Proceedings of the 40th IEEE
Conference on Decision and Control
Orlando, Florida USA, December 2001

A Webcast Virtual Laboratory on a Frequency Modulation Experiment

C.C. Ko, Ben M. Chen, K.P. Chan, C.D. Cheng, G.W. Zeng and J. Zhang
Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117576

Abstract

A number of Internet remote experimentation has been successfully developed for teaching and research purposes in the National University of Singapore (NUS). As only one user can assume control of the apparatus in any physical or remote experiment, the access to these experiments has hitherto been only been limited to single users one at a time. Without any increase in the hardware experimental apparatus needed, this paper presents a new webcast based approach for remote experimentation that allows several observing users to view an existing remote experimental session while it is being conducted by a main user. Multicast, the state of the art technology, is being adopted to implement the webcasting capability for remote experimentation purposes with some degree of reliability provided. In multicast, only one copy of the same data is sent to a group address, reaching all the observing users in a particular remote laboratory session. Thus, the system can be accessed by as many users as possible without overloading network and server resources. This is particularly useful as the number of users that may want to observe the experiment cannot be predicted in advance. The webcast virtual laboratory can be accessed at the following web site, http://vlab.ee.nus.edu.sg/vlab1/freqmod/index.html.

Keywords: Webcast Virtual Laboratory, Multicast Capable, On-line Experiment, Internet.

1. Introduction

Spurred by development in computer science and network technology, the use of the Internet has been expanding exponentially. It is now extensively used as a connectivity and reference tool for commercial, personal and educational purposes. In education, the Internet opens a variety of new avenues and methodologies for enhancing the experience of learning as well as expanding educational opportunities for a larger pool of students. Specifically, distance education and non-traditional classrooms have the capability to reach more students using specialized instruction and self-paced learning. As described by Poindexter and Heck [1], the integration of the Internet with education can be based on: (i) developing a course web site to centrally house various online functions and facilitate course management; (ii) creating a remote laboratory to replace physical experiments with multimedia animation or simulation; and (iii) developing a virtual laboratory for students to set up parameters and conduct experiments from a remote location.

Examples with the first two capabilities can be found in [2–8]. These examples give a good demonstration of how the web, and in particular, systems that are based only on theoretical/simulation materials and have capabilities (i) and (ii), can be explored for education. While a good learning experience can be obtained from such systems in many situations, it is commonly recognized that effective and complete learning, especially in engineering and science, requires a mixture of theoretical and practical sessions. In order to understand how theoretical knowledge can be applies to real world problems, practical exercises are essential [9-10].

To address this important issue, a number of attempts have been made to provide students with practical exercise or experimentation experience through the web. Generally, these are done through the use of web-based virtual laboratories that have capability (iii) and that enable students to set up parameters to run experiments from a remote location. Examples can be found in [11-16]. For instance, in [11], a virtual laboratory called SoftLab in Purdue University has been set up to provide an environment for both physical experiments and numerical simulation. Users are able to remotely control some real instruments, after installing SoftMedia, an exclusive software, for accessing the service in the laboratory.

Since systems with capability (iii) have to control instruments in real-time, they are inherently more complicated and it is not surprising that earlier systems have the deficiency of requiring the user to set up special software before access becomes possible. As issues such as installation time and system compatibility may crop up in setting up special software, it will be more convenient if it is possible to access virtual laboratories through common browsers such as Microsoft Internet Explorer or Netscape Navigator [17 - 20]. Due to the limitation of hardware resources, normally, these virtual laboratories can only be accessed by individual user one at a time.

In this paper, we present a webcast laboratory (WVLAB) on a frequency modulation experiment. In this experiment, users are able to control instruments and measure spectrum of modulated signals in real-time. However, since there is only one set of apparatus, only one user can be given access to it. Users who want to conduct the experiment will be denied access unless instruments are available and no user has been given control or is conducting the experiment. Since it will also be very meaningful for students and others to observe how an experiment is being carried out, the webcast laboratory is developed to serve this purpose and to provide students a good learning experience. However, there are several inherent difficulties in implementing webcasting capability for remote experimentation purposes. Since the number of users that may want to observe the experiment can be huge, the bandwidth required of the server and network can be significantly reduced when IP multicast is adopted. With IP multicasting, only one copy of the same data is sent to a group address (Class D IP address used for multicast transmission), reaching all the observing users in a particular remote laboratory session. Without multicasting, the same information must be either carried over the network multiple times, one time for each user, or broadcast to everyone on the network, consuming unnecessary bandwidth and processing, and limiting the number of participants. In addition, some degree of reliability is provided onto multicasting data by sending UTF 8-encoded characters [23]. If packets received are corrupted, the decoding of UTF 8-encoded characters would not succeed and hence cause problems in observer's program. Consequently, display of the observer would not change until next healthy packets arrive. Figure 1 shows the scenario for a webcast based laboratory session where the left screen is the main user instrument interface and the other two is for observation.

Figure 1. Scenario for a webcast laboratory session

The rest of the paper is organized as follows. In Section 2, the typical user interface is described to show how WVLAB helps students gain the same experience as with viewing experiments in the real laboratory. Section 3 describes the hardware and software architecture, and Section 4 discusses the impacts of the virtual experiment. Finally, conclusions are drawn in Section 5.

2. User Interface

The objective of the experiment is to study the frequency spectra of frequency modulated signals. A spectrum analyzer, a signal generator, a frequency counter, a voltmeter and a circuit board with a variable resistor are used. Figure 2 and Figure 3 show the typical screen display of the frequency modulation experiment for the main user and observer.

In order to give the observers an impression that they are actually viewing physically existing instruments, the user interface of a remote laboratory should be as realistic as possible. Also, the user interface should be virtually identical to the main user. However, the observer is not given control of the real instruments and therefore, mouse cursor will not be set as a hand icon when it is moved onto various lead and knobs as in the case for the main user.

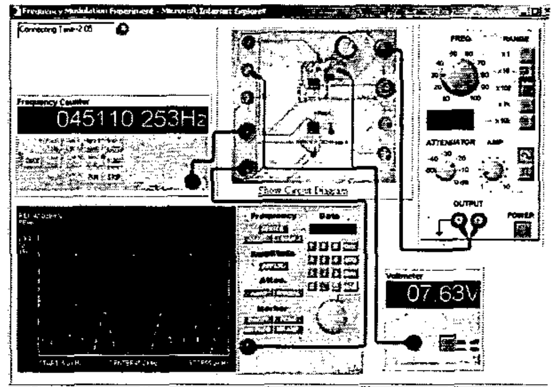

Figure 2. User Interface for Main User

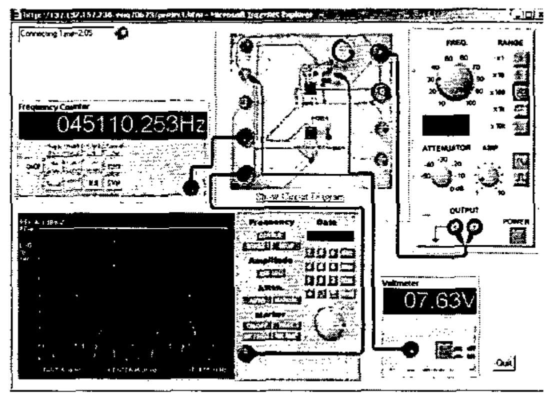

Figure 3. User Interface for Observer

The front panels of the instruments, circuit and connectors are all implemented in a graphical manner. Various leads and cables dragged by the main user to connect relevant points of the circuit to the instruments are shown to the observer. The observers also can see the main user turning the control buttons and knobs with the mouse. By doing so, a more realistic feel of operating the instrument is provided.

The feedback or measured results from the instruments are brought to the observer through the same interface. As shown in Figure 3, a curve representing the frequency spectrum is shown according to the data collected from the actual spectrum analyzer on a real-time basis. It is real-time in the sense that the displays of the conductor and observer are identical to the actual spectrum analyzer at anytime unless data is lost or corrupted. Similarly, the actual values measured by the voltmeter and the frequency counter are displayed on the appropriate panels. A clock in the observer's user interface keeps the observer informed of the remaining time before the system logs the main user out of the system. All these results are collected at the appropriate IP multicast addresses. Observer is free to log off the WVLAB at anytime by pressing the "quit" button at the bottom right corner in the same interface. After logging off, the window containing the instruments will be shut down.

An on-line instruction (lab manual) on experimental procedure is also provided. It serves to guide students to grasp the key concepts and attain knowledge in a "learn by seeing and doing" manner. Any interested user can freely browse the help and documentation pages, which are not protected and can be accessed without logging on, at the same time without interrupting the experiment.

2.1 A Typical Session

To view the remote experiment using the interface described, the user needs to have an Internet Explorer web browser installed on the client terminal. A typical session can then be initiated by setting the Uniform Resource Location (URL) to http://vlab.ee.nus.edu.sg/vlab/. From this web page on NUS virtual laboratories, the user can click the "Laboratory" button and choose "Frequency Modulation Experiment" from a list of available experiments.

Before watching the actual experiment, the user can view some relevant web pages of the experiment by clicking the links provided in the navigation menu on the left. The aim of these web pages is to give some preliminary information to the user on WVLAB, the experiment and the various instruments involved. Figure 4 shows an instruction page containing background information and procedure of the experiment.

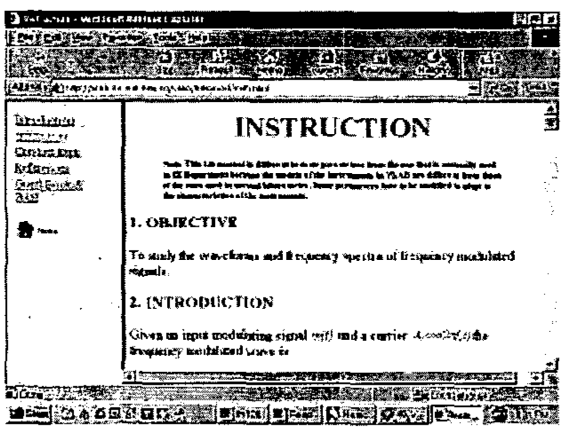

Figure 4. An Instruction Page

After getting familiar with the experiment, the user can click "Conduct Expt" to request to conduct the actual experiment. If there is already someone conducting the experiment in WVLAB, clicking on this link will bring the observer to the webcast user login page. Username and password for authentication will be requested from the observer. If the observer does not have an account, he or she can still log on using the guest account with a username of "guest" and a password of "welcome". For security, a guest or user can only access the system through the CGI programs developed for the experiment, and the system will not accept any other commands. Figure 5 and Figure 6 show the login interface for main user and observer separately.

After authentication, an internal session ID is granted, and the observer can now view the actual experiment in control by the main user and obtain results on a real-time basis. However, the observer can terminate the session at anytime. Once the session is terminated, all connections are released.

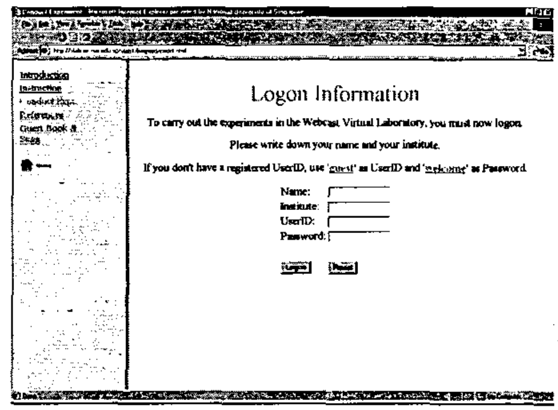

Figure 5. Main User Login Interface

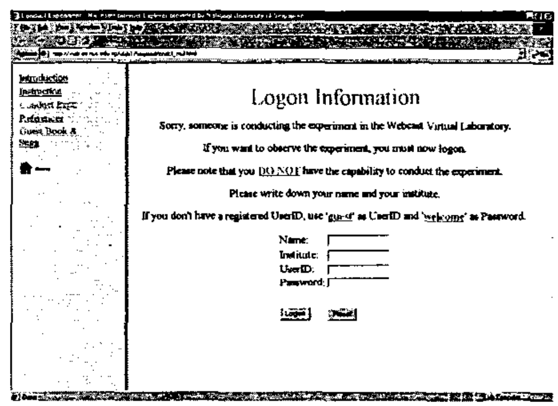

Figure 6. Observer Login Interface

3. System Architecture

3.1 Hardware System

Figure 7 gives a block diagram of the main hardware structure and components in the system, which can be divided into the following five subsystems.

(i) A PC with an Ethernet card works as the main controller, equipped with a GPIB card and a DAQ (Data Acquisition) card. It is connected to the Internet through the NUSNET-III network in the National University of Singapore (NUS). The main controller receives command strings from the WWW server through a TCP/IP channel for instrument control and analog input/output. Thus, besides being used as a controller PC, it represents the server side of the communication with the WWW server (which represents the client in this scenario).

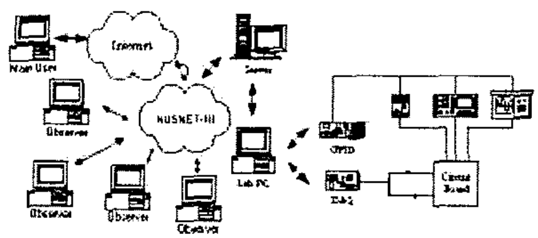

Figure 7. Hardware Structure

(ii) The programmable instruments, namely, the spectrum analyzer, frequency counter and signal generator, are connected to the controller PC through a GPIB card and cable. These GPIB-based instruments are ready to accept and execute SCPI (Standard Commands for Programmable Instruments) commands defined in IEEE488.2. Any data resulted from the execution of the command will also be sent back to the controller PC through the GPIB.

(iii) The DAQ card installed on the controller PC carries out analog input/output. Specifically, it sets or measures the voltage at a specific test point on the circuit board.

(iv) A WWW server hosts the web site of the virtual laboratory.

(v) NUSNET-III is a campus-wide network which inter-connects 104 departments in 90 buildings and covers a campus area of 150 hectares. It serves a population of 24000 students and 2700 staff. NUSNET-III network consists of multiple LANs connected together by local routers. A dedicated 3Com router is used to serve multicast for the entire SPnP segment, which use DVMRP instead of PIM-DENSE mode, as the rest of the campus routers do.

3.2 Software System

Figure 8 summarizes the software structure of the system. The control of local instruments and analog I/O is implemented under LabVIEW using the G programming language. For every instrument, there is one sub-module to process the commands from the user, while analog I/O is also handled under a sub-module within the main program. A WWW server with Red Hat Linux 7.0 and Apache HTTP Server host the web pages for the experiment. An mSQL database system is installed to manage user authentication. The GNU C program that transits command strings from the client side to the controller PC and passes sampled data in the reverse direction also runs on the Linux server. In addition, a Java server application that multicast the real-time spectrum analyzer display and other sampled data coming from the controller PC, as well as a simple queuing system server, are also residing on the Linux server. Java Applets embedded in HTML files are downloaded for running on the client machine to provide a user-friendly interface. Two Java applets are available for the main user and observers separately. For the implementation of multicast in applet, Java code needs to be signed in order to step out of the sandbox using the test certificate provided by Microsoft J++ [21-22]. An Internet Explore is sufficient to carry out the experiment.

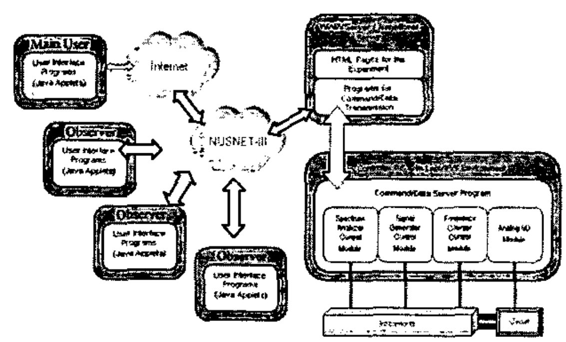

Figure 8. Software Structure

3.3 IP Multicast

At any one time only one user can be given control of the real instrument in VLAB as there is merely one set of apparatus. Without any increase in the hardware experimental apparatus needed, a new webcast based remote laboratory was brought to our mind to make VLAB accessible to as many users as possible. That is to allow several users to view how an existing remote experimental session is being carried out but without overloading network and server resources.

Hence, IP multicast is the solution. IP multicast is a bandwidth-conserving technology that reduces traffic by simultaneously delivering a single stream of information to thousands of recipients. It is designed to scale well as the number of participants expands, adding one more user doesn't amount to adding a corresponding amount of bandwidth.

However, IP multicast is built on top of the User Datagram Protocol (UDP). UDP provides only minimal services, such as port multiplexing and error detection. If a packet is detected in error under UDP, it is simply discarded. Packets may be received out of order, and some may be missing. Neither the sender nor the recipient will receive notification of any such network errors. This is one of the complexities of datagram-based networking.

It is up to the application to identify and overcome such problem. One of the solutions is to implement a protocol on top of it to ensure reliability. However, for our purposes, the occasional loss of data is quite acceptable. In order to keep things simple, the contents of our datagrams are simply a sequence of UTF 8-encoded characters. The decoding would not succeed and hence cause problems if packets are corrupted.

In order to communicate with multicast network more conveniently, two stream classes are defined [23]. DatagramOutputStream is used to translate a stream of output data into datagrams and then place it on the network for delivery to the destination. DatagramInputStream translates received datagrams from the network into a stream of input data. Besides the reason of ease to use, this approach also provides some degree of reliability onto the datagram.

As for TCP/IP sockets programming, Java provides substantial support for multicast as well. It simplifies the program development for us and hence reduces the amount of time needed to complete the program.

To initialize the network, we create a MulticastSocket listening on the chosen port, set the time-to-live to 63 to restrict the packets travelling around the same region (the default is 1), join the group that we are interested in receiving packets and then create our datagram streams (DatagramInputStream & DatagramOutputStream). To make textual communications easier, we attach an InputStreamReader and an OutputStreamWriter using UTF 8 encoding on top of these byte streams. We then wrap a BufferedReader around the InputStreamReader to provide for efficient reading of characters. By doing so, the input stream and output stream associated with the multicast socket are retrieved and we can start sending and receiving using multicast.

3.4 Simple Queuing Server

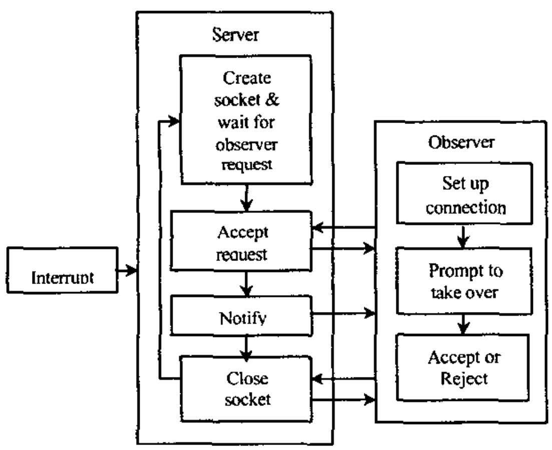

Figure 9. Observer-Server Interaction Using TCP

Since the webcast virtual laboratory is mainly multi-user based, it will be weird that nobody takes over the experimental session after the main user logs off. It doesn't make sense to have all observers waiting indefinitely for the next user logging on to carry out the experiment. Therefore, it is necessary to implement a simple queuing server for this purpose.

After the observers download the applet, it will create a TCP connection to the queuing server automatically at their start up. When the main user quits the experimental session, the first observer in the queue list will have the privilege to perform the next session. If he agrees so, he will be directed to conduct the next session straightforwardly. But if he rejects, the observer next to him will gain the privilege. However, he still remains in our queue list but now at the bottom.

As shown in Figure 9, when the main user quits, it will interrupt the server and in turn notify and prompt the first observer to take over. The server program will close the connection with the first observer after he answers. The server subsequently accepts the next observer connection and asks for taking over if the previous observer rejects.

3.5 Real-Time Transfer of Spectrum Analyzer Display

Instead of using live video streaming with multicast which involves some intolerable delays, the display of the spectrum analyzer is sampled and multicast to the observer program and in turn reconstructed in the graphical interface. The spectrum and other parameters to be displayed are sampled by GPIB from the spectrum analyzer. To reduce data traffic, a new spectrum is sent only when the server program detects that there is a significant change in the spectrum. Once the experiment is started, the controller keeps sampling the instrument and pumping the sampled data with the web server working like a pipe, passing exactly what it receives.

4. Impacts of Webcast Virtual Laboratories

The experiment described in this paper is currently used to support the teaching of an undergraduate course on Communication Principles in Department of Electrical and Computer Engineering, National University of Singapore. Since the number of students taking the course is of the order of a few hundreds, the available laboratory time slot is limited, and an expensive spectrum analyzer is involved, it is impossible for every student to have real hands-on experience of using the instrument to measure the spectra of frequency modulated signals.

With the launch of the webcast VLAB, the above problems are resolved to a certain extent. One student is conducting the experiment at anywhere while other students can observe his way to carry out the experimental work and hence gain hands-on experience of using an expensive spectrum analyzer. This can be done even if the physical laboratory is not open for student access. Webcast VLAB is designed to scale well as the number of participants expands, adding one more observers doesn't amount to adding a corresponding amount of bandwidth.

5. Conclusions

A general methodology to create a webcast based remote laboratory has been presented. Based on this methodology, a frequency modulation experiment for students taking a course on communication principles has been developed in National University of Singapore. The laboratory requires only an Internet Explorer to access, and incorporates schemes for reducing data traffic and authenticating users. Without any increase in the equipment needed, this webcast based approach for remote experimentation allows huge number of users to view how an existing remote experimental session is being conducted. Multicast is used to implement the webcasting capability for the remote experimentation purposes with the benefit that the network and server resources are greatly reduced. This design methodology can be extended to other experiments as well. The webcast virtual laboratory can be accessed at the web site http://vlab.ee.nus.edu.sg/vlab1/freqmod/index.html.

References

[1]. S.E. Poindexter and B.S. Heck, "Using the Web in Your Courses: What Can You Do? What Should You Do?" *IEEE Control Systems Magazine*, Vol. 19, No. 1, pp. 83-92, 1999.

[2]. A. Simione and J. Tuttle, "Designing and Implementing College-Wide Web-based Course Materials: A Case Study", *North American Web Conference '97*, New Brunswick, Canada, 1997. http://www.uvm.edu/~hag/naweb97/papers/tuttle.html

[3]. R. Pascoe, "Introducing WWW Technology into Tertiary Teaching: A Personal Perspective", *North American Web Conference'97*, New Brunswick, Canada, 1997. http://www.uvm.edu/~hag/naweb97/papers/pascoe.html.

[4]. J. Rosenblum and E. Healy, "Developing an Educational Intranet", *North American Web Conference'96*, Brunswick, Canada, Oct. 1996. http://www.uvm.edu/~hag/naweb96/zrosenblum.html

[5]. D. Tilbury, J. Luntz and W. Messner, "Controls Education on The WWW: Tutorials for MATLAB and SIMULINK - Web Technology for Controls Education", *Proc. of the 1998 American Control Conference*, Philadelphia, pp. 1304-1309, 1998.

[6]. S.G. Crutchfield and W.J. Rugh, "Interactive Exercises and Demonstrations on the Web for Basic Signals, Systems, and Controls," *Proc. 36th Conference on Decision & Control*, San Diego, California, pp. 3811-3815, 1997.

[7]. V. L. Stonick, "Teaching Signals And Systems Using The Virtual Laboratory Environment in ECE At CMU", *Proc. of the 1993 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 36-39, April 1993.

[8]. J. Pasquatette, *Activating the Internet For Virtual Instrumentation*, http://www.evaluationengineering.com/pctest/articles/c702pcni.htm.

[9]. P. Antsaklis, T. Basar, R.D. Carlo, N.H. McClamroch, M. Spong and S. Yurkovich, "Report on the NSF/CSS Workshop on New Directions in Control Engineering Education", *IEEE Control Systems*, Vol. 19, No. 5, pp. 53-58, 1999.

[10]. B. Foss, T. Eikass and M. Hovd, "Merging Physical Experiments back into the Learning Arena", *Proc. 2000 American Control Conference*, Chicago, Illinois, pp. 2944-2948, 2000.

[11]. A.C. Catlin, M.G. Gaitatzes, E.N. Houstis, Z. Ma, S. Markus, J. R. Rice, N.-H. Wang and S. Weerawarana, "The SoftLab Experience: Building Virtual Laboratories for Computational Science". http://www.cs.purdue.edu/research/cse/softlab/softlab-vlabs/softlab-framework/softlab_report/report.html

[12]. B. Aktan, C. Bohus, L. Crowl and M. Shor, "Distance Learning Applied to Control Engineering Laboratories", *IEEE Transactions on Education*, Vol. 39, No. 3, 1996.

[13]. S. Werges and D. Naylor, *A Networked Instructional Instrumentation Facility*. http://www.mal.uic.edu/marble/brief.htm.

[14]. D. Gillet, C. Salzmann, H. Latchman and O. Crisalle, "Recent Advances in Remote Experimentation", *Proc. of the 2000 American Control Conference*, Chicago, Illinois, pp. 2951-2954, 2000.

[15]. National Instruments Corporation, "Academic Solutions: Remote Experiment Control with LabVIEW". http://www.ni.com/academic/remote.htm.

[16]. Environmental Research Institute on Michigan (ERIM), *The Remote Scientist*. http://www.erim-int.com/CMAT/remote_scientist1.html

[17]. T. Junge and C. Schmid, "Web-Based Remote Experimentation Using a Laboratory-Scale Optical Tracker, *Proceedings of the 2000 American Control Conference*, Chicago, Illinois, pp. 2951-2954, 2000.

[18]. C. C. Ko, B.M. Chen, S.H. Chen, V. Ramakrishnan, R. Chen, S.Y. Hu and Y. Zhuang, "A large scale web-based virtual oscilloscope laboratory experiment", *IEE Engineering Science and Education Journal*, Vol. 9, No. 2, pp. 69-76, 2000.

[19]. C. C. Ko, B. M. Chen, J. Chen, Y. Zhuang and K. C. Tan, "Development of a Web-based laboratory for control experiments on a coupled tank apparatus", *IEEE Transactions on Education*, in press.

[20]. C. Krause, "From a Distance: Remote Operation of Research Equipment". http://www.ornl.gov/ORNLReview/v30n3-4/remote.htm.

[21]. "*HOW TO: Deploying java in Internet Explorer 4.0 and Netscape 4.0*". http://support.microsoft.com/support/kb/articles/q179/6/52.asp

[22]. Derrick Oswald, "*How to code your way out of the sandbox*". http://www.capitalnet.com/~oswald/Derrick

[23]. Merlin Hughes, "Multicast the chatwaves".

http://www.javaworld.com/javaworld/jw-10-1999/jw-10-step_p.html