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
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

![Diagram showing typical features of an analysis framework]

- Define element as an "Experimental Element"

![Diagram showing OpenFresco and Laboratory connections]
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

- **FE-Software**
  - provides all features of unmodified computational framework, including parallel and network computing

- **Experimental Element**
  - 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

- **Experimental Site**

- **Experimental Setup**

- **Experimental Control**

- **Control System in Laboratory**
  - 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
OpenFresco Class Diagram
OpenFresco Components

- **FE-Software**
- **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
- **Experimental Site**
- **Experimental Setup**
- **Experimental Control**
- **Control System in Laboratory**

Consider element in structure
Two coordinate systems used in FE analysis

Global System

Cantilever Basic System

controlled displacements and acquired forces
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

\[ T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & L_0 \\ 0 & 1 & L_1 \end{bmatrix} \]
OpenFresco Components

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

\[
M \ddot{u}_n + C \dot{u}_n + P_r (u_n) = P(t_n)
\]

- Mass matrix \( M \) is often singular
- \( \rightarrow \) second order differential equation
- infinitely stiff \( \rightarrow \) 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
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

```plaintext
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

- **FE-Software 1**
  - GenericElement
  - Experimental Element
  - Experimental Site
  - Experimental Setup
  - Experimental Control

- **Simulink**
  - OPFConnect
  - SimFEAdapter

- **FE-Software 2**
  - AdapterElement

- **ECSimSimulink** (new experimental control that talks to Simulink model)

- **OPFConncet** in Moment Resisting Frame Subassembly in Master FE-Software

- **AdapterElement** in Shear Wall Subassembly in Slave FE-Software
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

OpenFresco Express
Release 1.0: May 2012

Developed by
Andreas Schellenberg, Ph.D., P.E.
Dep. of Civil and Env. Eng.
University of California, Berkeley

Carl Misra, M.S.
Dep. of Civil and Env. Eng.
University of California, Berkeley

Stephen Mahin, Ph.D.
Dep. of Civil and Env. Eng.
University of California, Berkeley

OpenFresco Express is supported by funds from
The Pacific Earthquake Engineering Research Center (PEER)

Copyright © 2012, PEER. All Rights Reserved.
GUI: OpenFresco *Express*

Select Structure

Period is immediately calculated

Assign Mass

Assign Stiffness

Select & Assign Damping
Define Loading: Free Vibration

Select Type of Loading

Input Free Vibration Parameters

Select Loading

- Initial Displacement
- Ramp Time
- Free Vibration Time

Graph showing displacement over time.
Experimental Setup

Select ExpSetup
Real Experimental Control

Select ExpControl

Select Real Controller

Select Type of Controller
Real Experimental Control

Input Ramp Time

Browse for MTS CSI config file
Analysis

Select Analysis

Click here to write TCL input files

Optionally Generate Report
Structural Output
Error Monitors: dispError and FFT

- **DispError**
  - Plots the error between measured and command displacement over time. A positive error means the measured displacements are greater than the corresponding command values.

- **Fourier Amplitude**
  - Plots the Fourier amplitude of the error and can be used to identify the dominant frequencies in the error signal.
Error Monitors: Subspace Plot

The graph shows the relationship between measured and command displacements. A line at 45 degrees indicates perfect tracking. Less than 45 degrees means the response is undershooting, while more than 45 degrees means the response is overshooting. If the plot traces out an ellipse counter-clockwise, the response is lagging behind the command values. Similarly, a clockwise ellipse indicates a leading response.

The dialog box explains the interpretation of the plot:

- Plots measured and command displacements.
- A line at 45 degrees indicates perfect tracking.
- Less than 45 degrees means the response is undershooting.
- More than 45 degrees means the response is overshooting.
- If the plot traces out an ellipse counter-clockwise, the response is lagging behind the command values.
- Similarly, a clockwise ellipse indicates a leading response.

OK
Error Monitors: Tracking Indicator

The graph shows the Tracking Indicator over time, with two lines representing real-time data and data that is 10 times slower. The red line represents real-time data, while the blue line represents the 10-times slower data.

The Tracking Indicator displays the area under the measured vs. command displacement plot. A positive plot indicates added damping in the system, and a negative plot means less damping. Caution should be taken when the response is negative, as the system can become unstable. An increasing line shows response lead while a decreasing line shows lag.
OpenFresco is an experimental teaching technique where data is exected 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 structural engineering community to take full advantage of this technique, OpenFresco standardizes the deployment of hybrid simulation and enhances capabilities to applications where advanced numerical techniques are utilized, boundary conditions are imposed, 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 reusability to the researcher or developer interested in hybrid simulation.

OpenFresco Website

MTS/PEER Expert Seminar, UCB 44
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.
Users

OpenFresco

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

**OpenFrescoExpress** is a self-contained software package, including an 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.
Developers

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 can use the ``svn`` command on your local machine first. You can download SVN for all major operating systems, including Linux, Windows, and MacOSX. If you are not familiar with the command line, SVN is particularly nice and easy to use. It lets you control SVN functions via menus as you navigate the file system in Windows Explorer.

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

dir $HOME/local

tmp_dir=$HOME/tmp

cd $tmp_dir

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

cd openfresco

The checkout command makes a local copy of the entire OpenFresco tree in your current working directory. By requesting ./OpenFresco/trunk, which should have the latest stable source code.

Browsing the Code

You can browse the source code online using WebSVN. The tool provides a web-based interface to the OpenFresco repository that has been developed with the use of the latest version of the Subversion software. You can view the log of any file or directory of the repository, and see which files have been changed, added or deleted in any given revision. You can also compare two versions of a file to see exactly what has changed in a particular revision.
Command Language Manual

Advanced Implementation of Hybrid Simulation

Andreas H. Schellenberg
Stephen A. Mahin
Gregory L. Fenves
University of California, Berkeley
<table>
<thead>
<tr>
<th>Forum</th>
<th>Topics</th>
<th>Posts</th>
<th>Freshness</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OpenFresco User Discussions</strong></td>
<td>1</td>
<td>1</td>
<td>6 months, 2 weeks ago</td>
</tr>
<tr>
<td><strong>OpenFrescoExpress User Discussions</strong></td>
<td>0</td>
<td>0</td>
<td>No Topics</td>
</tr>
<tr>
<td><strong>Hybrid Simulation Discussions</strong></td>
<td>0</td>
<td>0</td>
<td>No Topics</td>
</tr>
<tr>
<td><strong>Framework Development</strong></td>
<td>0</td>
<td>0</td>
<td>No Topics</td>
</tr>
<tr>
<td><strong>Script &amp; Model Database</strong></td>
<td>0</td>
<td>0</td>
<td>No Topics</td>
</tr>
<tr>
<td><strong>Simulink Models</strong></td>
<td>0</td>
<td>0</td>
<td>No Topics</td>
</tr>
</tbody>
</table>
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.