13th World Conference on Earthquake Engineering
Vancouver, B.C., Canada
August 1-6, 2004
Paper No. 3048

DEVELOPMENT OF A NETWORK PLATFORM FOR REMOTE HYBRID DYNAMIC TESTING

Yan XIAO [1,2]; Qing HU [2,3]; Yurong GUO [2]; Pingsheng ZHU [1]; Weijian YI [2]

SUMMARY

This paper presents a network platform developed for remote pseudo dynamic testing of substructures and structural elements, and discusses the results of trial tests. The network platform, NetSLab, was developed based on client/server concept along with a proposed data model and communication protocols. The platform is capable of transferring control and feedback data among remote structural testing laboratories or computers connected by Internet. Several concepts were introduced to realize the platform and to provide relatively easier and friendlier interface for applications and further development by users. Trial tests were successfully carried out at the Hunan University, China and the University of Southern California, USA. In the trial tests, models simulating bridge piers or piles were subjected to recorded earthquake ground motions controlled remotely over the Internet through the platform.

INTRODUCTION

The development and evolution of modern structural engineering highly depend on experimentation. Particularly, due to the complexity of the earthquake mechanism and the highly nonlinear performance of structures under extreme earthquake loading, experimental research plays an important role in developing seismic design methodology for engineering structures. Experimental research serves as an efficient mean to fill the gap between any analysis and the reality, and meanwhile can lead the way to advance theories. For large-scale complicated structures or new systems which are not covered by existing design codes and specifications, or their behaviors can not be confidently predicted by existing analytical tools, experimental testing can provide performance data needed for the basis of design, construction as well as operation. Followed by the speedy advance of mechanical engineering, computer, control and sensor technologies, modern structural experiment also evolves to become capable of testing larger-scale system with more proper, realistic and accurate simulations of complicated working conditions, using advanced equipment such as actuators, shake tables, centrifuge, hybrid testing facilities, etc. [1~5].

[1] Dept. of Civil Eng., University of Southern California, Los Angeles, CA90089, USA, yanxiao@usc.edu
[2] School of Civil Eng., Hunan University, Changsha, China.
[3] FutureNet Technologies Corporation, Monrovia, California, USA.

The recent rapid development of Internet technology provides new opportunities of revolution in structural experimentation. The Internet based communication enables the transfer and sharing of vast data and particularly the control signals, feedback information among geographically separated laboratories and facilities in an almost real time fashion. Scientific laboratories thus networked can then be utilized for testing more complicated systems to advance our knowledge. Several attempts have been made to carry out collaborative tests on structural systems with remotely located laboratories connected by Internet [6-10]. The most comprehensive efforts might be the Network for Earthquake Engineering Simulation (NEES) project [7], which is aimed at more efficient sharing of the resources for earthquake engineering research by integrating laboratories, testing facilities, field observations, computer facilities, etc.

To facilitate the structural research collaborations between the University of Southern California (USC) and the Hunan University (HNU) of China as well as other institutes, a research effort was initiated in 2000 to establish an Internet based network platform, now named as NetSLab (Networked Structural Laboratories). This paper presents the basic concepts and features of the platform developed for remote pseudo dynamic testing of substructures and structural elements, and discusses the results of several trial tests.

NETWORKED STRUCTURAL LABORATORIES (NETSLAB)

Proposed Data Model for NetSLab

A distributed structural testing can be organized as a series of steps. In one step, each participant completes one event according to the mutual communication among the participants, generates and sends additional result data. In other words, a step can contain several events. In order to make this process work accordingly, some kinds of controller should be introduced. The controller organizes the testing procedure and coordinates the working flow of all participates as well as the data flow in the testing.

The authors proposed a generalized data model to abstract such distributed testing procedures and attempted to cover all possible situations in this area. The data model consists of the following three components: testing results; three types of participants; communication protocols and dataflow.

Testing Results
The ultimate goal of a distributed structure testing is to obtain a series of testing results. According to the proposed data model, the testing results are represented by an array, *TR*, such as,

$$TR=TR(i,j),\ i=1,\ N;\ j=1,\ M \tag{1}$$

where, N is maximum number of the steps and M is maximum number of the events within a step. Each row of the array represents the testing results in one step, and each element of a row represents the testing result of a testing event, i.e. $TR(i,j)$ is the result from the step i and event j. Note that, $TR(i,j)$ may depend on the previous stages' results.

By this definition, the testing process can be considered as how to generate the *TR*, according to certain initial conditions and certain roles in the proposed data model.

Three Types of Participants
Based on the definition for *TR*, the participants' tasks in a testing are processing the *TR* with various different means. By their roles in the testing, all participants can be classified into three groups:

1) Controller: who organizes the testing procedure, controls the testing progress and data communication among all participants, and stores or publishes the testing results. There is only one controller in one testing for current study.
2) Testers: who carry out the detailed task event(s) for each testing step to create new results according to previous results. There are multiple testers in one testing. The testers can also be categorized into two types: virtual testers, who use computation to provide analytical *TR*, and actual testers, who operate the actual testing equipment to generate results, *TR*.
3) Observers: who monitor the process and share the testing results without any interference of the test processing.

From the Data Model's point of view, controller initiates and maintains the *TR*, designates the data flow, decides how to transfer data among the testers and publishes the partial or final consolidated *TRs*. Each tester reads the data of partial or whole *TR*, creates one or more new values based on their testing facilities or algorithms and forwards the result to next testers (or controller). Figure 1 represents the relationships among the participants.

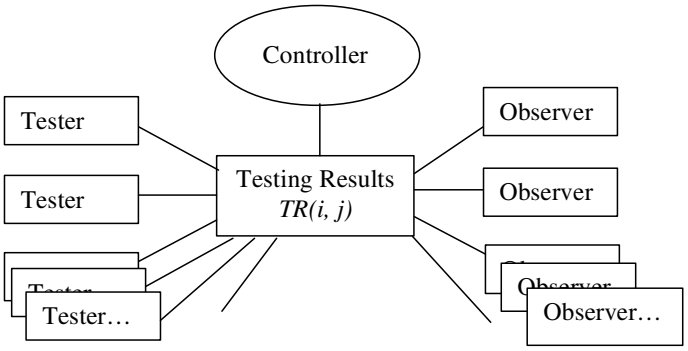


Fig.1. Roles of participants and interaction in proposed data model

Communication Protocols and Dataflow
The data to be circulated among all participants correspond to all kinds of testing results and control information. In order to keep the testing running smoothly and efficiently, it is very important to specify how these data to be circulated. We used the Finite State Automata (FSA) model to describe the dataflow during the testing process. In general, an FSA is decided by its state transfer function, defined as,

$$F(C, u)=N \tag{2}$$

where, C is the current state, u is input and N is the next state.

In the proposed data model, the state transfer function F is defined as,

$$F(C, u, H)= (N, \alpha) \tag{3}$$

where H is the historical data (partial *TR*), i.e. each node has a certain memory to maintain some historical data, and α is an action which specifies what to send and send to whom.

According to the definition and properties of action α, three types of data communication protocols are introduced in our data model:

1) Centralized sequential communication: controller sends control data (called request) to each tester one by one and a tester responds back to controller only after it finishes controller's request. The order of data transfer is pre-determined, i.e. F is independent to u and H, and at any moment, only one peer to communicate.
2) Centralized parallel communication: controller sends requests to multiple testers simultaneously and waits for their response. As soon as obtaining the responses from the testers, the controller decides and starts next step to send out next round of requests.
3) Distributed parallel communication: controller only sends out the start signals and initial data to certain testers. When a tester receives the partial *TR* and request, it executes certain action and decides its own action to send request and testing result to someone else. At certain point, a tester may report the results back to the controller.

Implementation of NetSLab

Since the final testing system will be used by many non-computer professionals for many different needs of structural testing, it is very important to keep its simplicity for understanding, developing and maintenance. Many tests require real-time response which also requires the final system to operate as quickly as possible under the given network environment. Therefore, we introduced two new concepts to fulfill these goals.

1) Dynamic Unified Data Packet (DUDP). This data packet covers all communication related components: current stage data of *TR*, controller's request, testers' current status and communication requirements, etc. With such concept, only one kind of data packet is needed for traveling among all participants, providing the simplicity and reliability to the system. DUDP Parser is the software module to parse a given DUDP and to help any entity (controller, tester or observer) understand DUDP efficiently and correctly.
2) Generalized Data Communication Agency (GDCA). This is the unique communication components in our testing system, which can be attached to its master entity, such as a controller, testers or observers. It accepts DUDP from its master entity, sends it to the given destination and activates the corresponding entity if necessary. On the other hand, any incoming DUDP also goes through this agency to activate its master entity.

Based on these two concepts, data communication in a testing procedure becomes clear and simple. Any entity, whether it is a controller, tester or observer, always contains a DUDP Parser and a GDCA. For a tester, it is always waiting there, and an incoming DUDP or a user command can activate this entity. It then parses the DUDP to understand the request and corresponding data, carries out all necessary operations (including using the actual testing equipment or theoretical model with the input data to create testing results *TR*), determines the destine entity, generates the new DUDP and submits it to GDCA. All the testers have the same procedure for data communication and the differences are their own operations. This scheme is very helpful for simplicity, reliability and efficiency. The controller also can be established on the tester model. Besides the tester's function, the controller has more functions, such as, designating the participants for a given testing, organizing the communication protocol, specifying the initial conditions, evaluating testing results and deciding the termination of the testing.

In NetSLab, DUDP is represented by an XML formatted structure data. XML format provides a good flexibility for data representation, well defined data format standard and a large number of processing

tools. It allows a large number of testers and observers and a tester specified special data structure to represent the results. The general format of a DUDP is as follows:

```
<DUDP>
    <Communication>
        <Source>…</Source>
        <Destine>…</Destine>
        <CommStatus>…</CommStatus>
        <Request>…</Request>
        … …
    </Communication>
    <CurTestResult>
        <Tester1>
            <Value11>…</Value11>
            … …
    </Tester1>
        <Tester2>
            <Value21>…</Value21>
            … …
    </Tester2>
    … …
        <TesterN>
            <ValueN1>…</ValueN1>
            … …
    </TesterN>
    </CurTestResult>
</DUDP>
```

Basically, DUDP consists of two sections: *Section DUDP.Communication* contains all communication and operation related parameters (control information) and *Section DUDP.CurTestResult* contains the complete data of one event for the *TR*. According to the property of XML data, each section can contain any number of subsection or unspecified number of data entries, which allow the maximum flexibility.

The proposed GDCA can be considered as an ISO layer 7 protocol, which has both data presentation and communication function. GDCA receives the standard XML formatted data DUDP from the application programs. By parsing the contents of *Section DUDP.Communication*, GDCA transfers the data from the given source to the given destine and delivers the *DUDP.CurTestResults* to corresponding parts. GDCA can be configurable to support the three communication models mentioned previously.

The implementation of GDCA is fully open to system designers. In NetSLab, GDCA has been implemented based on a general interface engine, UniPipe [11], which provides many sophisticated lower level communication functions and a very flexible ActiveX control (including event sink) interface. Using UniPipe's diagram based script language, Action Tree, the design and maintenance of GDCA become much easier. As an ActiveX controller, it naturally supports any application program language with standard ActiveX function, such as VB, VC++, VJ, etc.

In order to fulfill the task for distributed structures testing, the proposed GDCA has two interfaces, Client Agency and Control Center. The former is used for all testers (or some observers if necessary) to complete basic data transfer function, and the latter is used by the controller to carry out system configuration tasks besides basic communication functions. Such interfaces can be called by the host applications through

standard ActiveX control instance. It delivers the incoming information to the host application through predefined event sink mechanism.

Other system functions

As an efficient practical application system, many other issues also need to be considered in the implementation of NetSLab.

Data Repository
For any scientific experiment, data repository is an important issue. In NetSlab, the data repositories are considered at two levels. The controller establishes and maintains the overall data repository for the overall experiment. Each tester establishes and maintains its data repository for the assigned task. GDCA provides the database interface to allow applications to easily access certain database system to store/retrieve current and historical testing results, *TR*. Since UniPipe has a built-in database interface through Microsoft ADO, it is readily for GDCA to provide and expand its database functions. The ADO is the currently mainstream database interface in the Windows/Linux systems, which supports almost all commercial databases, such as Access, MS SQL Server, Oracle, MySQL, as well as any ODBC components. Thus, the NetSLab is a database independent system with the capability to readily use any kinds of database system.

Fault Tolerance
For a reliable application system, it is very important to handle the network errors during data communication. GDCA provides fault tolerance functions to application programs through two aspects:
1) The built-in fault tolerance communications inside Interface Engine. UniPipe utilizes all current reliable communication functions of TCP/IP networks to provide the most possibly reliable communications;
2) The complete error-handling capability to report the accurate error message and keep the on-error site. The application program can easily repeat/resume the last failed operation without any problem.

Based on GDCA's fault tolerance capability, NetSLab can handle the most Internet related communication errors quite smoothly without interfering the normal operations.

Multimedia Data Communications
During a remotely distributed testing, many video and audio information may need to be shared among participants besides normal DUDP. In order to handle such case, interface engine UniPipe provides the auxiliary data communication channel besides the XML data and GDCA. The auxiliary data channel can be established upon the request of application program between certain points. This is a fast binary data channel which can support the large amount of binary files, DirectX stream data, and any other user defined data. NetSLab uses such auxiliary channel to transfer A/V data among testers and/or controller. Controller also can post such A/V data to the Internet through the Web Servers.

Communication Security
Running over the public domain, secured communication in NetSLab is also a very important issue. NetSLab is built with its security functions over its interface engine. Currently, three security features have been introduced to secure the data communications.
1) *Windows level C3 security*: Interface engine UniPipe fully supports Windows system's native security feature, password protected access control by using Windows user access control as its own user access control. Each data communication can be configured as required user authentication.

2) *Standard HTTPS security channel*: Besides regular channels, interface engine UniPipe also supports an additional HTTPS channel to complete its data communication. In order to do so, a secured Web Server in the public domain will be required to serve as the data communication buffers.
3) *Capability to pass through firewalls*: With the help of HTTP or HTTPS data buffers, interface engine UniPipe also supports the communications between two parties inside different firewalls. With this function, NetSLab can allow some testers to be inside certain firewalls.

Obviously, implementation of such security function has the cost of communication speed. In the real system, users may need to settle with some kind of trade-offs between the speed and the security.

APPLICATION PROGRAMMING ISSUES OF NETSLAB

Standard Open Programs

According to the proposed Data Model, DUDP and GDCA provide the necessary foundation for NetSLab system's communication and data processing module. The remaining tasks are the application related programming, i.e. structural testing implementations. Specific application programs are required to be developed for testing or computation to handle different application needs.

Developed by the interface engine UniPipe script, GDCA has implemented an ActiveX Control Interface. This is a language independent programming interface to the high-level application programs. Any Windows programming tools can be used to develop such application program, as long as it fully supports ActiveX Controller, such as Visual Basic, VC++, Visual Java and C#. In the current study, most application programs are developed using Visual Basic and some VC++, where the ActiveX control is naturally integrated into the system.

Network Environment

The development of NetSLab is targeted to the flexibility and suitability in various network environments. This is considered to be one of the important factors for the technology transfer and the promotion for wide applications. A simple speed testing software was developed to test the data transfer speed using GDCA. At one end of the network, the program generates and sends a series of data for variable, x, and records the return time needed for the data of another variable y, calculated as the function of x at the other end of the network. Results of speed testing for roundtrip data transfer over different network environments are shown in Fig.2. As shown in Fig.2, the transfer speeds are fairly smooth with an average below 0.08 sec, for the networks of 1Mbps to 10Mbps, which are the environments within or among most academic units in the US and China. Interesting data were also recorded for the situation of a PC connected from a hotel room in China to another PC located at USC using a slow dialup connection with a speed of 24kbps. The roundtrip speed fluctuated between 0.4 to 0.7sec, however was not too drifted from the recorded average speed of about 0.4sec based on the MS Ping test. This situation may be considered as the possible worst-case application of NetSLab, in terms of network speed.

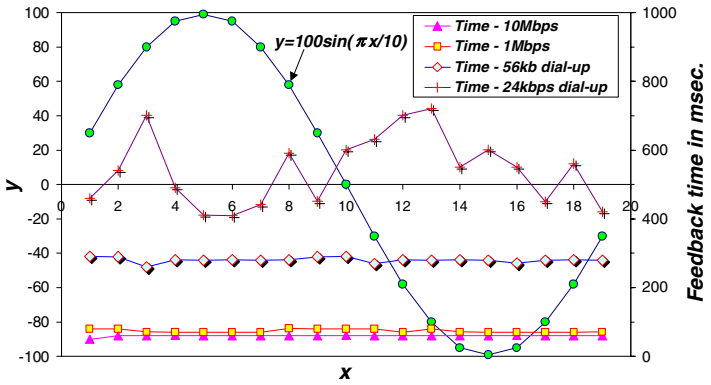


Fig.2. Speed test results in different network environments.

APPLICATION OF NETSLAB FOR REMOTE SIMULATION ANALYSIS OF EARTHQUAKE RESPONSE

Non-linear time history analysis is generally accepted as the best method to study the earthquake response of a structure. However, due to its complexity and large amount of computation, this method is currently mainly used in research or earthquake designs of some important and complex structures. For a non-linear analysis program, one of the most difficult parts is how to construct the hysteretic models for various structural elements. Since different types of structural members (beams, columns, and walls) use different materials and have many differences between their characteristics of restoring force, there exist many types of models to simulate earthquake response of structures. To develop a non-linear time history program, one needs to construct analysis model for structural system, select appropriate numerical integration method and equation solver, and also deal with structure hysteretic models.

Based on the characteristics of non-linear time history analysis, we can separate the hysteretic modeling from the main program. Computers with programs handling one or several types of hysteretic modeling is considered as virtual testers in NetSLab, and can be located remotely [12]. A remote analysis to simulate the earthquake response of a structure can be considered as a virtual remote testing.

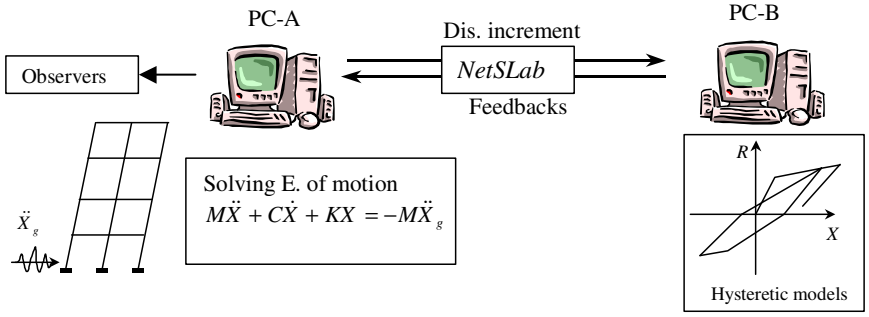


Fig.3. Schematic concept of remote earthquake response simulation using NetSLab

The concept of remote analysis of nonlinear time history simulation of structural response is schematically shown in Fig.3. As shown in Fig.3, the virtual testing controller located in PC-A handles the input of structure's data and ground accelerations, selects solution methods for the equation of motion, sends the

increment data to the remote virtual tester PC-B or others, reassembles the equation of motion once receives the feedback data, and decides the events for the next step. An example of the controller interface for remote analysis using NetSLab is shown in Fig.4. Programming for virtual tester(s) is essentially similar to that for controller, except only contains the hysteretic models.

In the authors' experiences, development of such virtual testing applications was particularly important and useful in evaluating the adequacy of the remote testing platform NetSLab. Remote analyses allowed the investigators to conveniently evaluate the platform under various simulated possible testing conditions, thus significantly reduced the need for the expensive actual tests during the development stage. Needless to say, remote analysis has its own significance in engineering applications.

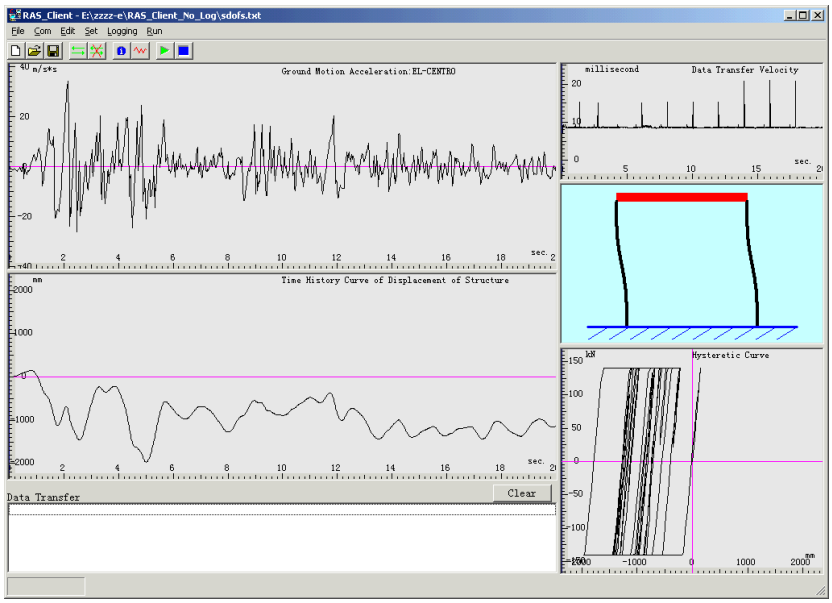


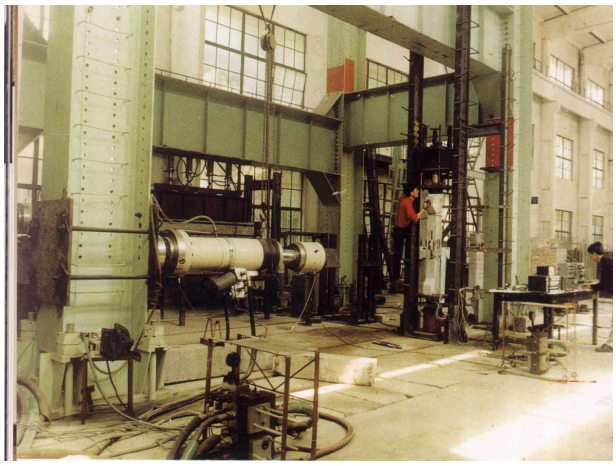
Fig.4. Example of controller interface for NetSLab [12]

APPLICATION OF NETSLAB FOR REMOTE PSEUDO DYNAMIC TESTING

Laboratory Facilities

Trial applications of NetSLab were carried out with development of several specific application programs for simulating earthquake response of structures. Current study is a collaboration effort between the University of Southern California and the Hunan University, both has adequate structural testing facilities. At this stage, only the large-scale pseudo dynamic testing capabilities of the two laboratories are mobilized. As illustrated in Fig.5(a), the participating USC equipment is a large-scale structural element testing frame with two actuators [13]. The two 150 ton capacity Parker actuators are controlled by a 2-axial control box. The participating equipment of the Hunan University is its multiple actuators and multi-axis loading system. The loading system currently has eleven Schenck actuators with capacities ranging from 10 ton to 65 ton, and two MTS actuators of 120 ton capacity. The control system includes a 4-axis MTS control card and a home-developed multi-axis control platform suitable for controlling actuators with different makers [14].

(a) USC test frame



(b) Hunan Univ. structural testing facility

Fig.5. Participating equipment for research collaboration

Pseudo-Dynamic Testing

The first trial application of NetSLab for remote pseudo dynamic testing was carried out using the large-scale testing frame at USC on a full-scale model of precast prestressed concrete pile to pile-cap assembly, as shown in Fig.6. The model specimen was first tested under cyclic loading for a separate study [15] and then reused to assess the applicability of NetSLab.

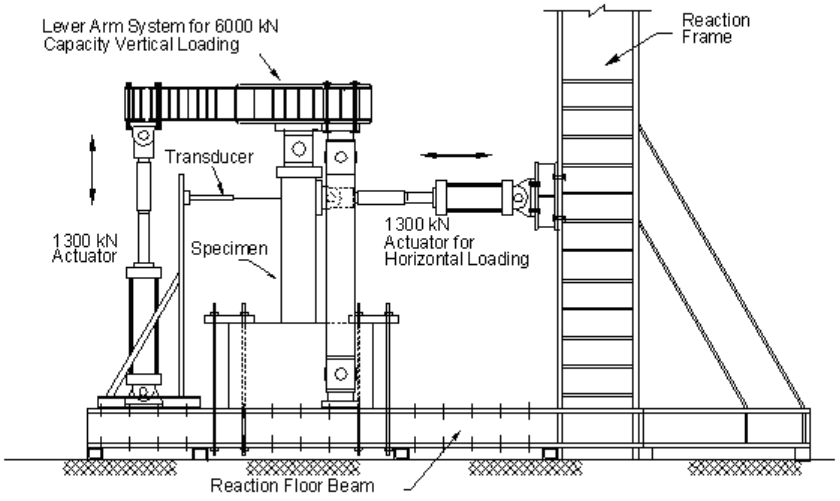


Fig.6. Test setup of precast concrete pile to pile-cap connection assembly model [15]

A program was written in Visual Basic for applying the NetSLab to the USC equipment, on the basis of the standard program Module for testers, discussed previously. This program is used to carry out pseudo-dynamic test through Compumotor 6270 motion control box for the Parker actuators, which are relatively inexpensive [16]. In the testing system, the control box controls motions of actuators and also collects

responses, which included displacement and force. The program for controller to organize the test is essentially the same as that used for remote analysis of earthquake response simulations. As shown in Fig.7, during a test, the controller first reads an earthquake record, which is composed by many ground acceleration data at a very small time interval. Based on the input information about testing structure, the controller can calculate the displacement for the testing element. The controller then sends the displacement command to the tester for operating the actuator. After the actuator achieves the command position, the tester sends the feedback data to the controller, and the controller then decides the command for the next step of testing.

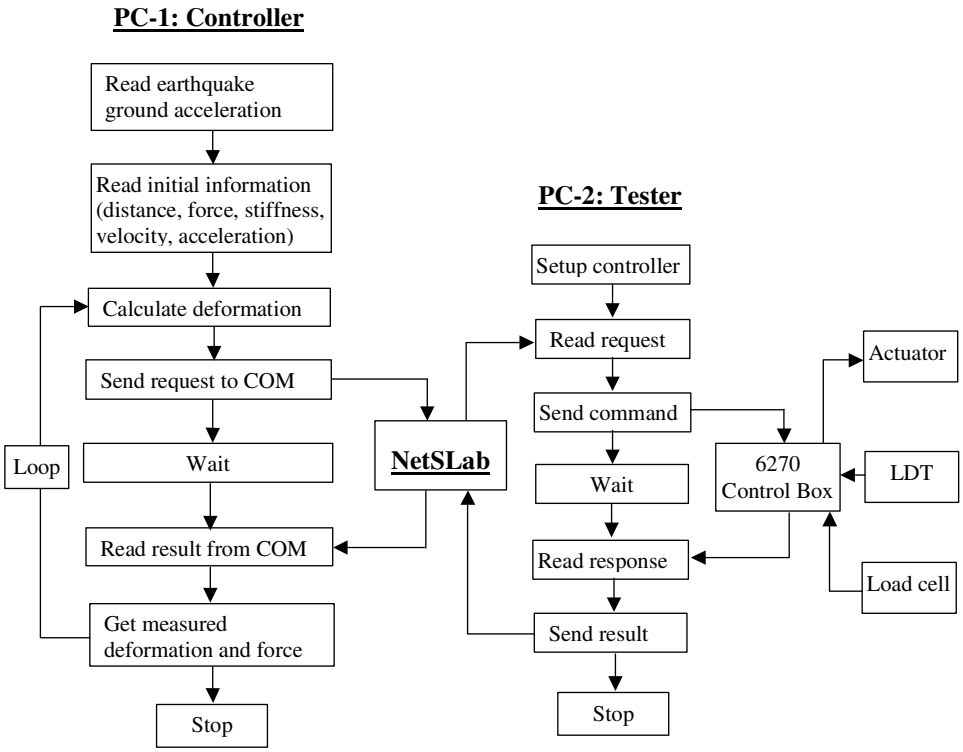
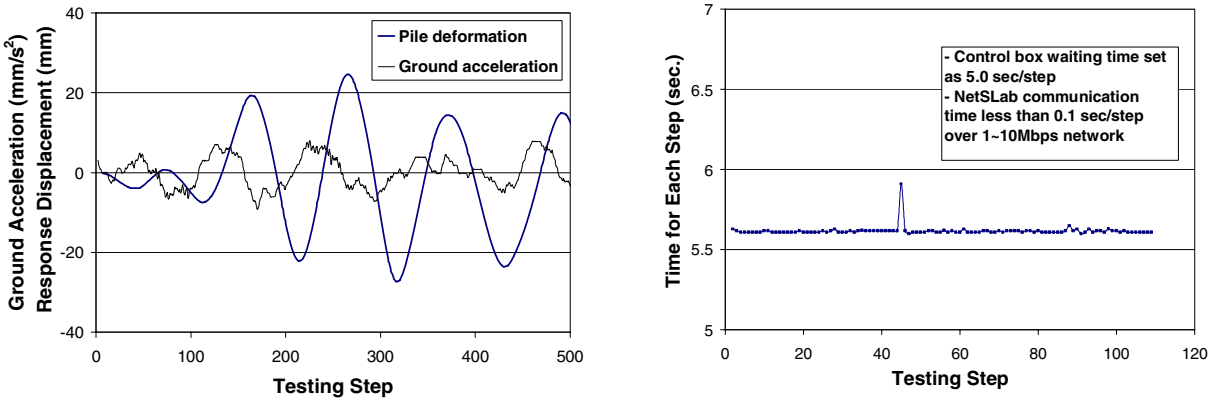


Fig.7. Main algorism for two-party pseudo dynamic testing [16].

Figure 8 shows an example chosen from a series of validation tests conducted. The test used a part of the ground acceleration from the September 19, 1985 Mexico City earthquake, which has about 3 minutes record at a 0.02sec interval. The input mass for the test was 1300 kg. The test was conducted for the first 500 steps of the ground acceleration record. The test took about 5.6sec for each step. The time required for each step was composed by three parts: communication time between remote controller PC and the local tester PC, actuator response time, and the waiting time set for the equipment control card, which for this case was 5.0sec, due to the limited capability in the hydraulic pump used for the actuator. Most of the time consumed was from the latter two, and the communication time between the controller and the tester using NetSLab for each step was essentially less than 0.1sec for the 1Mbps to 10Mbps network environment. The tests successfully validated the adequacy of the remote testing platform NetSLab. Further applications of NetSLab are currently underway at the Hunan University, the University of Southern California and elsewhere.



(a) **(b)**

Fig.8. Results from remote pseudo-dynamic testing over NetSLab on a pile specimen: (a) partial ground acceleration and response displacement; (b) recoded time per step (note: average communication time less than 0.1 sec)

CONCLUSIONS

A network platform (NetSLab) was developed for remote pseudo dynamic testing of substructures and structural elements under earthquake inputs, among geographically distributed structural laboratories or facilities. The NetSLab (Networked Structural Laboratories) was developed with a proposed data model and communication protocols. The concepts of Dynamic Unified Data Packet (DUDP), and Generalized Data Communication Agency (GDCA) were introduced to implement the platform along with an interface engine. Thus, the platform was made particularly easier to use by structural researchers using commonly available software. The adequacy of the NetSLab platform was validated by several remote virtual and actual tests at and between two universities in the US and China.

ACKNOWLEDGEMENTS

The research described in this paper has been jointly sponsored by the National Natural Science Foundation (NSFC) under the National Key Project; the Cheung Kong Scholarship from the Ministry of Education of China, the Hunan University and the University of Southern California. FutureNet Technologies Corporation, Monrovia, California, USA and its China subsidiary division provided generous supports under an agreement for joint development, academic promotion and applications.

REFERENCES

1. Mahin, S. A. and P.-S. B. Shing (1985) “Pseudo-dynamic Method for Seismic Testing.” Journal of Structural Engineering 111(7): 1482-1503.
2. Takanashi, K. (1987) “Japanese Activities on On-Line testing.” Journal of Engineering Mechanics 113(7): 1014-1032.
3. Mahin, S. A., P.-S. B. Shing, et al. (1989) “Pseudo-dynamic Test Method-Current Status and Future Directions.” Journal of Structural Engineering 115(8): 2113-2128.
4. Nakashima, M., H. Kato, et al. (1992) “Development of Real-Time Pseudo Dynamic Testing.” Earthquake Engineering and Structural Dynamics 21: 79-92.

5. "Assessment of Earthquake Engineering Research and Testing Capabilities in the United States," Earthquake Engineering Research Institute, Proceedings: Document WP-01A, Summary Report: Document WP-01. September 1995.
6. Nagata, K.; Watanabe, E.; Sugiura, K.; and Suzuka, Y. (1998), "Development of remote parallel pseudo-dynamic testing system by using INTERNET," Proc. of Fifth Pacific Structural Steel Conference, Oct. 13-16, 1998, Seoul, Korea, pp. 589-594.
7. Pauschke, J; Anderson, T.L.; Goldstein, S.N.; and Nelson, P. (2002), "Construction status of the George E. Brown, Jr. Network for Earthquake Engineering Simulation," Proceedings of the Seventh U.S. National Conference on Earthquake Engineering (7NCEE), July 21-25, 2002, Boston, MA.
8. Yang, Y.S.; Wang, S.J.; Wang, K.J.; Tsai, K.C.; and Hsieh, S.H. (2003), "ISEE: Internet-based Simulations for Earthquake Engineering, Part I: the Database Approach," Proceedings of the International Workshop on Steel and Concrete Composite Construction (IWSCCC-2003), Taipei, R. China, October 8-9, 2003, pp.301-310.
9. Wang, K.J.; Wang, S.J.; Yang, Y.S.; Cheng, W.C.; and Tsai, K.C. (2003), "ISEE: Internet-based Simulations for Earthquake Engineering, Part II: the Application Protocol Approach," Proceedings of the International Workshop on Steel and Concrete Composite Construction (IWSCCC-2003), Taipei, R. China, October 8-9, 2003, pp.311-320.
10. Xiao, Y. and Yi, W.J., (2002), "Proceedings of Workshop on Modern Structural Experiment (in Chinese)," Hunan University, Changsha, China, November 28-29, 2002.
11. Hu, Q and et al. (2001), "UniPipe: User's Manual," FutureNet Technologies Corporation, Monrovia, California, USA.
12. Guo, Y.R. and Xiao, Y. (2003), "Remote Collaborative Analysis of Structures (in Chinese)," Proceedings of Annual Conference of Engineering Mechanics, Chongqing, China.
13. Xiao, Y.; Yun, H.W., "Full-Scale Experimental Studies on High-Strength Concrete Columns," American Concrete Institute, *ACI Structural Journal,* Vol.99, No.2, March-April issue 2002, pp. 199-207.
14. Yi, W.J. and Liu, Y.J. (2003), "Development of Multi-Axis Control System of Hunan University (in Chinese)," Research Report of Structural Laboratory, School of Civil Engineering, Hunan University, Changsha, Hunan, China.
15. Xiao, Y., "Experimental Studies on Precast Prestressed Concrete Pile to CIP Concrete Pile-Cap Connections," *PCI JOURNAL*, Precast/Prestressed Concrete Institute, November-December Issue, 2003, Vol.48, No.6, pp.82-91.
16. Zhu, P.S. and Xiao, Y. (2003), "Manual for Remote Pseudo Dynamic Testing," Structural Engineering Research Report, University of Southern California.