

A Web-accessible Shaking Table Experiment for the Remote Monitoring of Seismic Effects in Structures

Mazen Manasseh, Eduardo Kausel, Kevin Amaratunga

**Department of Civil and Environmental Engineering
Massachusetts Institute of Technology**

Abstract

A remotely accessible system for controlling shake table laboratory experiments is presented. The Shake Table WebLab at MIT's Civil Engineering Department is implemented under the iLab initiative for the development of educationally oriented remote experiments. The fully functional system allows students and researchers to subject a model structure to vibration at its base while receiving accelerometer readings in real time from three different levels as the experiment proceeds. Registered Internet users can upload their own input data, such as seismic ground accelerations for a newly occurred earthquake, and therefore assess the behavior of a real structure to that earthquake. Two main characteristics of the developed web-based application are interactivity, provided through synchronized control/response processes, and sensor-based monitoring of the experiment, which are essential for real-time interactivity. The system is built on the dot-Net Framework through server-hosted Active Server Pages and browser-embedded Windows Form Controls. Web Service methods are implemented for initiating remote processes on the control server. In this paper, we state the motivations for this project, describe the various online activities and generic administrative features, and provide a description of the implemented technologies and system components.

1. Introduction

Recent efforts in engineering education have focused on supplementing traditional means of experimentation with virtual laboratories that are remotely accessible through the World-Wide Web (WWW), despite allegations that such a shift from the physical laboratory environment to an online experimentation space detaches students from irreplaceable real-world experiences gained through physical interaction with a laboratory setup². For instance, a laboratory presence is invaluable when conducting experiments on reinforced concrete members or soil samples, inasmuch as visualizing material behavior is crucial to learning. For these reasons, the primary goal of virtual experiments should be to improve the quality of learning, and not to supplant real

lab experience. Nonetheless, virtual or remote experiments may also allow students to familiarize themselves with experiments through virtual online versions before proceeding to actual laboratory sessions³. Depending on the type of experiment and the equipment involved, a remotely-conducted experiment can be more rewarding for student learning and simplify considerably the logistics of laboratory resources. Remotely controllable laboratories, either as replacement or supplement to traditional ones, have valuable benefits by allowing a more efficient management of the laboratories as well as facilitating distance-learning programs⁷. Moreover, online experimentation allows for lab-to-lab collaboration among universities and investigation centers by providing research and student groups access to a wide collection of experimental resources at geographically distant locations. An added benefit is the reduced costs incurred when different educational departments or institutions share facilities, since automated, remotely accessible systems are more cost effective than scheduled lab sessions conducted by salaried assistants and technicians, not to mention the cost and effort needed to maintain the laboratories. It should be emphasized, however, that virtual experiments are not simply canned computer simulations, but entail real laboratory equipment being controlled remotely. Thus, virtual experiments incorporate physical components providing a sense of reality⁷.

The iLab Group is an interdepartmental effort aimed at developing virtual and remotely managed experiments for both undergraduate and graduate courses at MIT. It is sponsored by the iCampus Project funded by the MIT-Microsoft Alliance. One of the first efforts of the iLab Group was to provide a shared architecture to facilitate the development of virtual laboratories. To this aim, experiments were classified in three groups, depending on the kind of interactivity between the user and the laboratory. The first type of experiments is defined as ‘batched’. Such an experiment expects the user to set values for input parameters at once before the experiment is started. After the experiment request has been processed, the user is provided with the values of the different output parameters. The Microelectronics WebLab (<http://weblab.mit.edu>), for experimenting with semiconductor devices, is an example of a virtual laboratory based on the notion of a batched experiment. The second category is termed ‘interactive’ and involves more interaction between the client and the laboratory server as various input parameters can be changed while the experiment proceeds. The Heat Exchanger Project (<http://heatex.mit.edu>) for experimenting with concepts in thermodynamics is representative of an interactive experiment. The third category includes experiments that focus on streaming sensor data. This category requires intensive data transfer from a server to a client whereby the delivered data is collected in real-time by different sensors. The flagpole project (<http://flagpole.mit.edu>), which is considered a preparatory step towards the monitoring of physical infrastructure¹, serves as an instance of a streaming sensor experiment. This paper presents a description of the Shake Table WebLab (<http://flagpole.mit.edu:8000/shaketable>), a successor to the Flagpole Project (a sensor lab for the monitoring of wind loading on a 31m tall flagpole).

2. Objective of the Shake Table WebLab

The Shake Table WebLab presents an online laboratory that is both interactive and heavily reliant on accelerometer sensor readings. Using the Shake Table WebLab, students can excite a two-story, 3ft tall structure by vibrating its base while simultaneously monitoring its behavior as they receive accelerometer readings from three different levels. Moreover, the Shake

Table WebLab allows any Internet user —both within and outside of MIT—to upload his/her own input data, such as the ground accelerations of a newly occurred earthquake, and observe the corresponding behavior in a real time. For instance, an interest to structural students is to explore the characteristics of the seismic motion as it propagates to the higher levels of the structure, and to assess the vibration signatures at various levels. In addition, a shaker table that is remotely accessible from the classroom can be of significant help in conveying concepts in structural dynamics. Accompanying class lessons with experimental demonstrations enhances the traditional means of conveying course material, and it adds interest to a class in structural dynamics, inasmuch as students are then able to observe in real-time the illustration of theoretical concepts. In addition, in-class demonstration allows experimentation to proceed in parallel to lecturing without the need to make arrangements for dedicated lab sessions.

A simple yet secure access via the Internet is essential for a successful design and implementation of a remotely operated shaking table. To illustrate, were students to conduct their experiments in the lab, they would need to devote precious time to get acquainted with the exacting procedures needed to operate the hardware components in a safe and error-free manner. By contrast, a web interface that is well-adapted to the specific needs of the users eliminates the need to spend course time instructing students on the proper operation of the test apparatus while shielding the users from the experimental hazards. In the case of the shaking table, this powerful equipment is vulnerable if it is not properly calibrated, or if it is subjected to loading above safe operating limits. Also, students are able to directly explore their ideas without being concerned about the technicalities of, or interfering directly with, the lab setup. The web application is configured so as to eliminate possibilities of human error while activating the shaking table. Such errors may lead to unsafe conditions that jeopardize the lab setup as well as anyone near it. These necessary precautions are carried automatically by the software-hardware interface whenever a web user conducts an experiment remotely.

3. Hardware Components

The hardware setup used for the Shake Table WebLab consists of a shake table (manufactured by Quanser Consulting), a power module equipped with a microcontroller-based safety circuit to drive the table, a data acquisition board (MultiQ) to drive the power amplifier and collect sensor data, three accelerometers (range of ± 5 g), and a two-floor test structure. The shaker table consists of an 18" x 18" linearly moving stage and operates under a maximum load of 33lb, a peak acceleration of 2.5g and a peak stroke of three inches⁵. The assembled structure, which is three feet tall, has

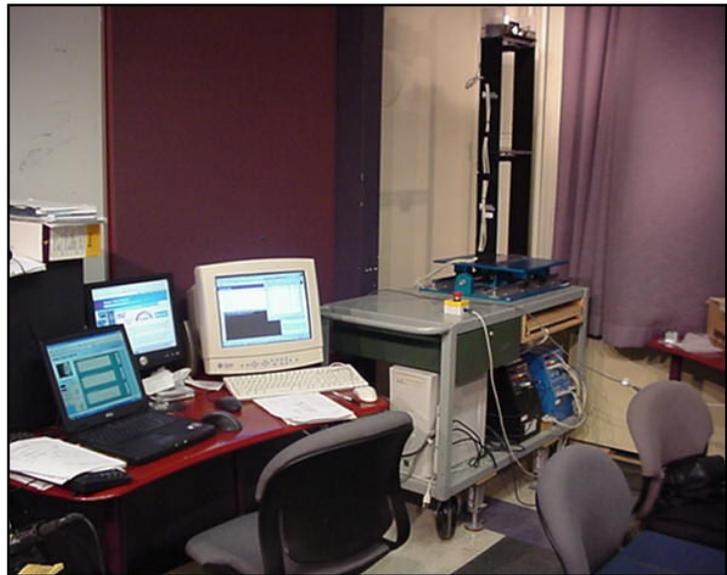


Figure 1. Shaker table test structure and hardware assembly on the right. Server computers on the left.

one accelerometer on each floor, while a third accelerometer is attached to the moving base of the table. Figure 1 shows the assembled hardware with the corresponding server computers.

4. Software interface

Building a remotely operated shaking table with a flexible yet stable architecture was one of the main challenges of this project. The shaking table and ancillary equipment is a product of Quanser Consulting, who also provides the WinCon control software needed to operate it. WinCon relies on a client/server setting that does allow carrying out experiments from outside the lab⁶. As delivered, however, such remote operation is solely based on a TCP connection that requires a client program to be installed on the user's computer. In addition, WinCon uses programmatic diagrams called Simulink models, which operate within the Matlab environment, each of which is in turn restricted to a single experiment. For instance, each stored earthquake has its own Simulink model that contains the seismic data embedded within. Each such Simulink representation of experiments must then be compiled with a C-compiler prior to running the executable file with WinCon. Clearly, this would require students and remote operators alike to install on their computers the proprietary WinCon as well as Matlab, not to mention the compiler, which would make the remote experiments with the shaking table largely impractical. In the light of these severe restrictions, it was necessary to significantly extend the original capabilities of WinCon by developing an external software interface that fully insulates users from the limitations and idiosyncrasies of both WinCon and Matlab, provides security, flexibility and multi-user capabilities, and is easily accessible from commonly used Internet browsers. In the ensuing, we describe some of the characteristics and capabilities of this software interface.

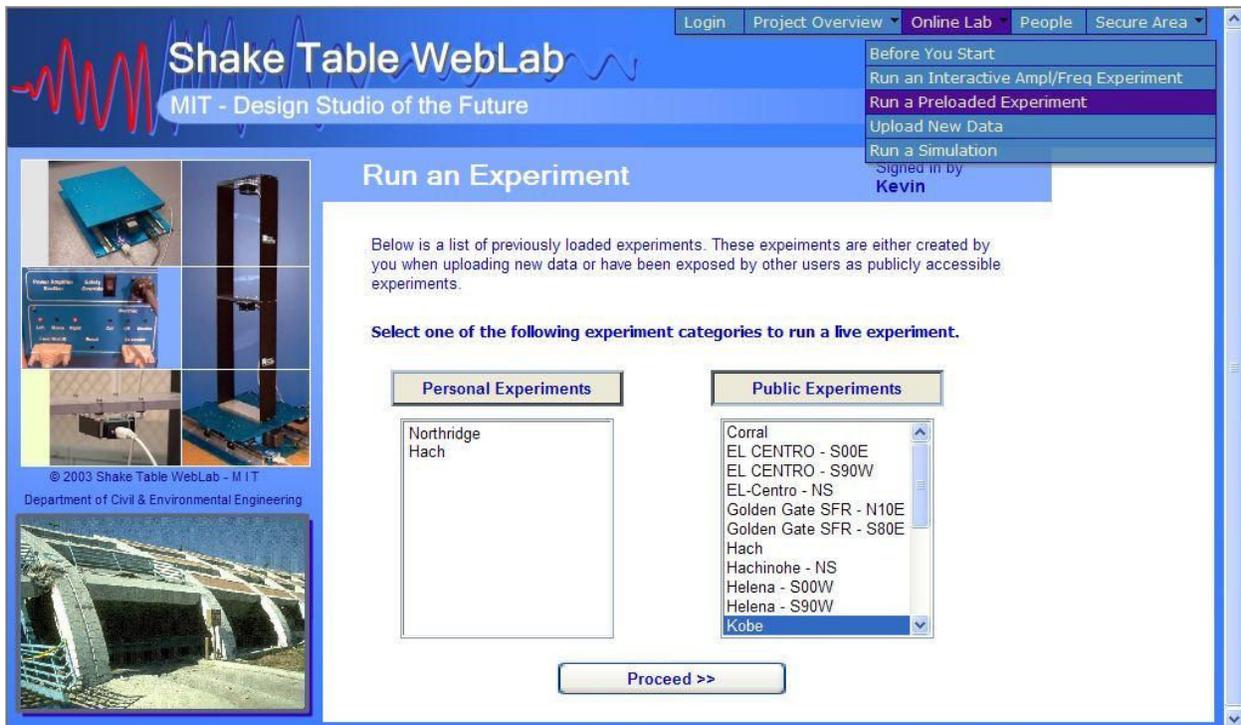


Figure 2. Shake Table WebLab menu page showing lists of publicly and privately accessible earthquake experiments.

Three types of online capabilities are available, namely: 1) Run an archived earthquake experiment; 2) Repeat a previously uploaded experiment, which the remote user may have declared as either publicly viewable and open to the whole user community, or privately accessible only by the experimenter, see figure 2 (more on user permissions and accessibility is described in the ‘*System Features*’ section); and 3) Run an interactive experiment in which the motion characteristics are modified while the experiment progresses, say by changing the amplitude and/or frequency content.

The first two types can be described as non-interactive in the sense that once the experiment is started, users have no further control on the shaker table other than observing (and perhaps manipulating) the stream-back of sensor responses. In these cases, the data consists of four streams: three accelerometer records and the table's base displacement record. The second type allows users to create new experiments to the preloaded experiments. This functionality is useful for experimenting with new earthquakes, which entails uploading seismic accelerations and running the corresponding experiments. However, since a universal standard for seismic data representation does not exist, it was necessary to establish a well-defined format for uploaded files. To avoid complications when preparing a new experiment, the chosen format can accommodate various numeric layouts in arbitrary units of measurement. On the other hand, uploading new earthquake data files requires server-sided checking to verify the physical realizability of the data as well as scaling down table displacements to a scale compatible with the capabilities of the equipment. Thus, while the experiments are exact representations of downloaded seismic accelerations, the motion intensities are scaled so that the table's maximum allowable stroke of 3 inches is not exceeded. With the newly developed web application, all necessary system processes for uploading new data are carried out at the click of a button.

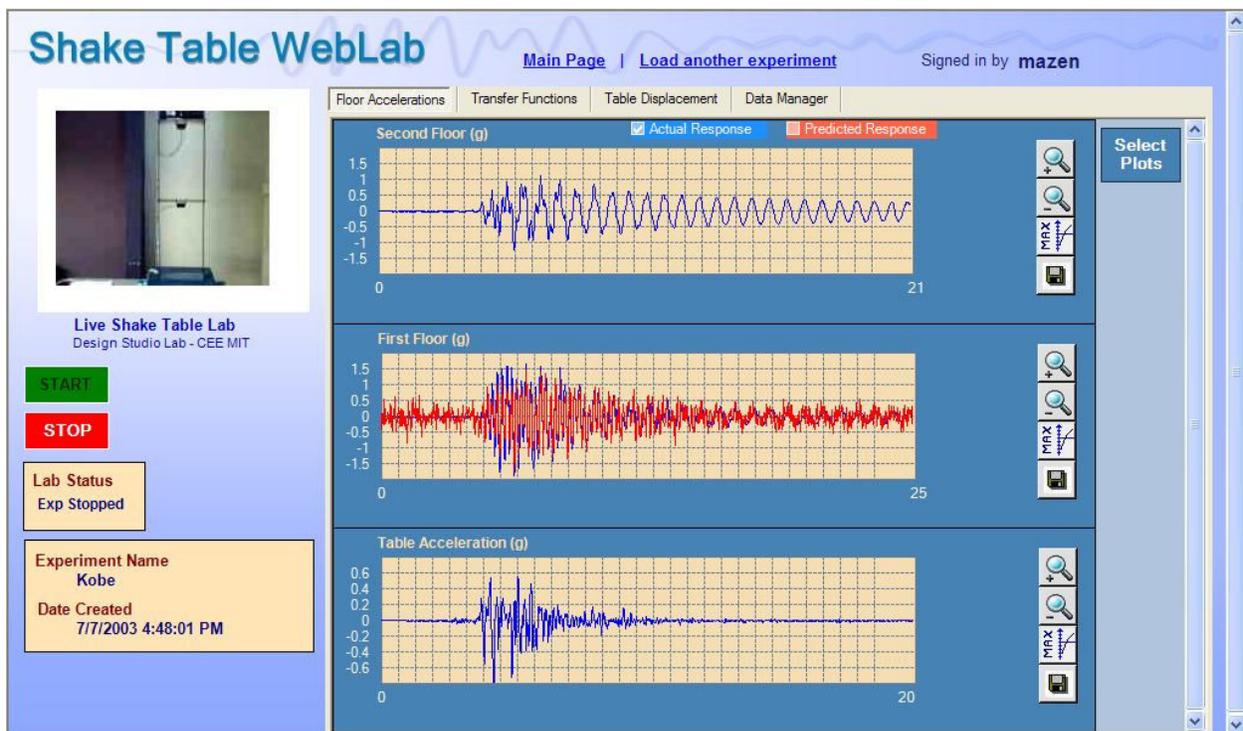


Figure 3. Snapshot of the plotted floor accelerations for the Kobe earthquake.

Figure 3 is a snapshot of the web environment that accesses the lab. The interface consists of tabbed pages that allow a user to dynamically shift from one page to another as the experiment is conducted. The 'Floor Accelerations' page (figure 3) displays three plot charts for each of the floor acceleration readings. Additional plots may be displayed for the Fast Fourier Transform spectra for each floor, which are incrementally updated while data is being received. The displayed plots allow choosing between measured and predicted acceleration plots. The latter are based on estimated transfer function, as explained in section 5.4. Note that figure 3 shows an overlap of both the actual and predicted responses for the first floor accelerations.

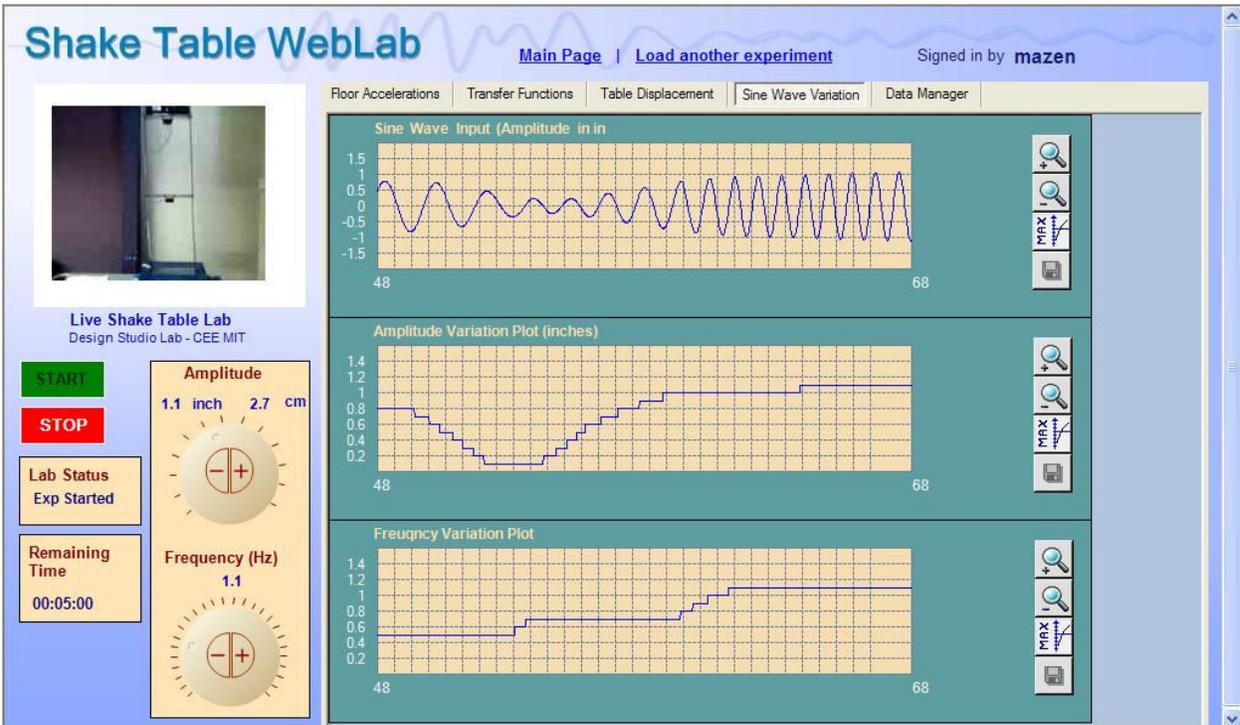


Figure 4. The WebLab's sine wave variation page implemented for the interactive experiment.

In contrast to the two previous categories, the third experiment type is interactive whereby displacements of the table are dynamically set by the user instead of being read from a file, see Fig. 4. In this case, the input motion is defined by a sine wave signal, the frequency and amplitude of which may be modified by the user as the experiment proceeds. In addition to the four monitoring streams described previously, the sine wave variation experiment provides three other signal readings, namely the amplitude, frequency and the input sine wave applied to the table. Accordingly, users are able to simultaneously change the input displacement signal and monitor the behavior of the mounted structure as it responds to instantaneous alterations on the client side. Figure 4 shows the web interface for the interactive sine wave experiment with adjustable knobs to change the input sine wave. The three plot charts display the variation of the amplitude and frequency of the wave as imposed by the user.

5. System Architecture

The Shake Table WebLab is built on a multi-tier architecture that supports client-server interaction. A client is representative of an Internet user computer – for example a student accessing the website. The term server incorporates computer systems located at the MIT Civil Engineering Labs that serve as intermediaries for establishing communication between a web client and the shaker table control apparatus. Server-sided components are distributed among three machines with well-defined interconnections: a shake table control server, a web/database server, and a streaming video server. Distributing server tasks among the three machines is essential for maintaining adequate system performance levels. Refer to figure 5 for an illustration of the overall system components and the channels of communication involved.

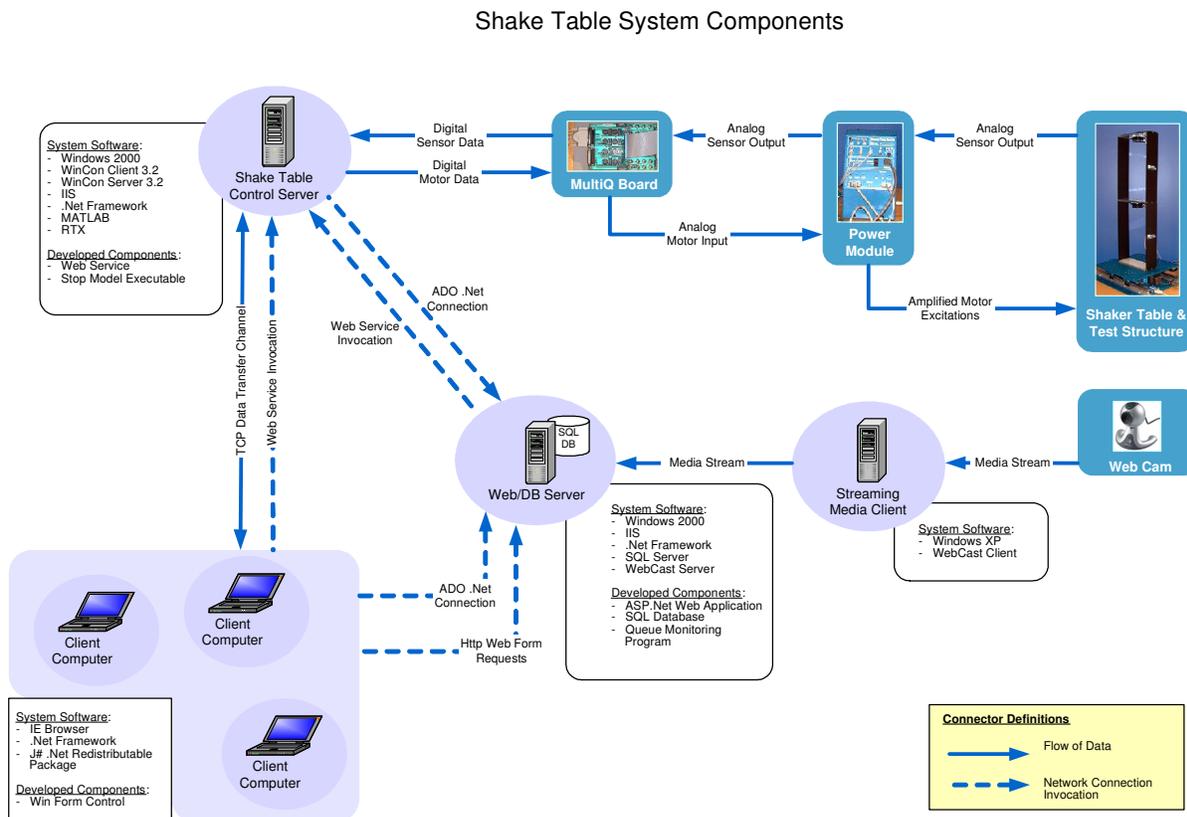


Figure 5. System architecture of the interrelationship between various hardware and software components.

5.1 Shake Table Control Server

The Control Server has direct access to the shaker apparatus through a data acquisition/controller board which is in turn connected to the power module. This same server runs Quanser Consulting's WinCon proprietary software package for driving the shaker. WinCon is originally aimed for client/server operation by installing WinCon server and WinCon client on separate computers with a TCP data transfer channel between the two. However, for the purpose

of the project, both of those programs are installed on the Shake Table Control Server. The WinCon Server application is accompanied by an external interface that provides an additional TCP channel for other applications to interact with the shaker. Since the aim of the project is to provide lab accessibility through an Internet browser, a user friendly client-side web application is needed. This application relies on WinCon as an intermediary for controlling/monitoring the shaker table equipment.

In order to establish a connection between the web application and WinCon, an essential mechanism is required to automatically carry out processes on the shake table control server. Accordingly, a Web Service⁴ is hosted in Windows' Internet Information Services (IIS) on the Control Server to translate client requests, carried out through the web application, into corresponding activities on the control server. For instance, a user request to start an experiment triggers a web service method to check the status of the control server, calibrate the shake table before initiating the experiment, and check that the calibration has succeeded. Another web method is then triggered to launch WinCon and load the corresponding experiment files.

Other major applications hosted on the Control Server include Matlab as a means for scaling down of original earthquake data —when a user chooses to experiment with new earthquakes— into intensities that comply with the shaker table capabilities. Functions within Matlab are triggered through a web method. This method uploads the new data file from the user's computer, launches Matlab as a background-running process on the control server and sends the necessary commands to the Matlab environment. Interfacing the MS .Net-based web service with Matlab is achieved through a dedicated CLR (Common Language Runtime) compliant type developed in VB .Net for interconnecting .Net applications with Matlab. The data file that results from the uploading and scaling procedures is stored under an identified directory on the Control Server in order to be read by WinCon when the corresponding experiment is requested.

5.2 Web/Database Server

The second server computer has two main functionalities: an IIS web server hosting an ASP .Net application⁴ and a database server implemented in MS SQL. The web application serves as the access point for Internet users to enter the Shake Table WebLab. Managing accessibility demand, such as user authentication and queuing of multiple client requests, is a functionality of the Web Server. For this purpose, a database is implemented to store user accounts, previously loaded experiment records, queued clients, and other modifiable system parameters such as the frequency of the sensors' sampling rate and the status of the lab.

The online web application allows a real-time control of the shaker table while receiving streaming sensor data from the Control Server. Accordingly, communication starts through http messages between a client – through an IE browser – and both the Web Server and the web service hosted on the Control Server. Through an http transfer protocol the client application calls web methods that launch the necessary processes on the Control Server. Afterwards, the web application maintains an open channel for data transfer to and from the Control Server. For this reason, the web application incorporates a Windows Form Control that establishes a TCP connection with WinCon's External Interface Window (EIW) hosted on the Control Server. The

Windows Form Control is embedded within an Active Server Page (aspx) and therefore runs on the client's machine in a CLR-compliant Internet browser such as Internet Explorer.

5.3 Streaming Video Server

The web laboratory environment includes a live video display of the shaker table located in the civil engineering lab. Live video is broadcast through a streaming media server called WebCast. A web cam is connected to this server which in turn publishes the video stream to be viewable in the user's browser. The video display enables a user to identify the different modes of vibrations of the structure as it undergoes dynamic excitations.

5.4 Client Application

Users access the lab through an Internet browser. An embedded Windows Form Control provides a rich interface with control and plotting features as described in the 'Online Activities' section. The client application is expandable in terms of the data processing tools that might be added depending on the purpose of the user. An already developed functionality is accessible through the 'Transfer Functions' tab shown in figure 6. The aim of this tool is to determine the transfer functions from the table motion to each of the floor motions. The two resonant frequencies of the installed structure are also extracted, and are found to be 1.5 and 5.3 Hz, respectively. Once the transfer functions have been estimated, they can be used to predict the structural response by means of standard methods in structural dynamics for any arbitrary table acceleration. Of course, the results of these predictions do not exactly agree with the measured response for a variety of reasons, but especially because the structure is not exactly a linear, two-degree of freedom system.

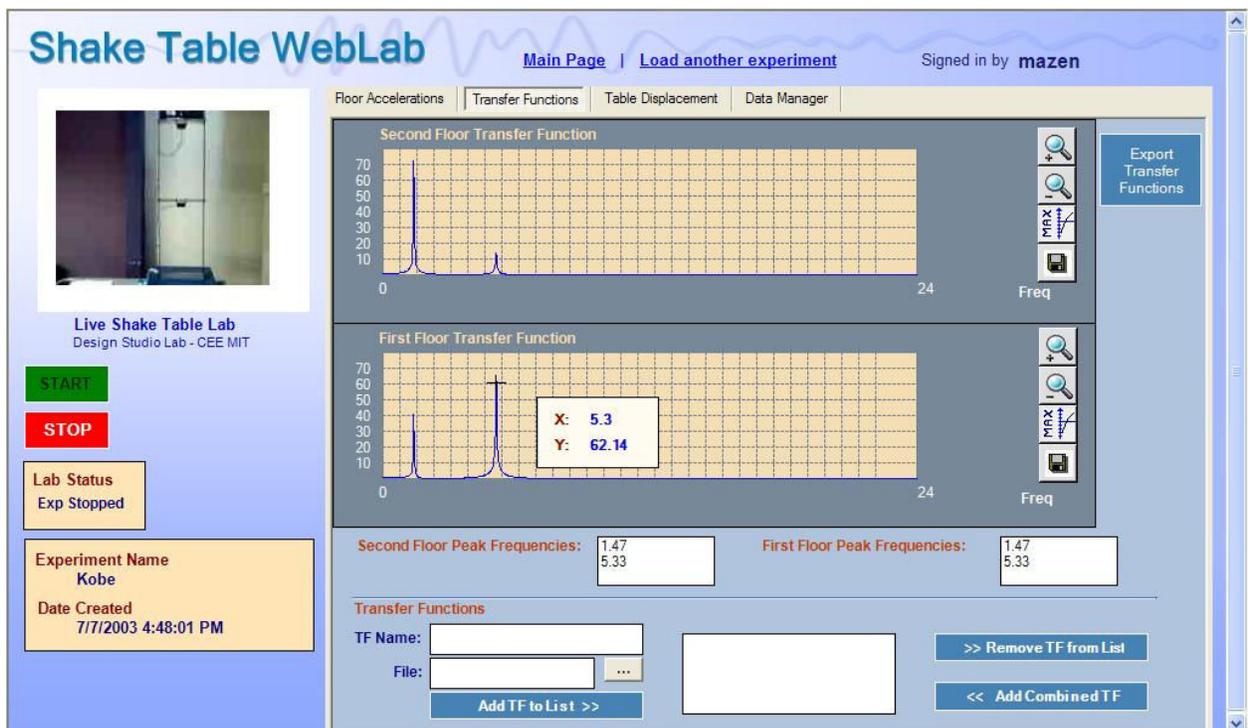


Figure 6. Transfer function tab view displays two plots showing the frequency peaks of each floor.

6. System Features

An essential feature for an online experimentation environment is the manageability of user accounts in terms of authenticating accessibility and authorizing permission to restricted activities. Moreover, a web channel to control laboratory equipment necessitates a request queuing mechanism that avoids the possibility of simultaneous multiple client-control scenarios. We explain next why these features are vital requirements for the Shake Table WebLab and how they have been implemented.

6.1 User Account Management

Access to the Shake Table WebLab requires registration which can be carried out online. Accessing any of the online lab activities entails a user login with a username and password. ‘Administrator’ and ‘Developer’ are two special types of user accounts that are given added accessibility privileges. A user with an administrative account can view, edit and remove other registered users. Another functionality restricted to an administrator is the ability to make uploaded earthquake experiments publicly accessible. Newly created experiments by ordinary users are only visible to the experiment owners who have uploaded the new data files. Administrators may, however, choose to configure new experiments as public. Consequently, the added experiment will be displayed under the public experiments list and may therefore be activated by any registered user. An administrator account is of value for a course instructor or an assistant who needs to manage student profiles and add to the public experiment archive. Creating a secondary administrative account is only possible when logged as administrator.

As to the ‘Developer’ account, it is intended for a technical developer who is to maintain and upgrade the system. To prevent access to the shaker table at times when the system is being upgraded, a protection key is implemented. Accordingly, an administrator can lock the online accessible channel to the shaker whereby none of the registered users except the ‘Developer’ can run experiments. This approach serves as a precaution against external interference when the table is being tested.

6.2 Queuing User Requests

In order to organize online requests for activating the table, a client monitoring solution is implemented with the following functionalities: 1) queue user requests in order to prevent more than one client from activating an experiment at the same time; 2) remove any disconnected client from the queue, and 3) stop the shaker table if an active user was accidentally disconnected. Queued users are also removed from the queue if and when the user closes his/her browser. Accordingly, a dynamic queuing mechanism continuously checks client status and grants lab access on a first-come-first-serve basis.

The aforementioned features are implemented with a queue-monitoring program hosted on the Web Server. When a logged-in user attempts to run an experiment, the client application (as a Windows Form Control) creates a unique ID which is logged into an SQL Server queue table with the corresponding time of request.

Process Model of User Requests

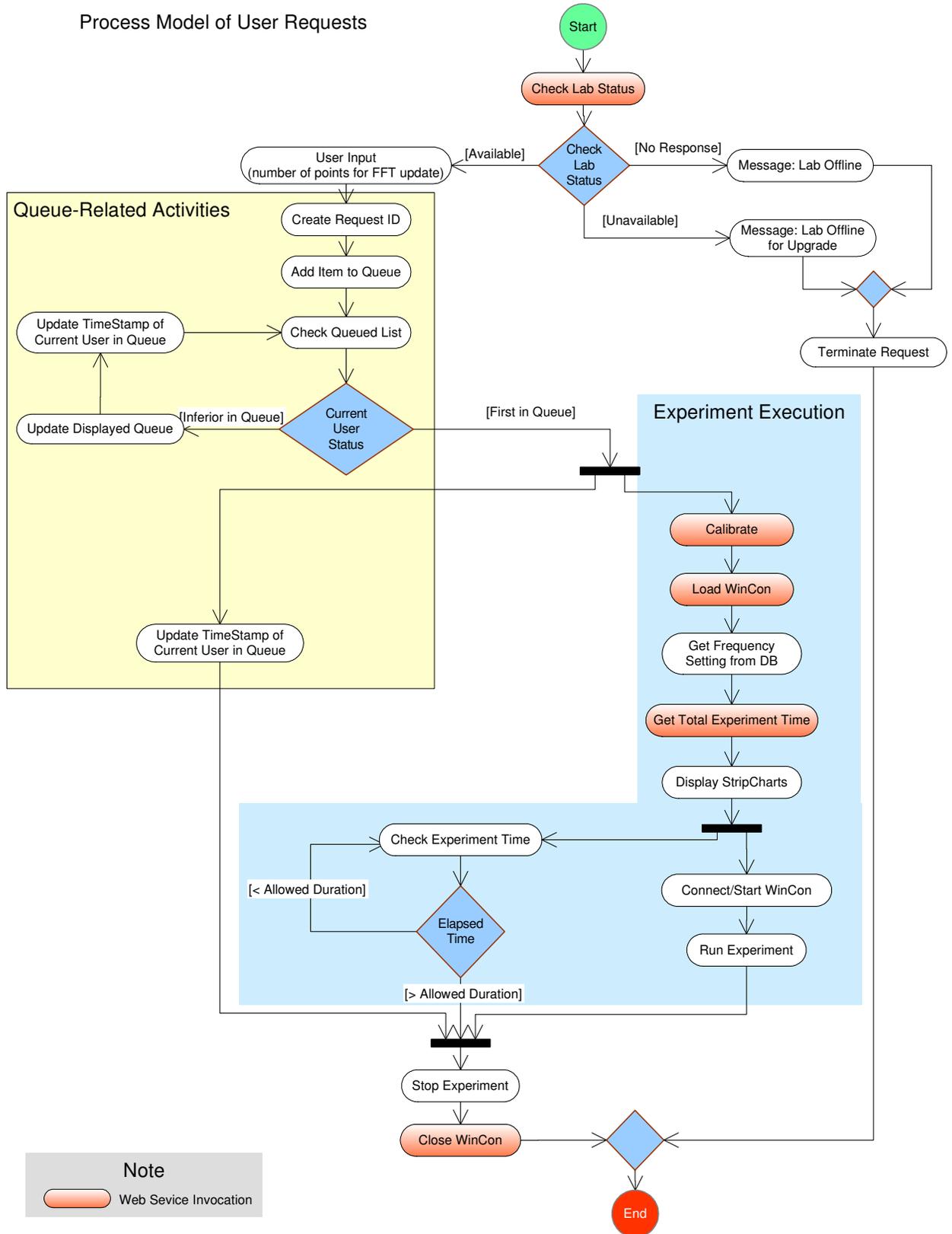


Figure 7. Process model for web-user requests to operate the shaker table.

During the time before a user is granted control permission, two different programmatic functions proceed: one running on each queued client computer, and one continuously monitoring program on the Web Server. The aim of the processes that run on the client side is to update the client status on the server at five second intervals by renewing the client's time stamp in the queue. Moreover, this same process keeps users informed of their rank in the queue and the estimated remaining time for their turn. The client-side processes are illustrated in figure 7 which presents the implemented model for handling user requests. In the meantime the monitoring program on the server checks for any updates in the database queue table at five second intervals. As a result, the queue monitoring program removes any queued item older than ten seconds; if the status of the user who is in charge of the shaker table hasn't been updated for more than ten seconds, the program disposes the corresponding user from the queue after stopping the table in case it was in operation mode.

7. Conclusion

This article presented various aspects of the design and implementation of a shaker table remote laboratory. The outcome of the project is an online space for the real-time interactive control of a shaker table. We discussed the effectiveness of online experimentation in general and focused on the deliverables of the project. We then described the various system components and the necessary features that are of valuable use in other similar remote laboratories.

The Shake Table WebLab is open to upgrades as additional course-specific applications can be easily added to the already implemented system. Since the foundation of a remotely accessible system is developed, future works are aimed at processing output data as the structure responds to different vibrations and therefore allow students to intelligibly compare theoretical means of understanding structural dynamics with real-case applications.

References

1. Amaratunga, K., and Sudarshan, R., "A Virtual Laboratory for Real-Time Monitoring of Civil Engineering Infrastructure", *International Conference on Engineering Education*, August 18-21, 2002.
2. Khan, R., "Software Architecture for Web-Accessible Heat Exchanger Experiment", *SM Thesis, Department of Civil and Environmental Engineering*, MIT, 2002.
3. Powell, R. et Al, "Using Web-Based Technology in Laboratory Instruction to Reduce Costs", *Computer Applications in Engineering Education*, Wiley Periodicals Inc., Volume 10, Issue 4, 2002.
4. Prosser, J., "Programming Microsoft .NET", p.521-560, 417-170, *Microsoft Press*, Washington, 2002.
5. Quanser Consulting Inc., "UCIST Shake Table".
6. Quanser Consulting Inc., "WinCon 3.2".
7. Wagner, B., and Tuttas, J., "Distributed Online Laboratories", *Engineering Education and Research*, iNEER, 2001.

Biographical Information

EDUARDO KAUSEL

Received his PhD from Massachusetts Institute of Technology in 1974 and currently a professor in the Civil and Environmental Engineering Department at the same university. His areas of specialization include: structural dynamics; soil dynamics; soil structure interaction; earthquake engineering; structural mechanics; machine vibration, non-destructive testing.



KEVIN AMARATUNGA

Received his PhD from Massachusetts Institute of Technology in 1996 and currently an associate professor in the Civil and Environmental Engineering Department at MIT. His areas of specialization include: Information technology, wavelets and multi-scale data representations, signal and image processing, multi-scale methods for large scale physical simulation, modeling and management of large-scale systems.



MAZEN MANASSEH

Obtained his B.E. in Civil Engineering from the American University of Beirut. Mazen is currently a graduate student and research assistant at the Information Technology Division of the Department of Civil and Environmental Engineering at MIT. His interests are in computer applications in civil engineering.

