범용 유한요소해석 프로그램을 이용한 분산 공유 하이브리드 해석 및 실험 시스템

Distributed Hybrid Simulation and Testing System using General-Purpose Finite Element Analysis Program

윤 군 진† 한 봉 구*

Yun, Gun-Jin Han, Bong-Koo

(논문접수일 : 2007년 10월 6일 ; 심사종료일 : 2007년 11월 13일)

요 지

본 논문에서는 지진하중하의 대형구조물의 시뮬레이션을 위해 실험과 해석을 병합한 분산공유 하이브리드 해석 및 실험 소프트웨어 framework를 개발하였다. 제안된 소프트웨어 framework은 별도의 동적 그리고 정적 해석을 위한 프로그램의 개발이 필요없기 때문에 일반 범용 유한요소해석 프로그램을 개발된 해석 및 실험 제어 프로그램과 interface API를 이용하여 사용할 수 있는 장점이 있다. 본 논문에서 개발된 소프트웨어 framework은 독자적 기능을 가진 module로 구성이 되어 있을 뿐만 아니라 개체지향형 프로그램 개념을 바탕으로 개발 되었다. 예제를 통하여 개발된 시스템의 기능과 분산공유 하이브리드 해석 및 실험에서의 유용성을 증명하였다.

핵심용어 : 하이브리드 해석 및 실험, 유한요소해석, 유사동적실험, 부구조기술

Abstract

In this paper, a software framework that integrates computational and experimental simulation has been developed to simulate and test a large-scale structural system under earthquake loading. The proposed software framework does not need development of the computer codes for both dynamic and static simulations. Any general-purpose software can be utilized with a main control module and interface APIs. This opens up a new opportunity to facilitate use of sophisticated finite elements into hybrid simulation regime to enhance accuracy and efficiency of simulations. The software framework described in the paper is modular and uses object oriented programming concepts. A series of illustrative examples demonstrate that the system is fully-functional and is capable of running any number of experimental and/or analytical components.

Keywords : *hybrid simulation and testing, finite element analysis, pseudo-dynamic testing, sub-structuring technique*

1. Introduction

Distributed hybrid testing methods(also called pseudo-dynamic test) are considered to be quite effective for evaluating the dynamic response of structures as they combine numerical simulation of a complex structure with experiments. With recent advancements in fast hybrid test, the potential application has been broaden greatly. For example, real-time hybrid test enables experimental verification of control devices such as MR or viscous dampers applied to structures showing nonlinear behavior.

Sub-structured hybrid testing method is a hybrid

† 책임저자, 정회원 · Washington University 기계항공구조공학과 박사후연구원
Tel: 02-970-6578 ; Fax: 02-975-6002
E-mail: gunjinyun@gmail.com
* 정회원 · 서울산업대학교 건설공학부 교수

• 이 논문에 대한 토론을 2008년 4월 30일까지 본 학회에 보내주시면 2008년 6월호에 그 결과를 게재하겠습니다.

computational and experimental method. Parts of the structure are numerically modeled through finite element method and its equation of motion combined with restoring forces coming from experimental component is solved by numerical time integration method. To obtain the restoring forces, parts that show complex behavior that can not be easily modeled are constructed and tested quasi-statically in laboratory.

In the early stages of the development of the hybrid test(Mahin Stephen *et al.*, 1985; Shing *et al.*, 1985) only simplified numerical models with a limited number of degrees of freedom were used. However, it is obviously advantageous to use advanced numerical models for efficiently simulating earthquake response of complex structures. Also tested parts can be eventually replaced by reliable numerical models to save costs. Recently, some software frameworks have been developed for applying advanced finite element analysis platforms in hybrid simulation and testing. For example, UI-SIMCOR has been developed to accommodate various finite element analysis platforms like, ZEUS-NL, OpenSees and ABAQUS(Kwon *et al.*, 2005) because they exhibit different strengths and weaknesses. In this approach, static analyses are conducted using different analysis platforms. Otherwise, actual experiment is conducted instead of the static analysis. The static analysis models transfer restoring forces at active dynamic degrees of freedom to dynamic analysis module. Therefore, a separate dynamic analysis program is required to numerically integrate the dynamic equation of motion. Moreover, because restart functionality of the static analysis is required, modification of the source code could be required depending on the software used. However, the modification of source code is not readily accomplished with ordinary users. In 2006, Wang, et al. also proposed an identical approach to utilize a commercial finite element code, ABAQUS (Wang *et al.*, 2006). However, it can not utilize advanced capabilities in dynamic analysis from the general-purpose finite element codes. In 2006, Takahashi, et al. developed an object-oriented software framework for distributed hybrid simulation and testing(Takahashi *et al.*, 2006). In the framework, OpenSees is used as a numerical integrator for dynamic simulation as well as a simulation coordinator that provides the co-ordination of the hybrid simulation. In particular, a special element, named the experimental element, was added into the source code of OpenSees, which requires a modification in Open Sees.

In this paper, a new software framework for the distributed hybrid simulation and testing has been developed. Unlike other software frameworks, the proposed software framework does not need any development of computer codes for both dynamic and static simulations. So any general-purpose finite element analysis code can be integrated with a main control module and customized interface APIs without modification of its source code as a dynamic computational module. Key function required from general-purpose finite element analysis codes is restarting option to accomplish such integration. The software framework consists of four modules: main control module(Server), computational dynamic module(Client), experimental module and Computational static module(Client) with Interface API. They can communicate with each other using a standard TCP/IP protocol in network environment. Two numerical integration schemes are implemented in the software framework: Predictor-Corrector method(Ghaboussi *et al.*, 2006) and α-Operator Splitting (OS) method(Nakashima *et al.*, 1990). It opens up a new opportunity to facilitate use of advanced numerical models within hybrid simulation regime for advanced dynamic and static models. The software framework allows the seamless integration of multiple physical and numerical simulations of structural and geotechnical components within a unified simulation of a full-scale system such as a bridge or a building with SFSI effects. In the future, given the appropriate network bandwidth and computational speed, fast rate hybrid test as well as real time dynamic tests can be incorporated. A future development of the software framework will include intelligent on-line model updating of material and structural component response in the numerical simulation based on physical component testing results, Yun *et al.*(2006, 2007).

2. DESIGN OF SOFTWARE FRAMEWORK AND IMPLEMENTATIONS

In this section, distinct features of the proposed software framework are described with respect to integration algorithm, configuration of software framework, implementations for network communication and sub-structuring techniques. Particularly, the sub-structuring technique that maintains dynamic characteristics is efficient in simulation of a large-scale ground structure because physical testing of such structures including soil-foundation- structure interaction(SFSI) remains extremely space- and equipment-intensive as well as costly. Section 2.1 introduces α-OS algorithm used for examples is introduced and explains how it can fit into the proposed software framework with a general- purpose finite element analysis platform, ABAQUS without modification of its source codes.

2.1 Integration Algorithm for Hybrid Simulation and Testing: α-OS method

Desirable integration algorithm for hybrid simulation and testing would be non-iterative, unconditionally stable and would not require initial stiffness values of experimental components for adjusting restoring forces because it can induce systematic error in severe nonlinear behavior. In α-OS method, the amount of adjustment of restoring forces is calculated at predictor stage of displacement and velocity to avoid iterative procedure. The main advantages of this method are that it requires no iteration and ensures unconditionally stability if the nonlinearity is of softening type, Nakashima *et al.* (1990). However, initial stiffness values are required for calculating restoring force adjustment before beginning the test. Equation of motion for a structure is expressed as follows

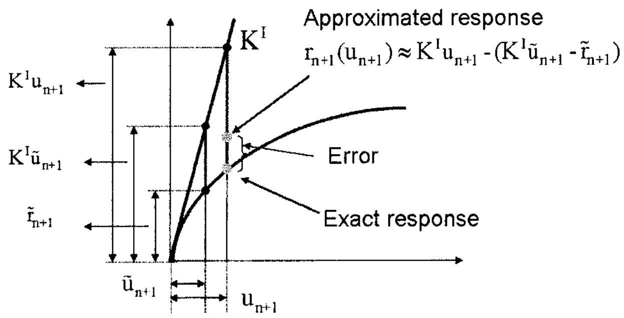


Figure 1 Approximation of Restoring Force in One-Dimensional Case for Operator Splitting Method

$$\mathbf{Ma}_{n+1} + \mathbf{Cv}_{n+1} + \mathbf{r}_{n+1}\left(t, \mathbf{u}_{n+1}, \mathbf{v}_{n+1}, \nabla\mathbf{u}\big|_{n+1}\right) = \mathbf{P}_{n+1} \quad (1)$$

where M and C are the mass and viscous damping matrices, a, v and u are the acceleration, velocity and displacement and P indicates the external force vector. The α-OS method is based on approximation of restoring forces as follows

$$\mathbf{r}_{n+1} \approx \mathbf{K}^I\mathbf{u}_{n+1} - \left(\mathbf{K}^I\tilde{\mathbf{u}}_{n+1} - \tilde{\mathbf{r}}_{n+1}\left(\tilde{\mathbf{u}}_{n+1}\right)\right) \quad (2)$$

where K^I indicates the initial stiffness matrix and n+1 indicates the (n+1)th time step. The approximation for the restoring force in one-dimensional case is depicted in Figure 1. As shown in Figure 1, error could be accumulated depending on the level of nonlinearity and accuracy of the initial stiffness value, K^I. However, because size of time step is often sufficiently small in dynamic simulation, negative effect of the accumulated error could be negligible.

The predicted displacement is calculated using explicit equation as follows

$$\tilde{\mathbf{u}}_{n+1} = \mathbf{u}_n + \Delta t\mathbf{v}_n + \frac{\Delta t^2}{2}(1-2\beta)\mathbf{a}_n \quad (3)$$

where $\tilde{\mathbf{u}}_{n+1}$ is the predicted displacement to be sent to experimental or numerical static models and β is the integration parameter. Substituting equation (2) into equation (1), the equation of motion can be expressed by

$$\mathbf{Ma}_{n+1} + \mathbf{Cv}_{n+1} + \mathbf{K}^{I}\mathbf{u}_{n+1} - \left(\mathbf{K}^{I}\tilde{\mathbf{u}}_{n+1} - \tilde{\mathbf{r}}_{n+1}\left(\tilde{\mathbf{u}}_{n+1}\right)\right) = \mathbf{P}_{n+1} \quad (4)$$

If the last two terms on left hand side of equation (4) are moved to right hand side as external forces, then the equation of motion can be solved using implicit Newmark method which is readily available in any general-purpose finite element analysis program. Because the initial stiffness is defined as linear elastic values in dynamic models, no iteration is required. Based on the equation of motion (4), procedures of hybrid simulation and testing using a general-purpose finite element analysis program are summarized as follows

Step 1 Compute initial stiffness K^{I} by pre-test in the laboratory or substructure analysis

Step 2 Construct a dynamic model using the initial stiffness matrix

Step 3 Set n=0 and initial conditions on u_0 and v_0

Step 4 Input excitation P_{n+1}

Step 5 Solve equation (4) using implicit Newmark method in Computational Dynamic Module

Step 6 Compute $\tilde{\mathbf{u}}_{n+1}$ in Computational Dynamic Module

Step 7 Impose $\tilde{\mathbf{u}}_{n+1}$ to tested model in the lab or numerical static model in Computational Static Module

Step 8 Measure $\tilde{\mathbf{r}}^{m}_{n+1}$ and $\tilde{\mathbf{u}}^{m}_{n+1}$ in the lab or Computational Static Module

Step 9 Compute $\tilde{\mathbf{r}}_{n+1} = \tilde{\mathbf{r}}^{m}_{n+1} - \mathbf{K}^{I}\left(\tilde{\mathbf{u}}^{m}_{n+1} - \tilde{\mathbf{u}}_{n+1}\right)$ in Computational Dynamic Module

Step 10 Compute adjustment of restoring force vector, $\mathbf{K}^{I}\tilde{\mathbf{u}}_{n+1} - \tilde{\mathbf{r}}_{n+1}$ in Computational Dynamic Module

Step 11 Solve equation (4) for u_{n+1} and v_{n+1} and a_{n+1} using implicit Newmark method

Step 12 Set n=n+1 and go to step 6.

Schematic outlines of the proposed software framework are illustrated in Figure 2 for a simplified soil-structure interaction problem. The structure is decomposed into two parts: the column structure and the soil-foundation structure. The entire structure is modeled in computational dynamic module using sophisticated finite elements and initial stiffness matrices K^{I} of substructures. The column is tested physically in the laboratory. The soil-foundation structure is simulated in computational static module.

As shown in Figure 2, targeted displacements are computed in Step 4 and 11 using implicit Newmark method available in ABAQUS. They are collected by main control module and redistributed to each experimental and computational static module. After imposing the target displacements, experimental

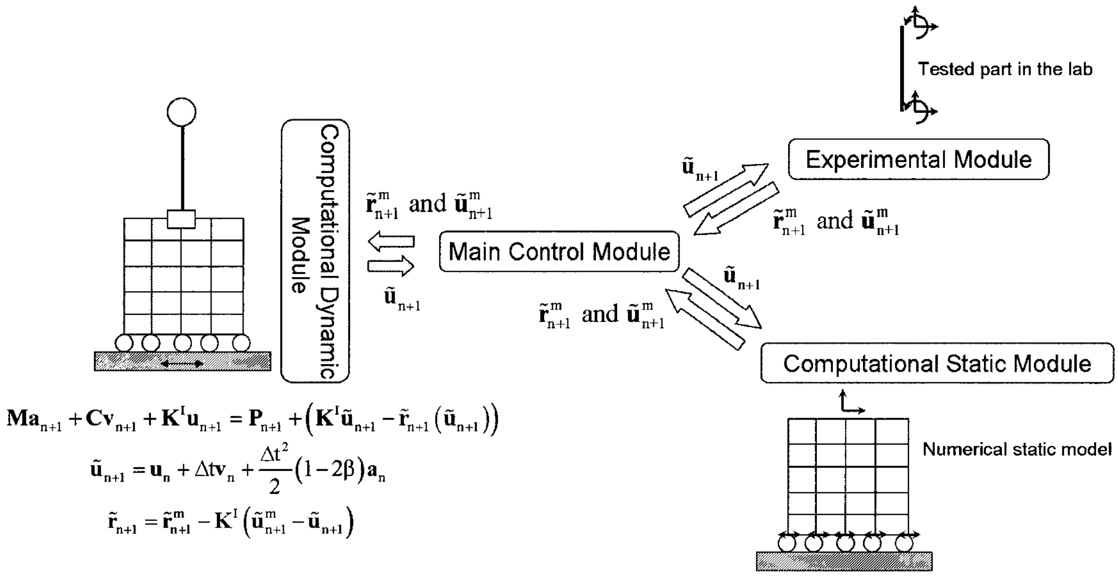


Figure 2 Schematic Outline of Proposed Software Framework

and computational static module send measured restoring force and displacement to main control module. Main control module sends the measurements to computational dynamic module. Thus, the time marching algorithm is repeated until the end of time duration is reached.

2.2 Sub-structuring Techniques

Level of sophistication of the dynamic finite element model in the computational dynamic module depends on how the mass and initial stiffness matrices for substructures are reduced. ABAQUS provides extensive functionalities for substructure analysis(ABAQUS/Standard HKS, 2004): 1) extraction of stiffness matrices of substructures using user-defined retained DOFs after preloading, 2) reuse of the substructures with the reduced stiffness and mass matrices in dynamic analysis and 3) reduction of mass matrix of substructures with generalized DOFs to enhance dynamic response within substructures.

Figure 3 depicts an example of hybrid simulation and testing of soil-foundation-structure interaction problem. LBC box indicates load boundary condition box which can apply 6 DOFs displacement to structures(Elnashai *et al.*, 2004). Geotechnical laminar box is a container which permits development of stresses and strains associated with wave propagation(Hex, 2005; Kagawa *et al.*, 2004).

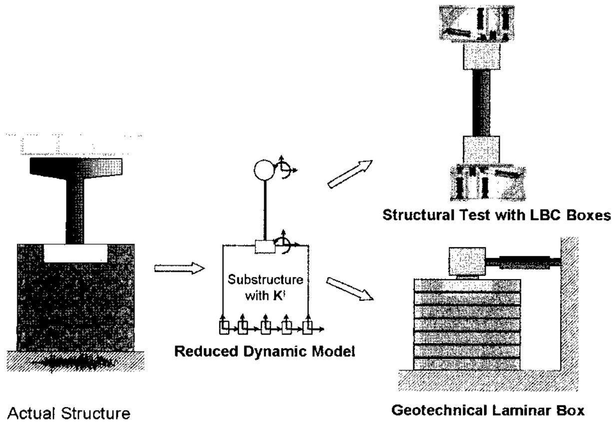


Figure 3 Sub-structured Hybrid Simulation and Testing of Soil-Structure Interaction Problem

Two reduction techniques are available in ABAQUS: Guyan reduction(Guyan, 1965; Irons, 1965) and component-mode synthesis method(Craig *et al.*, 1968). To enhance dynamic response within substructures, the component mode synthesis method is preferred. If the component-mode synthesis method is used, the transformation between original DOFs of substructure and DOF of synthesized substructure is obtained as follows(Cook *et al.*, 2001).

$$\{\mathbf{u}\} = \begin{Bmatrix} \mathbf{u}_n \\ \mathbf{u}_a \end{Bmatrix} = \begin{bmatrix} \mathbf{\Phi} & \mathbf{\Psi} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{Bmatrix} \mathbf{q} \\ \mathbf{u}_a \end{Bmatrix} = \mathbf{Tz} \;\; ; \;\; \mathbf{\Psi} = -\mathbf{K}_{nn}^{T}\mathbf{K}_{na} \quad (5)$$

where Φ is the mode shapes of the substructure, obtained with all retained DOFs constrained, Ψ is the static transformation matrix which is the ratio of sub stiffness matrices, q is the generalized DOFs, u_a is the retained DOF, subscript 'n' is internal DOFs within the substructure and subscript 'a' is the retained DOFs. The equation of motion is reduced in terms of retained DOFs within substructures and generalized DOFs.

$$\mathbf{M}_r\ddot{\mathbf{z}}_{n+1} + \mathbf{C}_r\dot{\mathbf{z}}_{n+1} + \mathbf{r}_{n+1}(t) = \mathbf{P}_{r,n+1} \;\; , \;\; \mathbf{z} = \{\mathbf{q} \quad \mathbf{u}_a\}^T \quad (6)$$

where M_r and C_r are the reduced mass and viscous damping matrices, respectively. The mass and initial stiffness matrices of the reduced dynamic model in Figure 3 are expressed as follows.

$$\mathbf{M}_r = \mathbf{T}^T\mathbf{M}\mathbf{T} = \begin{bmatrix} \mathbf{\Phi}^T\mathbf{M}_{nn}\mathbf{\Phi} & \mathbf{\Phi}^T\mathbf{M}_{nn}\mathbf{\Psi} + \mathbf{\Phi}^T\mathbf{M}_{na} \\ \mathbf{\Psi}^T\mathbf{M}_{nn}\mathbf{\Phi} + \mathbf{M}_{na}\mathbf{\Phi} & \mathbf{\Psi}^T\mathbf{M}_{nn}\mathbf{\Psi} + \mathbf{M}_{an}\mathbf{\Psi} + \mathbf{\Psi}^T\mathbf{M}_{na} + \mathbf{M}_{aa} \end{bmatrix} \quad (7)$$

$$\mathbf{K}_r = \mathbf{T}^T\mathbf{K}\mathbf{T} = \begin{bmatrix} \mathbf{\Phi}^T\mathbf{K}_{nn}\mathbf{\Phi} & \mathbf{\Phi}^T\mathbf{K}_{nn}\mathbf{\Psi} + \mathbf{\Phi}^T\mathbf{K}_{na} \\ \mathbf{\Psi}^T\mathbf{K}_{nn}\mathbf{\Phi} + \mathbf{K}_{na}\mathbf{\Phi} & \mathbf{\Psi}^T\mathbf{K}_{nn}\mathbf{\Psi} + \mathbf{K}_{an}\mathbf{\Psi} + \mathbf{\Psi}^T\mathbf{K}_{na} + \mathbf{K}_{aa} \end{bmatrix} \quad (8)$$

Because component modes are required in the calculation of reduced mass matrix, eigenvalue analysis should be carried out before constructing the synthesized mass and damping matrices. To enhance dynamic response of the substructure, the first lower modes are required. It is noteworthy that slight changes in mode shapes can be made due to locally induced nonlinear behavior in the soil-foundation structure. However, because sensitivities of structural mode shapes are far less than those of natural frequencies, enhancement of dynamics of the substructure compensate for the change in mode shapes due to local nonlinear behavior.

The key functionalities required for this seamless integration of dynamic and static modules with main control module are restart option in dynamic and static analysis and generalized capabilities of each module and interface API by which users can run a general-purpose finite element program, post-process the output and communicate with a main control module. In Section 2.3, design of the software framework is described.

2.3 Design of Software Framework

The proposed software consists of four modules and one interface API: main control module, computational dynamic module, experimental module, computational static module with an interface API. They are developed using Visual C^{++}6.0 in Window XP operating system. In order to render the software framework development flexible and efficient, object-oriented programming concepts were used for development of the software framework. Its object-oriented structure allows easier code maintenance and also reduces the possibility of errors during upgrades.

A schematic of the software framework is illustrated in Figure 4. In its simplest application, the software framework is capable of performing hybrid simulation and testing of a selected structural component. However, as the framework is capable of representing an entire structural system, it is used to pseudo-dynamically test more than one component or a combination of several experimental facilities or numerical static models. Therefore, any number of experimental facilities and numerical static models can be incorporated within a simulation.

The generalization capability in configurations of a dynamic model, numerical static models and pseudo-dynamically tested models could be accomplished owing to 1) data structures and protocol, 2) unique design of software framework(separation of computational dynamic module from main control module) and 3) use of a general-purpose finite element analysis codes.

Main Control Module

The module plays a key role in setting up the overall structure to be simulated and initializing all other modules. It identifies the parts of the structure that will be simulated in physical experiments and in numerical simulations. The controller also acts as a coordinator between computational dynamic, static and experimental modules. It controls

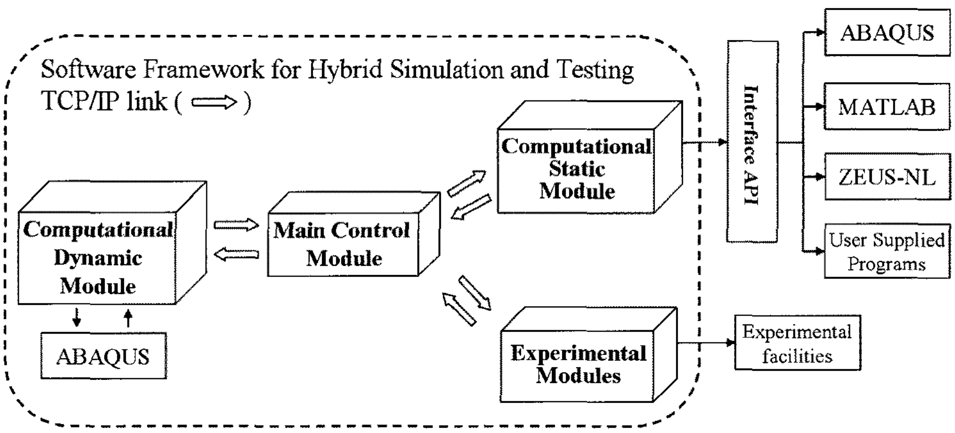


Figure 4 Modules in Proposed Software Framework for Hybrid Simulation and Testing

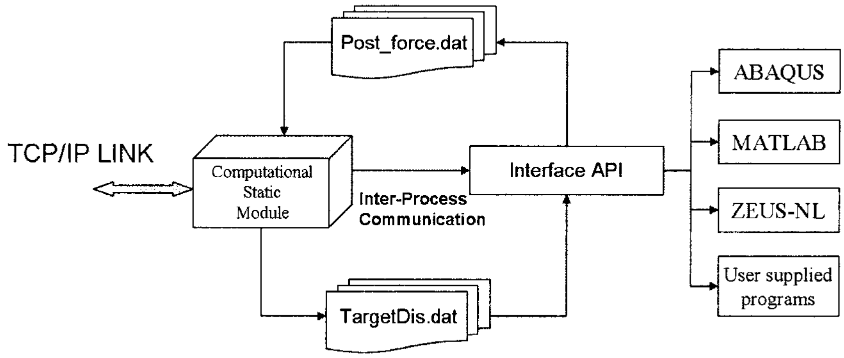


Figure 5 Inter-Process Communication (IPC) between Computational Static Module and Interface API

overall testing and simulation by providing scenario of communication between modules. Main control module can control more than one component of either computational or experimental module. During simulation, main control module prepares streams of string data to be sent to each module with action commands. For a hybrid test, main control module receives target displacements at control points from computational dynamic modules and distributes them to each associated experimental module at every time step. The main control module retrieves measured restoring forces at control points from experimental modules and distributes them to associated computational modules. At each time step, restoring forces or restoring force balances are updated for solving equation of motion at next time step.

Computational Dynamic Module

Portions of the structure that are not tested in an experimental simulation are simulated in a computational dynamic module. Unlike other software frameworks, a general-purpose finite element analysis program, ABAQUS is adopted. Because iteration in time integration of the equation of motion is not desirable for hybrid testing, dynamic models are often assumed to be linearly elastic. ABAQUS provides very extensive functionalities such as restart option, viable definition of dynamic substructures using user-defined stiffness matrices and sub-structuring techniques previously described. Computational dynamic module is an independent program which can seamlessly communicate with main control module and interface with ABAQUS. ABAQUS provides a general finite element tool for time integrator based on implicit time integration scheme with extensive element library.

Computational Static Module

This module provides interface with various computational platforms for replacing physical experiments in the laboratory with nonlinear static analysis of substructures. Because each computational platform has different strengths and weaknesses, various computational tools are adopted so that they can be used in computational static module. The proposed software framework is designed to incorporate any number of available simulation codes. The current implementation includes ABAQUS, ZEUS-NL(Elnashai *et al.*, 2002) and MATLAB. Other codes including OpenSees will be implemented in the near future. The communication between computational static module and interface API is established by Inter- Process Communication (IPC) as shown in Figure 5. IPC is a set of techniques for the exchange of data among multi-threads in one process. Data exchange between two applications is carried out by sharing files that contain restoring force and targeted displacement data. Advantage of having the interface API is that users can plug in any customized computational tool by slightly modifying the interface API without change of the computational static module. Instead, the computational static module plays a role in communicating with main control module.

Experimental Module

To accommodate various experimental set-ups, a middleware to transform DOFs in computational static module to specific experimental set-up should be developed. The middleware application is under development. To verify performance of the proposed software framework, simulation-based hybrid testing is performed in this paper. In this paper, computational static module with ABAQUS and MATLAB applications was successfully connected with MOST experiment set-up which was conducted.

2.4 Network Communication between Modules and Data Protocol

Generally, there are three methods of communication for multi-threaded programs: TCP/IP link, window message passage (IPC API), and sharing of data files. TCP/IP link method is used for each module within the proposed software framework. However, because the computational static module and interface API are run in a single computer, the communication between them is established through IPC API functions provided by MFC. Winsock library (ws2_32.lib) is used for implementation of the communication methods following TCP/IP protocol. For data exchange through network environment, data string is parsed following data protocol set-up. The data protocol consists of data length, command ID number, node number and restoring force or displacement data. To minimize amount of data to be exchanged between modules, ASCII code is used for command ID and node label. Scenario defined in

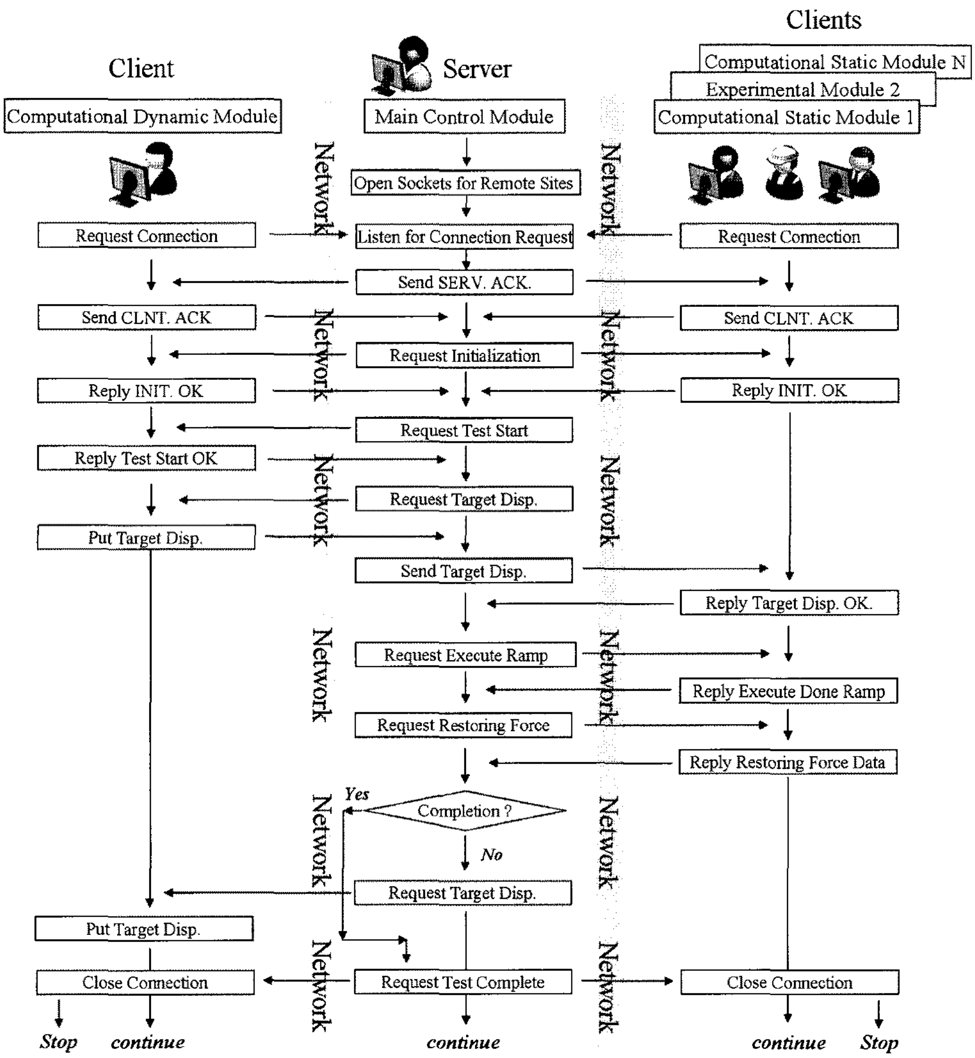


Figure 6 Scenario Defined in Main Control Module for Distributed Hybrid Simulation and Testing

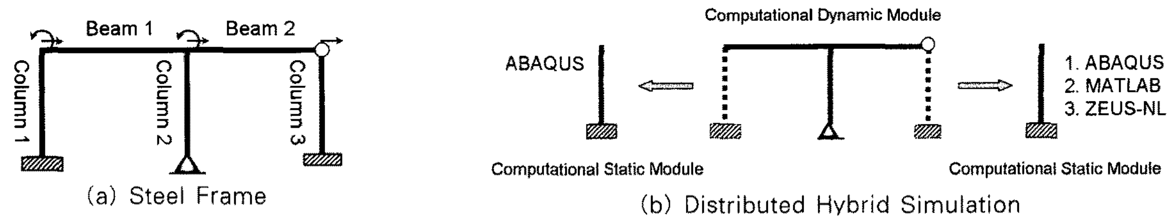
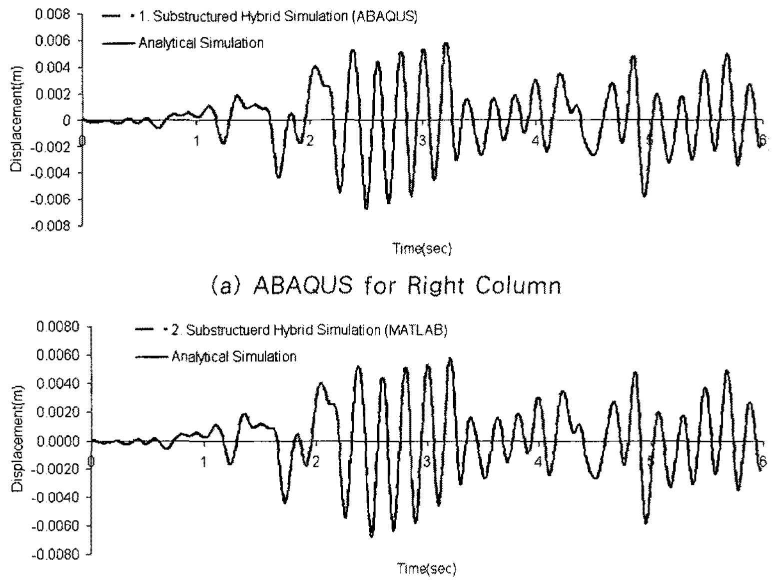


Figure 7 MOST Experiment



(b) Matlab Code for Right Column

Figure 8 Time History of Horizontal Displacement at Top Node of Column 1 for Elastic Case with 5.0% Viscous Damping

main control module and interactions with other modules are illustrated in Figure 6. The methods defined in client sides are performed by commands sent from main control module. However, the scenario defined in main control module is coded in sequential procedure. As illustrated in Figure 6, the hybrid test starts after connections are established and initialization is completed. Then a command requesting target displacements with measured data from previous step is sent to computational dynamic module. After main control module receives the target displacement, it sends it to experimental module. Experimental module receives the target displacement from main control module and wait for a command to execute the ramp action. After execution of ramp, the measured data are sent to main control module. The procedure is repeated until the end of the entire time duration.

As mentioned previously, the generalization capability is due to efficient data protocol and data structures, unique design of software framework and use of a general-purpose finite element analysis tool. Therefore, any type of structures can be simulated with sub-structured hybrid testing in network environment. Although minimal constraints exist in preparation of input data file to use ABAQUS in computational dynamic module, those are related to format of input data lines. Detail of the constraints can be found in Reference(Hashash *et al.*, 2004).

3. Distributed simulations using SOFTWARE framework

A series of examples are presented to demonstrate the versatility of the proposed software framework. The first example is a two-bay steel frame which was tested in MOST experiment in July 2003 (Spencer *et al.*, 2004). In addition, a complex soil-pile foundation interaction problem is demonstrated in the second example.

3.1 Multi-site On-line Simulation Test

Distributed hybrid simulation(physical and analytical) of a steel frame structure is described in a reference by Spencer *et al.*(2004) and is illustrated in Figure 7. Figure 7(a) shows a steel frame tested in MOST experiment. It has 5 active dynamic DOFs. Standard section W14×120 and W14×60 are selected for column and beam, respectively. One of the El Centro earthquake records is used as the input ground motion. For distributed hybrid simulation, three computers are utilized for left column, middle column and beams and right column. The left column and middle column with beams are modeled by ABAQUS in computational static and dynamic modules, respectively. To compare performance of various computational static analysis tools in distributed simulation, ABAQUS and customized Matlab code, and ZEUS-NL are used for right column. As shown in Figure 8, results from hybrid simulations show very good agreements with those from single dynamic analysis.

In the case of ZEUS-NL, bilinear material assumption is made for both left and right column. Because of error in adjusting restoring force vectors due to nonlinearity, small difference is observed as shown in Figure 9. However, response is shown to be reliably accurate. For testing performance of the computational module in hybrid testing, MOST experiment was performed in which left column was tested as shown in Figure 10. Figure 11 shows comparison of time history of horizontal displacement at node 2 for two platforms. Because the hybrid test was carried out with incomplete calibra-

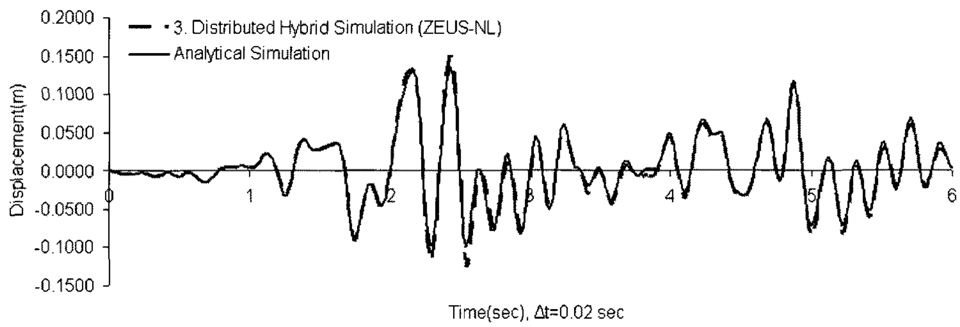


Figure 9 Time History of Horizontal Displacement at Top Node of Column 1 for Inelastic Case with 5.0% Viscous Damping

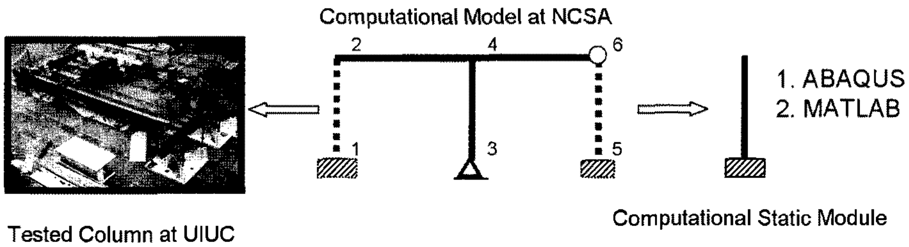


Figure 10 Distributed Hybrid Testing with Physical Testing of Left Column

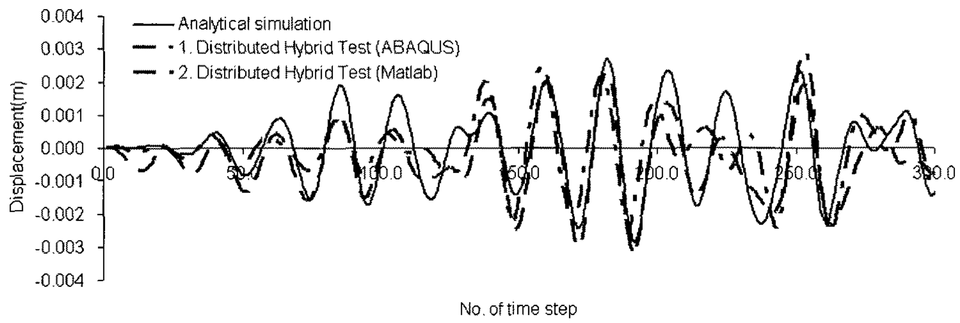


Figure 11 Time History of Horizontal Displacement at Node 2

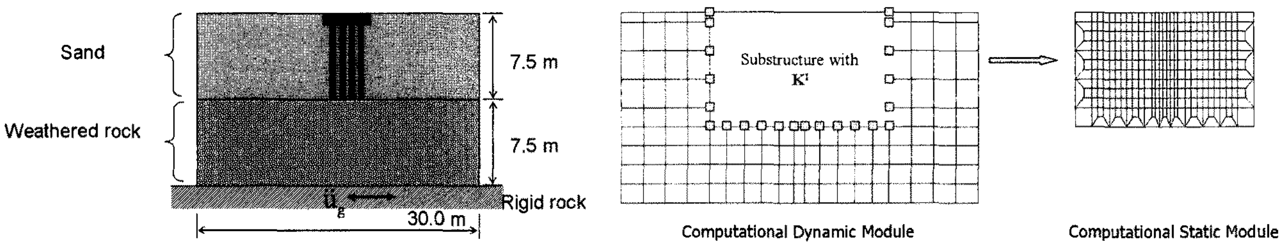
tion of experimental set-up, two hybrid tests show different discrepancy. The experimental set-up shown in Figure 11 had difficulties because translation and rotation should be controlled at the same time.

Current example proves potential applications of the proposed software framework in distributed hybrid simulation and testing. Because a general-purpose finite element analysis tool(ABAQUS) is employed, the proposed software framework can be readily applied to more complex structures. Implementation of α-OS method within computational dynamic module also provides an unconditionally stable time integration scheme for hybrid simulation and testing.

3.2 Soil-Pile Foundation Interaction Problem

There are few limitations in the level of complexity of the physically tested substructures because of a general-purpose finite element analysis tool. Moreover, the extended sub-structuring techniques provide an efficient method for sub-structured hybrid testing as long as experimental set-up for each substructure is feasible in the laboratory. To show its potentiality, complex soil-Pile foundation interaction problem is demonstrated in this example. Two-layered soil body and footing with piles are invented. Figure 12(a) depicts geometrical dimensions and configuration of the structure. Foundation and RCD piles are set to have physical properties: Young's modulus=2.9×10^7kN/m^2, poisson ratio=0.25, density=24.53kN/m^3 and damping ratio=5%. Sand layer is set to have physical properties: shear velocity=250m/sec, shear modulus=1.25×10^5kN/m^2, poisson ratio=0.4, density=19.61kN/m^3 and damping ratio=20%. Weathered rock is also set to have physical properties: shear velocity=400m/sec, shear modulus=4.094×10^5kN/m^2, poisson ratio=0.3, density=25.1kN/m^3 and damping ratio=10%. Initial stiffness of the substructure is calculated by Guyan reduction with DOFs along the boundary retained as shown in Figure 12(b). Bed rock is excited in horizontal direction using Kobe earthquake record. Single dynamic simulation also uses the same reduced substructure.

As shown in Figure 13, sub-structured hybrid simulation shows good agreement with analytical simulation results.



(a) (b)

Figure 12 Soil-Pile Foundation Structure Interaction Problem
(a) Configuration of Structure and (b) Configuration of Computational Dynamic and Static Models

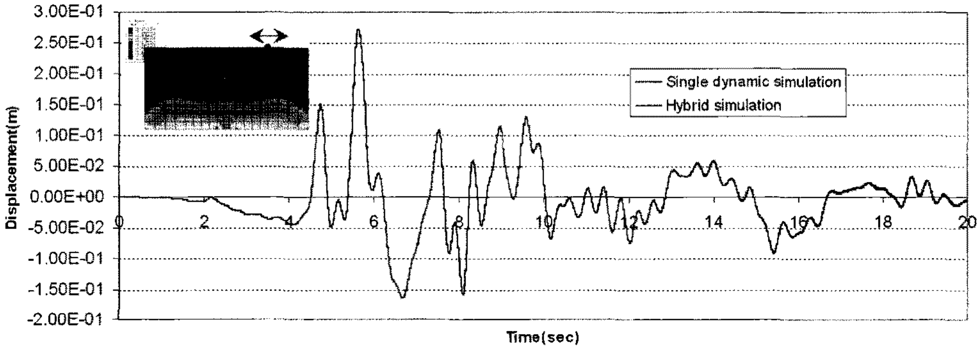


Figure 13 Comparison of Time History of Displacement at Surface

4. conclusions

In this paper, a software framework that integrates computational and experimental simulation has been developed to simulate and test a large-scale structural system under earthquake loading. The software framework consists of four modules; main control module, computational dynamic module, experimental module and computational static module with interface API. It has been developed using Visual C^{++}6.0 language in Window XP operating system. Currently, the proposed software framework supports simulation using ABAQUS deployed as computational dynamic model and ABAQUS, ZEUS NL and MATLAB, deployed as computational static components of the structure under investigation. Two numerical time integration algorithms are implemented in the system; the conventional α-OS method and new Predictor-Corrector method. However, this study focuses on use of the α-OS method. Network communication between modules is established using TCP/IP link method. A series of examples including soil-structure interaction problems demonstrate accurate and efficient simulation capability of the software framework. The accurate results from distributed hybrid simulations, compared to a single-model analysis, confirm that the system is fully-functional and is capable of running any number of experimental and/or analytical components.

Unique advantage of the proposed software framework is that it does not require separate development or modification of dynamic and static analysis codes for the purposed hybrid simulation and testing. It is mainly because a general-purpose finite element analysis tool, ABAQUS is employed for computational dynamic and static modules. Sophisticated finite elements and sub-structuring techniques can potentially enhance efficiency and accuracy of the hybrid simulation and testing. Currently the computational static module can use ABAQUS, ZEUS-NL and Matlab codes through interface API. Distinct advantages of using the proposed software framework are as follows: 1) Generalization capability in configuring structures under investigation, 2) Extensibility of the computational static platforms, and 3) Scalability of the dynamic model, numerical static models and pseudo-dynamically tested models. Those advantages are realized by efficient data structure/protocol and management, unique design of the overall software framework and use of a general-purpose finite element analysis tool.

Acknowledgement

This work was supported by the research fund of Seoul National University of Technology.

References

ABAQUS/Standard HKS (2004) *A General Purpose Finite Element Code*. Karlsson & Sorense, Inc: Hibbitt.

Cook, R.D., Malkus, D.S., Plesha, M.E., Witt, R.J. (2001) *Concepts and Applications of Finite Element Analysis*. John Wile &Sons Inc.

Craig, J.R., Bampton, M.C.C. (1968) Coupling of Substructures for Dynamic Analysis, *AIAA Jounral*, 6, pp.1313~1319.

Elnashai, A., Spencer, B.F., Kuchma, D., Hashash, Y., Ghaboussi, J., Gan, G. (2004) Multi-axial full-scale sub-structured testing and simulation (MUST-SIM) facility at the University of Illinois at Urbana-Champaign. *13th World Conference on Earthquake Engineering* Vancouver, BC Canada.

Elnashai, A.S., Papanikolaou, V., Lee, D.H. (2002) eds. *ZEUS NL, A Program of the Inelastic Dynamic Analysis of Space Frames*. Vol. User Manual. Mid-America Earthquake Center, Civil and Environmental Engineering Department, University of Illinois at Urbana-Champaign.

Ghaboussi, J., Yun, G.J., Hashash, Y.M.A. (2006) A Novel Predictor-Corrector algorithm for substructure pseudo dynamic testing. *Earthquake Engineering and Structural Dynamics* 35, pp.453~476.

Guyan, R.J. (1965) Reduction of stiffness and mass matrices. *AIAA Jounral*, 3, p.380.

Hashash, Y., Yun, G.J., Ghaboussi, J., Elnashai, A. (2004) *Integrated Computational and Experimental Simulation Framework*, User Manual, Department of Civil and Environmental Engineering, University of Illinois Urbana-Champaign, Urbana

He, L. (2005) Liquefaction-Induced Lateral Spreading and Its Effects on Pile Foundations, PhD Thesis *Department of Structural Engineering University of California, San Diego.*

Irons, B. (1965) Structural eigenvalue problems: Elimination of unwanted variables, *AIAA Jounral*, 3, pp.961~962.

Kagawa, T., Sato, M., Minowa, C., Abe, A., Tazoh, T. (2004) Centrifuge Simulations of Large-Scale Shaking Table Tests: Case Studies, *Journal of Geotechnical and Geoenvironmental Engineering*, 130, pp.663~672.

Kwon, O.S., Nakata, N., Elnashai, A.S., Spencer, B. (2005) A framework for multi-site distributed simulation and application to complex structural systems, *Journal of Earthquake Engineering*, 9, pp.741~753.

Mahin, Stephen, A., Shing, P.B. (1985) Pseudodynamic Method for Seismic Testing, *Journal of Structural Engineering*, 111, pp.1482~1503.

Nakashima, M., Kaminosono, T., Ishida, M., Ando,. K. (1990) Integration techniques for substructure pseudo dynamic test, *4th U.S. national conference on earthquake engineering*, CA, pp.515~524

Shing, P.B., Mahin Stephen, A. (1985) Computational Aspects of a Seismic Performance Test Method using On-line Computer Control, *Earthquake Engineering and Structural Dynamics*, 13, pp.507~526.

Spencer, B., Elnashai, A., Kuchma, D., Abrams, D. (2004) NEESGrid Multi-site online simulation test (MOST) at the University of Illinois at Urbana-Champaign, *13th World Conference on Earthquake Engineering*, Vancouver, BC Canada.

Takahashi, Y., Fenves, G.L. (2006) Software framework for distributed experimental-computational simulation of structural systems, *Earthquake Engineering and Structural Dynamics*, 35, pp.267~291.

Wang, T., Nakashima, M., Pan, P. (2006) On-line hybrid test combining with general-purpose finite element software, *Earthquake Engineering and Structural Dynamics*, 35, pp.1471~1488.

Yun, G.J., Ghaboussi, J., Elnashai, A.S. (2006) Development of neural network based hysteretic models for steel beam-column connections through self-learning simulation, *Journal of Earthquake Engineering*, Accepted.

Yun, G.J., Ghaboussi, J., Elnashai, A.S. (2007) A New Neural Network-based Model for Hysteretic Behavior of Materials, *International Journal for Numerical Methods in Engineering* In Press.

Yun, G.J., Ghaboussi, J., Elnashai, A.S. (2007) Self-learning Simulation Method for Inverse Nonlinear Modeling of Cyclic Behavior of Connections, *Computer methods in applied mechanics and engineering* Accepted.