

MTS/PEER Expert Seminar: Hybrid Simulation Technologies & Methods for Civil Engineering, UC Berkeley, 3/20-3/21/2018

Implementation Frameworks: OpenFresco & OpenFresco *Express*

Andreas Schellenberg, Ph.D., P.E.

Senior Structural Engineer, Maffei Structural Engineering

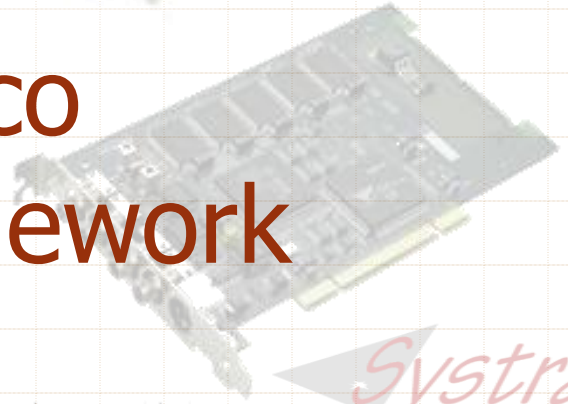


Outline of Presentation

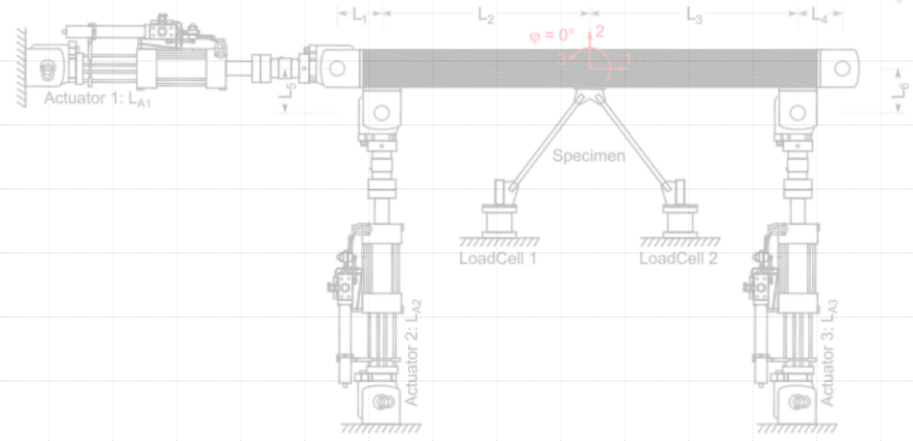
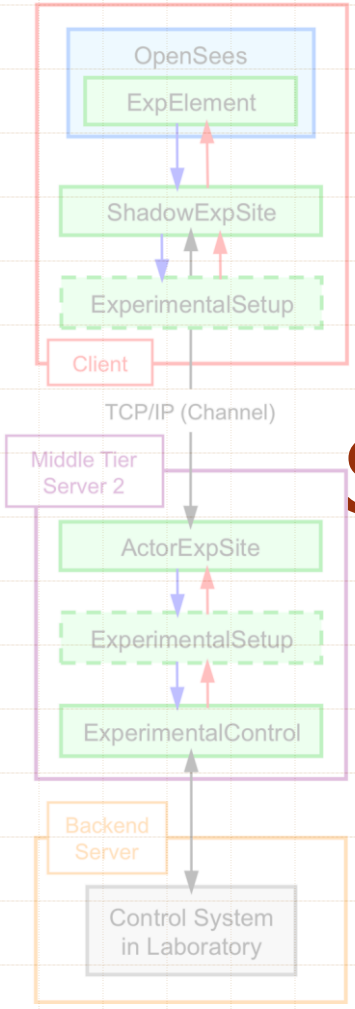
1. OpenFresco Software Framework
2. Unique OpenFresco Features
3. OpenFresco Express GUI
4. Website and Resources
5. Summary & Conclusions



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
 - ANSYS
 - LS-DYNA
 - Matlab
 - Simulink
 - UI-SimCor
 - ★ Control Systems
 - dSpace
 - MTS
 - ◆ STS family
 - ◆ Flextest/CSI
 - ◆ Flextest/SCRAMNet
 - ◆ 469D
 - ◆ SRMD
 - National Instruments
 - Pacific Instruments
 - ADwin

Why a Software Framework?

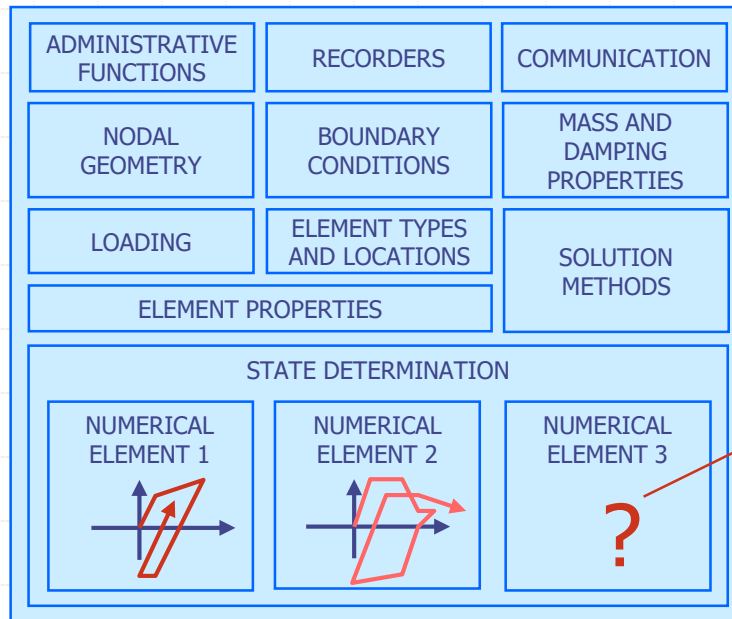
- ★ Lack of a common framework for development and deployment of HS
 - ★ Problem specific implementations which are site and control system dependant
 - ★ Such highly customized software implementations are difficult to adapt to different structural problems
- ➔ Need a robust, transparent, adaptable,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

What is a Software Framework?

- ★ A reusable design for a software system, or subsystem
- ★ Defines overall architecture of a software system, meaning its basic components and the relationships among them
- ★ Expressed as a set of abstract classes and the way their instances collaborate
- ★ Loose-coupling of components within the framework is essential for extensibility and reusability

Rethinking implementation strategies

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

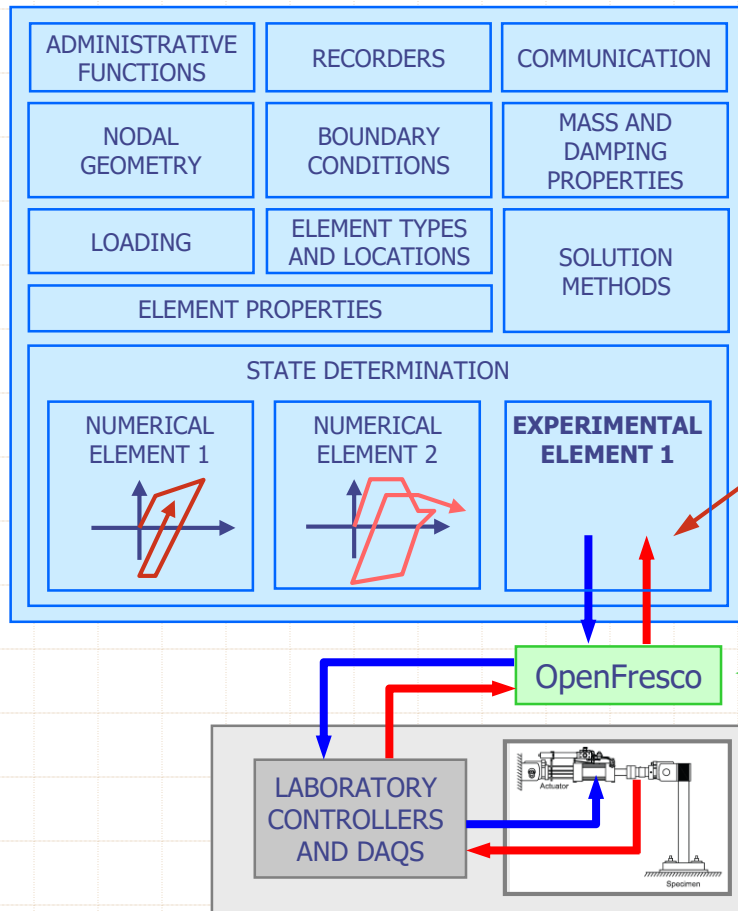


Typical features of an analysis framework

Proper numerical model uncertain

Rethinking implementation strategies

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



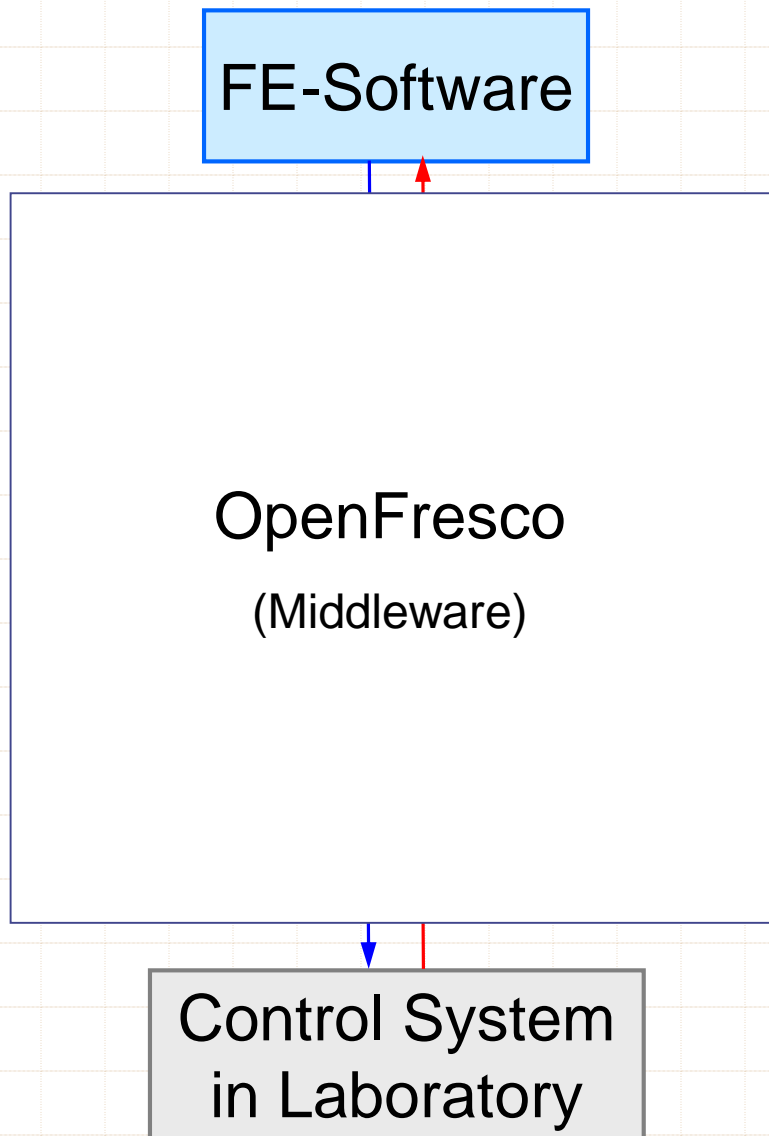
Typical features of an analysis framework

Define element as an "Experimental Element"

OpenFresco

Laboratory

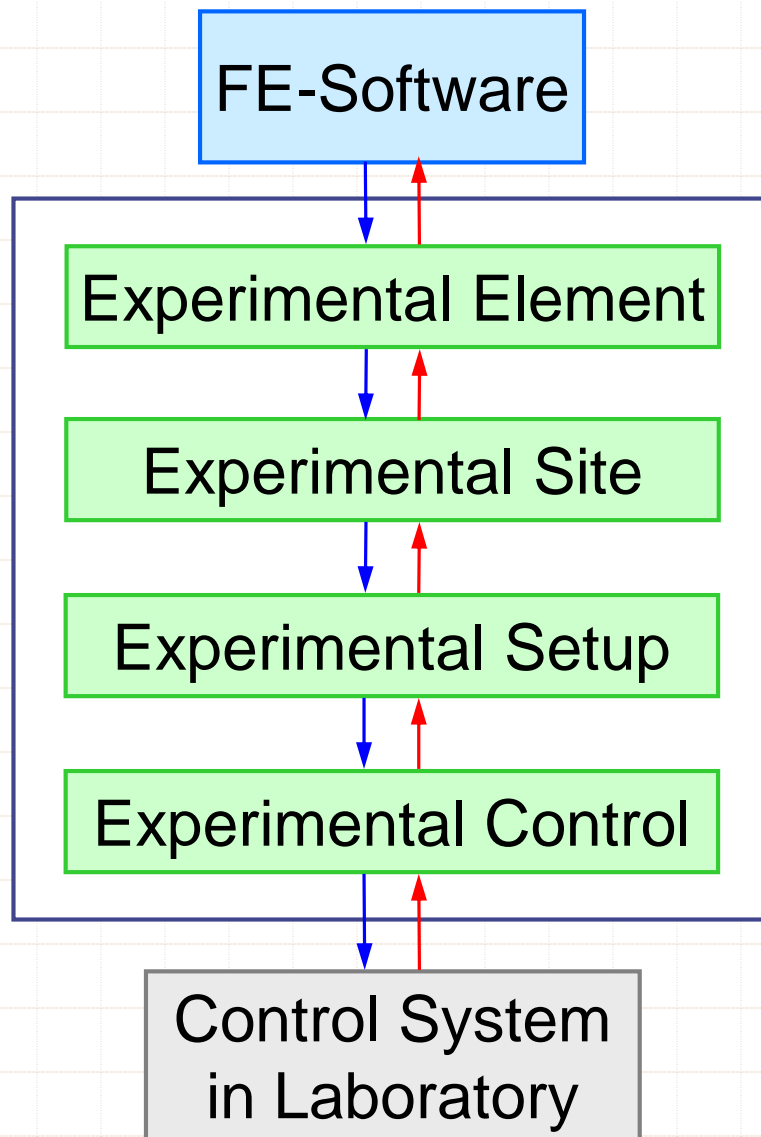
OpenFresco Components



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

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

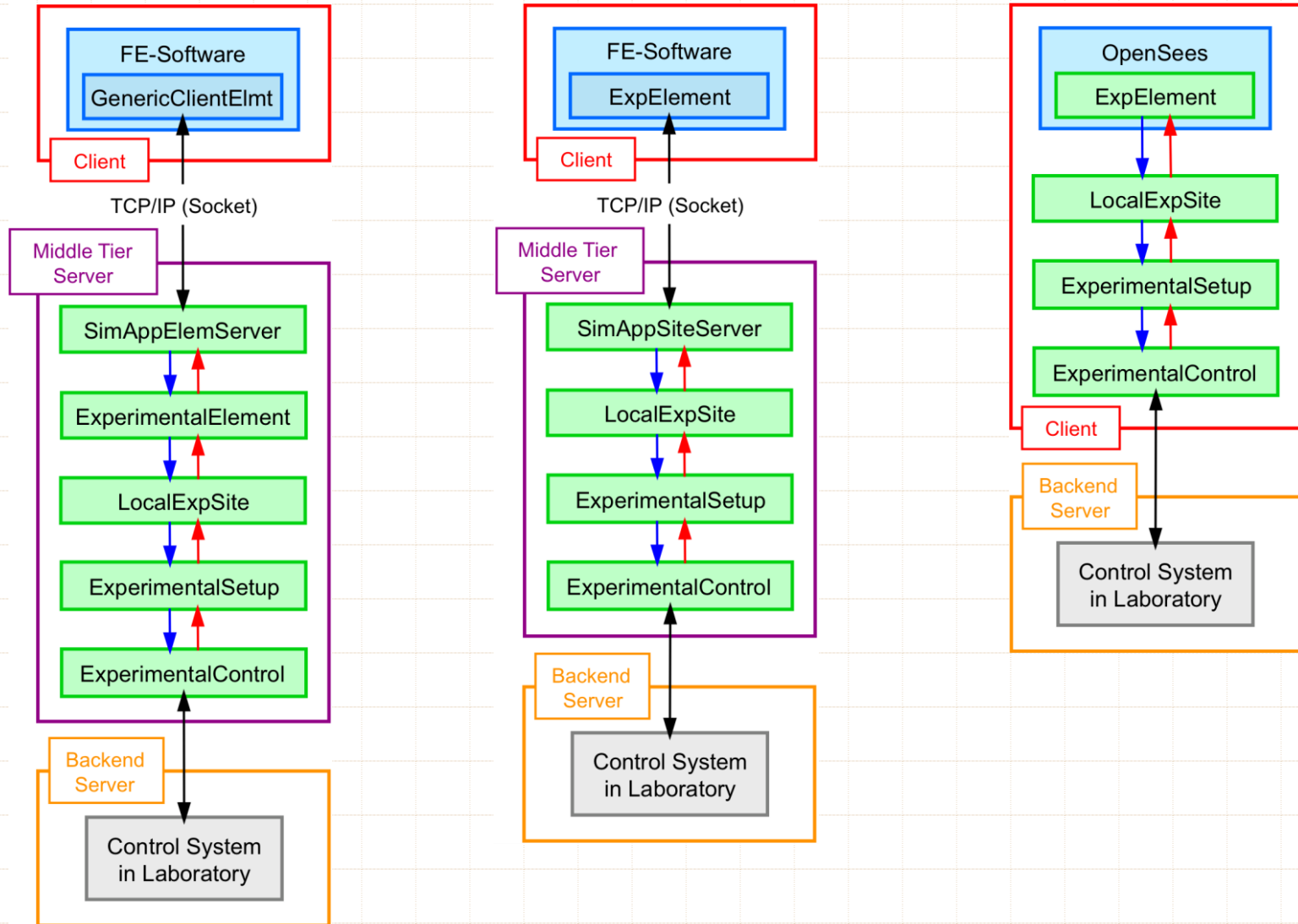
Requirements for Architecture

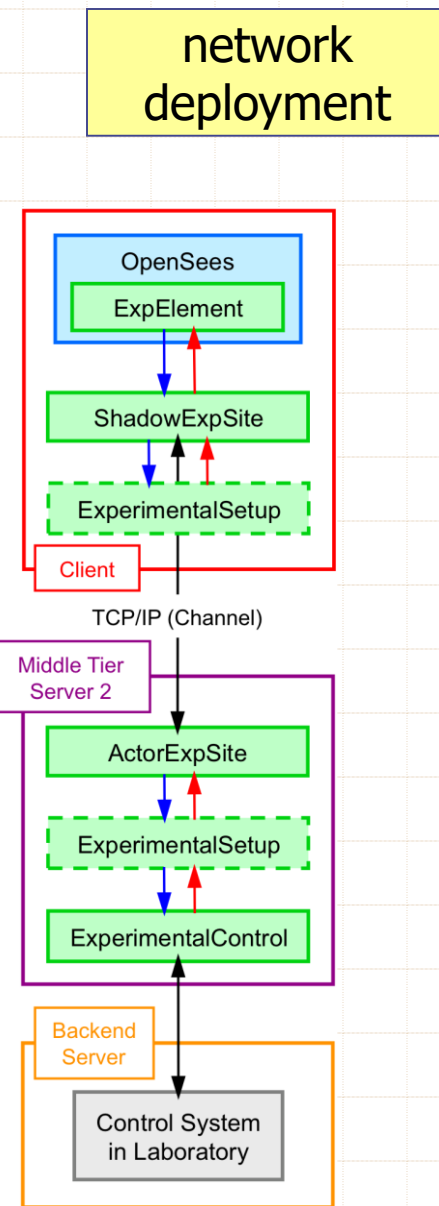
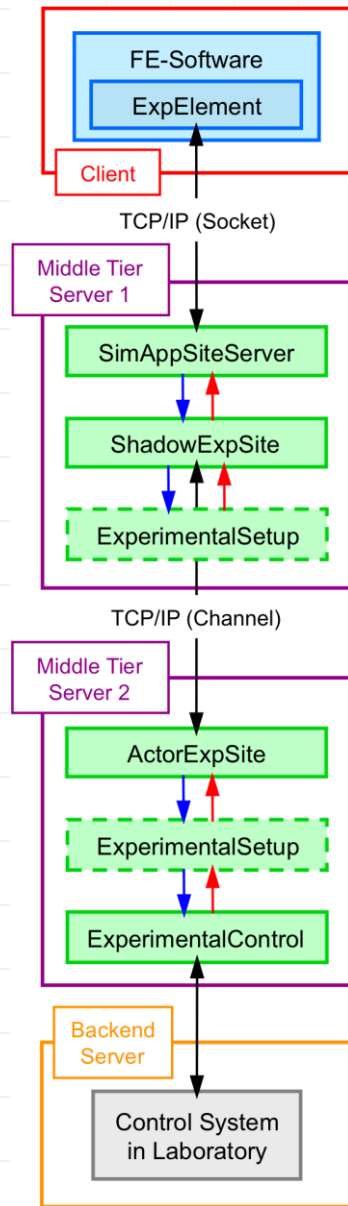
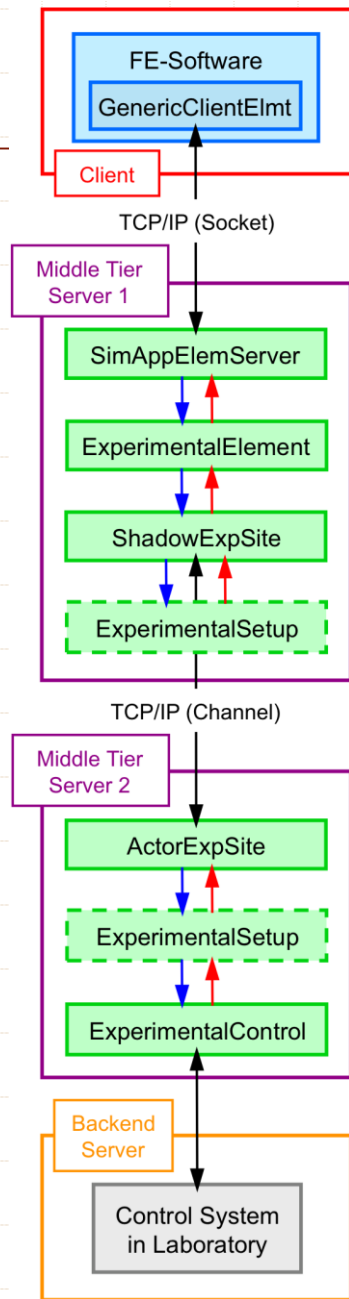
- ✦ Provide connectivity to a wide variety of FE-software (clients), independent of the language, such analysis software is programmed in
- ✦ Enable distributed testing and support different communication protocols
- ✦ Interface with rapidly evolving control and data acquisition systems deployed at testing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

➔ Multi/Three-Tier Software Architecture

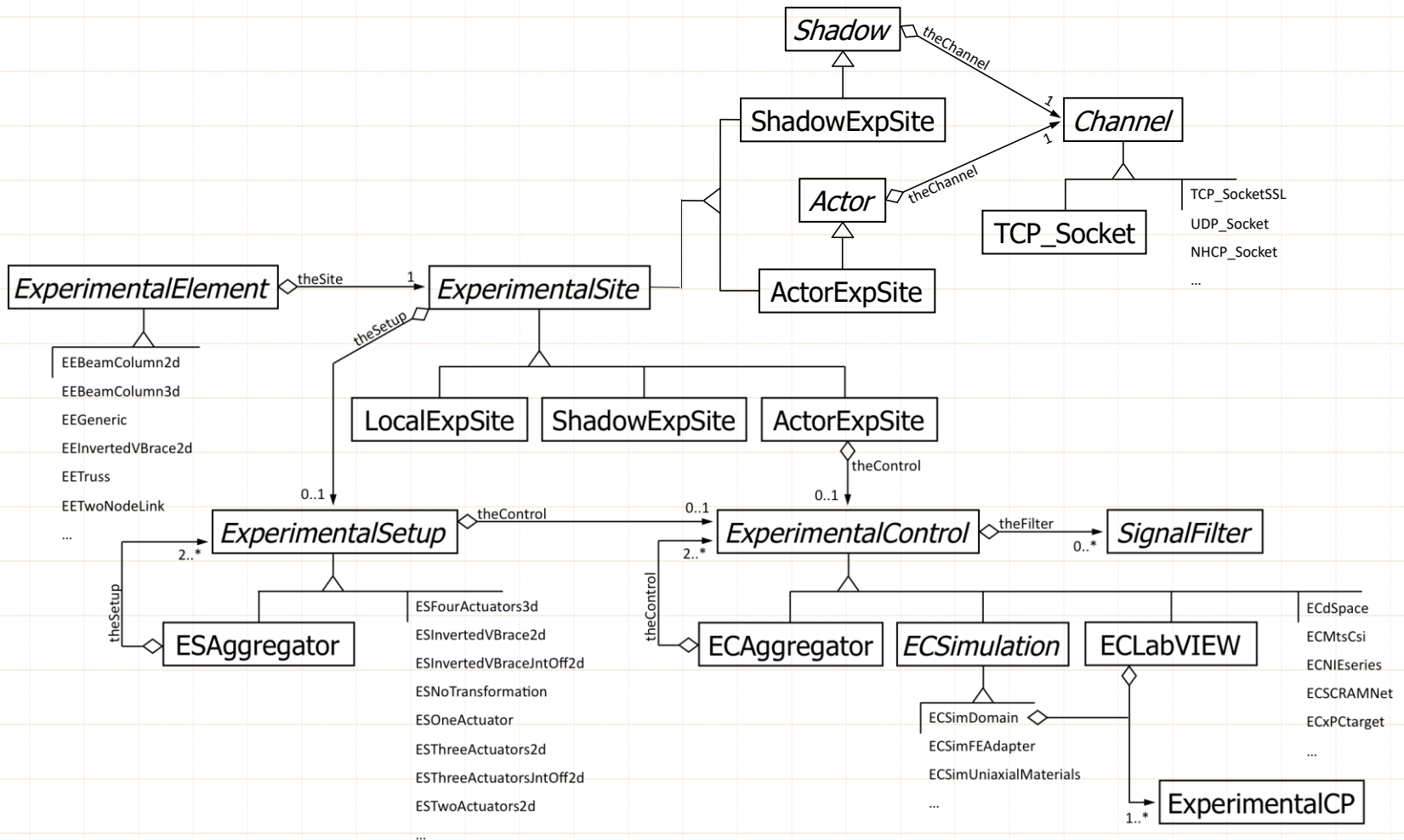
OpenFresco Components

local deployment

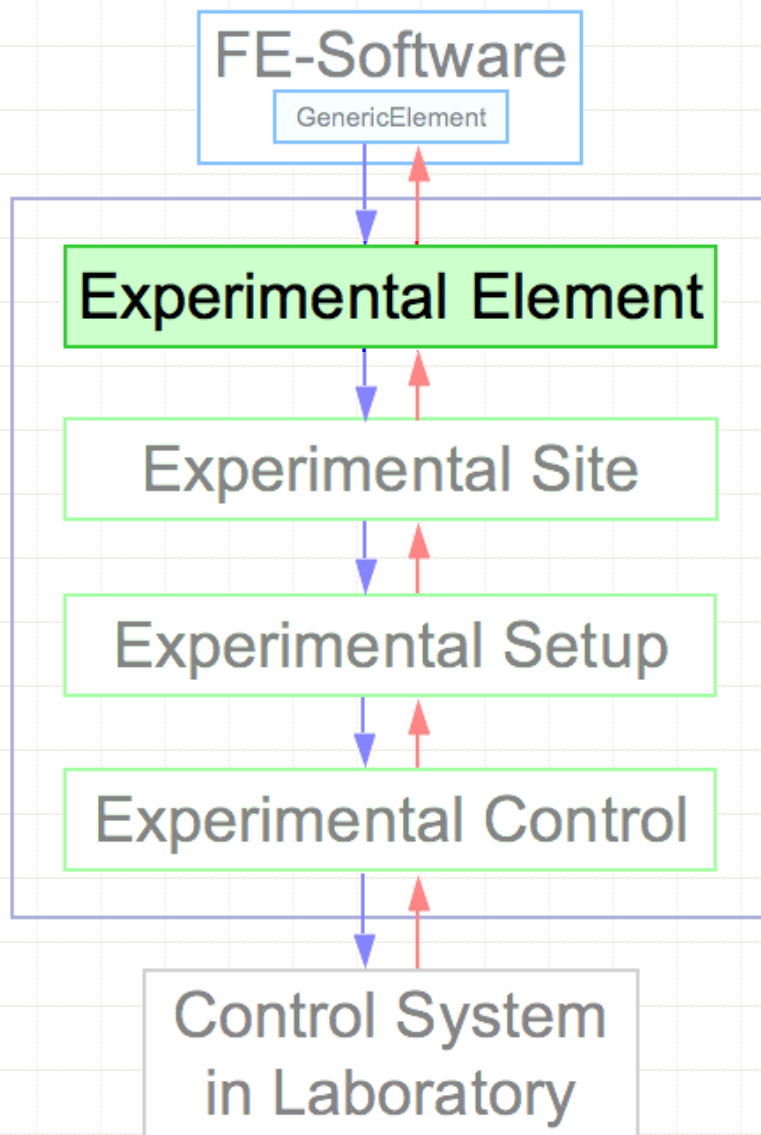




OpenFresco Class Diagram

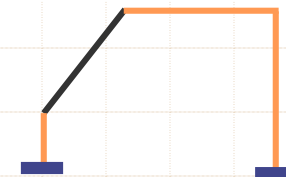


OpenFresco Components

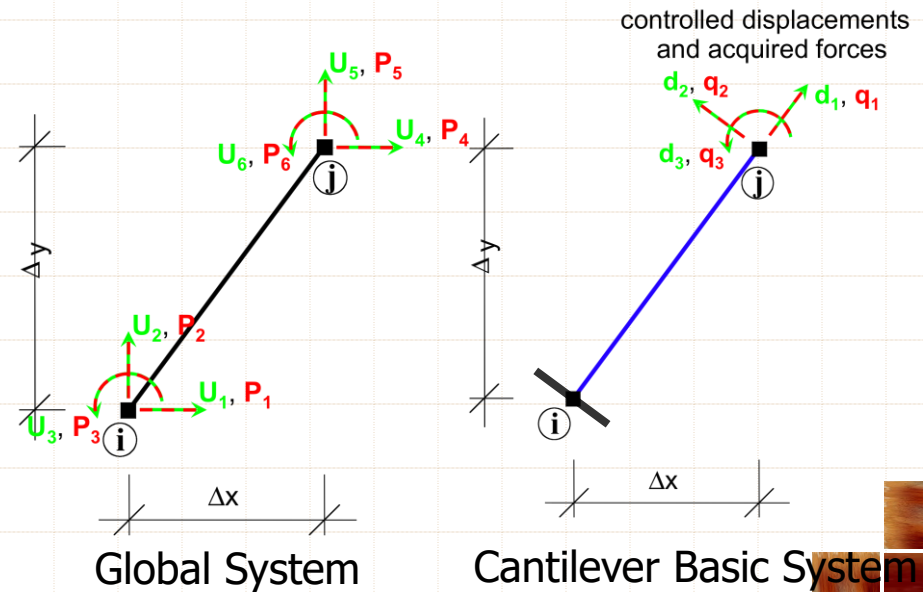


Experimental Element

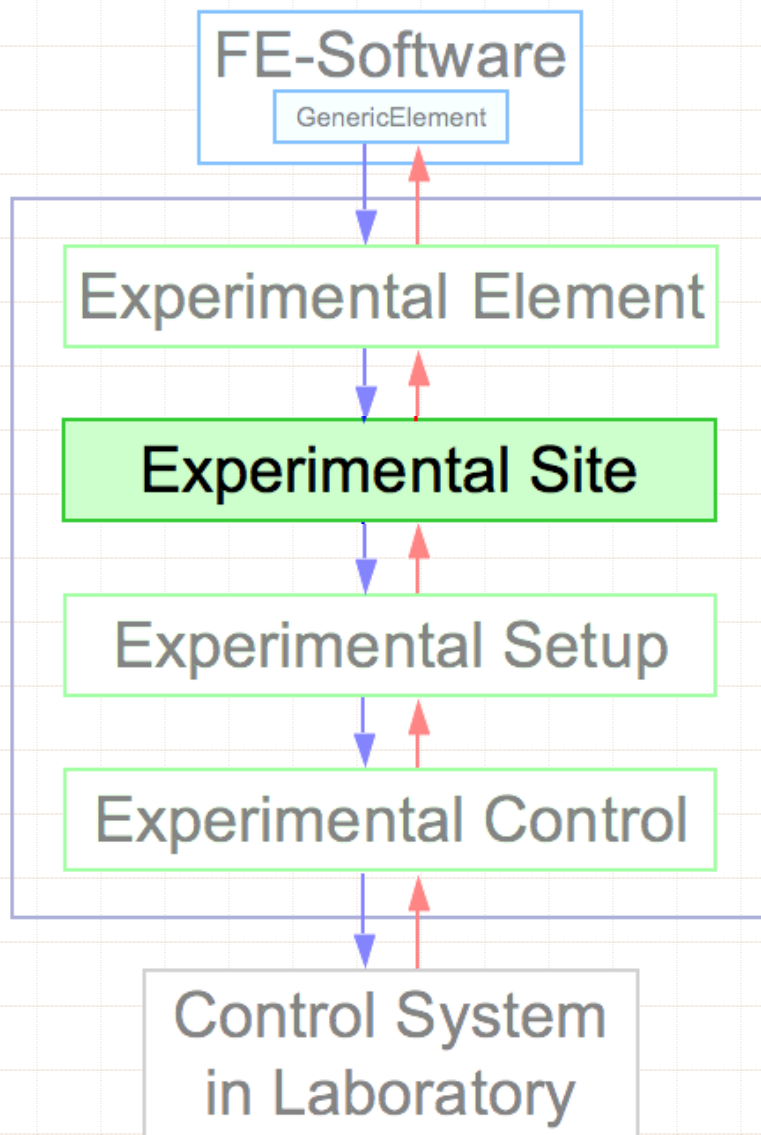
Transforms between the global element degrees of freedom in the FE-Software and the basic element degrees of freedom in the experimental element



Consider element in structure
Two coordinate systems used in FE analysis



OpenFresco Components



Experimental Site

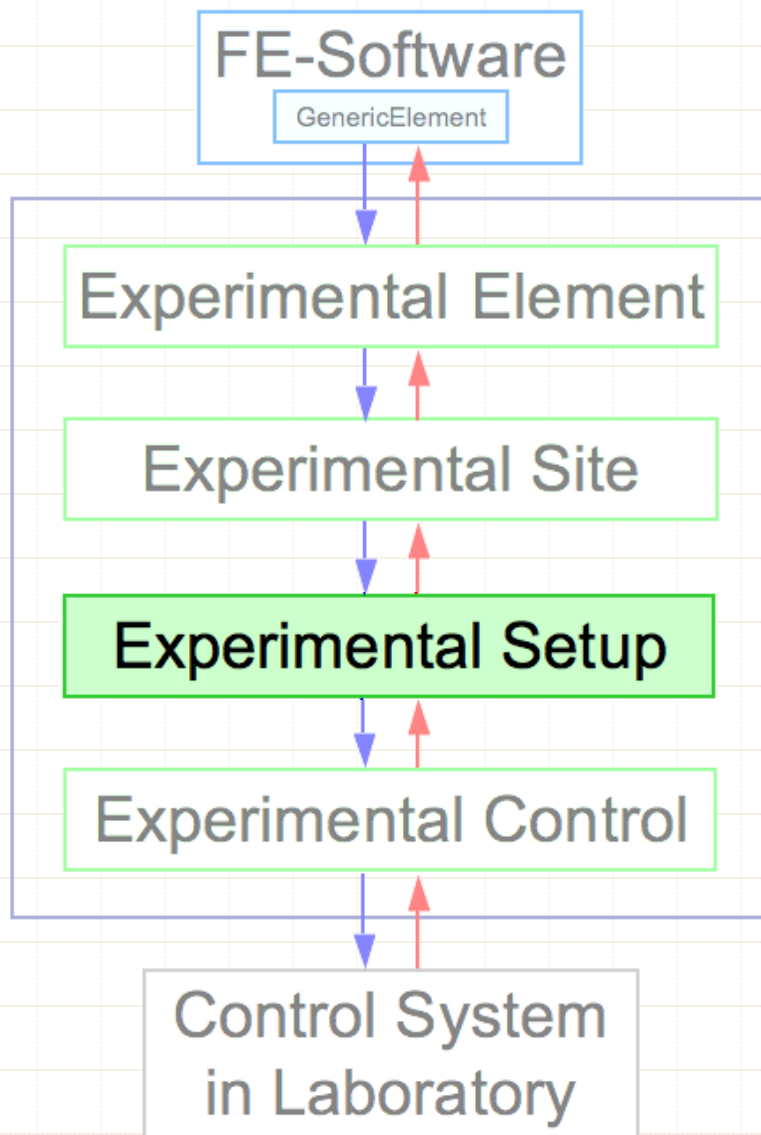
Stores data and provides communication methods for distributed testing

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

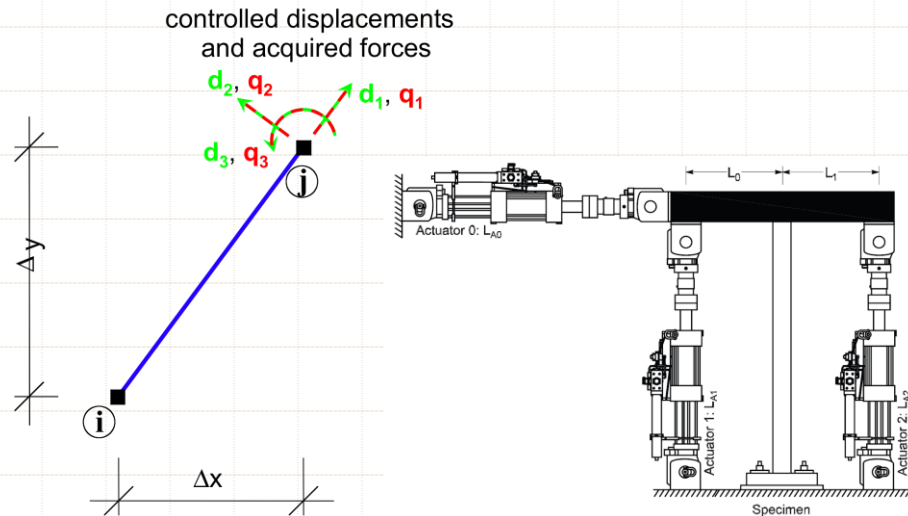


OpenFresco Components



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)

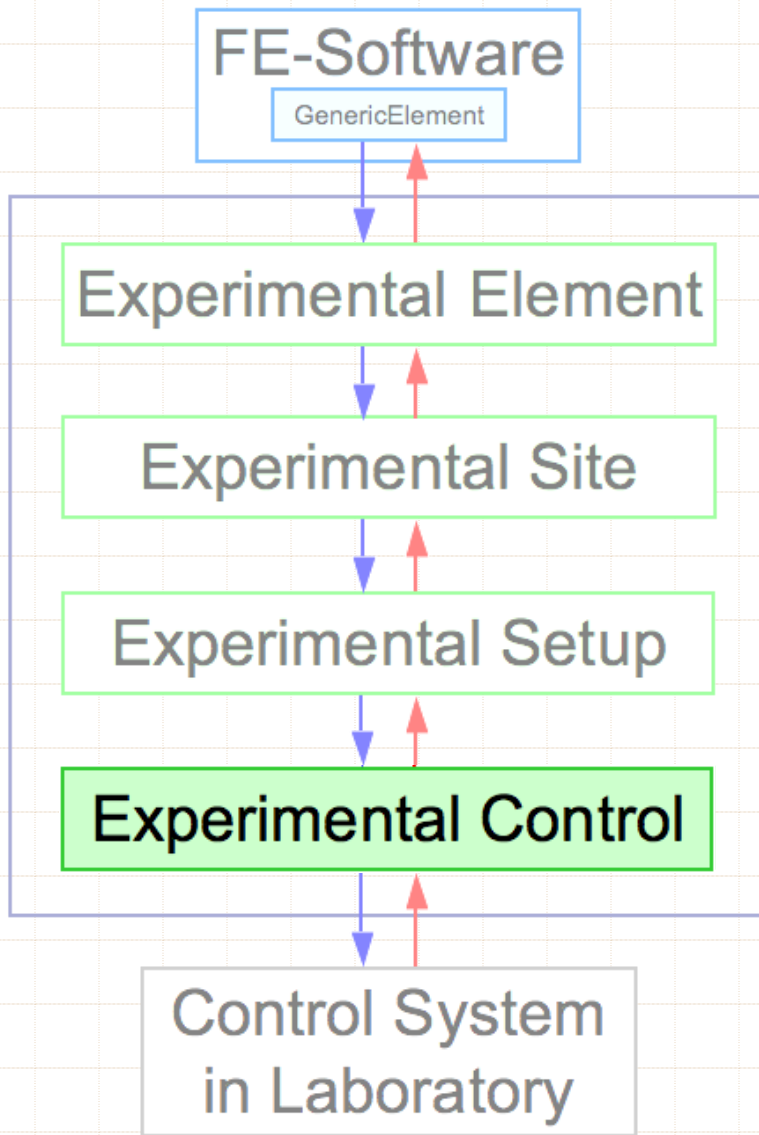


Cantilever Basic System

Actuator Setup

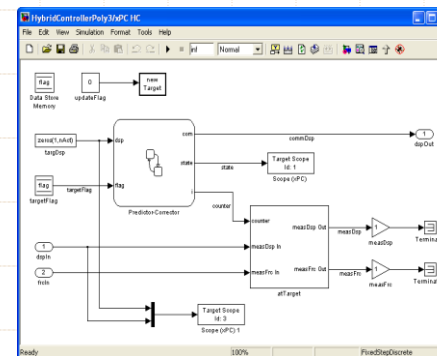
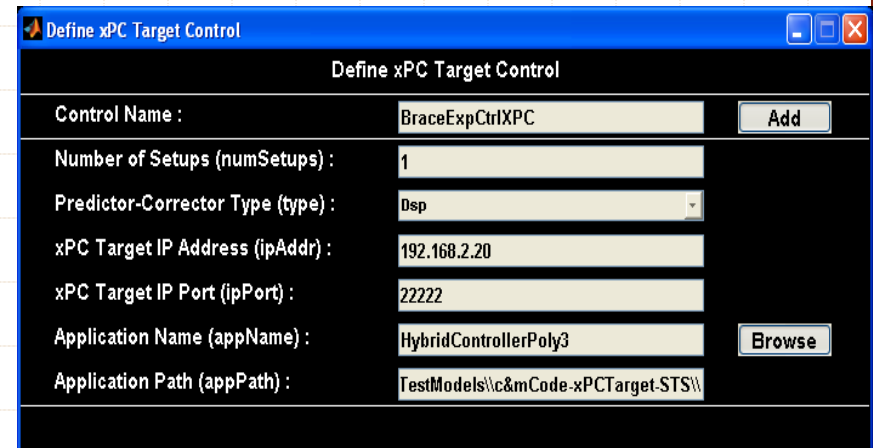
$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -L_0 \\ 0 & 1 & L_1 \end{bmatrix}$$

OpenFresco Components

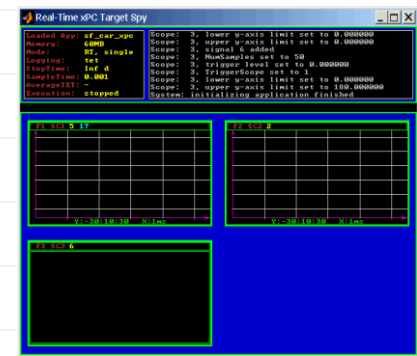


Experimental Control

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



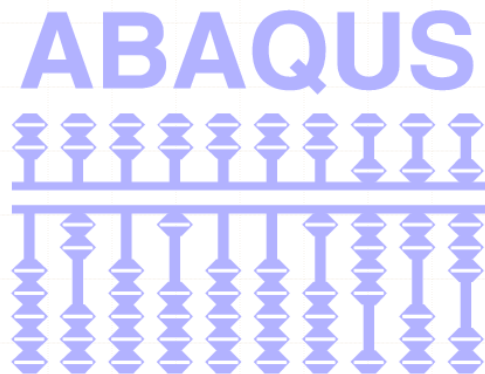
Predictor-Corrector



xPC Target

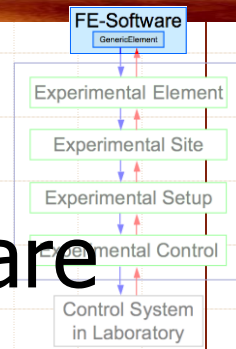


Computational Drivers



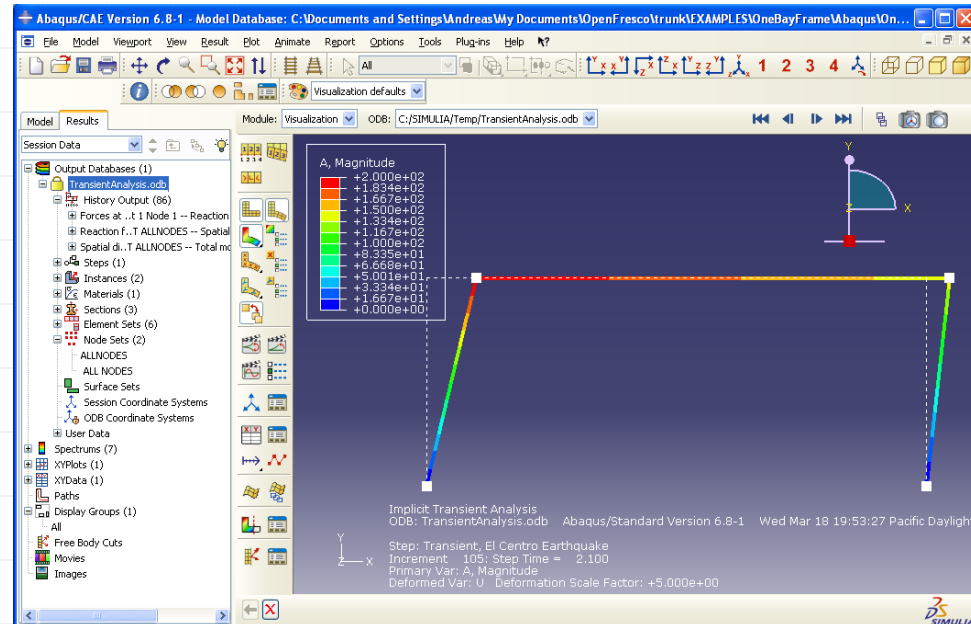
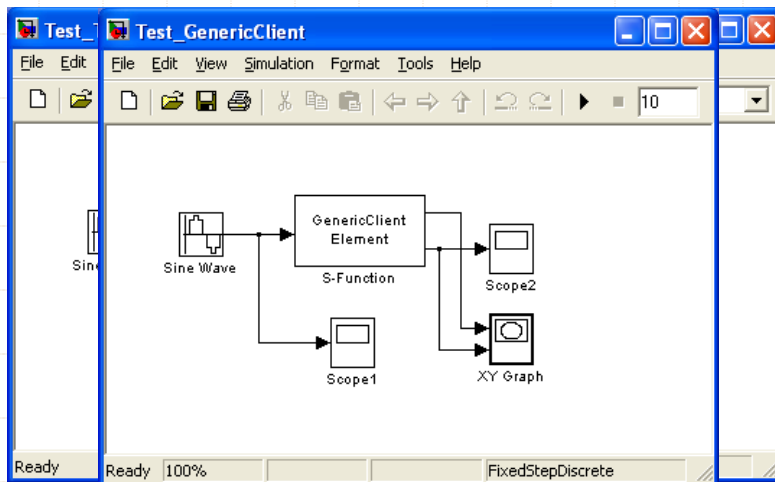
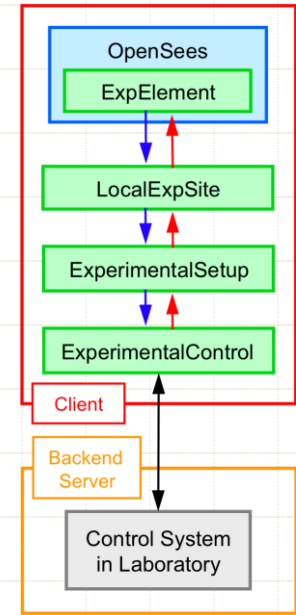
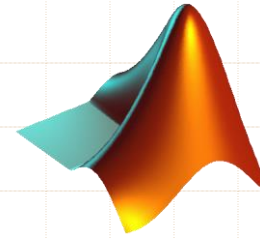
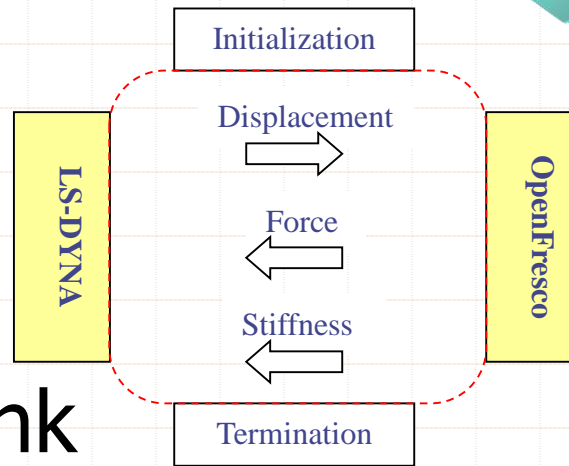
How to Interface

- ★ 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

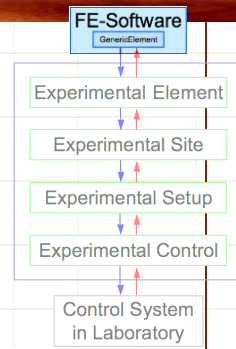
- ✦ OpenSees
- ✦ LS-DYNA
- ✦ Abaqus
- ✦ Matlab/Simulink
- ✦ ANSYS

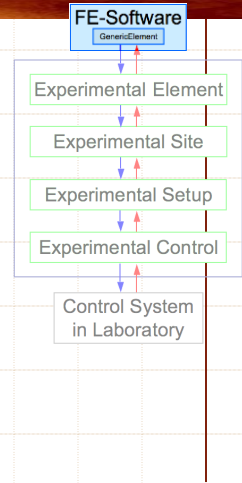


Integration Methods

$$\mathbf{M} \times \ddot{\mathbf{u}}_n + \mathbf{C} \times \dot{\mathbf{u}}_n + \mathbf{P}_r(\mathbf{u}_n) = \mathbf{P}(t_n)$$

- ★ Mass matrix \mathbf{M} is often singular
-> second order differential equation
infinitely stiff -> fully implicit numerical methods
- ★ Make as few function calls as possible
- ★ Use constant Jacobian in the numerical methods since tangent stiffness is not available





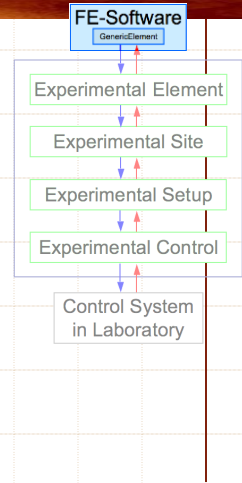
Direct Integration Methods

★ Explicit Integrators

- explicit Newmark Method
- Central-Difference Method
- explicit Alpha Method
- explicit Generalized-Alpha Method
- KR-Alpha Method

★ Implicit Integrators (do not use for HS)

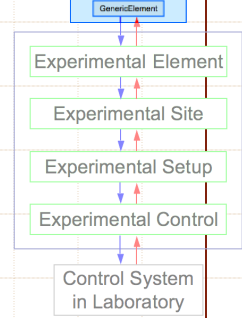
- Newmark Method
- Alpha Method
- Generalized-Alpha Method
- Collocation Method



Direct Integration Methods

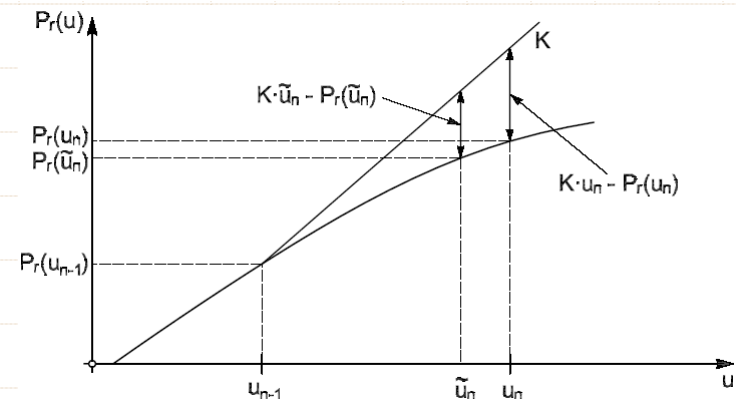
- ★ Implicit Integrators with increment reduction factors
 - Newmark HS IncrReduct Method
 - Generalized-Alpha HS IncrReduct Method
 - Collocation HS IncrReduct Method

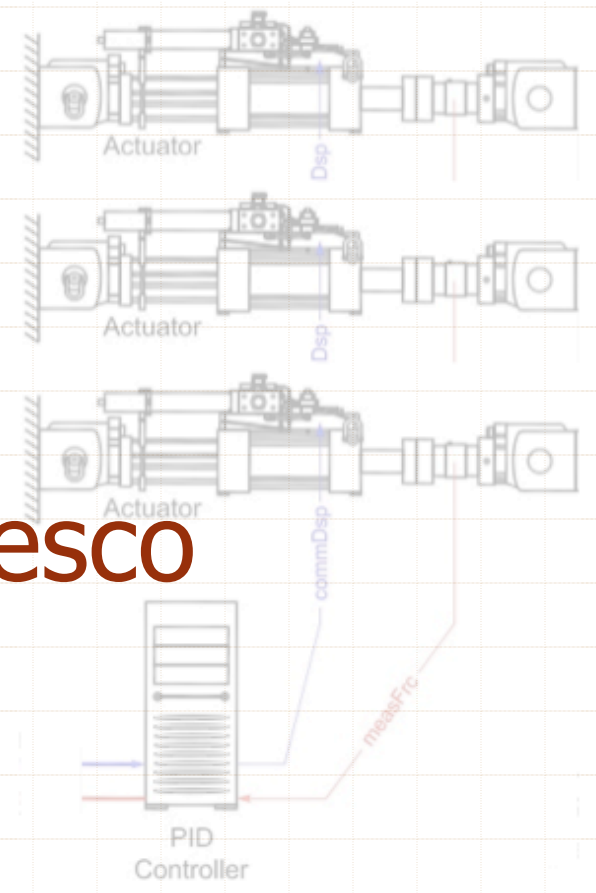
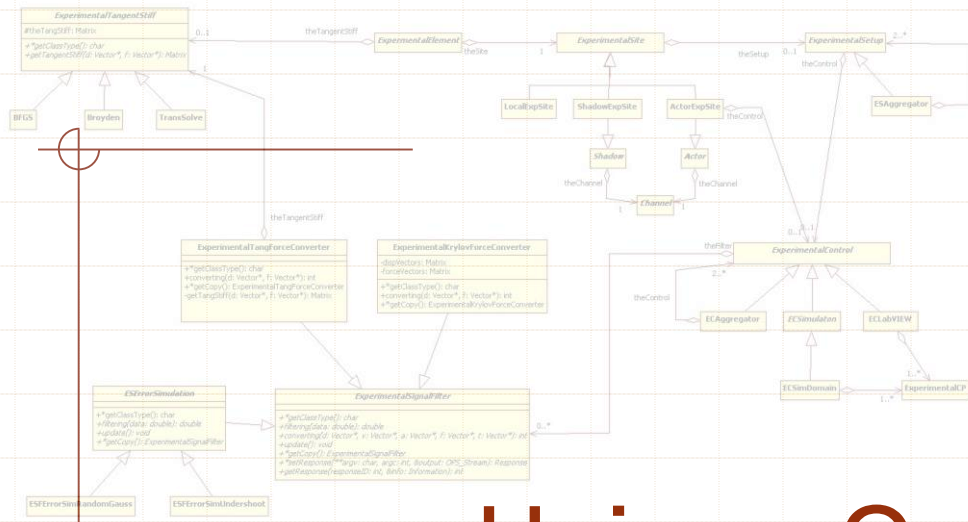
- ★ Implicit Integrators with increment limits
 - Newmark HS IncrLimit Method
 - Generalized-Alpha HS IncrLimit Method
 - Collocation HS IncrLimit Method



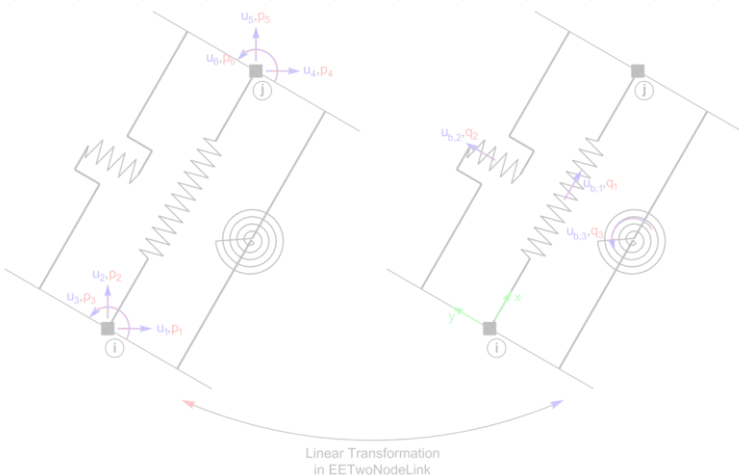
Direct Integration Methods

- ★ Implicit Integrators with sub-stepping (constant number)
 - Newmark HS FixedNumIter Method
 - Generalized-Alpha HS FixedNumIter Method
 - Collocation HS FixedNumIter Method
- ★ Predictor-Corrector Integrators
 - Alpha-OS Method
 - Generalized-Alpha-OS Method

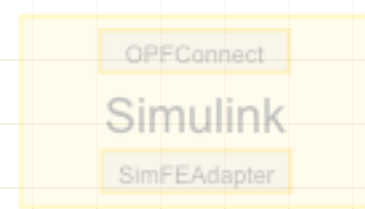




Unique OpenFresco Features



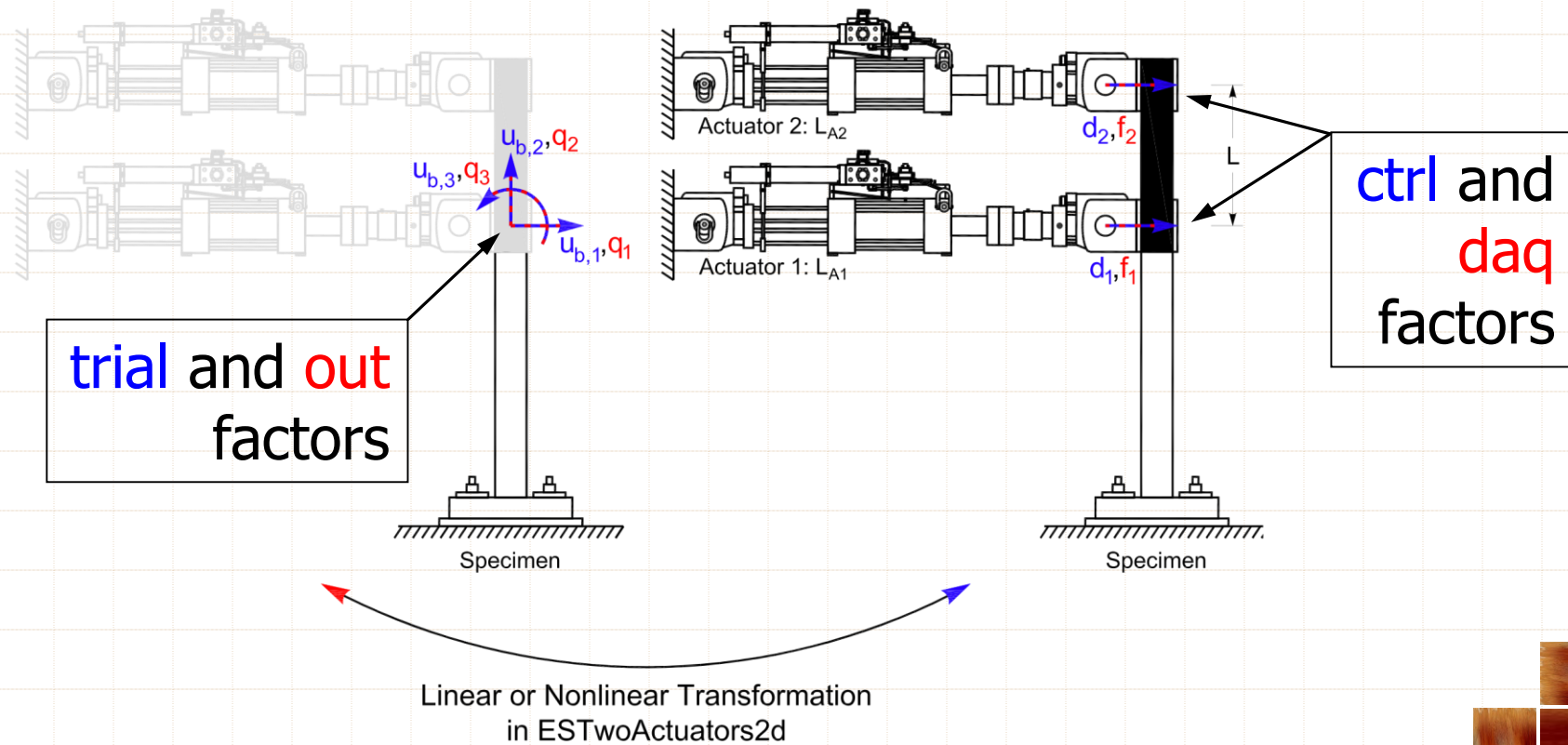
$$\begin{aligned}
 u_{b,1} &= u_4 - u_1 \\
 u_{b,2} &= u_5 - u_2 \\
 -0.5L(u_3 + u_6) \\
 u_{b,3} &= u_6 - u_3
 \end{aligned}$$



Similitude and Scaling

★ Response Factors:

factors can now be applied to trial and output response data in addition to control and daq response data

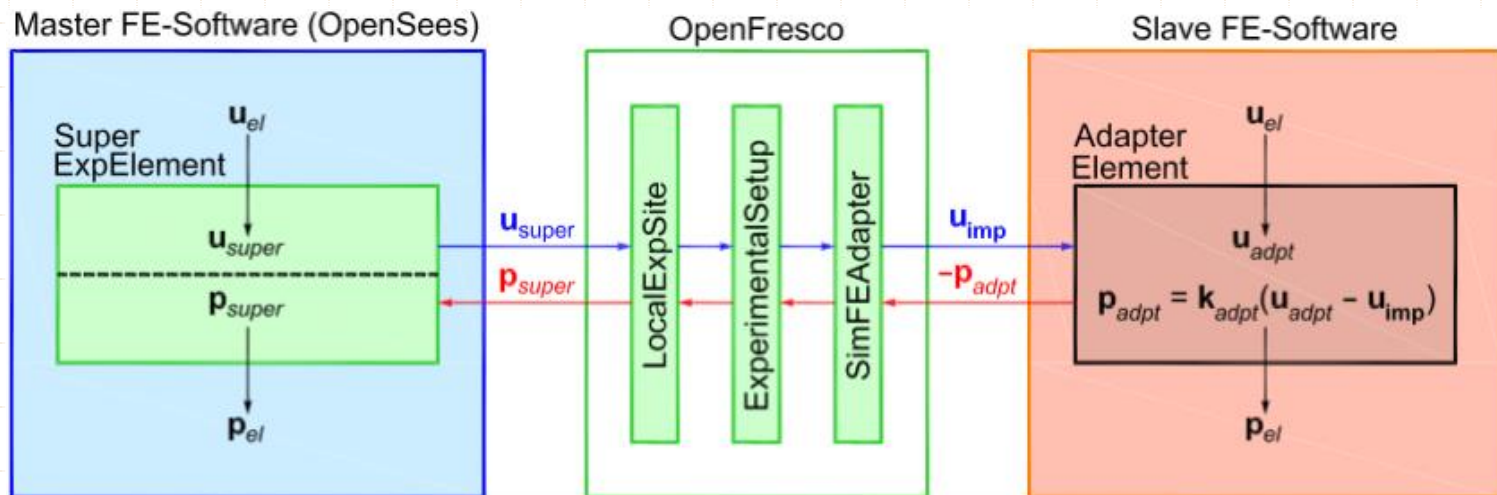


Co-Simulation (Software Coupling)

ECSimFEAdapter

```
expControl SimFEAdapter $tag ipAddr $ipPort
```

\$tag unique control tag
ipAddr IP address of slave process
\$ipPort IP port of slave process



Co-Simulation (Software Coupling)

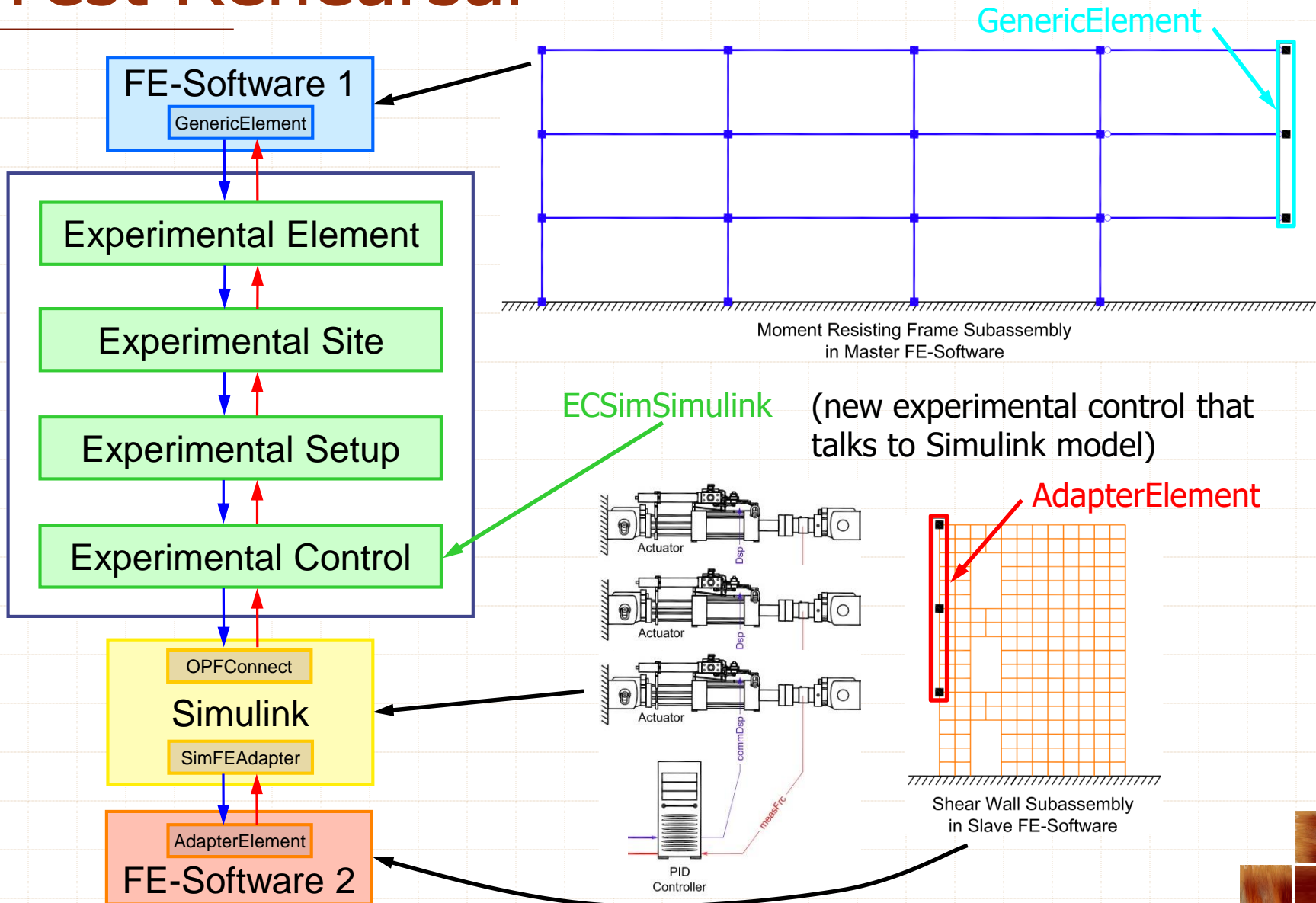
★ Master Programs

- OpenSees
- OpenFresco *Express*
- Abaqus
- LS-DYNA
- Matlab
- Simulink
- ANSYS
- UI-SimCor

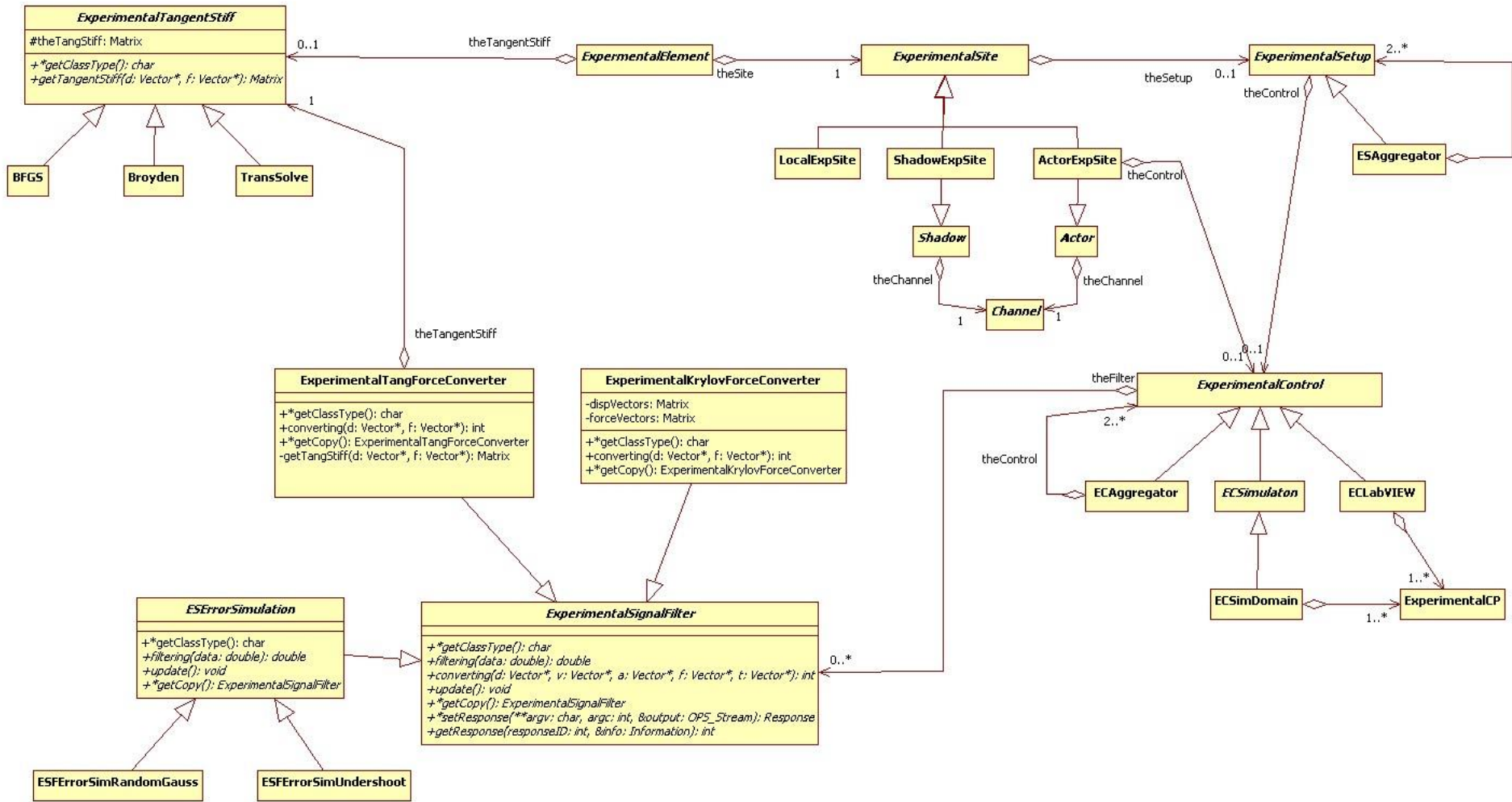
★ Slave Programs

- OpenSees
- Abaqus
- LS-DYNA
- Matlab
- Simulink
- ANSYS

Test Rehearsal



Force and Mixed Control



Some useful utility commands

- ★ `logFile $fileName <-append> <-noEcho>`
- ★ `defaultUnits -force $type -length $type
-time $type -temp $type`
- ★ `metaData -title $txt -contact $txt -description
$txt -modelType $txt -analysisType $txt ...`
- ★ `recordExp`
- ★ `removeExp recorder $tag`
- ★ `wipeExp`
- ★ `setupLabServer $siteTag`
- ★ `stepLabServer $siteTag $numSteps`
- ★ `stopLabServer $siteTag`

GUI: OpenFresco *Express*



GUI: OpenFresco *Express*

The screenshot displays the OpenFresco Express v1.0 software interface. On the left is a vertical menu with buttons for 'Structure', 'Loading', 'ExpSetup', 'ExpControl', 'Analysis', and 'User Tips'. The main area is titled 'Structural Properties' and contains a 3D model of a vertical cantilever beam with a mass m at the top and a spring constant k . A red arrow labeled 'DOF_x' indicates the direction of motion. Below the model are input fields for mass m (0.02), stiffness k (2.8), and a calculated 'Period: 0.53103'. There is also a 'Mass Proportional' dropdown menu set to 'Mass Proportional' and a corresponding input field with the value 0.02. Red callout boxes with arrows point to these elements: 'Select Structure' points to the 'Structure' button; 'Period is immediately calculated' points to the period value; 'Assign Mass' points to the mass input field; 'Assign Stiffness' points to the stiffness input field; and 'Select & Assign Damping' points to the mass proportional input field. The PEER logo is visible in the bottom left corner, and the version number 'Version 1.0' is in the bottom right corner.

Structure

Loading

ExpSetup

ExpControl

Analysis

User Tips

Structural Properties

Assign Mass

Assign Stiffness

Select & Assign Damping

Select Structure

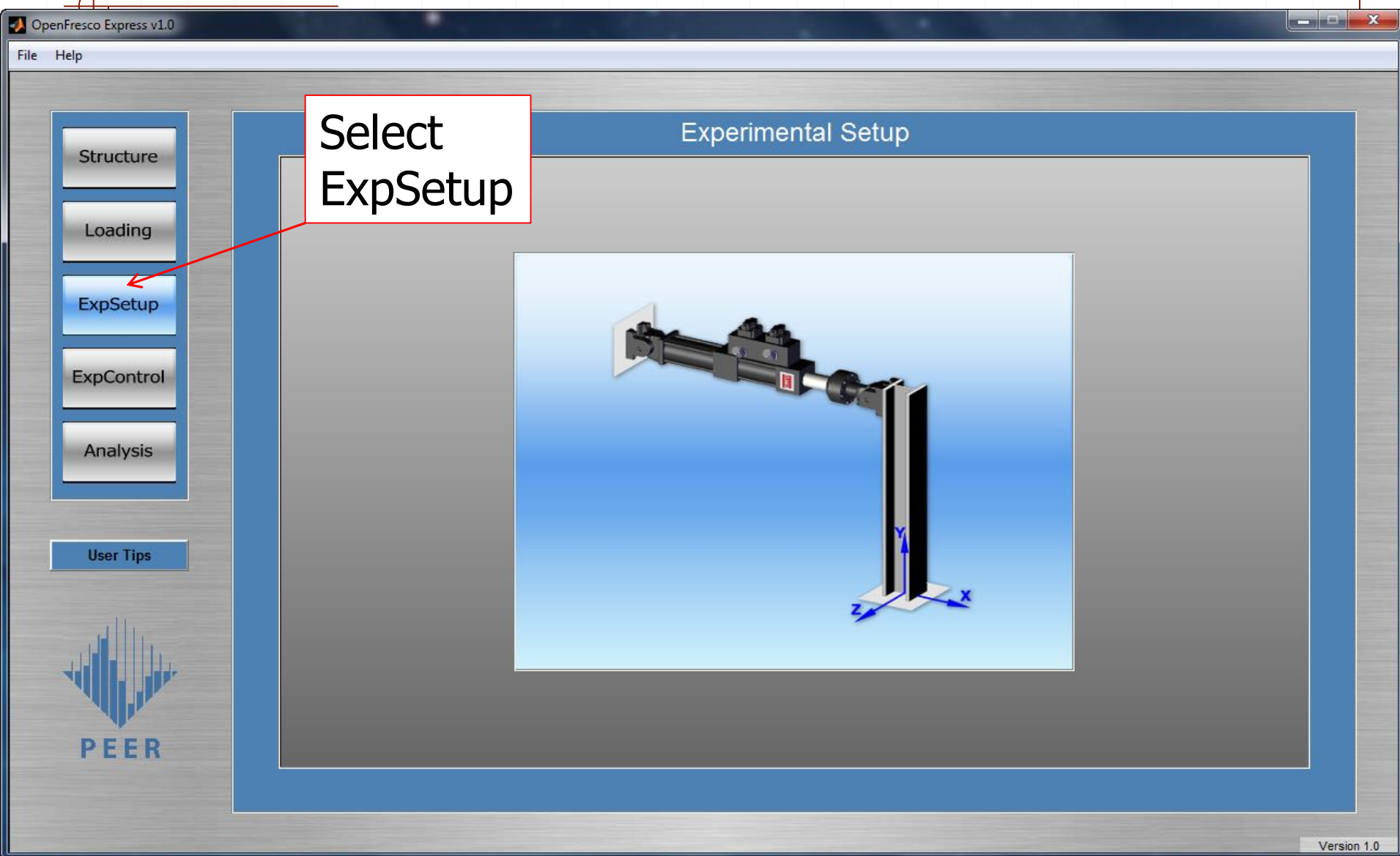
Period is immediately calculated

Version 1.0

Define Loading: Free Vibration

The screenshot shows the 'OpenFresco Express v1.0' software interface. On the left is a vertical navigation menu with buttons for 'Structure', 'Loading', 'ExpSetup', 'ExpControl', 'Analysis', and 'User Tips'. The 'Loading' button is highlighted with a red box and an arrow pointing to it from a text box labeled 'Select Loading'. The main workspace has two tabs: 'Ground Motions' and 'Free Vibration', with the latter selected. A red box labeled 'Select Type of Loading' has an arrow pointing to the 'Free Vibration' tab. Below the tabs are three input fields: 'Initial Displacement' (set to 'Mode 1', '0.1', and '0.167'), 'Ramp Time' (set to '10'), and 'Free Vibration Time' (set to '20'). A red box labeled 'Input Free Vibration Parameters' has an arrow pointing to these fields. At the bottom of the main workspace is a graph titled 'Displacement [L]' vs 'Time [sec]'. The graph shows a blue line that ramps up linearly, then drops sharply to a peak, and then oscillates with decreasing amplitude. A red dashed vertical line marks the start of the vibration. A red box labeled 'Select Loading' has an arrow pointing to the graph area. The graph includes labels for 'Initial Displacement', 'Ramp Time', and 'Vibration Time'.

Experimental Setup



Real Experimental Control

The screenshot displays the OpenFresco Express v1.0 software interface. On the left, a vertical sidebar contains buttons for Structure, Loading, ExpSetup, ExpControl, Analysis, and User Tips. The ExpControl button is highlighted with a red arrow pointing to a callout box that says "Select ExpControl". The main window area is titled "Experimental Control" and features two tabs: "Simulation" and "Real Controller". The "Real Controller" tab is selected and highlighted with a red arrow pointing to a callout box that says "Select Real Controller". Below the tabs is a "Define Control" section containing a dropdown menu labeled "Control Types...". The dropdown menu is open, showing options: LabVIEW, MTSCsi, SCRAMNet, dSpace, and xPCtarget. The "MTSCsi" option is highlighted with a blue bar and a mouse cursor, with a red arrow pointing to a callout box that says "Select Type of Controller". The PEER logo is visible in the bottom left corner, and the version number "Version 1.0" is in the bottom right corner.

Real Experimental Control

The screenshot displays the OpenFresco Express v1.0 software interface. The main window is titled "Experimental Control" and contains two tabs: "Simulation" and "Real Controller". The "Real Controller" tab is active. The configuration area includes:

- Define Control:** A dropdown menu set to "MTSCsi".
- Configuration File Name:** A text box containing "OpenFresco_mNEES" and a "Browse" button to its right.
- Configuration File Path:** A text box containing "C:\Users\Andreas\Documents\MTS_CSI\Op".
- Ramp Time:** A text box containing "0.1".

Annotations with red arrows point to the "Browse" button and the "Ramp Time" input field. The "Browse" button is annotated with the text "Browse for MTS CSI config file". The "Ramp Time" input field is annotated with the text "Input Ramp Time".

On the left side of the interface, there is a vertical menu with buttons for "Structure", "Loading", "ExpSetup", "ExpControl", "Analysis", and "User Tips". The "ExpControl" button is highlighted. At the bottom left, there is a logo for "PEER" (Professional Engineering Expertise Research) featuring a stylized bar chart.

Version 1.0

Analysis

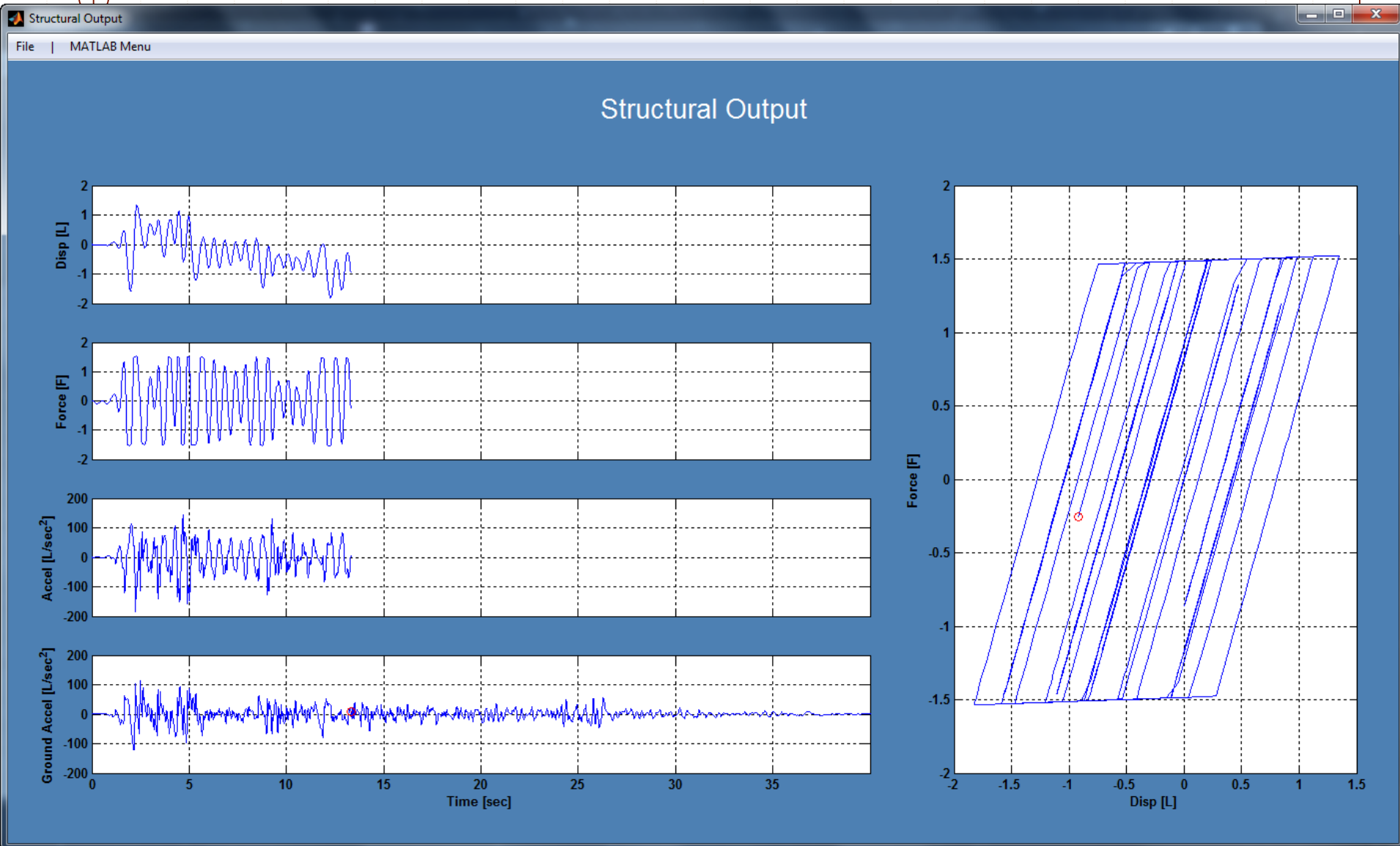
The screenshot shows the OpenFresco Express v1.0 software interface. The main window is titled "Analysis" and contains the following elements:

- Left Sidebar:** A vertical menu with buttons for "Structure", "Loading", "ExpSetup", "ExpControl", "Analysis" (highlighted in blue), and "User Tips". At the bottom of the sidebar is the PEER logo.
- Analysis Window:**
 - A text field for "dt" with the value "0.01" and a comparison "< 0.16903".
 - A "Write TCL File" button.
 - A "Generate Report" button.
 - A question: "Would you like to animate the response?" with radio buttons for "No" (selected) and "Yes".
 - Three large control buttons: a play button, a stop button, and a refresh button.
 - Buttons for "Run New Test" and "Quit" at the bottom right.

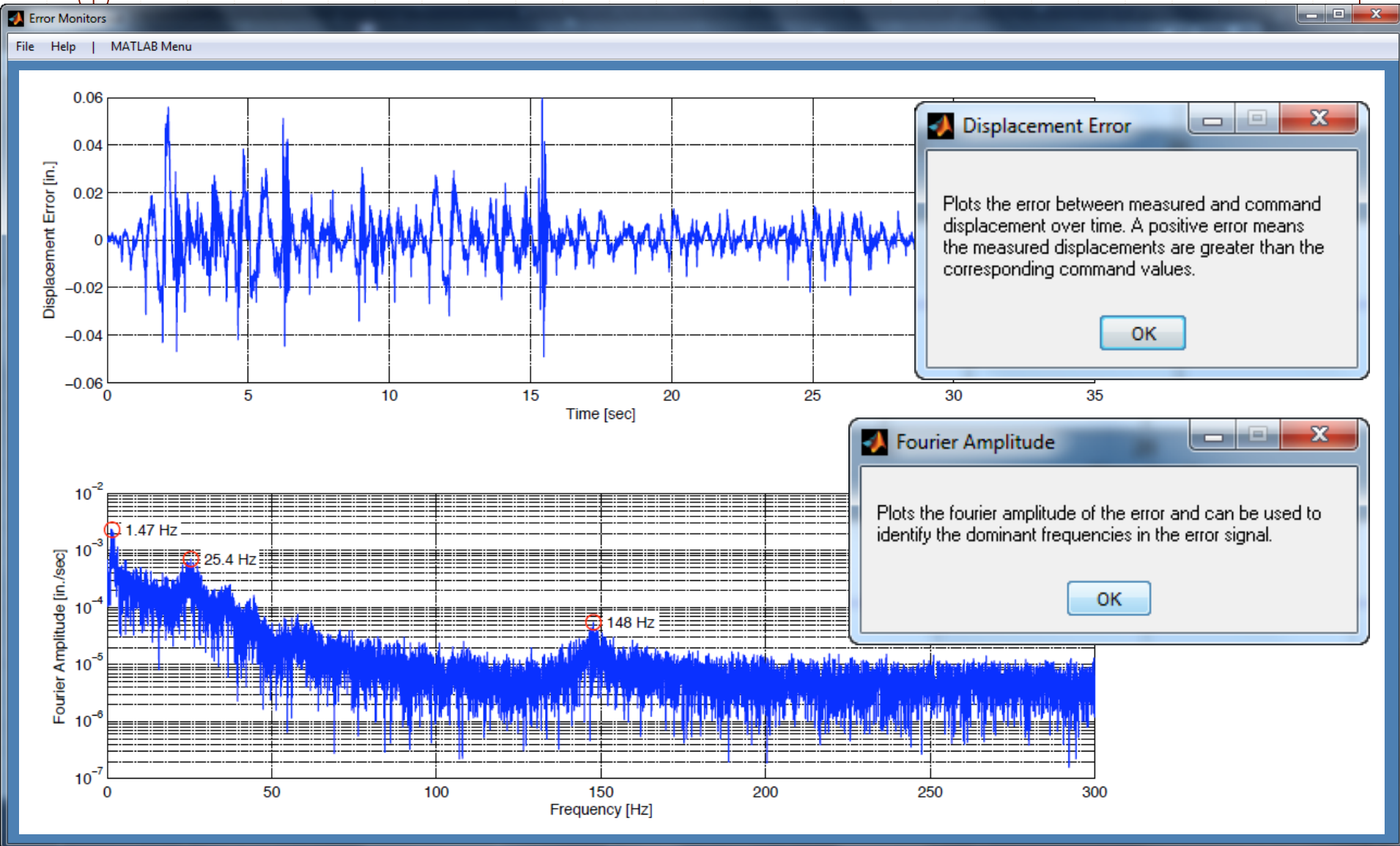
Callouts and annotations:

- A red box labeled "Select Analysis" has an arrow pointing to the "Analysis" button in the sidebar.
- A red box labeled "Click here to write TCL input files" has an arrow pointing to the "Write TCL File" button.
- A red box labeled "Optionally Generate Report" has an arrow pointing to the "Generate Report" button.
- A red box with a question mark icon contains the text: "In order to ensure stability when using the explicit Newmark method, dt must be less than the indicated maximum value." An arrow points from this box to the "dt" field.

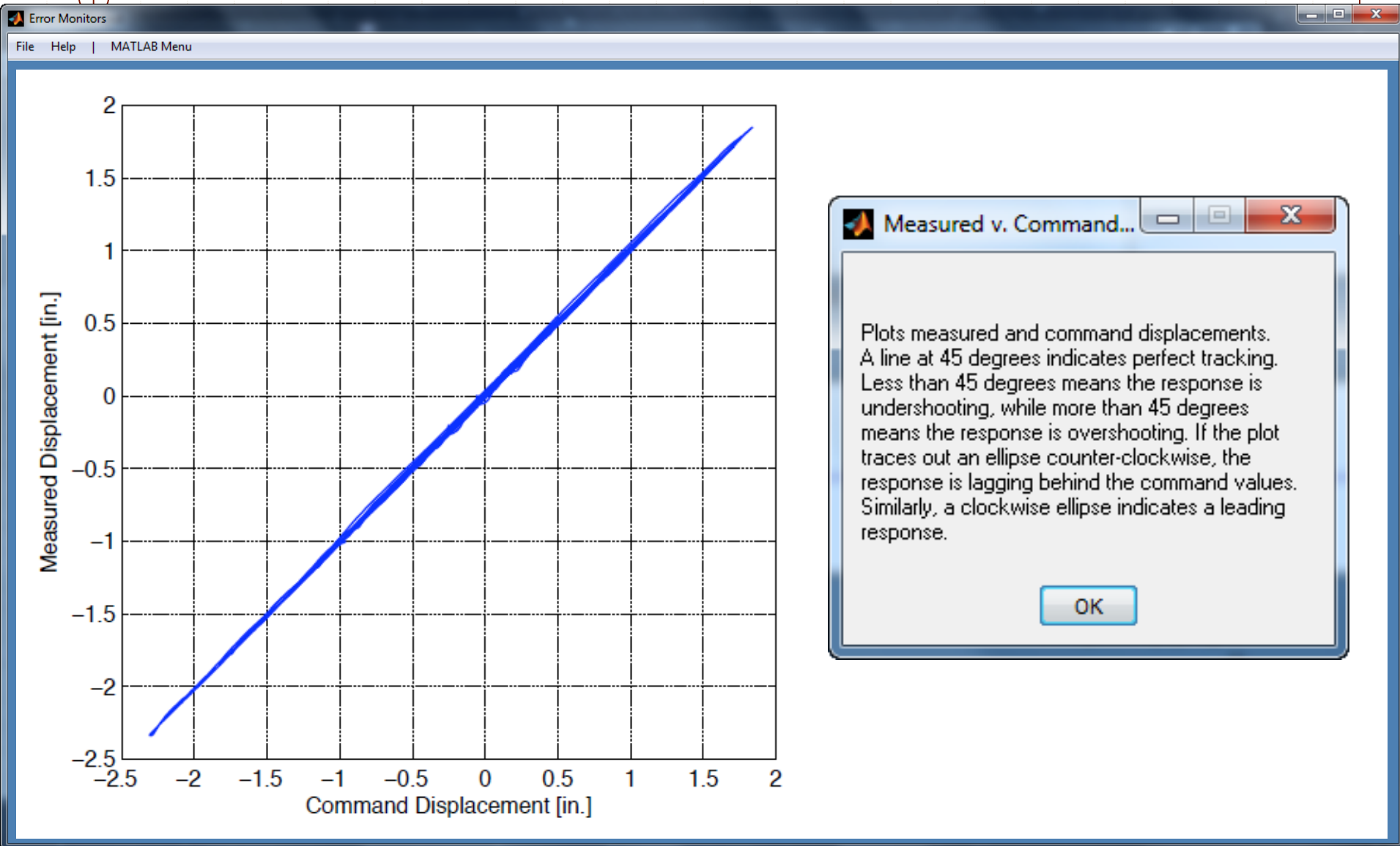
Structural Output



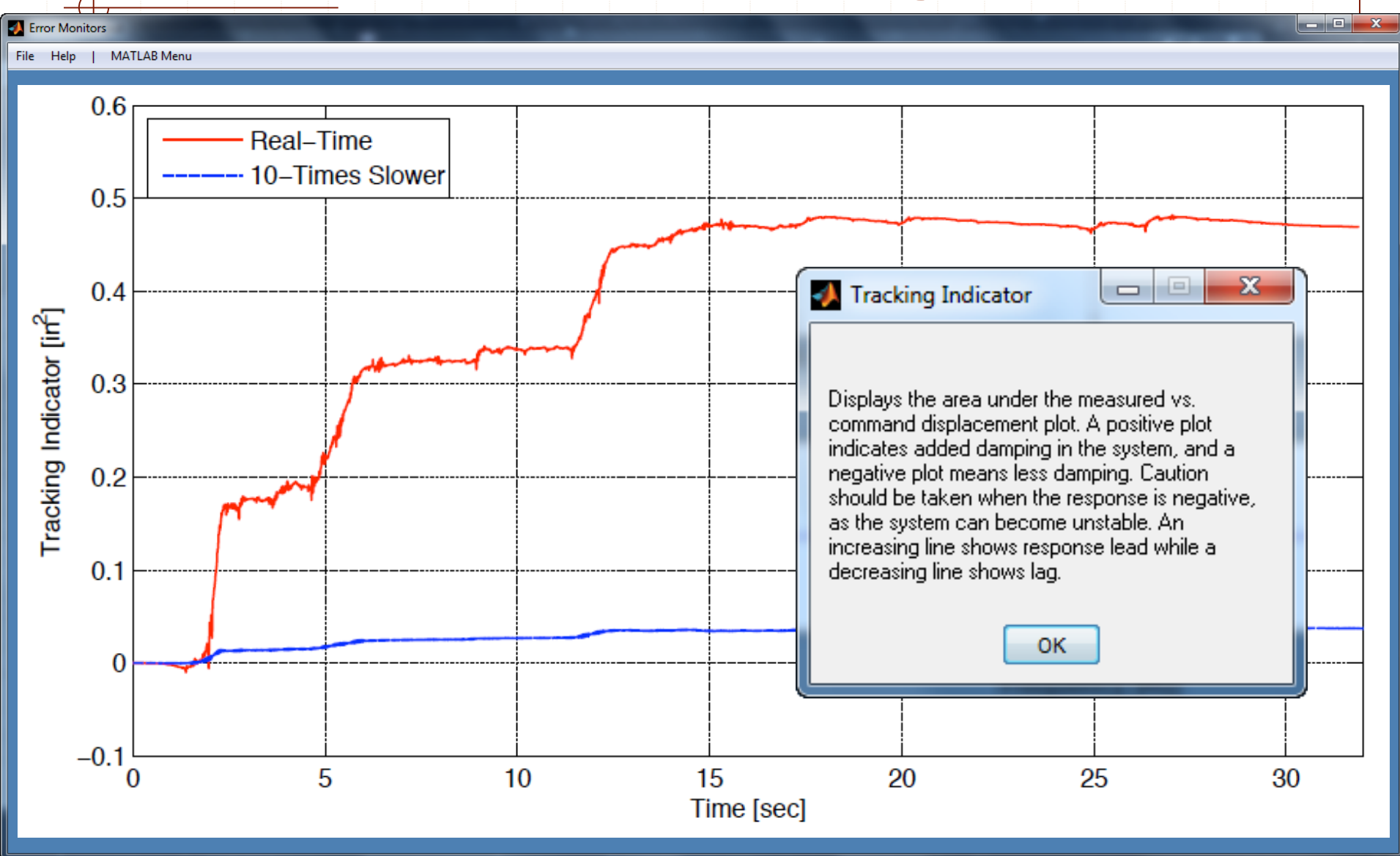
Error Monitors: dispError and FFT



Error Monitors: Subspace Plot



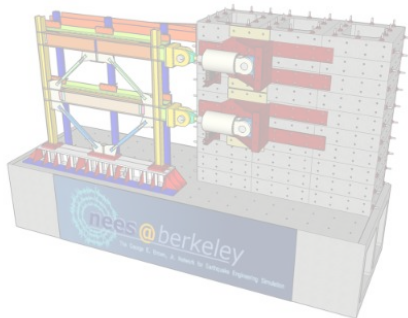
Error Monitors: Tracking Indicator



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

Search

Menu

- Log in
- Register

Downloads

- OpenFresco
- OpenFrescoExpress
- OpenSees Navigator



OpenFresco Website

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	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

http://openfresco.berkeley.edu



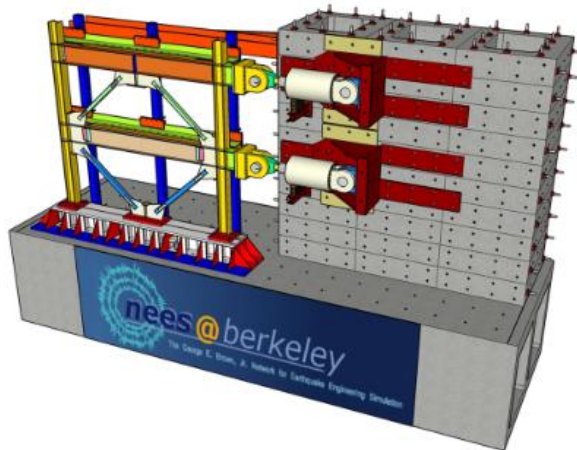
OpenFresco

[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](#)



Users



The screenshot shows the OpenFresco website. The header features the OpenFresco logo, which includes a circular image of Michelangelo's 'The Creation of Adam' and the text 'OpenFresco'. The navigation menu includes links for HOME, USERS, DEVELOPERS, DOCUMENTATION, REFERENCES, and COPYRIGHT. The 'USERS' link is highlighted, and a dropdown menu is open, showing options for OpenFrescoExpress, OpenFresco, Examples, and Feedback. The main content area is titled 'USERS' and contains a paragraph about the software, a list of user types, and a description of OpenFresco. On the right side, there is a search bar, a menu with 'Log in' and 'Register' options, and a 'Downloads' section with links to OpenFresco, OpenFrescoExpress, and OpenSees Navigator. At the bottom right, the PEER logo is displayed, and at the bottom left, contact information for the PEER Center is provided.

USERS

OpenFresco and OpenFrescoExpress are developed to allow users to conduct hybrid simulation tests. Both are designed to connect finite element models with laboratory control and data in an extensible manner to facilitate performing local and geographically distributed simulations of structural systems.

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.

OpenFresco is a robust middleware software package for performing hybrid simulations involving numerical models, test specimens, experimental setups and loading conditions that are larger and more complex than those considered by OpenFrescoExpress. It targets researchers, graduate students and laboratory staff that are more experienced in the concept and application of the hybrid simulation method. It is suited for advanced hybrid simulation when the users have analytical model and/or experimental specimen configurations that exceed the capabilities of the OpenFrescoExpress version, and/or have the desire and need to create their own custom graphical user interface.

Search

Menu

- Log in
- Register

Downloads

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

PEER

PEER Center - 325 Davis Hall, University of California, Berkeley, CA 94720-1792 - (510) 642-3437 - peer_center@berkeley.edu

Developers

The screenshot shows the OpenFresco website with the 'DEVELOPERS' menu item highlighted. Below it, the 'SVN' section is visible, containing text about the Apache Subversion (SVN) software and its use for the OpenFresco project. A blue arrow points from the 'Browsing the Code' section to a screenshot of a Subversion repository browser interface.

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.

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: 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 `../OpenFresco/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, see 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 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

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

PEER Center - 325 Davis Hall, University of California, Berkeley, CA 94720-1792 - (510) 642-3437 - peer_center@berkeley.edu

Command Language Manual



OpenFresco

HOME NEWS USERS DEVELOPERS DOCUMENTATION

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)



PACIFIC EARTHQUAKE ENGINEERING RESEARCH CENTER

Advanced Implementation of Hybrid Simulation

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

Created on

PEER 2009/104
NOVEMBER 2009

Forum



OpenFresco

[HOME](#) [NEWS](#) [USERS](#) [DEVELOPERS](#) [DOCUMENTATION](#) **[FORUM](#)** [REFERENCES](#) [COPYRIGHT](#)

FORUM

[HOME](#) > [FORUM](#)

Search


Menu

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

Downloads

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



Forum	Topics	Posts	Freshness
OpenFresco User Discussions If you have questions on how to use a certain feature of OpenFresco, or how to define OpenFresco objects in your hybrid model using the TCL language, then this is the right place to discuss them with other OpenFresco users.  PaniCesar	1	1	6 months, 2 weeks ago
OpenFrescoExpress User Discussions If you have questions on how to use a certain feature in the graphical user interface OpenFrescoExpress, then this is the right place to discuss them with other OpenFrescoExpress users.	0	0	No Topics
Hybrid Simulation Discussions Here you can discuss anything related to hybrid simulation, such as planning for a test, executing a test, assessing accuracy of results, collaborating with others, etc.	0	0	No Topics
Framework Development If you are a developer and would like to contribute to the OpenFresco source code, please discuss questions you might have or make comments here.	0	0	No Topics
Script & Model Database If you have a script that could be helpful to others conducting hybrid simulations or you have a cool model that you are really proud of and you would like to share it with the community, then please post them here.	0	0	No Topics
Simulink Models	0	0	No Topics

- OpenFresco User Discussions
- OpenFrescoExpress User Discussions
- Hybrid Simulation Discussions
- Framework Development
- Script & Model Database
- Simulink Models
- Documentation
- Bug Reports
- Brainstorming & Ideas

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.
- ★ Speed up HS from beginning of planning until end of testing

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:
 - More user feedback on refinements and new features
 - Developer contributions to extend libraries

Questions?
Thank you!



<http://openfresco.berkeley.edu/>

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