APPLICATION OF AN EXPERIMENTAL SOFTWARE FRAMEWORK FOR INTERNATIONAL HYBRID SIMULATION

Andreas Schellenberg, Stephen A. Mahin and Gregory L. Fenves

ABSTRACT

Hybrid simulation provides a versatile, realistic and cost-effective method for simulating seismic response experimentally. However, hybrid simulation has not yet achieved its full potential. Although hybrid simulation is undergoing widespread development worldwide, each site generally follows a different approach to implementation. As such, it is difficult for researchers to work together to accelerate overall progress and to undertake important geographically distributed experiments that can take advantage of unique experimental, computational and intellectual resources worldwide. This paper describes the continuing evolution and a testbed application of Open Framework for Experimental Setup and Control (OpenFresco), an environment independent, modular, and open source software framework for deploying hybrid simulation worldwide in a robust, transparent, scalable and easily extensible fashion (Takahashi and Fenves, 2005). OpenFresco is based on a system’s analysis of common operations and services needed to implement hybrid simulations locally as well as across a wide area network. It utilizes object-oriented software design methodologies to define a set of interrelated software classes, which in their unity form a framework for hybrid simulation. While OpenFresco is independent of the finite element analysis software used to carry out the computational aspects of the hybrid simulation, it supports applications that allow the user to take direct advantage of the interface, computational, visualization and other capabilities of modern analysis frameworks to model and execute a hybrid simulation. This is achieved by permitting those portions of the structure to be tested to be represented as special finite elements within the computational framework used. A simple, but important example of this approach is presented in which the dynamic response of a one-story, one-bay frame is simulated through collapse.

Keywords: Hybrid simulation, dynamic analysis, test methods, collapse, geometric nonlinearities

INTRODUCTION

Experiments are an essential aspect of understanding the effects of earthquakes on the built environment, and for developing and evaluating design and analysis procedures for use in earthquake engineering. At present, three well-established experimental methods are employed. In the most common method, actuators impose a predefined history of loads or displacements on the test specimen being investigated. By imposing the same load or displacement history on a series of specimens, the effect of systematic changes in material properties, details, boundary conditions, loading rates, and other factors can be readily identified. While such tests are relatively easy and economical to perform, the demands imposed on the test specimens are not directly related to the damage observed, raising questions about whether the specimens were under- or over-tested for project specific situations.

Shaking table tests are a second form of tests, and they are able to produce conditions that closely resemble those that would exist during a particular earthquake. These tests provide important data on dynamic response to specific ground motions considering the inertial and damping characteristics of the structure tested and the consequences on response of geometric nonlinearities and localized yielding and damage. For shaking table tests, a complete structural system is required, and the specimen needs to be constructed carefully following the rules of dynamic similitude. The limited

1 Graduate Student, Dept. of Civil and Env. Eng., Univ of California, Berkeley, CA, USA, andreas.schellenberg@gmx.net
2 Byron and Elivra Nishkian Prof., Dept. of Civil and Env. Eng., Univ of Calif., Berkeley, CA, mahin@berkeley.edu
3 T.Y. and Margaret Lin Prof., Dept. of Civil and Env. Eng., Univ. of Calif., Berkeley, CA, fenves@berkeley.edu
capacity and size of most available shaking tables place significant restrictions on the size, weight and strength of a specimen that can be tested. As a result, reduced scale or highly simplified specimens are commonly necessitated, which call into question the realism of many shaking table tests.

The third method is hybrid simulation, where the simulation is performed based on a step-by-step numerical solution of the governing equations of motion for a model formulated considering both numerical and physical components. As implemented for a typical quasi-static, displacement-based hybrid simulation, the mass and viscous damping characteristics of the specimen are numerically modeled, and the incremental displacement response of the structure to a specified ground motion is computed at each step based on the current state of the physically and numerically modeled portions of the structure. The physically represented portions of the overall hybrid model can be tested in one or more laboratories using computer-controlled actuators. The hybrid simulation method therefore gives the researcher the ability to use large reaction wall and actuator facilities to simulate large specimens or to subdivide a large structure into portions where the behavior is either (1) well understood and can be modeled reliably using finite element models, and (2) highly nonlinear and/or numerically difficult to simulate and are thus physically simulated (Campbell and Stojadinovic, 1998). Since dynamic aspects of the simulation are handled numerically, such tests can be conducted quasi-statically using standard computer-controlled actuators. As such, hybrid simulation may be viewed as an advanced form of actuator based testing but where the loading history is determined during the course of an experiment for a given system subjected to a specific ground motion. Alternatively, hybrid simulation can also be considered as a conventional finite element analysis where physical models of some portions of the structure are embedded in the numerical model.

Considerable research is being conducted worldwide to extend hybrid simulation to applications where advanced numerical techniques are utilized (high performance computing, including parallel processing and realistic treatment of three-dimensional effects, soil-structure interaction, multiple support excitations, geometric nonlinearities, etc.), deformations are imposed dynamically, and dynamic loading conditions are caused by wind, blast, impact, waves and traffic. Similarly, efforts are underway to optimize capabilities for testing portions of the structure in geographically distributed laboratories, including techniques to overcome network quality of service deficiencies. Fundamental studies are underway related to improving or extending approaches used to define and solve the equations of motion, and characterizing and correcting for errors associated with numerical modeling, numerical simulation, experimental control, scaling, strain rate effects and network communication loss and latency.

While only a few basic operations and communication protocols are needed to carry out a hybrid simulation, to date few efforts have been made to develop a common framework for implementing and deploying systems that can carry out hybrid tests (e.g., Takahashi and Fenves, 2005; Kwon et al, 2005). Typically, each implementation of hybrid simulation is problem specific and is strongly dependent on the testing site and the actuator control system used. Such highly customized software implementations are difficult to adapt to different structural problems and even harder to port to different laboratories. Thus, the lack of a common framework poses a hurdle for those who wish to learn how to conduct a hybrid simulation and has limited collaboration among experts in the field. In this paper, several improvements to the OpenFresco framework for hybrid simulation are described, and its use in carrying out tests that have been heretofore difficult to conduct is presented.

**EXPERIMENTAL SOFTWARE FRAMEWORK - OpenFresco**

To accelerate the development and refinement of hybrid simulation techniques, there is a need for an environment independent software framework that is robust, transparent, scalable and easily extensible. The framework should allow domain researchers to carry out hybrid simulations without specialized knowledge about the underlying software. It should also enable hybrid simulation and IT specialists to extend the frontiers of the methodology, by permitting the addition of new control systems, integration methods, communication strategies, and computing resources. Lastly, it should support use of advanced computing features of modern finite element analysis methods.

To achieve these desired attributes, the software framework should have a very modular architecture. By properly structuring this modularity, modules can be modified and new ones added without much dependence on other modules. Since such modularity is the core of object-oriented software design methodologies, Takahashi and Fenves (2005) proposed a set of interrelated software classes, which in their unity form a framework for experimental testing. `ExperimentalElement`, `ExperimentalSite`, `ExperimentalSetup` and `ExperimentalControl` classes define the various operations, data, and
relationships needed during a hybrid simulation to provide a bridge between a standard finite element analysis program and laboratory control and data acquisition systems, as schematically shown in Fig. 1.

![Diagram](image)

Figure 1: OpenFresco software framework components configured for local deployment

OpenFresco is independent of the finite element (FE) software used. However, for its ideal realization, the software must allow the addition of new elements. As such, Abaqus (Abaqus, 2006), LS-Dyna (Livermore Software, 2006), Matlab (Mathworks, 2006), OpenSees (Fenves et al., 2004), ZeusNL (Elshahi, Papanikolaou and Lee, 2006) and similar programs can in principle be used with OpenFresco. Where specialized numerical integration operators are needed for the type of hybrid simulation planned, the finite element software must also allow new integration operators to be added.

In the illustrative examples presented, the software framework Open System for Earthquake Engineering Simulation, OpenSees (McKenna, 1997; Fenves et al., 2004) is used. Other FE software could be employed in a similar manner. Because of its adherence to an object oriented design methodology, OpenSees can be easily used with OpenFresco to implement and extend hybrid simulation technologies. No changes are required to any existing classes in OpenSees, except for augmenting the existing `Element` class with the subclass `ExperimentalElement` and adding the numerical integration operators needed for the hybrid simulation. The `ExperimentalElement` objects act within the FE software to represent truss, beam, column, or other parts of the structure that are physically tested and provide the interface between the FE analysis core and the experimental software framework OpenFresco, as described by Schellenberg and Mahin (2005).

OpenFresco handles three common, repetitively imposed sets of tasks needed to implement computer-controlled tests. The first involves transforming actions at the boundary nodes for the experimental elements (e.g., displacement, velocity, acceleration or force) from the coordinate system used by the FE software to those used in the laboratory. Objects implemented within the `ExperimentalSetup` class are thus responsible for transforming the degrees-of-freedom for each `ExperimentalElement` into actuator degrees-of-freedom, utilizing the geometry and the kinematics of the loading and instrumentation system, and back again. These transformations can be implemented either as simple linear transformation matrices (small displacements) or they can be implemented as complex algebraic transformations taking large displacement effects into account. The second basic task involves communicating with the laboratory control and data acquisition systems so that the command actions in the actuator coordinate system are imposed and measured ones are returned to the finite element program. The OpenFresco `ExperimentalControl` class is responsible for interfacing with specific laboratory control and data acquisition systems and performing these functions. The advantage of this abstraction and the encapsulation of these operations is that the `ExperimentalSetup` class separates the details of the loading system configuration from the `ExperimentalControl` class. Thus, those responsible for the IT aspects of the control and data acquisition systems need only be concerned with the `ExperimentalControl` class; while those configuring the actuators and sensors can focus on the `ExperimentalSetup` class. To enable geographically distributed testing, the `ExperimentalSite` class provides the services for communicating between the experimental site (as a server) and the computational software (as a client). Other classes are available for utilizing communication protocols between the two, such as standard TCP/IP sockets or specialized protocols such as NTCP, the NEESgrid Tele-operations Control Protocol (Pearlman et al., 2004). A `LocalExpSite` subclass is available for a local (non-networked) implementation, as schematically shown in Fig. 1. A more detailed class diagram (using the unified modeling language) is provided in the Appendix.
Because of the modularity and flexibility of the OpenFresco architecture, a wide variety of geographically distributed testing configurations are possible. Four of several basic configurations are illustrated in Figure 2, where the large box in the upper part of the figure resembles the client program and the four smaller boxes in the lower part of the figure resemble server programs. Client and server communicate with each other by using one of several available communication protocols. The first two configurations in Figure 2 illustrate use of TCP/IP Socket for communication. The transformation from the element degrees-of-freedom to the actuator degrees-of-freedom (e.g., ExperimentalSetup) is either performed on the server side or on the client side. In the other two configurations shown, the NTCP protocol is used for communication; and, depending on the control plug-in used, the experimental setup transformations take place on the client side or inside the control plug-in on the server side. With the flexibility of OpenFresco, any of the existing control plug-ins (e.g., the Matlab plug-in, the Labview plug-in, the Shore-Western plug-in and the Fresco plug-in) can be used on the NTCP server.

![OpenFresco software framework components configured for distributed deployment](image)

**Figure 2:** OpenFresco software framework components configured for distributed deployment

Several specific ExperimentalElement, ExperimentalSite, ExperimentalSetup and ExperimentalControl classes are already available. To facilitate integrated use of OpenFresco with OpenSees, a scripting language based on Tcl has been implemented for all of these objects. It should be understood that with appropriate realizations of ExperimentalElement, ExperimentalSite, ExperimentalSetup and ExperimentalControl objects, portions of the overall hybrid model can be modeled and simulated numerically using the special capabilities of different FE software packages.

![Examples of OpenSees experimental element objects](image)

**Figure 3:** Examples of OpenSees experimental element objects

As can be seen from Fig. 3, four OpenSees Experimental Elements are available thus far. The experimental truss element has one axial degree-of-freedom and can be used in 2D- and 3D-problems. Depending on the dimension of the problem selected, the experimental beam-column element formulation is based on the three or six collocated degrees-of-freedom of the cantilever basic system.
The experimental zero-length element has up to six collocated degrees-of-freedom and can, for example, be used to test plastic hinges. Finally, the experimental chevron brace subassembly element has a total of nine degrees-of-freedom that are not collocated. This element highlights the versatility of the approach followed by having displacements imposed on the three degrees-of-freedom at node k and forces are obtained on the six degrees-of-freedom at nodes i and j. These realizations provide a template for users to extend the range of available experimental elements.

Available OpenFresco experimental setups are shown in Fig. 4. The no-transformation setup assumes that the control system automatically takes care of all geometric transformations and therefore sends the experimental element response quantities without any transformation to the experimental control object. It is obvious from Fig. 4 that the one-, two- and three-actuator setups are able to control one, two or three element degrees-of-freedom, respectively. The last experimental setup was developed specifically for a chevron brace subassemblage test. Three actuators, connected by a large (nearly rigid) steel beam, are used to control the three degrees-of-freedom at the top node and two VPM load cells are placed at the two bottom nodes of the specimen in order to measure resisting forces. Linear and nonlinear geometric transformations have been implemented for most of the experimental setups.

![Figure 4: OpenFresco experimental setup objects](image)

Experimental control class implementations in OpenFresco are available for dSpace and xPCTarget digital signal processors and for the National Instruments E-Series AD/DA boards (Fig. 5). The implementation of experimental control classes to interface with the Systran shared common RAM network (SCRAMNet) and MTS FlexTest controllers is under development.

![Figure 5: OpenFresco experimental control objects](image)

The modularity and transparency of OpenFresco allows users to easily add new experimental control, experimental setup and experimental element objects as required. A stand-alone graphical user interface (Schellenberg and Yang, 2005) enables users to define all aspects of the simulation, control execution of the experiment, and view results before, during and after the test.

**HYBRID SIMULATION OF STRUCTURAL COLLAPSE**

One of the advantages in hybrid simulation is that geometric nonlinearities, three-dimensional effects, multiple support excitation and soil-structure interaction can be investigated by incorporating them into the analytical portion of the hybrid model. To demonstrate how geometric nonlinearities can be accounted for in the numerical part of the model, a hybrid simulation, wherein a portal frame is tested by consistently accounting for second-order effects due to gravity loads, is carried out using OpenSees and OpenFresco. Since the second-order effects are caused by gravity loads, it is important to understand how such loads affect the response of a structure and what the limitations can be imposed.
by the hybrid simulation test-setup. In general, gravity loads influence the response of a structure on three levels that can be categorized as shown in Table 1.

<table>
<thead>
<tr>
<th>Level</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Section</td>
<td>Gravity loads cause axial forces that interact with bending moments and shears at the cross section, commonly known as P-M and P-V interaction.</td>
</tr>
<tr>
<td>Element</td>
<td>Gravity loads affect the stability of structural elements and critical regions, causing softening and local and/or lateral-torsional buckling of elements, commonly known as the small P-Delta (P-δ) effects.</td>
</tr>
<tr>
<td>Structure</td>
<td>Gravity loads influence the stability of a structure, causing additional overturning moments due to large displacements. These effects are commonly known as the large P-Delta (P-Δ) effects and are captured by a geometric stiffness matrix.</td>
</tr>
</tbody>
</table>

The gravity loads on the simple one-story, one-bay portal frame shown in Fig. 6 are to be entirely treated in the analytical portion of the hybrid model, with no axial loads imposed on the experimental parts of the structure. Thus, only large P-Delta effects are accounted for in this particular example.

The hybrid model of the steel single-story, single-bay portal frame consists of a numerically modeled elastic beam (W6x12) and two physically tested S4x7.7 beam-columns. The connecting beam, the viscous energy dissipation and mass properties as well as the gravity and earthquake loads are modeled analytically in OpenSees. The Experimental Element EEBeamColumn (see Fig. 3) is used in OpenSees to represent the two columns. For convenience in the lab, the two beam-column specimens are inverted with the pin end at the top and the fixed end at the bottom. Special clevises incorporating replaceable steel coupons are used to produce stable bilinear hysteretic behavior in the plastic hinges.

This experimental beam-column element is formulated in the element’s basic coordinate system without any rigid body modes. As with other analytical elements in OpenSees, the geometric transformation object in OpenSees is then used to transform displacements from the global to the basic coordinate system and to transform forces and stiffness from the basic back to the global coordinate system. Three geometric coordinate transformations are available in OpenSees: (1) the Linear transformation, (2) the PDelta transformation that accounts for the large P-Delta effects due to the axial load in the element, and (3) the Corotational transformation which is a second-order large displacement formulation that performs an exact transformation of the displacements, resisting forces and stiffness between the two coordinate systems (de Souza, 2000; Filippou, 2004). In the portal frame example, the PDelta transformation as well as the Corotational transformation are employed to compare methods for treating gravity load effects in hybrid simulations. It is important to emphasize that because the basic coordinate system moves as the elements deform, the two experimental ESOOneActuator setups (see Fig. 4) utilized in these tests also behave as if they were attached to and move with the elements.
Considering the axial and flexural deformations in the elements, the portal frame model has a total of 8 degrees-of-freedom (4 translational and 4 rotational). We will assume that only the two horizontal degrees-of-freedom have mass \((m = 0.0149 \text{ slug})\), which makes the 8x8 mass matrix singular with rank equal to two. To achieve an unconditionally stable direct integration method for the hybrid simulation, the Newmark method with a constant number of 4 sub-steps and the Newton-Raphson algorithm are employed. The fundamental period of the structure is 0.458 sec and the viscous energy dissipation is modeled considering 2% mass proportional damping. The magnitude of the gravity loads \((P = 5.75 \text{ kips})\) is set at 50% of the critical side-sway buckling load of the S4x7.7 columns assuming that the beam is flexurally rigid and the base supports for the columns are pinned \((k = 2)\). The hybrid portal frame model is subjected to the 1978 Tabas earthquake scaled to a peak ground acceleration of 0.755g. The integration time step is chosen to be \(\Delta t_{\text{int}} = 0.01\) seconds. On the other hand, the simulation time step is chosen to be \(\Delta t_{\text{sim}} = 2^5 = 0.0625\) seconds and in combination with the constant number of 4 sub-steps per time step, a time scale factor of 25 is obtained. This means that the hybrid simulation is performed 25 times slower than real-time and each test lasts 8.3 minutes.

To investigate the effect of gravity loads on the response of the portal frame, two pairs of simulations are performed. One simulation is performed without any gravity loads applied, and then gravity loads are added to the analytical portion of the hybrid model for the second test. Both tests use the \textit{PDelta} geometric coordinate transformation object. The story-drift time-histories as well as the story shear-drift hysteresis loops are compared in Fig. 7. Even though the frame drifts and the residual displacements are substantial for the first test without gravity loads, the hybrid model does not collapse. In the second hybrid test, the presence of the gravity loads is clearly evident from the negative post-yield tangent stiffness exhibited by the combined hybrid model. As can be seen from Figure 7a, story drifts increase rapidly at 6.5 seconds in the test which can be considered as the initiation of frame collapse. With each additional cycle, the portal frame shifts further over until the hybrid simulation is finally terminated when the actuators reach their stroke limit of 7.5 inches, which corresponds to an inter-story drift ratio of 13.9%. The second simulation demonstrates that the hybrid model can be tested all the way to collapse of the portal frame, which is difficult or dangerous to accomplish in shaking table tests.

![Figure 7: Comparison of story-drift time-histories and story shear-drift hystereses for tests with PDelta geometric transformations](image)

Figure 8 compares hysteresis loops for the two experimental beam-column elements. The shear-force shear-deformation relationships are plotted in the basic cantilever coordinate system as they are acquired from the laboratory, before the gravity load effects are applied by the geometric transformation object. Thus, no negative post-peak stiffnesses are observed in the actual test. The differences between the hysteresis loops of Test 1 and Test 2 are entirely due to the consistently applied displacements. Furthermore, it can be seen that for large displacements of the cantilever column tips (respectively large rotations of the clevises) the hystereses exhibit some degradation. This degrading behavior is due to the local buckling of the replaceable steel coupons as the clevis goes...
through large rotations. The last large drops of the shear forces are due to the self-shutdown of the actuator control system when the preset displacement limits of 7.5 inches are exceeded.

Figure 8: Comparison of element hystereses in basic coordinate system for tests with P-Delta geometric transformations

In a second series of tests, the PDelta geometric coordinate transformation in the hybrid model is replaced by the Corotational transformation. The story-drift time-histories as well as the story hysteresis loops are compared in Fig. 9. The effect of the gravity loads is again evident from the large increase of the inter-story drift ratio at 6.5 seconds as well as from the negative post-peak stiffness. Even though the initiation of collapse occurs at the same time as in the tests with the PDelta transformation, the strength degradation with each cycle is less pronounced when using the exact Corotational transformation. The actuator displacement limits of 7.5 inches in this test are not reached after 20 seconds of earthquake simulation.

Figure 9: Comparison of story-drift time-histories and story shear-drift hystereses for tests with Corotational geometric transformations

Figure 10 shows the comparison of the hystereses of the two experimental beam-column elements for the hybrid simulations using the Corotational coordinate transformations. The shear-force shear-deformation relationships are again plotted in the basic cantilever coordinate system as they are acquired from the laboratory, before the gravity load effects are applied by the Corotational transformation objects. Again, no negative post-yield tangent stiffnesses are observed in the actual tests. The differences between the hysteresis loops of Tests 1 and 2 are entirely due to the applied displacements. Because the degradation is less severe with the Corotational transformation, the actuator limits of 7.5 inches are never reached.
The ability to correctly account for the second-order effects in hybrid models is crucial for simulating collapse of structures under gravity loads. Hybrid simulation of collapse behavior offers three significant advantages over conventional testing methods: 1) the gravity loads and the resulting geometric nonlinearities are represented in the analytical portion of the hybrid model in the computer, eliminating the need for complex active or passive gravity load setups; 2) there is no need to protect expensive test equipment from specimen impact during collapse because the actuator control system will limit the movements of the test specimens; and 3) only the critical, collapse-sensitive, elements of the structure need to be physically modeled allowing for a substantial increase in the number of different collapse tests afforded by the same budget.

**CONCLUSIONS**

The OpenFresco framework for setup and control of experiments provides a structured set of modules for implementing and deploying hybrid simulation applications. By using software modules to isolate the various tasks that need to be carried out in a test, site personnel can focus on the ExperimentalControl classes needed to operate a particular site. Those conducting tests can focus primarily on the ExperimentalSetup and the numerical model of the specimen to be tested. With the concept of ExperimentalElements, the full computational capabilities of modern simulation frameworks, including parallel and network computation, can be brought to bear to define and control the hybrid simulation. As such, OpenFresco provides a mechanism that can encourage synergistic collaboration at a national and international level.

An example is presented to indicate the benefit of using a general-purpose computational platform, and some of the practical and safety advantages of conducting hybrid simulations. The example, suggests that hybrid simulation provides a versatile, convenient and cost effective means of investigating complex aspects of dynamic behavior.

**ACKNOWLEDGMENTS**

Support for this work was provided in part by the Earthquake Engineering Research Centers Program of the National Science Foundation, under Award Number EEC-9701568, through the Pacific Earthquake Engineering Research Center (PEER). The authors thank Prof. Yoshikazu Takahashi of Kyoto University and Dr. Frank McKenna of UC Berkeley for initiating and cooperating with the OpenSees and OpenFresco developments, and Dr. Tony Yang, Profs. Bozidar Stojadinovic (UC Berkeley) and Ken Elwood (Univ. of British Columbia) for their help with the portal frame example.
APPENDIX

Figure 11: OpenFresco class diagram

REFERENCES


