construction and testing of the highly nonlinear and/or numerically hard to model parts of the structure in one or more laboratories
- ★ Can be considered as a conventional finite element analysis where physical models of some portions of the structure are embedded in the numerical model

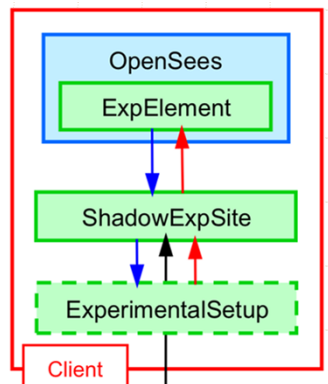
Required Components

1. Discrete model of the structure to be analyzed, including the static and dynamic loading
2. Servo-hydraulic control system with static or dynamic actuators
3. Physical test specimen, including a reaction-frame
4. Data acquisition system with instrumentation

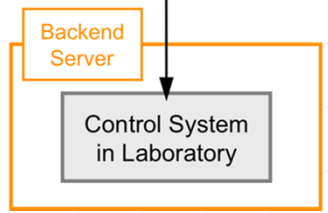
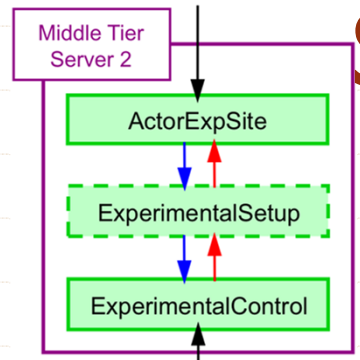


Testing Methods

- ★ Conventional hybrid simulation test where specimen is loaded using a ramp-and-hold loading procedure
- ★ Continuous test where specimen is loaded at a continuous slow to moderately slow rate to avoid load relaxations
- ★ Real-time test where specimen is loaded at correct velocities to account for rate-dependent material behaviors
- ★ Geographically distributed network test



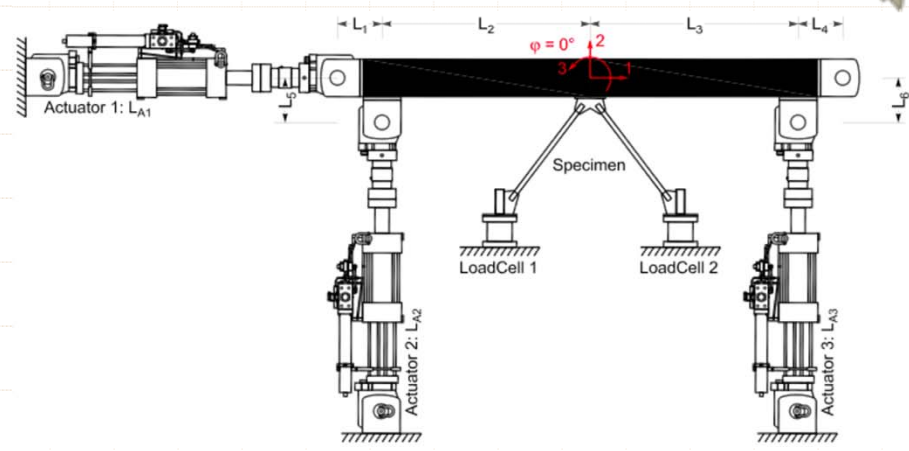
TCP/IP (Channel)



OpenFresco Software Framework



Sysstran[™]



What is OpenFresco?

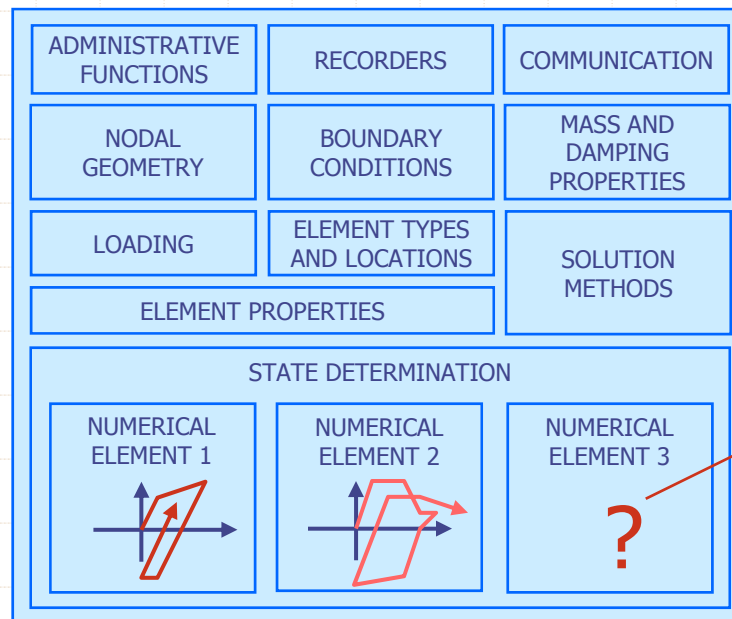
- ★ Open source Framework for Experimental Setup and Control
- ★ Secure, object oriented, network enabled “**middleware**” -- Pairs computer analysis software with laboratory control systems and other software to enable hybrid and collaborative computing:
- ★ Computational Drivers
 - OpenSees
 - OpenFresco *Express*
 - Abaqus
 - LS-DYNA
 - Matlab
 - Simulink
 - Ansys
 - UI-SimCor
- ★ Control Systems
 - dSpace
 - MTS
 - ◆ STS family
 - ◆ Flextest/CSI
 - ◆ Flextest/Scramnet
 - National Instruments
 - Pacific Instruments
 - ADwin

Why a Software Framework?

- ★ Lack of a common framework for development and deployment of HS
 - ★ Problem specific implementations which are site and control system dependant
 - ★ Such highly customized software implementations are difficult to adapt to different structural problems
- ➔ Need a robust, transparent, adaptable, and easily extensible software framework for research and deployment

Rethinking implementation strategies

- ★ Embed test specimen(s) in an existing computational framework of user's choice

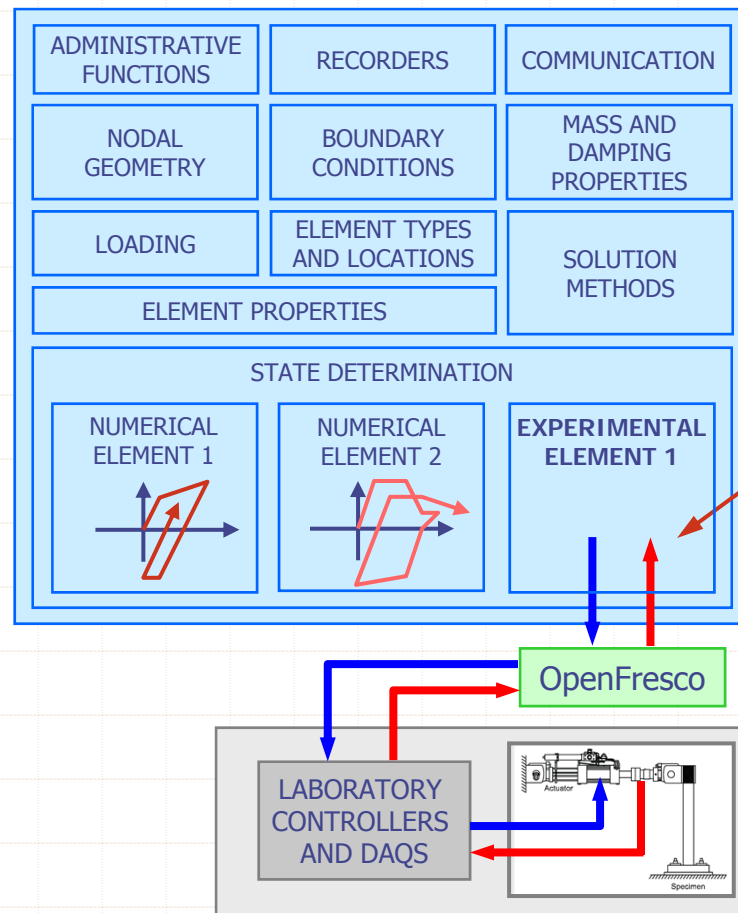


Typical features of an analysis framework

Proper numerical model uncertain

Rethinking implementation strategies

- ★ Embed test specimen(s) in an existing computational framework of user's choice



Typical features of an analysis framework

Define element as an "Experimental Element"

OpenFresco

Laboratory

OpenFresco Components

FE-Software

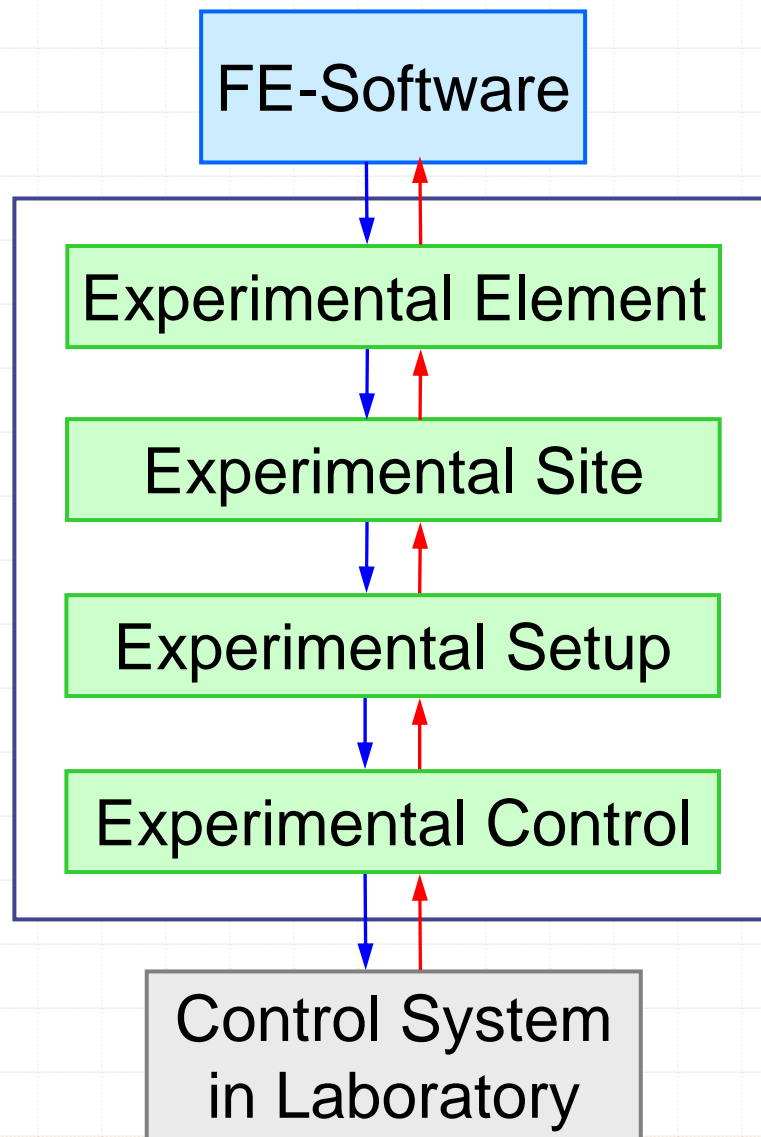
provides all features of unmodified computational framework, including parallel and network computing

OpenFresco
(Middleware)

Control System
in Laboratory

provides control of physical actuators as well as data acquisition using physical instrumentation devices

OpenFresco Components



provides all features of unmodified computational framework, including parallel and network computing

represents the part of the structure that is physically tested and provides the interface between the FE-software and the experimental software framework

stores data and provides communication methods for distributed testing

transforms between the experimental element degrees of freedom and the actuator degrees of freedom (linear or non-linear transformations)

interfaces to the different control and data acquisition systems in the laboratories

provides control of physical actuators as well as data acquisition using physical instrumentation devices

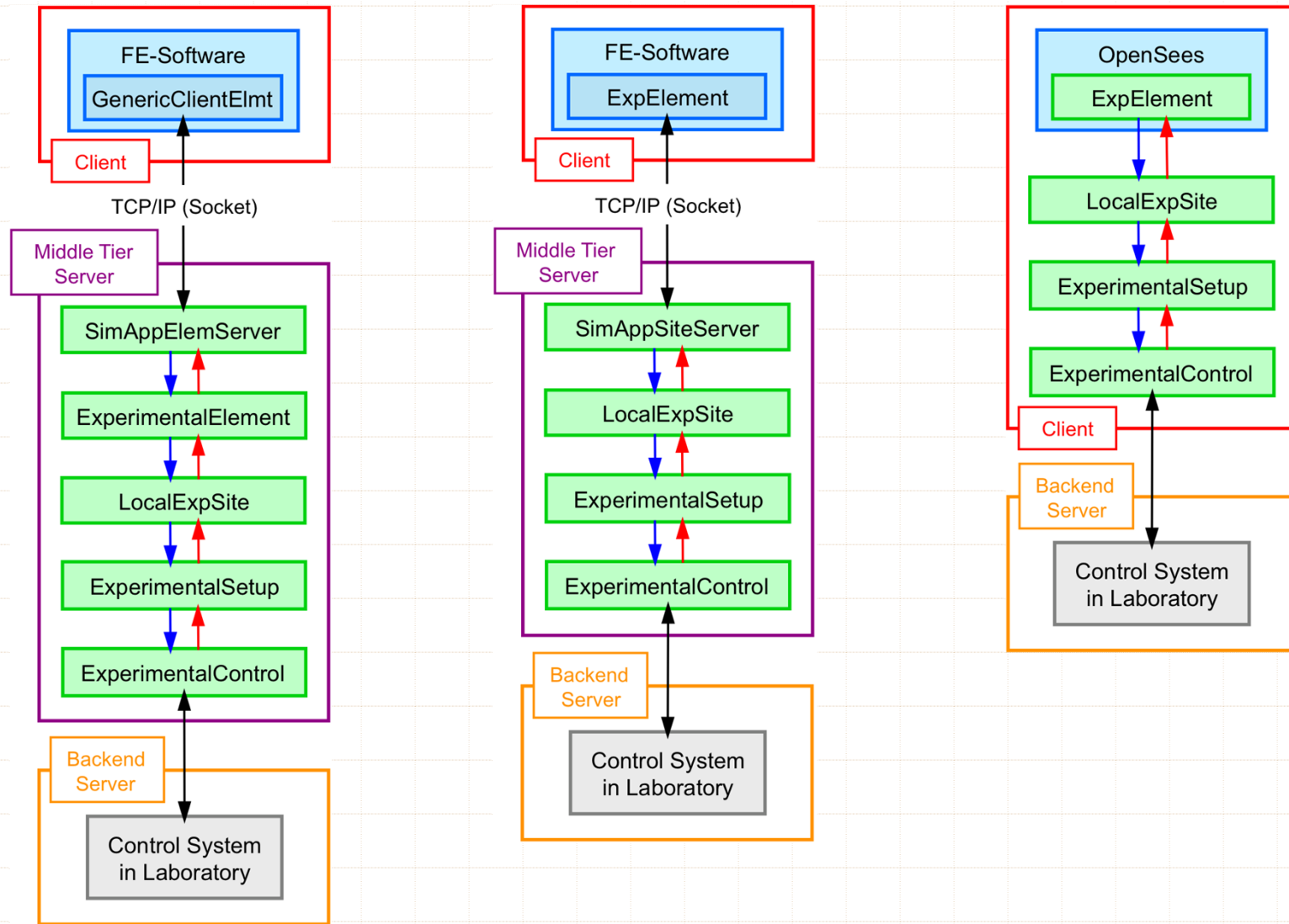
Requirements for Architecture

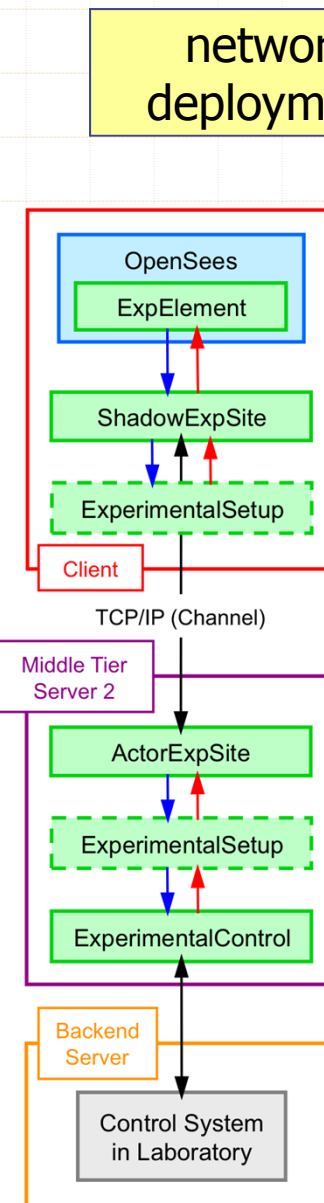
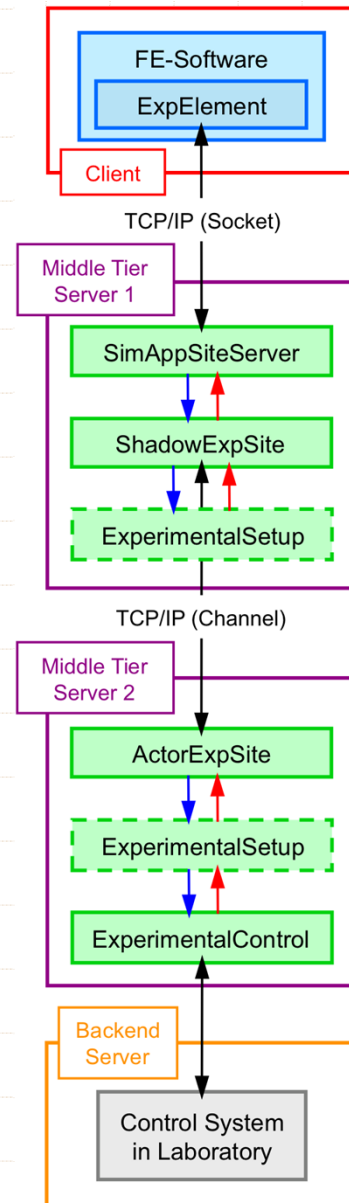
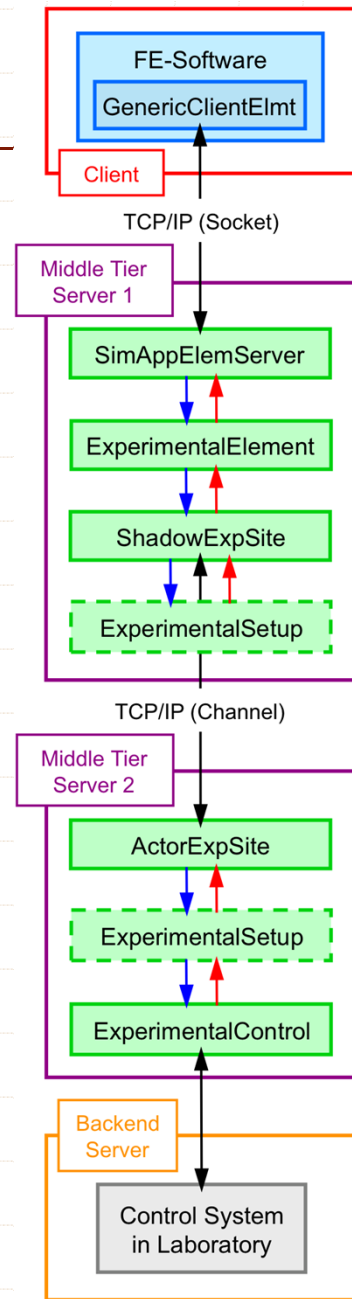
- ✦ Provide connectivity to a wide variety of FE-software (clients), independent of the language, such analysis software is programmed in
- ✦ Enable distributed testing and support different communication protocols
- ✦ Interface with rapidly evolving control and data acquisition systems deployed at testing facilities all over the world

➔ Multi/Three-Tier Software Architecture

OpenFresco Components

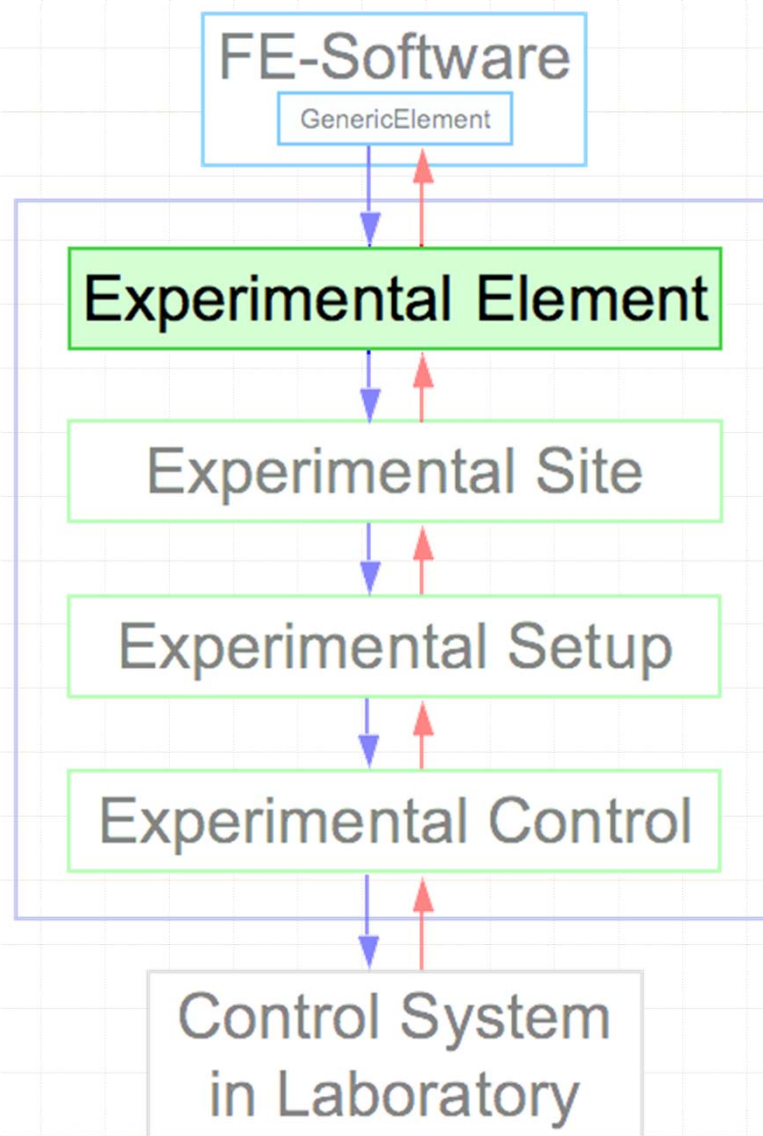
local deployment





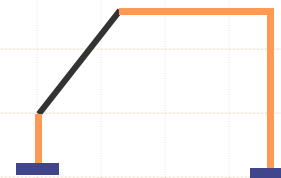
network deployment

OpenFresco Components

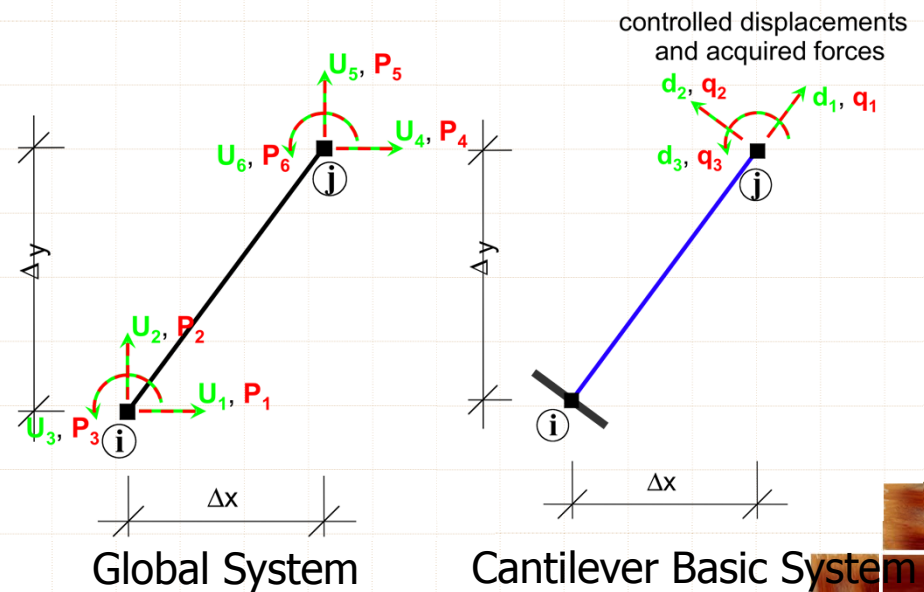


Experimental Element

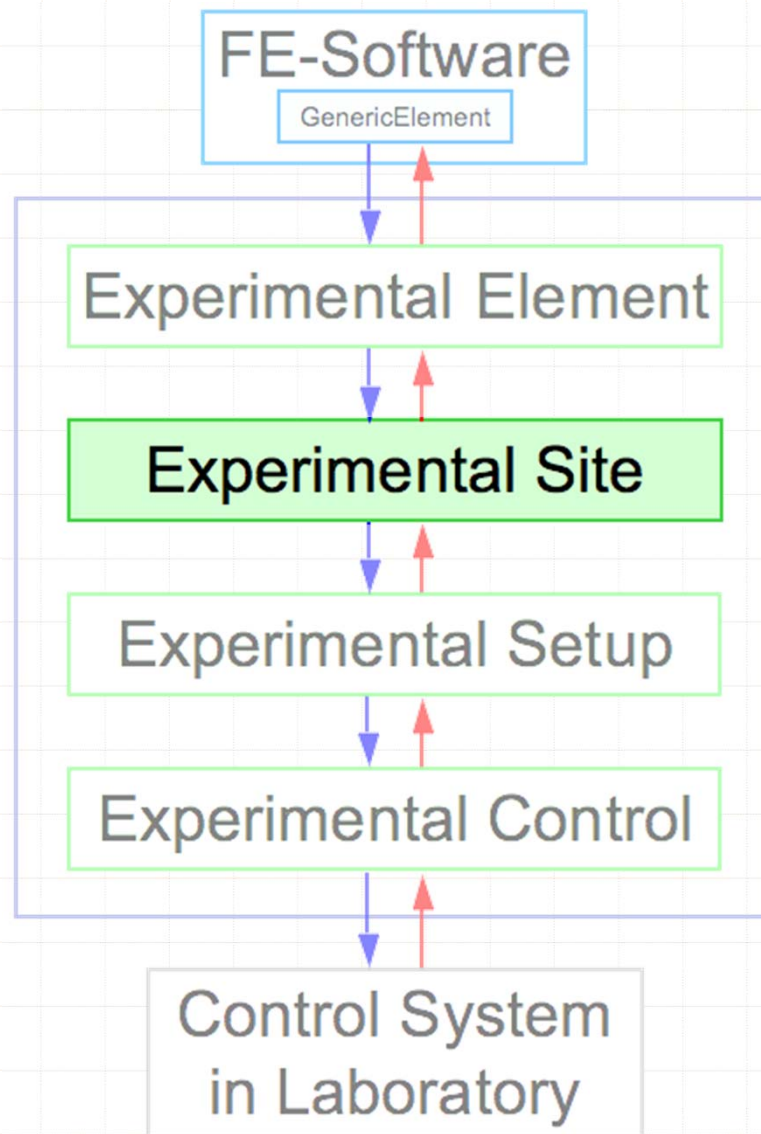
Transforms between the global element degrees of freedom in the FE-Software and the basic element degrees of freedom in the experimental element



Consider element in structure
Two coordinate systems
used in FE analysis



OpenFresco Components



Experimental Site

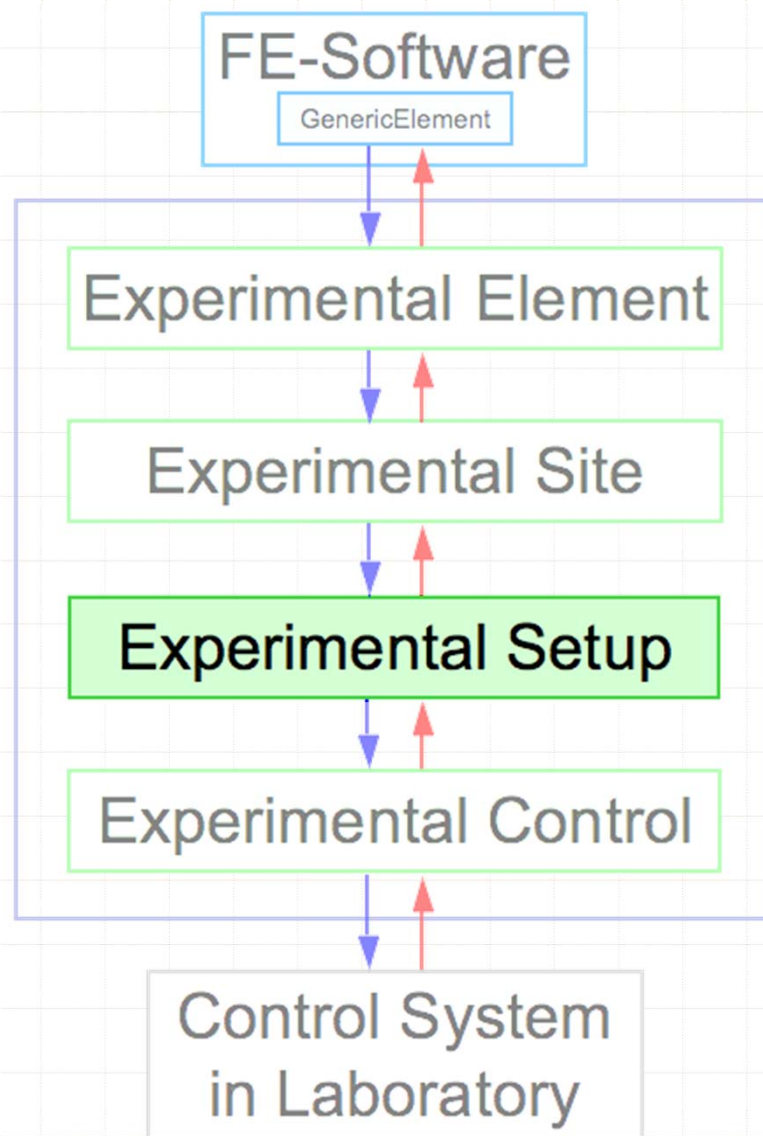
Stores data and provides communication methods for distributed testing

LocalExpSite available for local testing and RemoteExpSite/ActorExpSite pair available for geographically distributed testing

Utilizes communication channels with TCP, TCP+SSL or UDP communication protocols

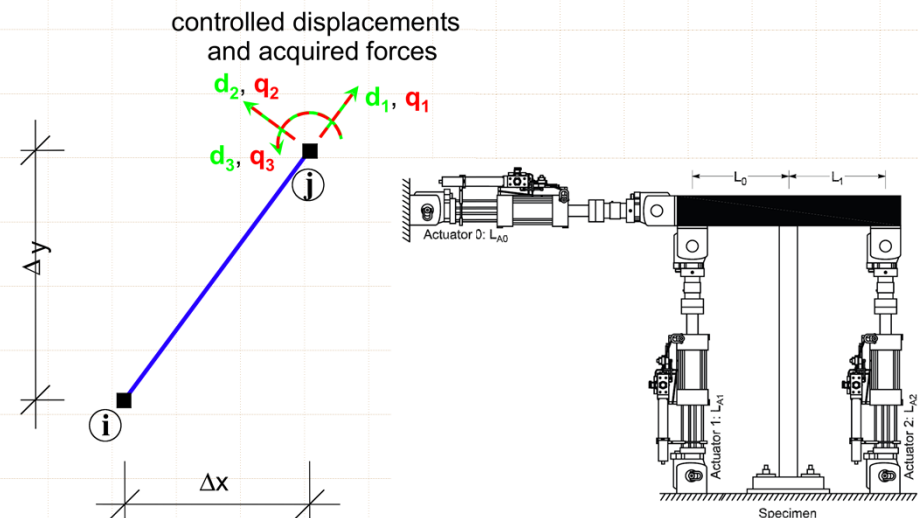


OpenFresco Components



Experimental Setup

Transforms between the basic experimental element degrees of freedom in OpenFresco and the actuator degrees of freedom in the laboratory (linear vs. non-linear transformations are available)

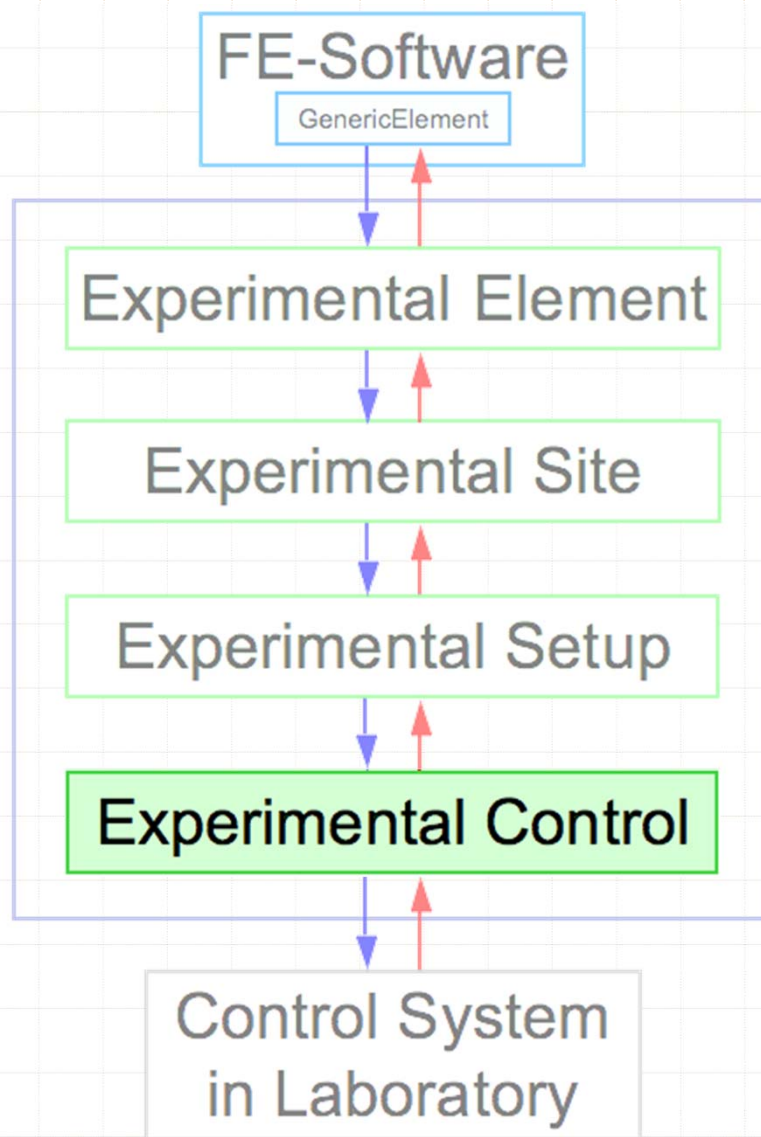


Cantilever Basic System

Actuator Setup

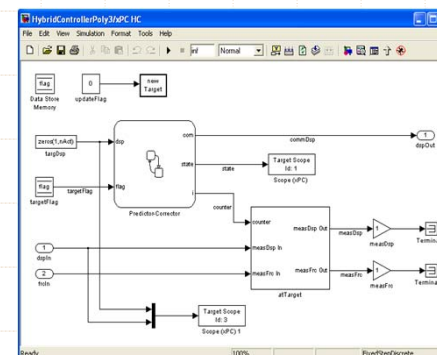
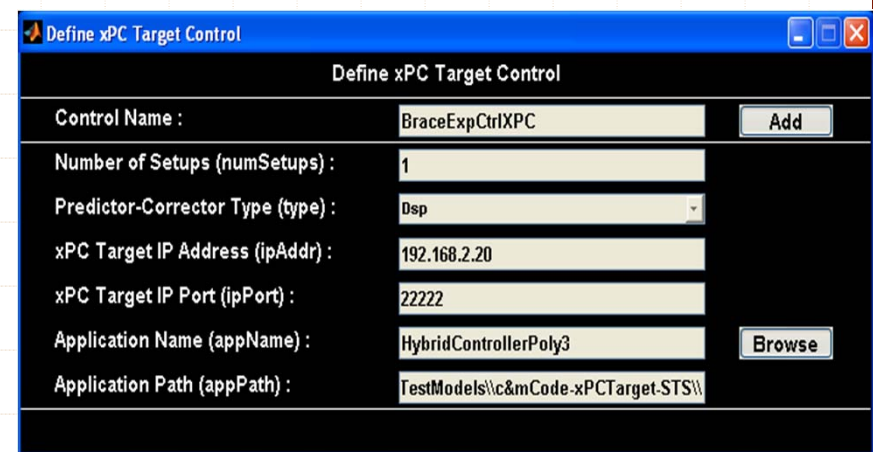
$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -L_0 \\ 0 & 1 & L_1 \end{bmatrix}$$

OpenFresco Components

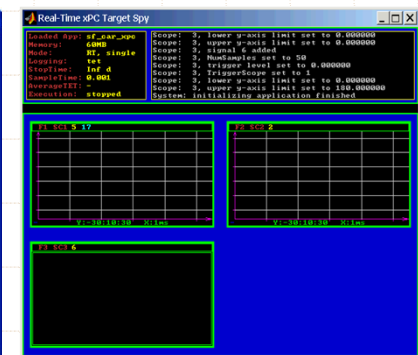


Experimental Control

Interfaces to the different control and data acquisition systems in the laboratories (IP addresses and port numbers)



Predictor-Corrector



xPC Target

OpenFresco TCL-Commands



```
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 OneRayFrame_Local_SimAppServer.tcl
Channel successfully created: Waiting for Simulation Application Client...
SimAppSiteServer with ExpSite 1 now running...
SimAppSiteServer with ExpSite 1 shutdown
OpenFresco >
```

```
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 OneRayFrame_Disco_SimAppServer.tcl
Connected to ActorExpSite 1
Channel successfully created: Waiting for Simulation Application Client...
SimAppSiteServer with ExpSite 1 now running...
Disconnected from ActorExpSite 1

SimAppSiteServer with ExpSite 1 shutdown
OpenFresco >
```

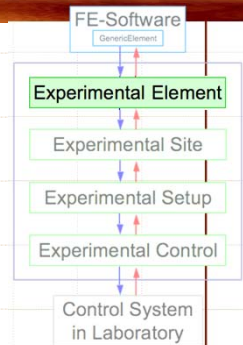
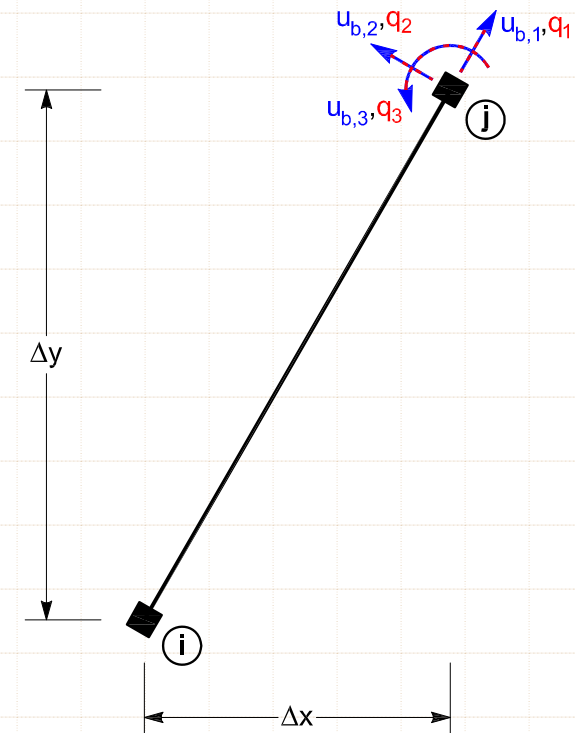
Experimental Elements

EEBeamColumn (2D,3D)

```
expElement beamColumn $eleTag $iNode $jNode
  $stranTag -site $siteTag -initStif $Kij
  <-iMod> <-rho $rho>
```

\$eleTag	unique element tag
\$iNode, \$jNode	end nodes
\$stranTag	tag of previously defined crd-transf object
\$siteTag	tag of previously defined site object
\$Kij	initial stiffness matrix elements (ndf x ndf)
-iMod	flag for I-Modification (optional, default=false)
\$rho	mass per unit length (optional, default=0.0)

Cantilever Basic DOF

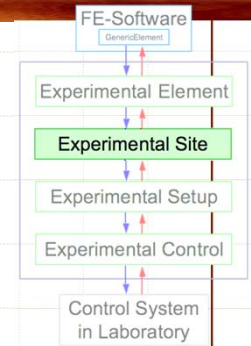


Experimental Sites

LocalExpSite

`expSite LocalSite $tag $setupTag`

`$tag` unique site tag
`$setupTag` tag of previously defined setup object

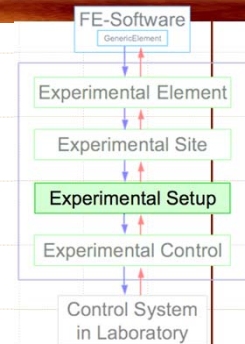
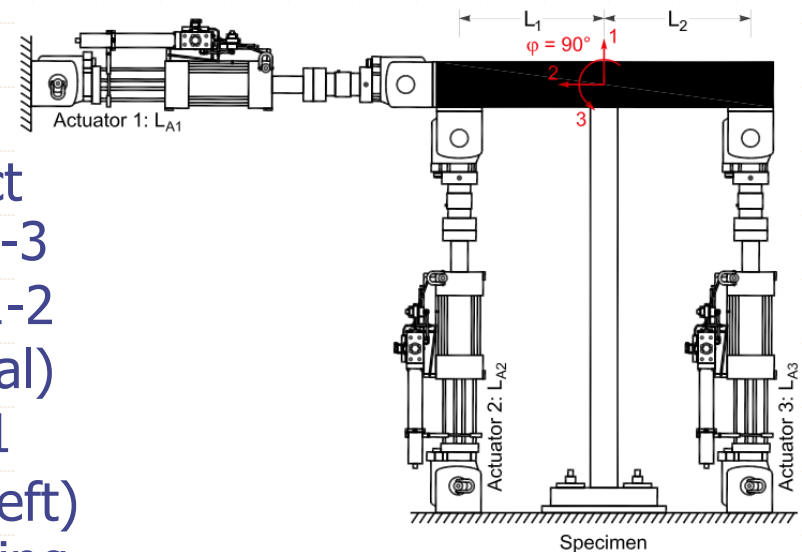


Experimental Setups

ESThreeActuators

```
expSetup ThreeActuators $tag <-control $ctrlTag>  
  $La1 $La2 $La3 $L1 $L2 <-nlGeom> <-posAct1 $pos>  
  <-phiLocX $phi> <-trialDispFact $f> ...
```

\$tag	unique setup tag
\$ctrlTag	tag of previously defined control object
\$La1-3	length of actuators 1-3
\$L1-2	length of rigid links 1-2
-nlGeom	NL-geometry (optional)
\$pos	position of actuator 1 (optional, default = left)
\$phi	angle from rigid loading beam to local x-axis (optional, default = 0.0)

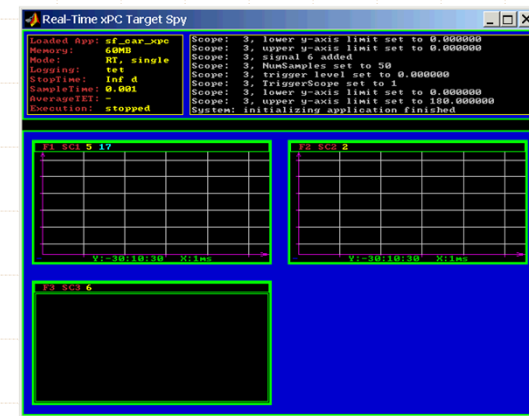
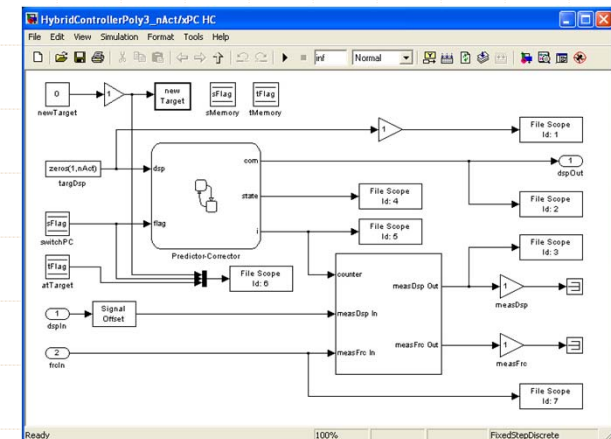
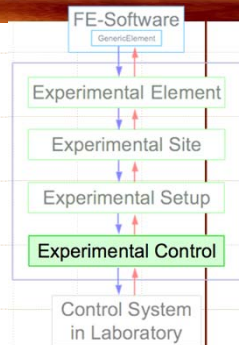


Experimental Controls

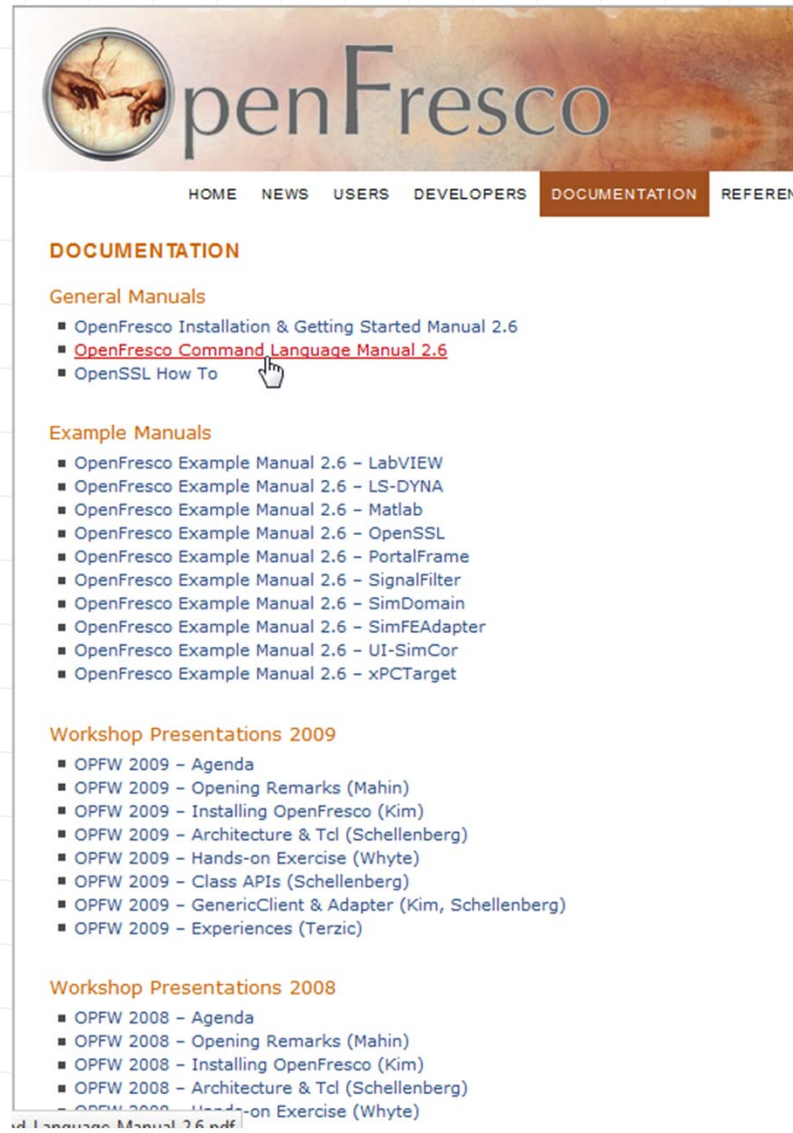
ECxPCtarget

`expControl` `xPCtarget` \$tag \$type ipAddr \$ipPort
appName appPath

\$tag unique control tag
\$type predictor-corrector type
ipAddr IP address of xPC Target
\$ipPort IP port of xPC Target
appName name of Simulink application to be loaded
appPath path to Simulink application



Command Language Manual



The screenshot shows the OpenFresco website's documentation page. At the top, there is a navigation menu with links for HOME, NEWS, USERS, DEVELOPERS, DOCUMENTATION (highlighted), and REFEREN. Below the navigation, the page is titled "DOCUMENTATION" and is divided into three main sections: "General Manuals", "Example Manuals", and "Workshop Presentations 2009". Under "General Manuals", there are three items: "OpenFresco Installation & Getting Started Manual 2.6", "OpenFresco Command Language Manual 2.6" (which is highlighted with a mouse cursor), and "OpenSSL How To". Under "Example Manuals", there are ten items, each representing a different software application or framework. Under "Workshop Presentations 2009", there are ten items, each representing a presentation from the OpenFresco Framework Workshop 2009. At the bottom of the screenshot, there is a small text label "Command Language Manual 2.6.pdf".

OpenFresco

HOME NEWS USERS DEVELOPERS DOCUMENTATION REFEREN

DOCUMENTATION

General Manuals

- OpenFresco Installation & Getting Started Manual 2.6
- OpenFresco Command Language Manual 2.6**
- OpenSSL How To

Example Manuals

- OpenFresco Example Manual 2.6 - LabVIEW
- OpenFresco Example Manual 2.6 - LS-DYNA
- OpenFresco Example Manual 2.6 - Matlab
- OpenFresco Example Manual 2.6 - OpenSSL
- OpenFresco Example Manual 2.6 - PortalFrame
- OpenFresco Example Manual 2.6 - SignalFilter
- OpenFresco Example Manual 2.6 - SimDomain
- OpenFresco Example Manual 2.6 - SimFEAdapter
- OpenFresco Example Manual 2.6 - UI-SimCor
- OpenFresco Example Manual 2.6 - xPCTarget

Workshop Presentations 2009

- OPFW 2009 - Agenda
- OPFW 2009 - Opening Remarks (Mahin)
- OPFW 2009 - Installing OpenFresco (Kim)
- OPFW 2009 - Architecture & Tcl (Schellenberg)
- OPFW 2009 - Hands-on Exercise (Whyte)
- OPFW 2009 - Class APIs (Schellenberg)
- OPFW 2009 - GenericClient & Adapter (Kim, Schellenberg)
- OPFW 2009 - Experiences (Terzic)

Workshop Presentations 2008

- OPFW 2008 - Agenda
- OPFW 2008 - Opening Remarks (Mahin)
- OPFW 2008 - Installing OpenFresco (Kim)
- OPFW 2008 - Architecture & Tcl (Schellenberg)
- OPFW 2008 - Hands-on Exercise (Whyte)

Command Language Manual 2.6.pdf

Open Framework for Experimental Setup and Control (OpenFresco)

OpenFresco Command Language Manual

Andreas Schellenberg, Hong K. Kim, Yoshikazu Takahashi,
Gregory L. Fenves, and Stephen A. Mahin

OpenFresco.exe & OpenFresco.dll
Version 2.6

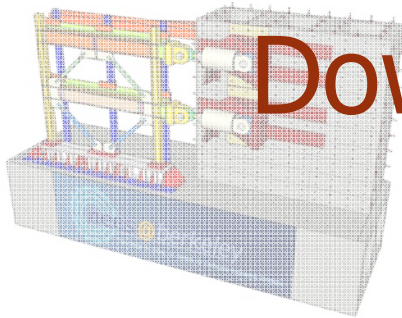
July 2009

Created on 9/26/07

HOME

OpenFresco (the Open-source Framework for Experimental Setup and Control) is an environment-independent software framework, that connects finite element models with control and data acquisition systems in laboratories to facilitate hybrid simulation of structural and geotechnical systems.

Hybrid simulation is an experimental testing technique where a test is executed based on a step-by-step numerical solution of the governing equations of motion for a hybrid model, formulated considering both the numerical and physical portions of a structural system. In order for the earthquake engineering community to take full advantage of this technique, OpenFresco standardizes the deployment of hybrid simulation and extends its capabilities to applications where advanced numerical techniques are utilized, boundary conditions are imposed in real-time, and dynamic loading conditions caused by wind, blast, impact, waves, fire, traffic, and, in particular, seismic events are considered. Accordingly, the architecture of the OpenFresco software package provides a great deal of flexibility, extensibility, and re-usability to the researcher or developer interested in hybrid simulation.



Search

Menu

- [Log in](#)
- [Register](#)

Downloads

- [OpenFresco](#)
- [OpenFrescoExpress](#)
- [OpenSees Navigator](#)



DOCUMENTATION

General Manuals

- [OpenFresco Installation & Getting Started Manual 2.6](#)
- [OpenFresco Command Language Manual 2.6](#)
- [OpenSSL How To](#)

Example Manuals

- [OpenFresco Example Manual 2.6 - LabVIEW](#)
- [OpenFresco Example Manual 2.6 - LS-DYNA](#)
- [OpenFresco Example Manual 2.6 - Matlab](#)
- [OpenFresco Example Manual 2.6 - OpenSSL](#)
- [OpenFresco Example Manual 2.6 - PortalFrame](#)
- [OpenFresco Example Manual 2.6 - SignalFilter](#)
- [OpenFresco Example Manual 2.6 - SimDomain](#)
- [OpenFresco Example Manual 2.6 - SimFEAdapter](#)
- [OpenFresco Example Manual 2.6 - UI-SimCor](#)
- [OpenFresco Example Manual 2.6 - xPCTarget](#)

- [OpenFresco Presentation 2009](#)
- [OPFW 2009 - Architecture & TdJ \(Schellenberg\)](#)
- [OPFW 2009 - Opening Remarks \(Mahin\)](#)
- [OPFW 2009 - Installing OpenFresco \(Kim\)](#)
- [OPFW 2009 - Architecture & TdJ \(Schellenberg\)](#)
- [OPFW 2009 - Hands-on Exercise \(Whyte\)](#)
- [OPFW 2009 - General Comment & Adapter \(Kim, Schellenberg\)](#)
- [OPFW 2009 - Presentation \(Terzic\)](#)

Search

Menu

- [Log in](#)
- [Register](#)

Downloads

- [OpenFresco](#)
- [OpenFrescoExpress](#)
- [OpenSees Navigator](#)



Downloading and Installing OpenFresco

SUBVERSION REPOSITORIES [OPENFRESKO](#)

OpenFresco

calm

English - English

(root)/trunk/SRC - Rev 332

Rev HEAD Go

[Rev 329](#) |
 [Last modification](#) |
 [Compare with Previous](#) |
 [View Log](#) |
 [RSS feed](#)

LAST MODIFICATION

Rev 332 2012-06-15 10:50:44
 Author: aschell
 Log message:
 updated copyright information

Path	Revision	Date	Author	Log	RSS
branches/	288	5896 07h	hongkim	Log	RSS
tags/	206	1042d 09h	aschellenberg	Log	RSS
trunk/	332	12d 11h	aschell	Log	RSS
EXAMPLES/	321	312d 23h	aschellenberg	Log	RSS
GUI/	332	12d 11h	aschell	Log	RSS
MAKES/	314	402d 08h	aschellenberg	Log	RSS
SRC/	332	12d 11h	aschell	Log	RSS
WIN32/	329	159d 00h	aschell	Log	RSS
COPYRIGHT	332	12d 11h	aschell	Log	RSS
Makefile	314	402d 08h	aschellenberg	Log	RSS

[Compare Paths](#)

http://openfresco.berkeley.edu

HOME [USERS](#) [DEVELOPERS](#) [DOCUMENTATION](#) [REFERENCES](#) [COPYRIGHT](#)

HOME

OpenFresco (the Open-source Framework for Experimental Setup and Control) is an environment-independent software framework, that connects finite element models with control and data acquisition systems in laboratories to facilitate hybrid simulation of structural and geotechnical systems.

Hybrid simulation is an experimental testing technique where a test is executed based on a step-by-step numerical solution of the governing equations of motion for a hybrid model, formulated considering both the numerical and physical portions of a structural system. In order for the earthquake engineering community to take full advantage of this technique, OpenFresco standardizes the deployment of hybrid simulation and extends its capabilities to applications where advanced numerical techniques are utilized, boundary conditions are imposed in real-time, and dynamic loading conditions caused by wind, blast, impact, waves, fire, traffic, and, in particular, seismic events are considered. Accordingly, the architecture of the OpenFresco software package provides a great deal of flexibility, extensibility, and re-usability to the researcher or developer interested in hybrid simulation.

Search

Menu

- [Log in](#)
- [Register](#)

Downloads

- [OpenFresco](#)
- [OpenFrescoExpress](#)
- [OpenSees Navigator](#)

PEER

http://openfresco.berkeley.edu/users/openfresco

HOME

OpenFresco (the Open-source Framework for Experimental Setup and Control) is an environment-independent software package that integrates finite element models with control and data acquisition systems to facilitate hybrid simulation of structural and geotechnical systems.

Hybrid simulation is an experimental testing technique where a test is executed based on a step-by-step numerical solution of the governing equations of motion for a hybrid model, formulated considering both the numerical and physical portions of a structural system. In order for the earthquake engineering community to take full advantage of this technique, OpenFresco standardizes the deployment of hybrid simulation and extends its capabilities to applications where advanced numerical techniques are utilized, boundary conditions are imposed in real-time, and dynamic loading conditions caused by wind, blast, impact, waves, fire, traffic, and, in particular, seismic events are considered. Accordingly, the architecture of the OpenFresco software package provides a great deal of flexibility, extensibility, and re-usability to the researcher or developer interested in hybrid simulation.

FE-Software
GenericElement

Experimental Element

Experimental Site

Experimental Setup

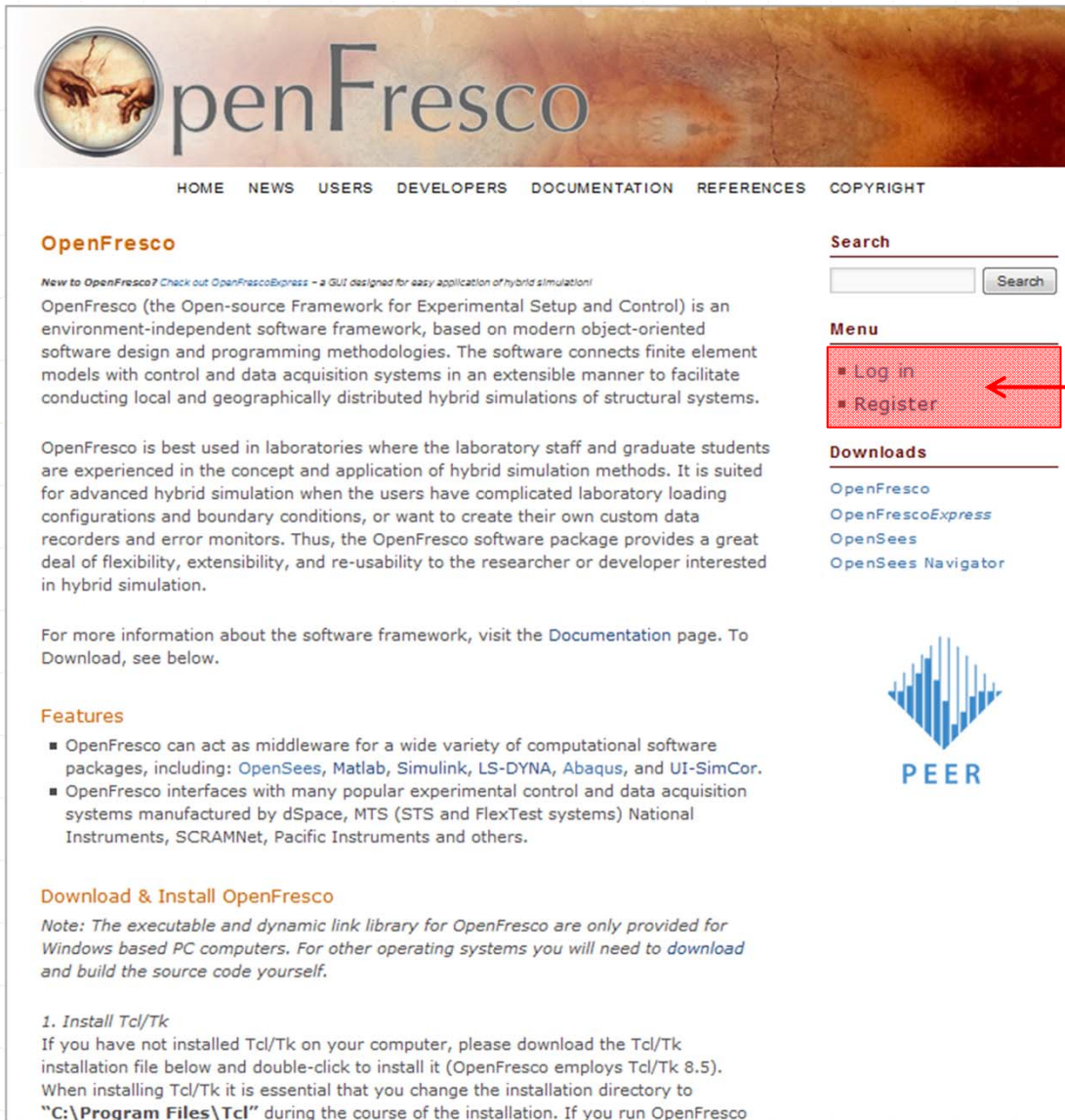
Experimental Control

Control System in Laboratory

PEER

2nd Floor Target
Strain Gages
DC-FPS
DCDT
5 Component Load Cell
1st Floor Target

Login or Register



OpenFresco

New to OpenFresco? Check out OpenFrescoExpress - a GUI designed for easy application of hybrid simulation!

OpenFresco (the Open-source Framework for Experimental Setup and Control) is an environment-independent software framework, based on modern object-oriented software design and programming methodologies. The software connects finite element models with control and data acquisition systems in an extensible manner to facilitate conducting local and geographically distributed hybrid simulations of structural systems.

OpenFresco is best used in laboratories where the laboratory staff and graduate students are experienced in the concept and application of hybrid simulation methods. It is suited for advanced hybrid simulation when the users have complicated laboratory loading configurations and boundary conditions, or want to create their own custom data recorders and error monitors. Thus, the OpenFresco software package provides a great deal of flexibility, extensibility, and re-usability to the researcher or developer interested in hybrid simulation.

For more information about the software framework, visit the [Documentation](#) page. To Download, see below.

Features

- OpenFresco can act as middleware for a wide variety of computational software packages, including: [OpenSees](#), [Matlab](#), [Simulink](#), [LS-DYNA](#), [Abaqus](#), and [UI-SimCor](#).
- OpenFresco interfaces with many popular experimental control and data acquisition systems manufactured by [dSpace](#), [MTS \(STS and FlexTest systems\)](#) National Instruments, [SCRAMNet](#), [Pacific Instruments](#) and others.

Download & Install OpenFresco

Note: The executable and dynamic link library for OpenFresco are only provided for Windows based PC computers. For other operating systems you will need to download and build the source code yourself.

1. Install Tcl/Tk

If you have not installed Tcl/Tk on your computer, please download the Tcl/Tk installation file below and double-click to install it (OpenFresco employs Tcl/Tk 8.5). When installing Tcl/Tk it is essential that you change the installation directory to "C:\Program Files\Tcl" during the course of the installation. If you run OpenFresco

First, log in if you already have an account or register for a new account

Download Files

Download & Install OpenFresco

Note: The executable and dynamic link library for OpenFresco are only provided for Windows based PC computers. For other operating systems you will need to download and build the source code yourself.

1. Install Tcl/Tk

If you have not installed Tcl/Tk on your computer, please download the Tcl/Tk installation file below and double-click to install it (OpenFresco employs Tcl/Tk 8.5).

When installing Tcl/Tk it is essential that you change the installation directory to "C:\Program Files\Tcl" during the course of the installation. If you run OpenFresco and you see an error message to the effect, "Cannot find tcl85.dll", you have skipped this step and must reinstall Tcl/Tk. Note that you will probably have to uninstall the version you just installed first. [Download Tcl/Tk 8.5](#)

2. Install OpenSSL (optional)

If you would like to use OpenFresco's capability to encrypt signals during geographically distributed hybrid simulations through Secure Socket Layer, please download the OpenSSL installation file from the website below and install it (OpenFresco employs the full (not light) version of OpenSSL 1.0). When installing OpenSSL it is essential that you change the installation directory to "C:\Program Files\OpenSSL" during the course of the installation. [Download OpenSSL](#)

3. Install Computational Driver

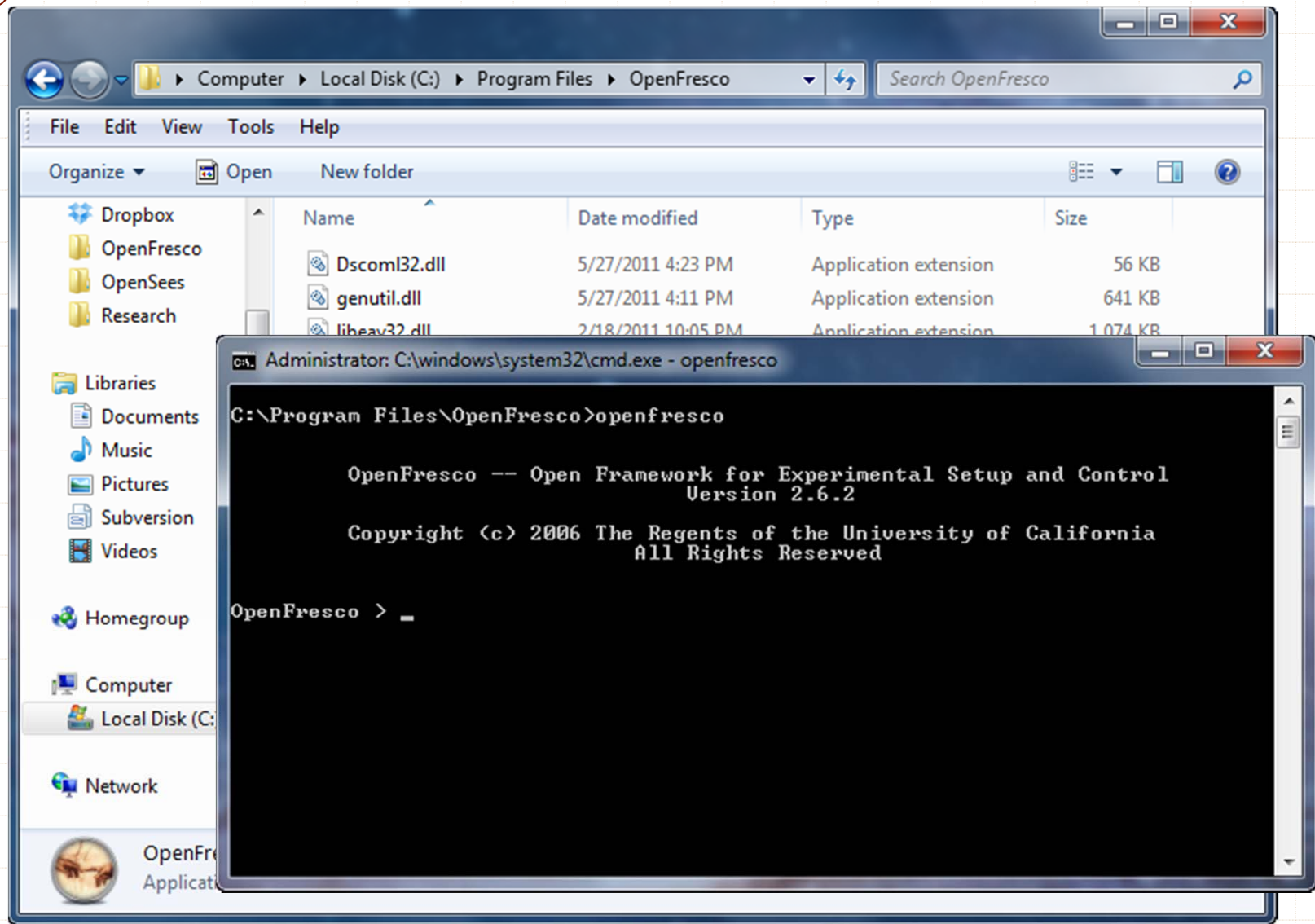
Install the structural analysis software package of your choice. OpenFresco can interface with many software packages including OpenSees, Abaqus, LS-DYNA, Matlab, Simulink, UI-SimCor, ANSYS (coming soon!)

4. Download & Extract OpenFresco

 [Download OpenFresco 2.6.2](#) (.zip file, 5 MB)

1. Download and install Tcl/Tk 8.5
Important: install in C:\Program Files\Tcl
2. Download and install OpenSSL (optional)
3. Install OpenSees
4. Download and extract OpenFresco in a convenient directory

Run OpenFresco



openfresco.berkeley.edu/developers/svn



openFresco

HOME USERS **DEVELOPERS** DOCUMENTATION REFERENCES COPYRIGHT

SVN

The OpenFresco source code is stored using the Apache Subversion (SVN) software. SVN provides the means to store not only the current version of a piece of source code, but a record of all changes that have occurred to that source code over time and a record of who made those changes. The use of SVN is particularly common for software projects with multiple developers, because SVN guarantees that changes made by one developer are not accidentally removed when another developer commits changes to the source code. For the OpenFresco software project anyone can check out the code via anonymous SVN access, but only trusted developers have the ability to commit changes and additions to the code repository.

Search

Menu

- Log in
- Register

Downloads

Getting the Code

To download the OpenFresco source code from the repository you need to have SVN installed on your local machine first. You can download SVN for all major operating systems including Linux, Windows, and MacOSX. If you are working on Windows, TortoiseSVN is particularly nice and easy to use. It lets you control SVN functions directly from the menus as you navigate the file system in Windows Explorer.

Once you have SVN installed, you can download the OpenFresco source code using the following command:

```
svn co svn://openfresco.berkeley.edu/usr/local/svn/
```

The checkout command makes a local copy of the entire OpenFresco source code into your current working directory. By requesting `svn co svn://openfresco.berkeley.edu/trunk` you get the development trunk, which should have the latest stable source code.

Browsing the Code

You can browse the source code online using WebSVN. The application provides a real-time view onto the OpenFresco repository that has been developed using the Subversion methodology. You can view the log of any file or directory, and you can view the files changed, added or deleted in any given revision. You can also view differences between two versions of a file so as to see exactly what changed in that particular revision.

SUBVERSION REPOSITORIES OPENFRESCO

OpenFresco | calm | English - English

(root)/trunk/SRC - Rev 332

Rev HEAD Go

Rev 329 | Last modification | Compare with Previous | View Log | RSS feed

LAST MODIFICATION

Rev 332 2012-06-15 10:50:44

Author: aschell

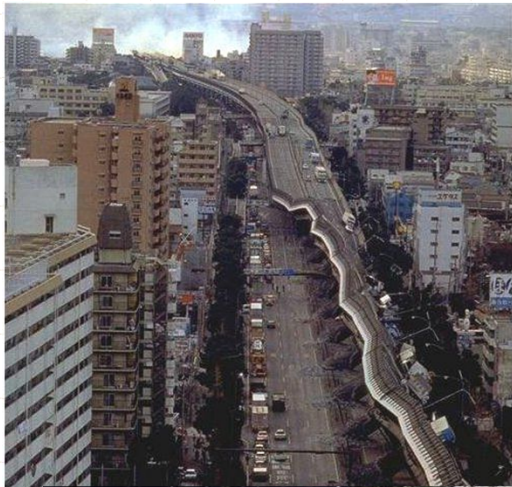
Log message:
updated copyright information

Path	Last modification	Log	RSS
branches/	288 589d 07h hongkim	Log	RSS
tags/	206 1042d 09h aschellenberg	Log	RSS
trunk/	332 12d 11h aschell	Log	RSS
EXAMPLES/	321 312d 23h aschellenberg	Log	RSS
GUI/	332 12d 11h aschell	Log	RSS
MAKES/	314 402d 08h aschellenberg	Log	RSS
SRC/	332 12d 11h aschell	Log	RSS
WIN32/	329 159d 00h aschell	Log	RSS
COPYRIGHT	332 12d 11h aschell	Log	RSS
Makefile	314 402d 08h aschellenberg	Log	RSS

Compare Paths

Powered by WebSVN [trunk] and Subversion 1.6.11 | XHTML & CSS

HS of Structural Collapse



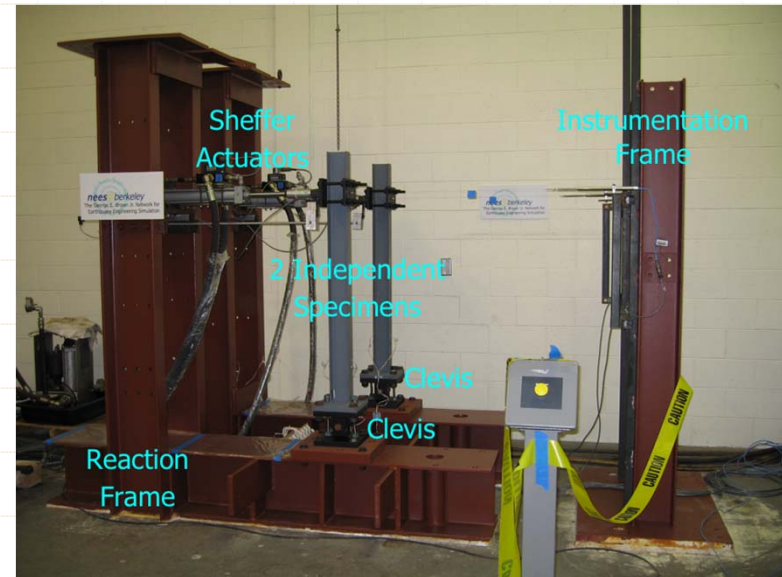
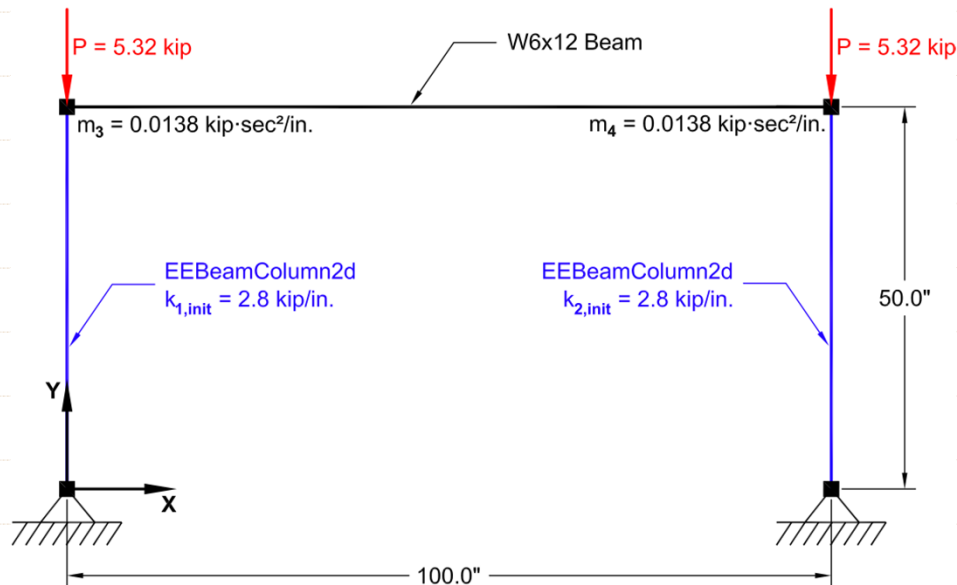
Structural Collapse

- ✦ On shaking tables, simulation of collapse is dangerous and expensive
- ✦ In hybrid simulations
 - Gravity loads and resulting geometric nonlinearities can be modeled analytically
 - ◆ Therefore, no complex active or passive gravity load setups are necessary
 - Actuator movements will limit displacements during collapse (safety)
 - ◆ Thus, there is no need to protect expensive test equipment from specimen impact
 - Only critical, collapse-sensitive elements of a structure need to be physically modeled

Implementation in a Hybrid Model

- ★ Physical portion of the model:
 - Test material and cross-section level response
- ★ Analytical portion of the model:
 - Apply the gravity and/or prestress loads
 - Provide the geometric transformations such that the second-order effects due to axial loads are accounted for
 - Model the rest of the structure

Structural Collapse of Portal Frame



Properties of Model:

- NDOF = 8 (4 with mass)
- Period: $T_1 = 0.49$ sec
- Damping: $\zeta_1 = 0.05$
- $P = 50\%$ of ϕP_n
- Crd-Trans: P-Delta, Corotational
- ExpElements: EEBeamColumn2d
- ExpSetups: ESOOneActuator
- ExpControl: ECxPtarget
- SACNF01: $p_{ga} = 0.906g$

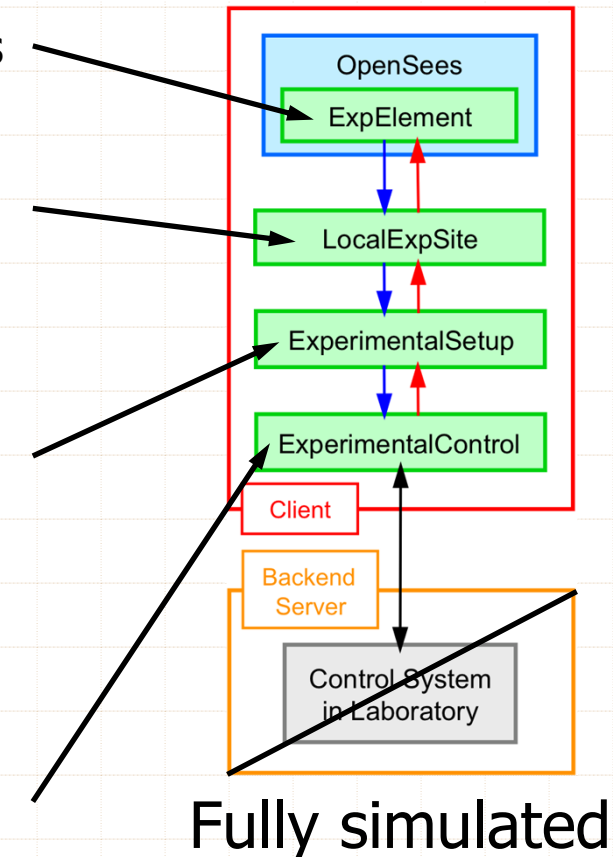
OpenFresco Local Architecture

Exp. beamColumn element defined in OpenSees

Communication methods for distributed testing.
In this case, we are using a local site.

Transforms between experimental element DOFs in OPF and the actuator DOFs in the laboratory. Linear and non-linear transformations are available.

Interface with control and data acquisition systems. In this example, SimUniaxialMaterials will simulate the response of the experimental element using a material defined in OpenSees, Steel02



Tcl File Components

Geometry

Materials

Experimental Control

Experimental Setup

Experimental Site

Geometric Transformation

Experimental Elements

Numerical Elements

Gravity Loads

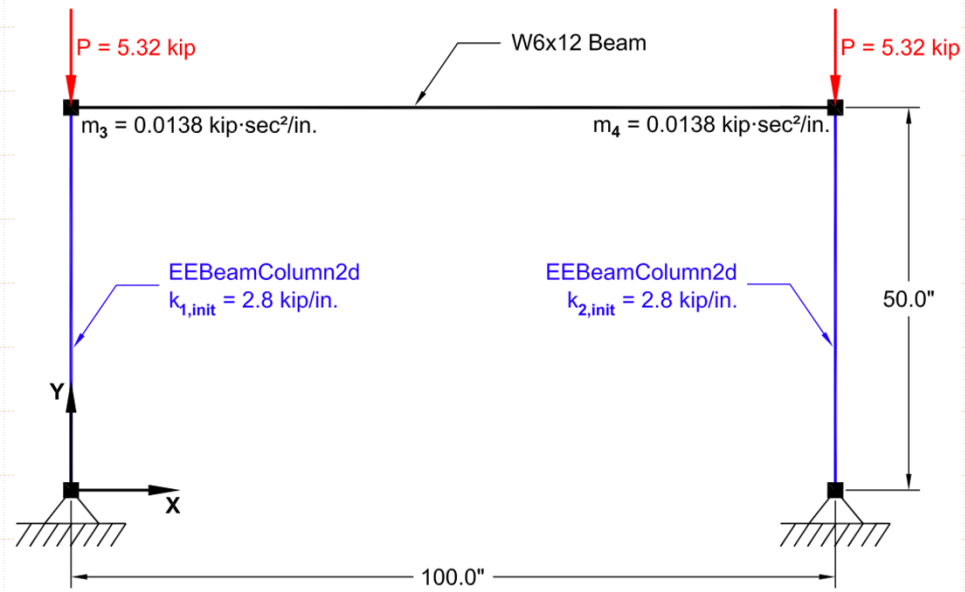
Gravity Analysis

Dynamic Loads

Dynamic Analysis

Portal Frame Model

```
19 # -----
20 # Start of model generation
21 # -----
22 # create ModelBuilder (with two-dimensions and 3 DOF/node)
23 model BasicBuilder -ndm 2 -ndf 3
24
25 # Load OpenFresco package
26 # -----
27 # (make sure all dlls are in the same folder as openSees.exe)
28 loadPackage OpenFresco
```

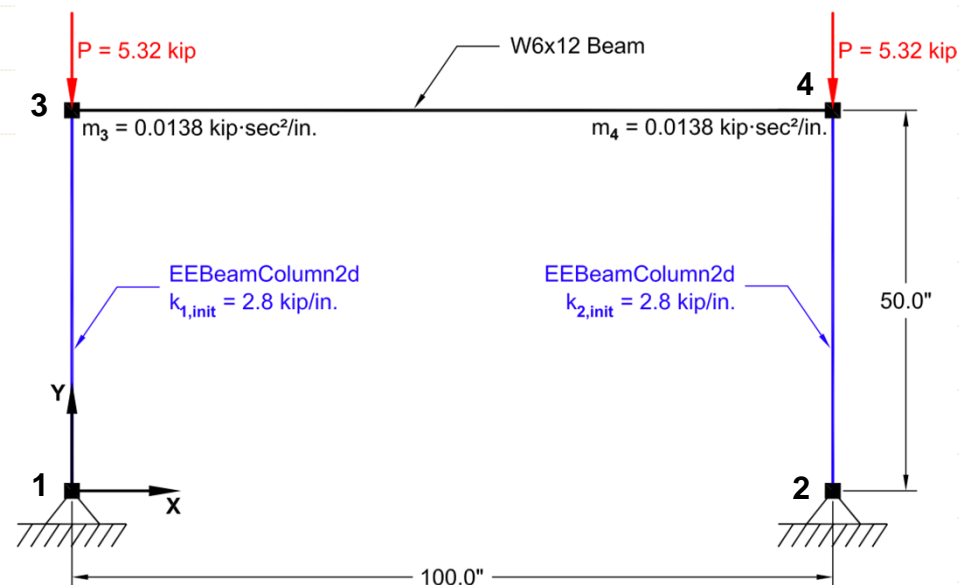


Geometry

```

30 # Define geometry for model
31 # -----
32 set withGravity 1;
33 set Pc 10.638;
34 set P [expr 0.5*$Pc];
35 set mass3 [expr $P/386.1];
36 set mass4 [expr $P/386.1];
37 # node $tag $xCrd $yCrd $mass
38 node 1 0.0 0.0
39 node 2 100.0 0.0
40 node 3 0.0 50.0 -mass $mass3 $mass3 0.0
41 node 4 100.0 50.0 -mass $mass4 $mass4 0.0
42
43 # set the boundary conditions
44 # fix $tag $DX $DY $RZ
45 fix 1 1 1 0
46 fix 2 1 1 0

```



- ✦ withGravity 0: turns off gravity loads (no P-delta)
- ✦ withGravity 1: turns on gravity loads (P-delta)
- ✦ Assigned no rotational mass – must use implicit integration method

Tcl File Components

Geometry

Materials

Experimental Control

Experimental Setup

Experimental Site

Geometric Transformation

Experimental Elements

Numerical Elements

Gravity Loads

Gravity Analysis

Dynamic Loads

Dynamic Analysis

Materials/Experimental Control

```
48 # Define materials
49 # -----
50 # uniaxialMaterial Steel02 $matTag $Fy $E $b $R0 $cR1 $cR2 $a1 $a2 $a3 $a4
51 uniaxialMaterial Steel02 1 1.5 2.8 0.01 18.5 0.925 0.15 0.0 1.0 0.0 1.0
52 #uniaxialMaterial Elastic 1 2.8
53
54 # Define experimental control
55 # -----
56 # expControl SimUniaxialMaterials $stag $matTags
57 expControl SimUniaxialMaterials 1 1
58 expControl SimUniaxialMaterials 2 1
```

Column 1 (left)
Column 2 (right)

- ★ Want to control two columns
- ★ SimUniaxialMaterials used to simulate a specimen
- ★ Need to create a separate experimental control for each element so create experimental control with tags "1" and "2"
- ★ Assign a material tag to each

Tcl File Components

Geometry

Materials

Experimental Control

Experimental Setup

Experimental Site

Geometric Transformation

Experimental Elements

Numerical Elements

Gravity Loads

Gravity Analysis

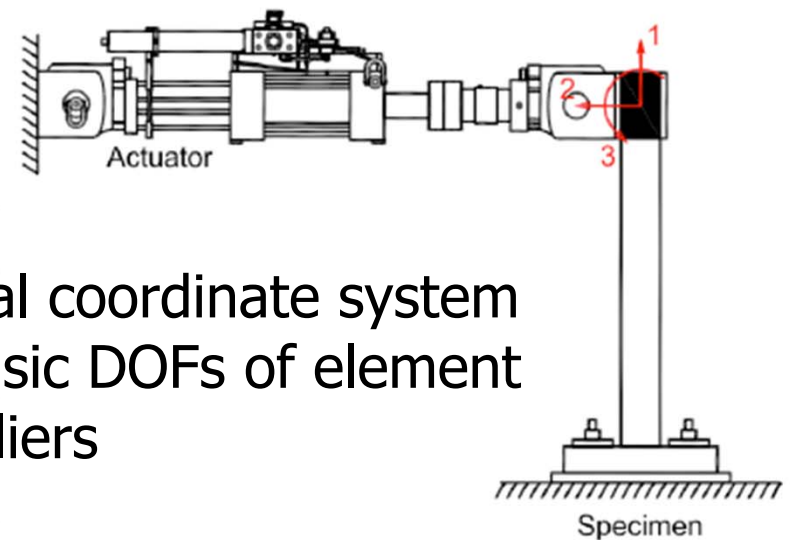
Dynamic Loads

Dynamic Analysis

Experimental Setup and Site

```
61 # Define experimental setup
62 # -----
63 # expSetup OneActuator $tag <-control $ctrlTag> $dir -sizeTrialOut $t $o <-trialDispFact $f>
64 expSetup OneActuator 1 -control 1 2 -sizeTrialOut 3 3
65 expSetup OneActuator 2 -control 2 2 -sizeTrialOut 3 3
66
67 # Define experimental site
68 # -----
69 # expSite LocalSite $tag $setupTag
70 expSite LocalSite 1 1
71 expSite LocalSite 2 2
```

Refers back to SimUniaxialMaterials Experimental Control
Left and right columns (tags 1 and 2)



- ★ OneActuator direction in element's local coordinate system
- ★ sizeTrialOut are equal to number of basic DOFs of element
- ★ Optional Factors: all factors are multipliers
- ★ e.g.
- ★ -ctrlDispFact 2 = target displacement x 2
- ★ -daqDispFact 0.5 = measured displacement x 0.5

Tcl File Components

Geometry

Materials

Experimental Control

Experimental Setup

Experimental Site

Geometric Transformation

Experimental Elements

Numerical Elements

Gravity Loads

Gravity Analysis

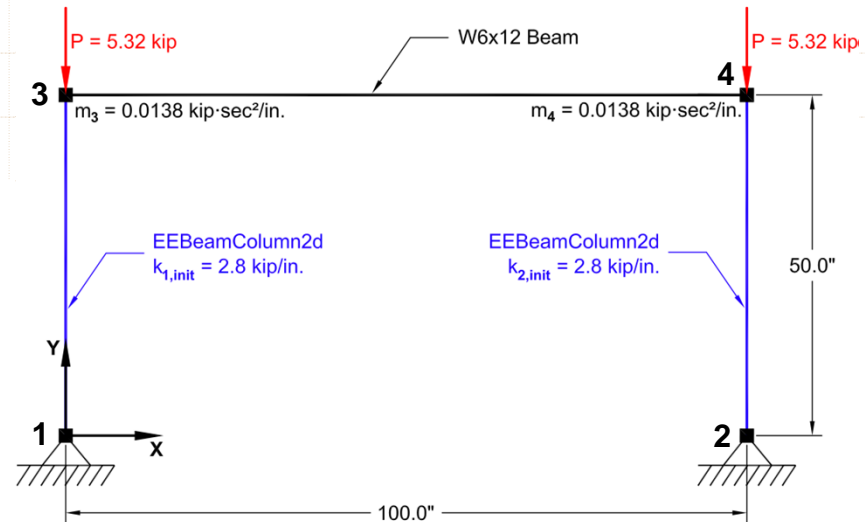
Dynamic Loads

Dynamic Analysis

Geometric Transf & Exp. Elements

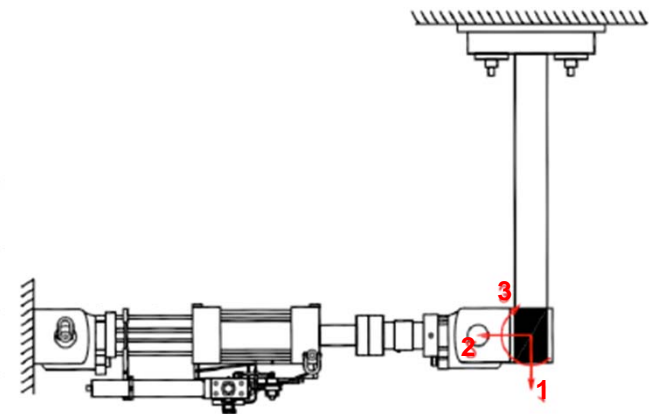
```

73 # Define geometric transformation
74 # -----
75 #geomTransf PDelta 1
76 geomTransf Corotational 1
77
78 # Define experimental elements
79 # -----
80 # left and right columns
81 # expElement beamColumn $eleTag $iNode $jNode $transTag -site $siteTag -initStif $Kij <-iMod
82 expElement beamColumn 1 3 1 1 -site 1 -initStif 1310.8 0 0 0 11.2 -280.0 0 -280.0 9333.3333
83 expElement beamColumn 2 4 2 1 -site 2 -initStif 1310.8 0 0 0 11.2 -280.0 0 -280.0 9333.3333
84
85 # Define numerical elements
86 # -----
87 # element elasticBeamColumn $eleTag $iNode $jNode $A $E
88 element elasticBeamColumn 3 3 4 3.55 29000 22.1 1
    
```



$$K_{init} = \begin{bmatrix} \frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} = \begin{bmatrix} 1310.8 & 0 & 0 \\ 0 & 11.2 & -280.0 \\ 0 & -280.0 & 9333.3 \end{bmatrix}$$

- ✦ L=50"
- ✦ A=2.26 in²
- ✦ I=4.02 in⁴
- ✦ E=29000 ksi



Tcl File Components

Geometry
Materials
Experimental Control
Experimental Setup
Experimental Site
Geometric Transformation
Experimental Elements
Numerical Elements
Gravity Loads
Gravity Analysis
Dynamic Loads
Dynamic Analysis

Gravity Loads

```
90  if {$withGravity} {
91      # Define gravity loads
92      # -----
93      # Create a Plain load pattern with a Linear TimeSeries
94      pattern Plain 1 "Linear" {
95          # Create nodal loads at nodes 2
96          #   nd      FX          FY  MZ
97          load 3    0.0  [expr -$P] 0.0
98          load 4    0.0  [expr -$P] 0.0
99      }
100     # -----
101     # End of model generation
102     # -----
```

Loads in the negative Y-direction at nodes 3 and 4

Gravity Analysis Options

```
105 # -----  
106 # Start of analysis generation  
107 # -----  
108 # Create the system of equation  
109 system BandGeneral  
110 # Create the DOF numberer  
111 numberer Plain  
112 # Create the constraint handler  
113 constraints Plain  
114 # Create the convergence test  
115 test EnergyIncr 1.0e-6 10  
116 # Create the integration scheme  
117 integrator LoadControl 0.1  
118 # Create the solution algorithm  
119 algorithm Newton  
120 # Create the analysis object  
121 analysis Static  
122 # -----  
123 # End of analysis generation  
124 # -----
```

Banded General SOE



DOFs assigned arbitrarily (ok for small models)

Only using homogeneous single point constraints

Test EnergyIncr \$tol \$maxNumIter

Load Control with 10 steps

Newton-Raphson algorithm

Perform a static analysis

Gravity Recorders and Analysis

```
127 # -----
128 # Start of recorder generation
129 # -----
130 # create a Recorder object for the nodal displacements at node 2
131 recorder Node -file Gravity_Dsp.out -time -node 3 4 -dof 1 2 3 disp
132 recorder Element -file Gravity_Frc.out -time -ele 1 2 3 force
133 # -----
134 # End of recorder generation
135 # -----
136
137
138 # -----
139 # Perform the gravity analysis
140 # -----
141 # perform the gravity load analysis, requires 10 steps to reach the load level
142 if {[analyze 10] == 0} {
143     puts "\nGravity load analysis completed"
144 } else {
145     puts "\nGravity load analysis failed"
146     exit -1
147 }
```

Tcl File Components

Geometry
Materials
Experimental Control
Experimental Setup
Experimental Site
Geometric Transformation
Experimental Elements
Numerical Elements
Gravity Loads
Gravity Analysis
Dynamic Loads
Dynamic Analysis

Dynamic Loads

```
158 # Define dynamic loads
159 # -----
160 # set time series to be passed to uniform excitation
161 set dt 0.01
162 set scale 1.2
163 timeSeries Path 1 -filePath SACNF01.txt -dt $dt -factor [expr 386.1*$scale]
164
165 # create UniformExcitation load pattern
166 # pattern UniformExcitation $tag $dir -accel $tag <-vel0 $vel0>
167 pattern UniformExcitation 2 1 -accel 1
168
169 # calculate the rayleigh damping factors for nodes & elements
170 set alphaM 1.2797; # D = alphaM*M
171 set betaK 0.0; # D = betaK*Kcurrent
172 set betaKinit 0.0; # D = betaKinit*Kinit
173 set betaKcomm 0.0; # D = betaKcomm*KlastCommit
174
175 # set the rayleigh damping
176 rayleigh $alphaM $betaK $betaKinit $betaKcomm
177 # -----
178 # End of model generation
179 # -----
```

Place ground motion file in the same folder as the PortalFrame_Local.tcl file

Dynamic Analysis Options

```
182 # -----
183 # Start of analysis generation
184 # -----
185 # create the system of equations
186 system BandGeneral
187
188 # create the DOF numberer
189 numberer Plain
190
191 # create the constraint handler
192 constraints Plain
193
194 # create the convergence test
195 test FixedNumIter 5
196
197 # create the integration scheme
198 integrator NewmarkHSFixedNumIter 0.5 0.25
199
200 # create the solution algorithm
201 algorithm Newton
202
203 # create the analysis object
204 analysis Transient
205 # -----
206 # End of analysis generation
207 # -----
```

} Same as gravity analysis

5 iterations/time-step

NewmarkHSFixedNumIter: implicit
Newmark method with 5
iterations/time-step

$\gamma=0.5$: second order accuracy, no
numerical damping

$\beta=0.25$: average acceleration,
unconditional stability

Dynamic Recorders

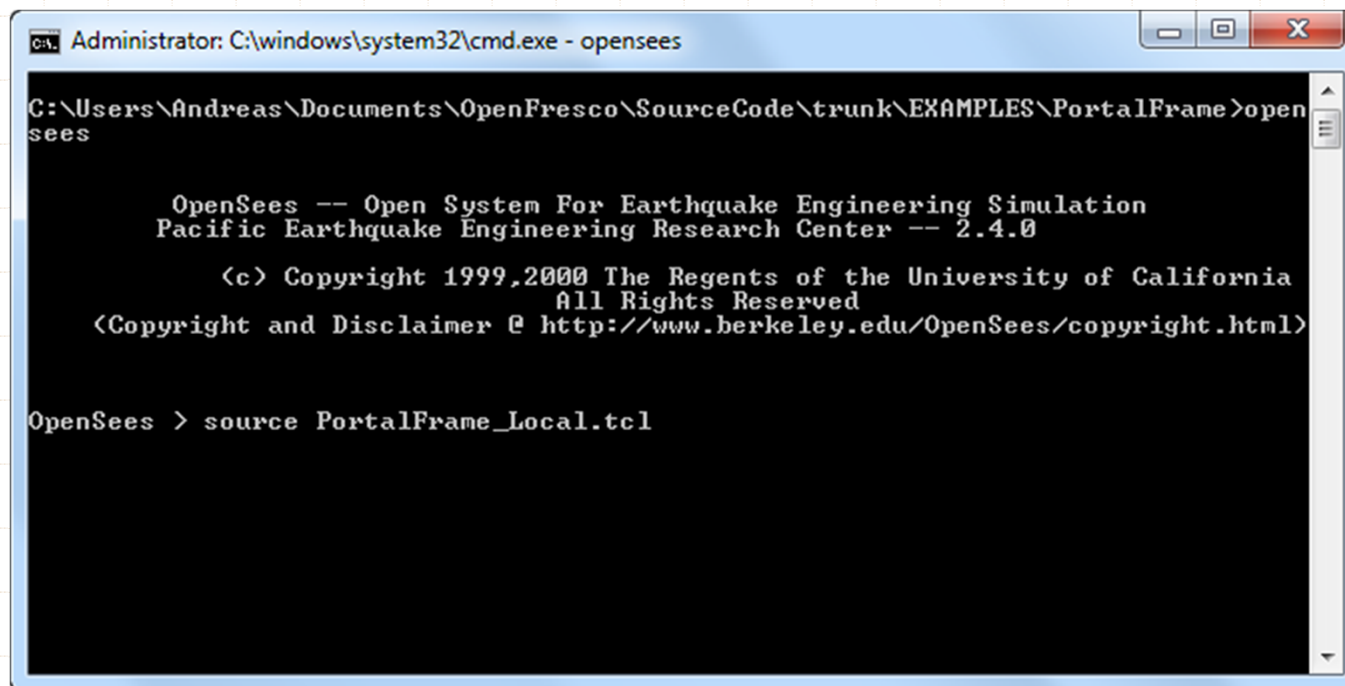
```
218 # -----
219 # Start of recorder generation
220 # -----
221 # create the recorder objects
222 recorder Node -file Node_Dsp.out -time -node      3 4 -dof 1 2 3 disp
223 recorder Node -file Node_Vel.out -time -node      3 4 -dof 1 2 3 vel
224 recorder Node -file Node_Acc.out -time -node      3 4 -dof 1 2 3 accel
225 recorder Node -file Node_Rxn.out -time -node 1 2 3 4 -dof 1 2 3 reactionIncludingInertia
226
227 recorder Element -file Elmt_glbFrc.out -time -ele 1 2 3 forces
228 expRecorder Control -file Control_ctrlDsp.out -time -control 1 2 ctrlDisp
229 expRecorder Control -file Control_daqDsp.out -time -control 1 2 daqDisp
230 expRecorder Control -file Control_daqFrc.out -time -control 1 2 daqForce
231 # -----
232 # End of recorder generation
233 # -----
```

Dynamic Analysis

```
236 # -----
237 # Finally perform the analysis
238 # -----
239 # perform an eigenvalue analysis
240 set pi [expr acos(-1.0)]
241 set lambda [eigen -fullGenLapack 4]
242 puts "\nEigenvalues at start of transient:"
243 puts "|  lambda  |  omega  |  period  |  frequency  |"
244 foreach lambda $lambda {
245     set omega [expr pow($lambda,0.5)]
246     set period [expr 2.0*$pi/$omega]
247     set frequ [expr 1.0/$period]
248     puts [format "| %5.3e | %8.4f | %7.4f | %9.4f |" $lambda $omega $period $frequ]
249 }
250
251 # open output file for writing
252 set outFileID [open elapsedTime.txt w]
253 # perform the transient analysis
254 set tTot [time {
255     for {set i 1} {$i < 2500} {incr i} {
256         set t [time {analyze 1 [expr $dt/1.0]}]
257         puts $outFileID $t
258         #puts "step $i"
259     }
260 }]
```

Running the Hybrid Simulation

- ★ Start the OpenSees executable file from the directory where you saved PortalFrame_Local.tcl
- ★ At the prompt, type source PortalFrame_Local.tcl and press enter



```
Administrator: C:\windows\system32\cmd.exe - opensees
C:\Users\Andreas\Documents\OpenFresco\SourceCode\trunk\EXAMPLES\PortalFrame>open
sees

OpenSees -- Open System For Earthquake Engineering Simulation
Pacific Earthquake Engineering Research Center -- 2.4.0

(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 PortalFrame_Local.tcl
```

Run Simulation

```
Administrator: C:\windows\system32\cmd.exe

OpenSees > source PortalFrame_Local.tcl

      OpenFresco -- Open Framework for Experimental Setup and Control
                    Version 2.6.2

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

WARNING EEBeamColumn2d::getTangentStiff() - Element: 1
TangentStiff cannot be calculated.
Return InitialStiff including GeometricStiff instead.
Subsequent getTangentStiff warnings will be suppressed.

WARNING EEBeamColumn2d::getTangentStiff() - Element: 2
TangentStiff cannot be calculated.
Return InitialStiff including GeometricStiff instead.
Subsequent getTangentStiff warnings will be suppressed.

Gravity load analysis completed















WARNING: NewmarkHSFixedNumIter::domainChanged() - assuming Ut-1 = Ut

Eigenvalues at start of transient:
|  lambda  |  omega  |  period  |  frequency  |
| 1.639e+002 | 12.8040 | 0.4907 | 2.0378 |
| 9.515e+004 | 308.4630 | 0.0204 | 49.0934 |
| 9.532e+004 | 308.7421 | 0.0204 | 49.1378 |
| 1.496e+005 | 386.7612 | 0.0162 | 61.5550 |

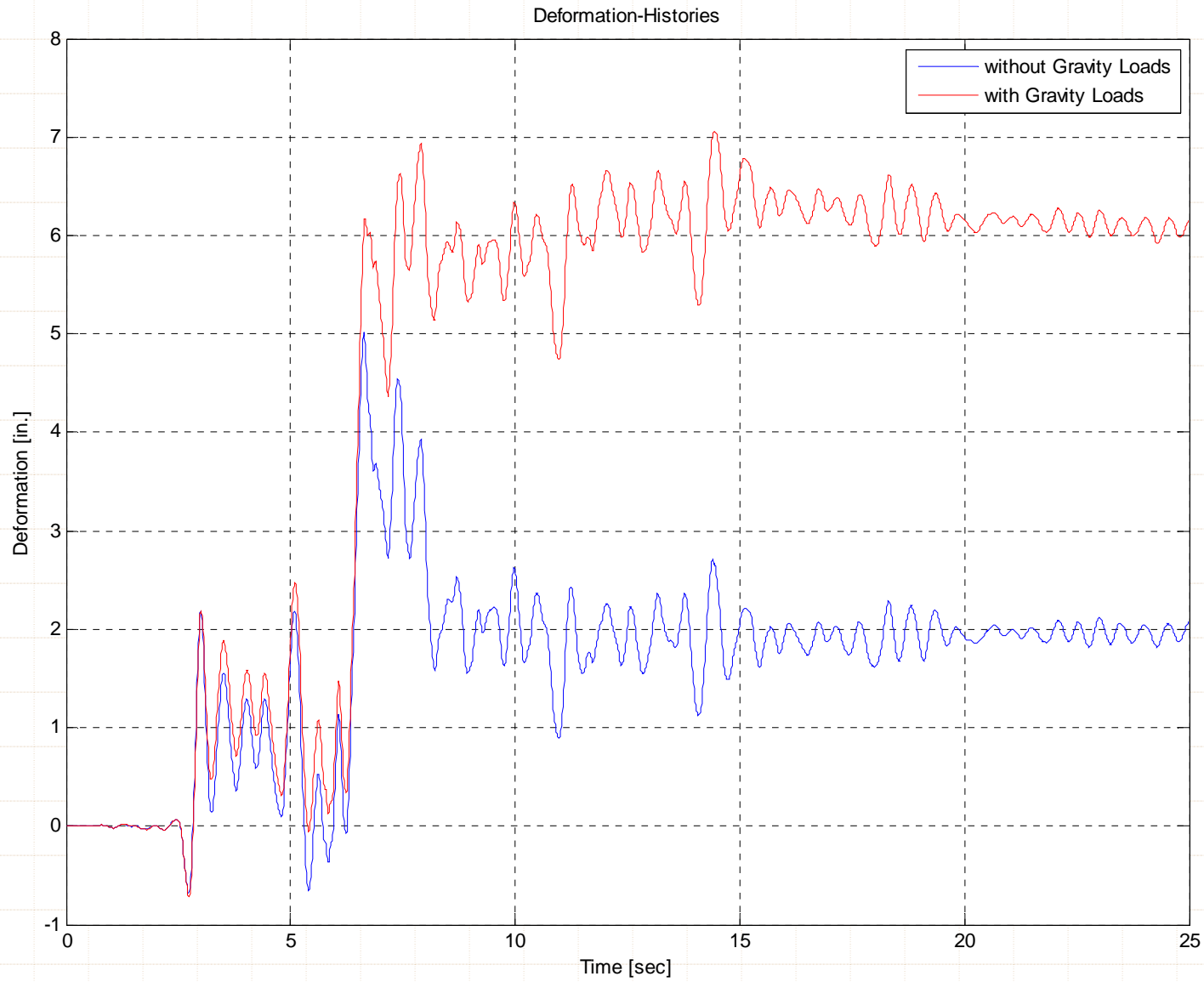
Elapsed Time = 815445 microseconds per iteration

C:\Users\Andreas\Documents\OpenFresco\SourceCode\trunk\EXAMPLES\PortalFrame>
```

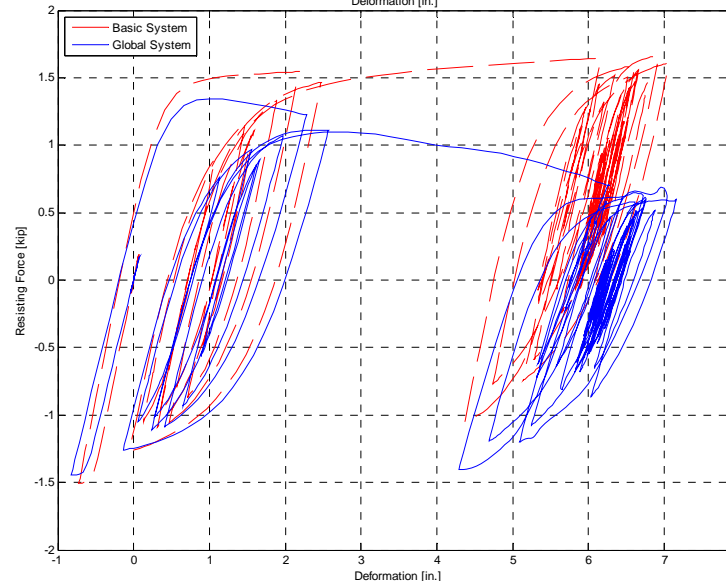
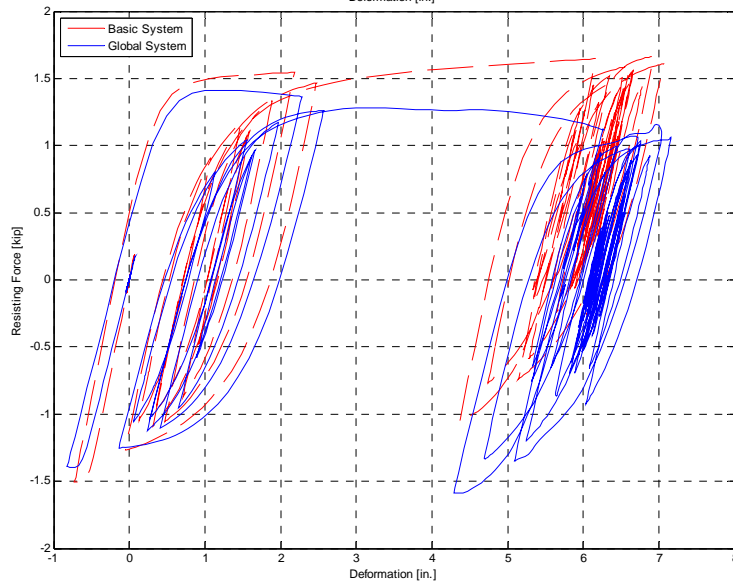
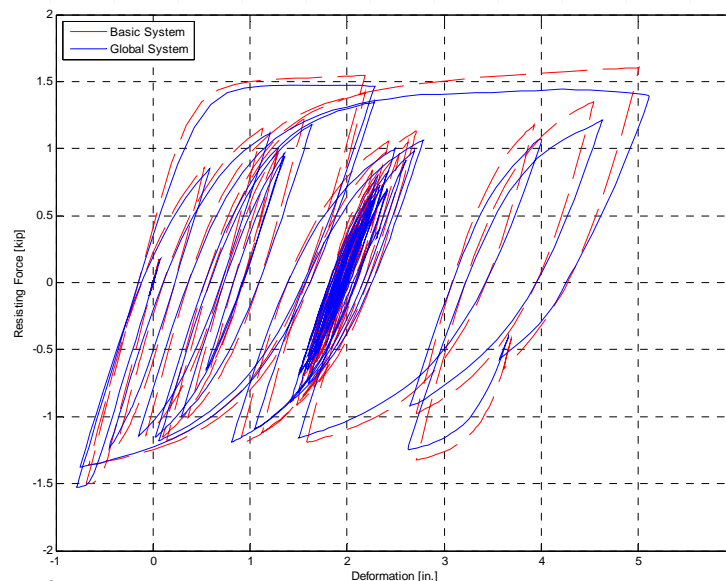
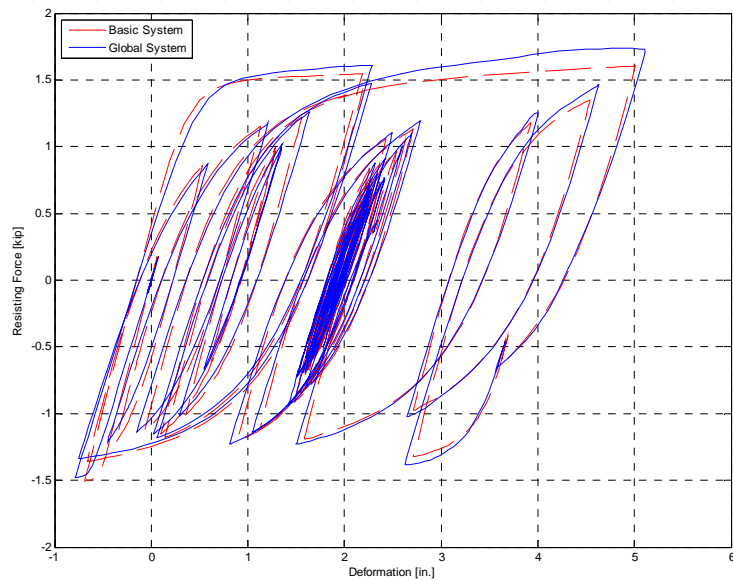

Recorders Save Output Files

Name	Date modified	Type	Size
 PortalFrame_Local.tcl	3/6/2013 7:20 PM	TCL File	9 KB
 Node_Vel.out	3/6/2013 8:50 PM	OUT File	164 KB
 Node_Rxn.out	3/6/2013 8:50 PM	OUT File	286 KB
 Node_Dsp.out	3/6/2013 8:50 PM	OUT File	168 KB
 Node_Acc.out	3/6/2013 8:50 PM	OUT File	144 KB
 Gravity_Frc.out	3/6/2013 8:50 PM	OUT File	2 KB
 Gravity_Dsp.out	3/6/2013 8:50 PM	OUT File	1 KB
 Elmt_glbFrc.out	3/6/2013 8:50 PM	OUT File	381 KB
 Control_daqFrc.out	3/6/2013 8:50 PM	OUT File	63 KB
 Control_daqDsp.out	3/6/2013 8:50 PM	OUT File	61 KB
 Control_ctrlDsp.out	3/6/2013 8:50 PM	OUT File	61 KB
 SACNF01.txt	10/13/2010 10:47 ...	Notepad++ Document	78 KB
 elapsedTime.txt	3/6/2013 8:50 PM	Notepad++ Document	79 KB
 PlotOutput.m	3/6/2013 8:59 PM	MATLAB Code	9 KB

Compare Deformations

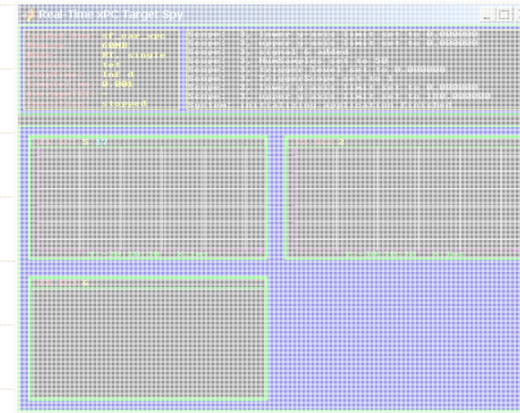
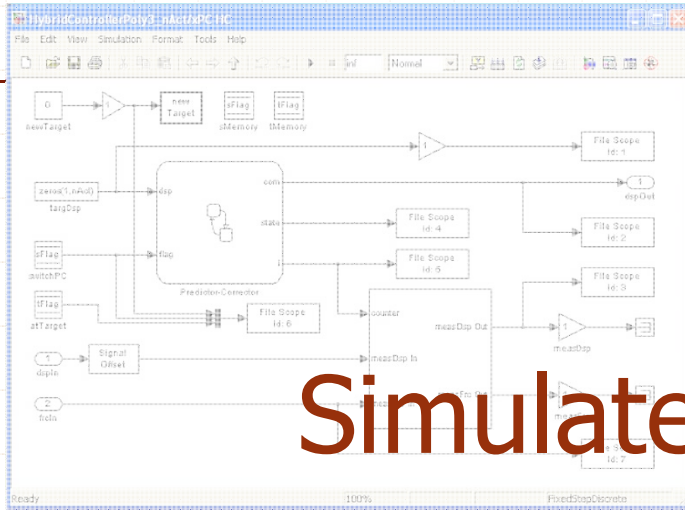


Compare Hysteresis Loops



w/o Gravity

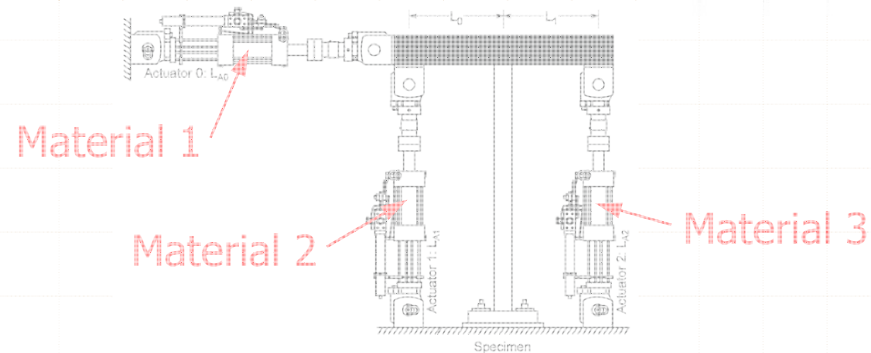
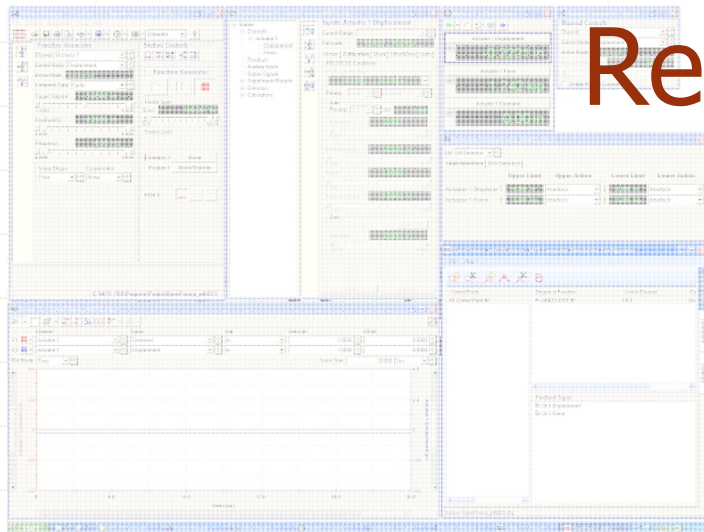
with Gravity



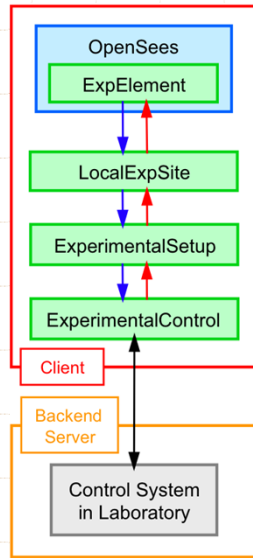
Simulated Controllers VS. Real Controllers



Sysstran



Connecting to MTS 493 controller



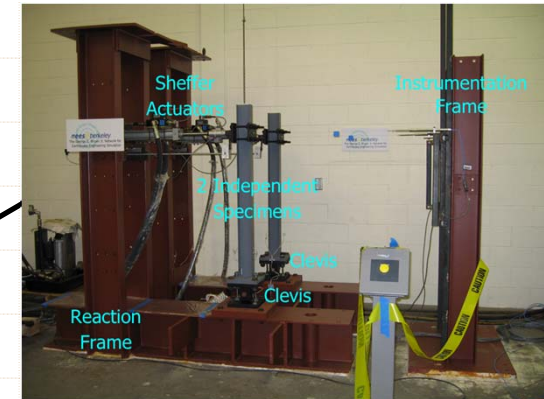
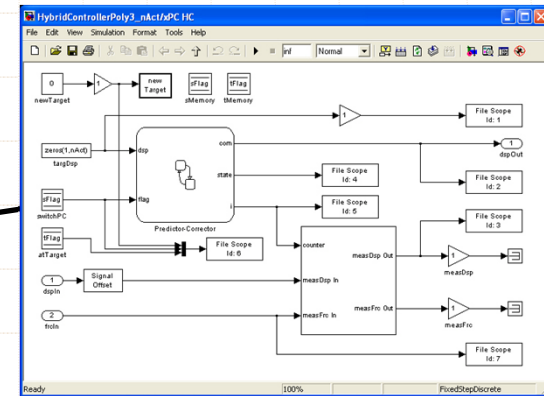
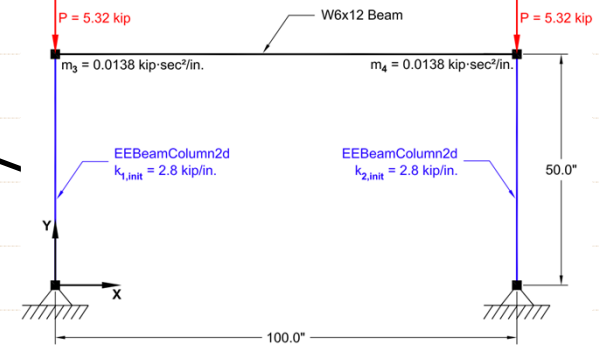
OpenSees Finite Element Model

OpenFresco Middleware

xPC-Target real-time Predictor-Corrector

MTS 493 real-time Controller

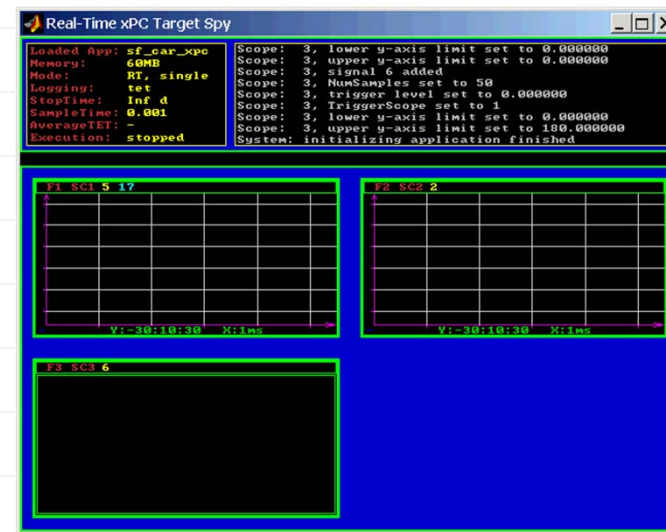
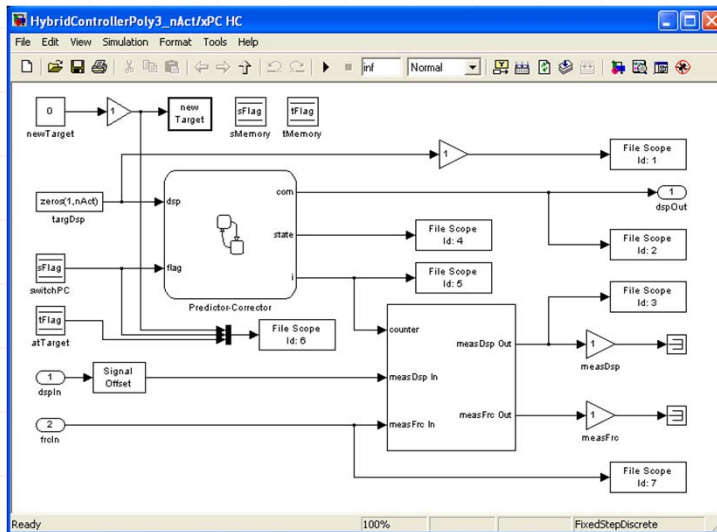
Physical Specimen in μ NEES Lab

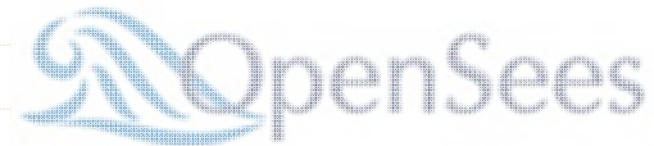
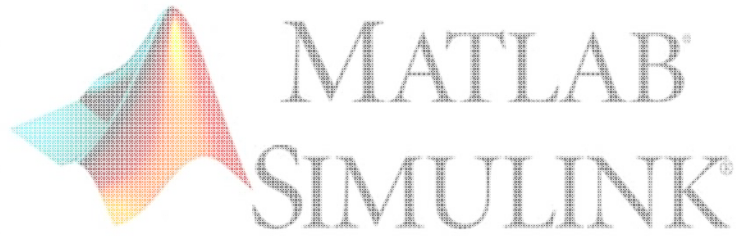


Modify Experimental Control

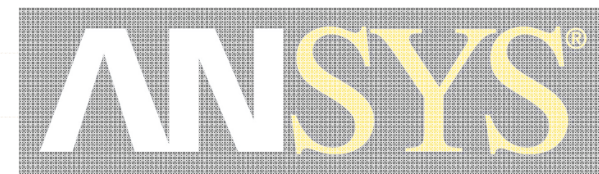
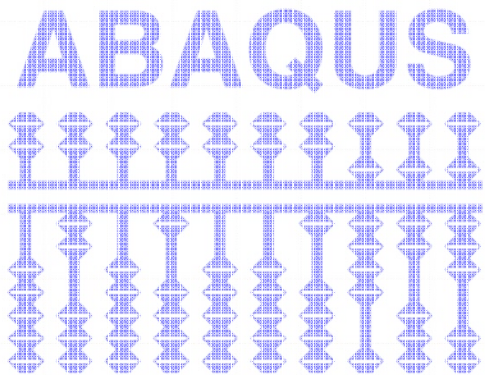
```
54 # Define experimental control
55 # -----
56 # expControl SimUniAxialMaterials $stag $matTags
57 #expControl SimUniAxialMaterials 1 1
58 expControl xPCtarget 1 1 "192.168.2.20" 22222 HybridControllerD3D3_1Act "D:/PredictorCorrector/
59 expControl SimUniAxialMaterials 2 1
```

- ✦ Want to control two columns
- ✦ xPCtarget used for left column
- ✦ SimUniAxialMaterials used to simulate right column





Computational Drivers

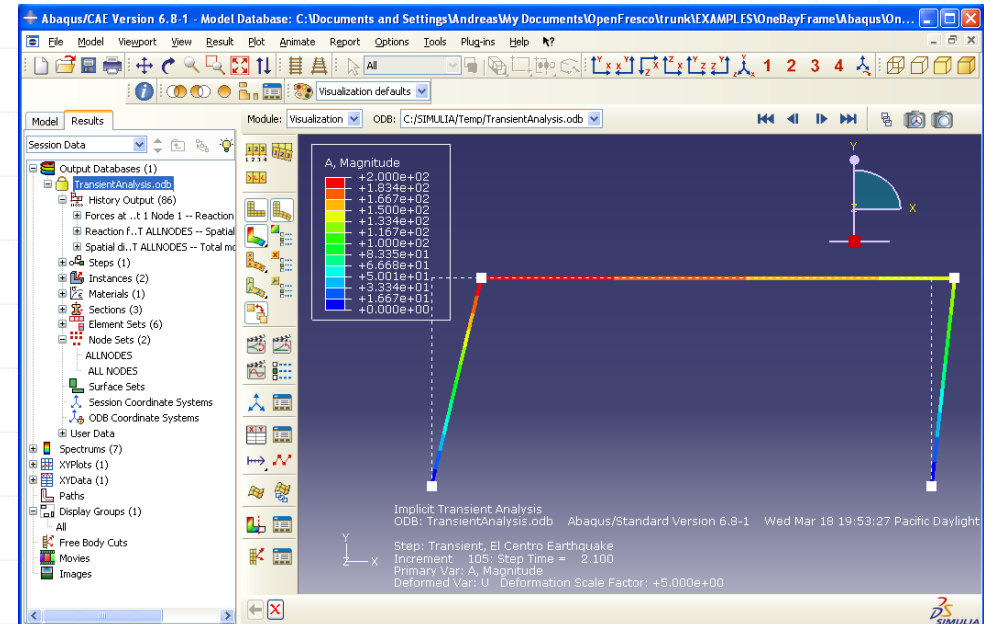
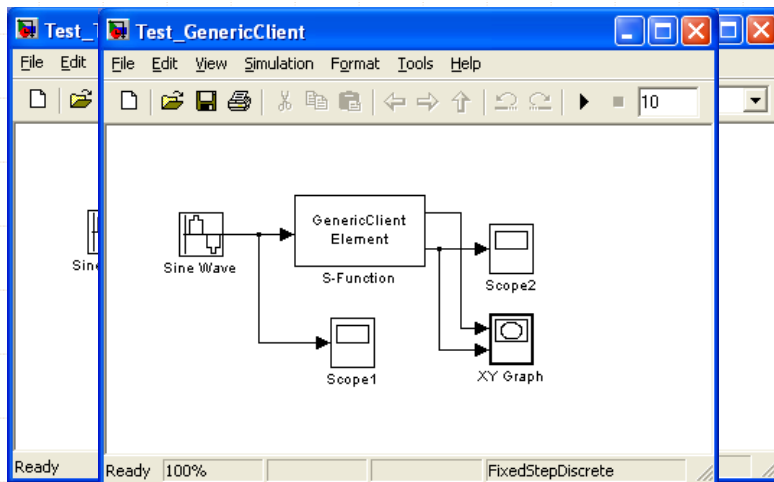
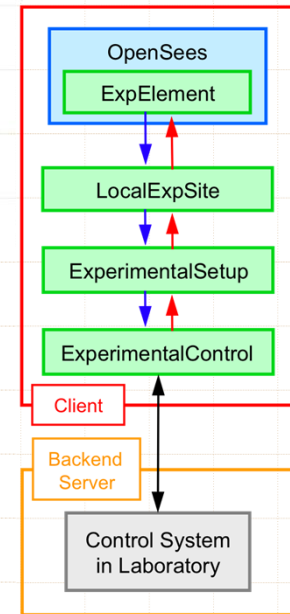
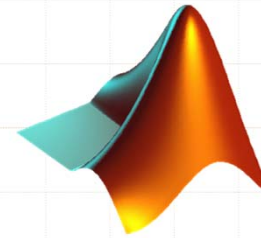
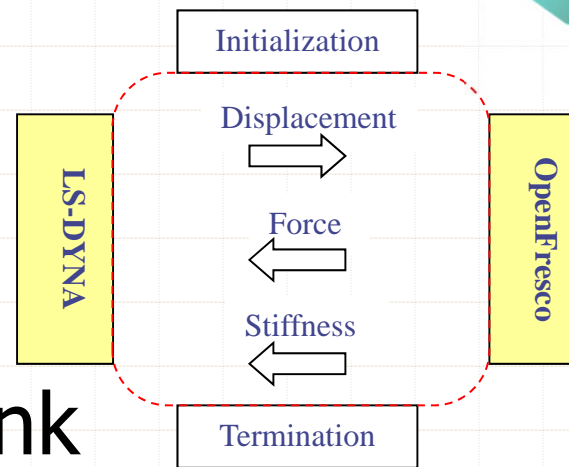


How to Interface

- ★ Two Ways to Interface with FE-Software
 - Generic Client Element
 - Experimental Element Directly in FE-Software
- ★ Generic Client Element to be Programmed by the Developers
- ★ Several generic client elements available:
`/trunk/SRC/simApplicationClient`

Computational Drivers

- ✦ OpenSees
- ✦ LS-DYNA
- ✦ Abaqus
- ✦ Matlab/Simulink
- ✦ UI-SimCor



More Information

OpenFresco

HOME USERS DEVELOPERS DOCUMENTATION REFERENCES COPYRIGHT

HOME

OpenFresco (the Open-source Framework for Experimental Setup and Control) is an environment-independent software framework, that connects finite element models with control and data acquisition systems in laboratories to facilitate hybrid simulation of structural and geotechnical systems.

Hybrid simulation is an experimental testing technique where a test is executed based on a step-by-step numerical solution of the governing equations of motion for a hybrid model, formulated considering both the numerical and physical portions of a structural system. In order for the earthquake engineering community to take full advantage of this technique, OpenFresco standardizes the deployment of hybrid simulation and extends its capabilities to applications where advanced numerical techniques are utilized, boundary conditions are imposed in real-time, and dynamic loading conditions caused by wind, blast, impact, waves, fire, traffic, and, in particular, seismic events are considered. Accordingly, the architecture of the OpenFresco software package provides a great deal of flexibility, extensibility, and re-usability to the researcher or developer interested in hybrid simulation.

Search

Menu

- Log in
- Register

Downloads

OpenFresco
OpenFrescoExpress
OpenSees Navigator

Hybrid Simulation

$$M \cdot \ddot{u} + C \cdot \dot{u} + P_r(u) = P(t)$$

Dynamic Loading:

- Seismic
- Wind
- Blast/Impact
- Wave
- Traffic

Static Loading:

- Gravity
- Prestress

Download documentation and software:
<http://openfresco.berkeley.edu>

energy dissipation and inertia
physical model of structural resistance

PEER

OpenFrescoExpress is a self-contained software package, including a easy-to-use graphical user interface, that facilitates hybrid testing of systems having up to two degrees of freedom. OpenFrescoExpress addresses the needs of a wide range of users including:

- laboratory staff and research students learning about hybrid simulation and starting to use this experimental testing method.
- staff and students at laboratories that regularly use hybrid simulation but desire a tool for quick demonstration of the hybrid simulation testing method.
- researchers who are conducting simple tests and would like to take advantage of a graphical user interface that quickly and easily displays useful real-time test data.
- graduate students and researchers who are not at a laboratory but wish to run the software as a pure simulation tool to learn more about hybrid simulation and how it works.



PACIFIC EARTHQUAKE ENGINEERING
RESEARCH CENTER

Advanced Implementation of Hybrid Simulation

Andreas H. Schellenberg
Stephen A. Mahin
Gregory L. Fenves
University of California, Berkeley

Download report from:
<http://peer.berkeley.edu/publications/>

PEER 2009/104
NOVEMBER 2009

Summary & Conclusions

- ★ Hybrid simulation inherently requires close collaboration amongst experts from many different fields.
 - Structural behavior
 - Laboratory testing and control
 - Computational simulation
 - Information technology
- ★ Hence, hybrid simulation fosters collaboration and communication among distant researchers in different labs.

Summary & Conclusions

- ★ OpenFresco, the environment-independent software framework for the development and deployment provides an excellent platform for this collaboration (on-site and geographically distributed)
- ★ The modularity and transparency of the framework permits existing components to be modified and new components to be added without much dependence on other objects.

Summary & Conclusions

- ★ Large libraries of hybrid simulation direct integration methods, experimental elements, experimental setups, controller models, and event-driven solution strategies are available to the researchers to choose or adapt from.
- ★ Needs:
 - User feedback on refinements and new features
 - Developer contributions to extend libraries

Questions?
Thank you!

The development of OpenFresco has been sponsored in parts by the National Science Foundation through grants from the NEES Consortium, Inc.



OpenFresco