

A Web-Based Virtual Laboratory on a Frequency Modulation Experiment

Chi Chung Ko, *Senior Member, IEEE*, Ben M. Chen, Shaoyan Hu, Vikram Ramakrishnan, Chang Dong Cheng, Yuan Zhuang, and Jianping Chen

Abstract—With the rapid proliferation of Internet technologies, accessing and operating engineering instruments remotely anytime anywhere is fast becoming a reality. This paper presents a new web-based virtual laboratory on a frequency modulation experiment for the teaching of an undergraduate course on communication principles in the National University of Singapore (NUS). The laboratory requires only a common web browser to access and incorporate schemes for reducing data traffic and authenticating users. It enables students to have a natural hands-on experience of using an expensive spectrum analyzer on a one-to-one basis and provides a solution for distant engineering education. The system uses a double client–server structure where access to the experiment is via two rounds of client–server processing. The virtual laboratory can be accessed at the web site <http://vlab.ee.nus.edu.sg/vlab/fre-qmod/index.html>.

Index Terms—Internet, online experiment, virtual laboratory, web-based laboratory.

I. 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 nontraditional 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 the following.

- 1) Developing a course web site to centrally house various online functions and facilitate course management.
- 2) Creating a remote laboratory to replace physical experiments with multimedia animation or simulation.
- 3) 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]–[6]. In [2], Simone and Tuttle describe a user-centered web

page construction and maintenance model for the development of the web-based course materials at the University of Minnesota College of Education and Human Development. In [3], Pascoe discusses several methods for students to interact with the course material to enhance their learning on the web. These include interactive exercises, facilities for annotating course notes, and automatic feedback tailored to specific student needs. In [4], Rosenblum and Healy describe a web-based communication tool for providing instructors with private course discussion areas. With an intuitive chat interface, these areas allow instructors to give students a platform-independent ability to communicate in as many groups as is needed by the class and open the door for a flexible online collaborative learning environment. In [5], Tilbury *et al.* describe an effective paradigm for using the web as an education aid for tutoring students on the use of Matlab and Simulink. Students are expected to run Matlab or Simulink in one window of their own computers and a web browser in another. While surfing the tutorial, the student can download examples from the web page and run them locally. In a similar development [6], Crutchfield and Rugh describe a “laboratory” where, using Java applets, students can run the simulation object directly through the web browser without downloading the source code. Other forms of virtual laboratories and/or virtual instruments can be found in [7] and [8]. In [7], an early version of a virtual laboratory is discussed under a Windows 3.1 environment, while in [8], a technique of creating virtual instruments using ActiveX is introduced.

The above examples give a good demonstration of how the web, and in particular, systems that are based only on theoretical/simulation materials and have capabilities 1) and 2), 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 applied 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 3) and that enable students to set up parameters to run experiments from a remote location. At Purdue University, a virtual laboratory called SoftLab has been set up to provide an environment for both physical experiments and numerical simulation [11]. Users are able to remotely control some real

Manuscript received April 1, 2000; revised July 2, 2001. This paper was recommended by Associate Editor R. Rada.

C. C. Ko and B. M. Chen are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117576 (e-mail: elekoc@nus.edu.sg).

S. Hu, V. Ramakrishnan, C. D. Cheng, Y. Zhuang, and J. Chen were with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117576.

Publisher Item Identifier S 1094-6977(01)08909-X.

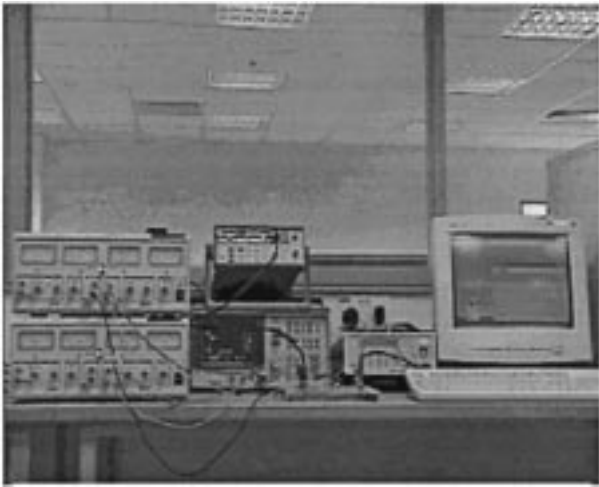


Fig. 1. Hardware setup.

instruments, after installing SoftMedia, an exclusive software, for accessing the service in the laboratory. In [12], Aktan *et al.* describe a virtual laboratory that enables users to control a 3 degree of freedom (DOF) robot arm remotely. A client-server structure is used, but a software X-terminal has to be properly set up. Also, communication between the client and server, realized using user datagram protocol (UDP), may be unreliable. In [13], Werges and Naylor describe a virtual laboratory called The Networked Instructional Instrumentation Facility, which is a prototype web system allowing multi-user access for carrying out measurements within a library of test equipment and devices. Again, the user must install some software built with Java using Microsoft J++ on the client side. Other similar systems can be found in [14]–[16].

Since systems with capability 3) 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 (see [17]–[20]).

In this paper, we present a web-based laboratory (VLAB) on a frequency modulation experiment. With a graphical user interface designed to be as natural to using real instruments as possible, remote users are able to gain access to actual physical apparatus, set experimental parameters, and conduct and monitor the experiment in the laboratory through browsing the relevant web pages. Microsoft Internet Explorer, a common web browser, is all that is needed to access all the service in the virtual laboratory (VLAB). Compared with other virtual laboratories, VLAB incorporates schemes for reducing data traffic and authenticating users, and enables students to have a natural hands-on experience of using an expensive spectrum analyzer on a one-to-one basis. To accomplish these design objectives, VLAB uses a double client-server structure (DCSS), where access to the experiment is via two rounds of client-server processing. The instruments in VLAB are controlled using IEEE 488.2, and technologies based on transmission control protocol

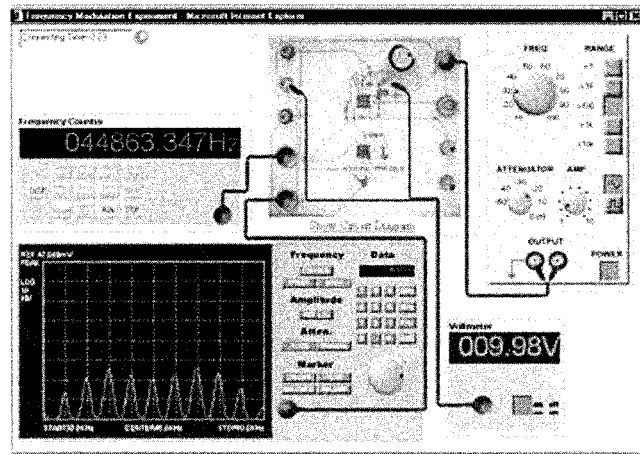


Fig. 2. Main user interface.

(TCP), hypertext transfer protocol (HTTP), common gateway interface (CGI), Java, JavaScript, LabVIEW, and G [21]–[25] are used. Fig. 1 shows the actual physical laboratory setup for VLAB.

The rest of the paper is organized as follows. In Section II, the typical user interface is described to show how VLAB helps students gain the same experience as with conducting experiments in the real laboratory. Section III describes the hardware and software architecture and Section IV discusses the impacts of the virtual experiment. Finally, conclusions are drawn in Section V.

II. USER INTERFACE

In order to give users an impression that they are actually operating on physically existing instruments, the user interface of a remote laboratory should be as realistic as possible. Also, unless the interface is user friendly, it will be difficult to conduct the experiment remotely. In VLAB, the GUI is implemented using Java, JavaScript, and Hypertext Markup Language (HTML) technologies, which are supported by popular web browsers. Fig. 2 shows the typical screen display while conducting the frequency modulation experiment online.

A. Brief Overview

Java Applets embedded in HTML pages are used to construct the main interface of the experiment. Java is the natural programming language of choice on the client side because of its flexibility in GUI design, convenient network programming, and platform independence. The last property is the most significant since it allows the same applet program to run on client machines under different platforms.

The object 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. Figs. 3 and 4 give separated views of these instruments and circuit board, which are set up as depicted in Fig. 2. Various leads and cables can be dragged to connect relevant points of the circuit to the instruments so as to

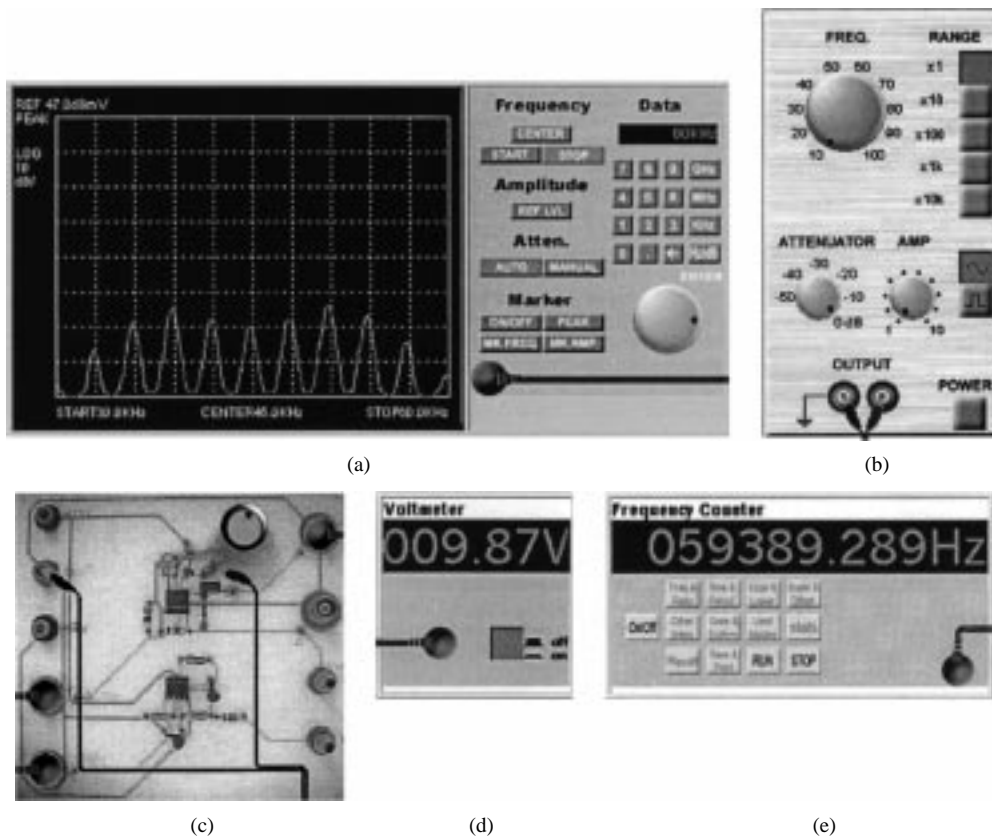


Fig. 3. Instruments and circuit board. (a) Spectrum analyzer. (b) Signal generator. (c) Circuit board with variable resistor. (d) Voltmeter. (e) Frequency counter.

create a realistic impression that connections have to be made before making measurements in an actual experiment.

The front panels of the instruments, circuit, and connectors are all implemented in a graphical manner. Specifically, the mouse can be used to turn the control buttons and knobs through dragging so that a more realistic feel of operating the instrument is provided. As an example, the output amplitude of the signal generator can be changed by using the mouse to drag and turn the AMP knob on the panel (Fig. 2).

The feedback or measured results from the instruments are brought to the user through the same interface. As shown in Fig. 3(a), a curve representing the frequency spectrum is shown according to the real-time data collected from the actual spectrum analyzer. Similarly, in Fig. 3(d) and (e), the actual values measured by the voltmeter and the frequency counter are displayed on the appropriate panels.

The user interface has been designed to work logically. For example, the frequency counter will show an appropriate reading only after the user has connected the frequency counter to a certain point in the circuit by dragging the two ends of the coaxial cable to connect the INPUT on the frequency counter panel [Fig. 3(e)] to the SQUARE WAVE OUTPUT on the circuit board [Fig. 3(c)].

An online 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. Implemented through a logon process, an authentication scheme is used so that only authorized users can access the important web pages that control the real instruments. Each user is allocated 30 min after a successful logon. A clock in the main

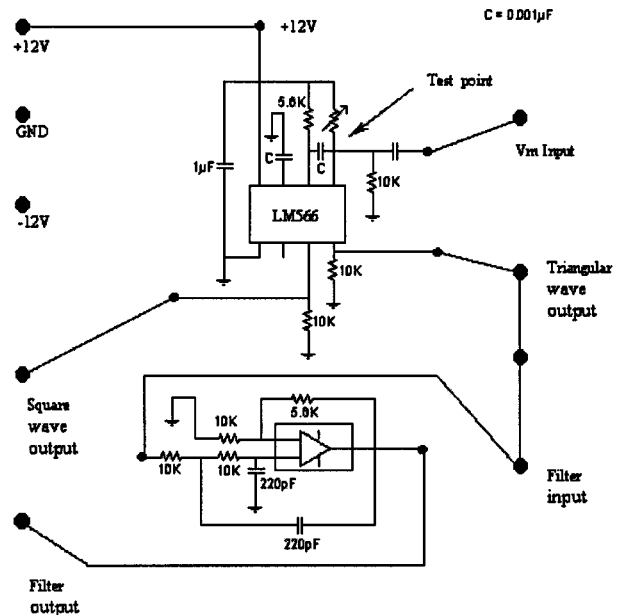


Fig. 4. Circuit diagram of circuit board.

user interface keeps the user informed of the remaining time before the system logs the user out of the system. As the user conducts the experiment, he or she can also access the help and documentation pages without killing the experiment. In addition, 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

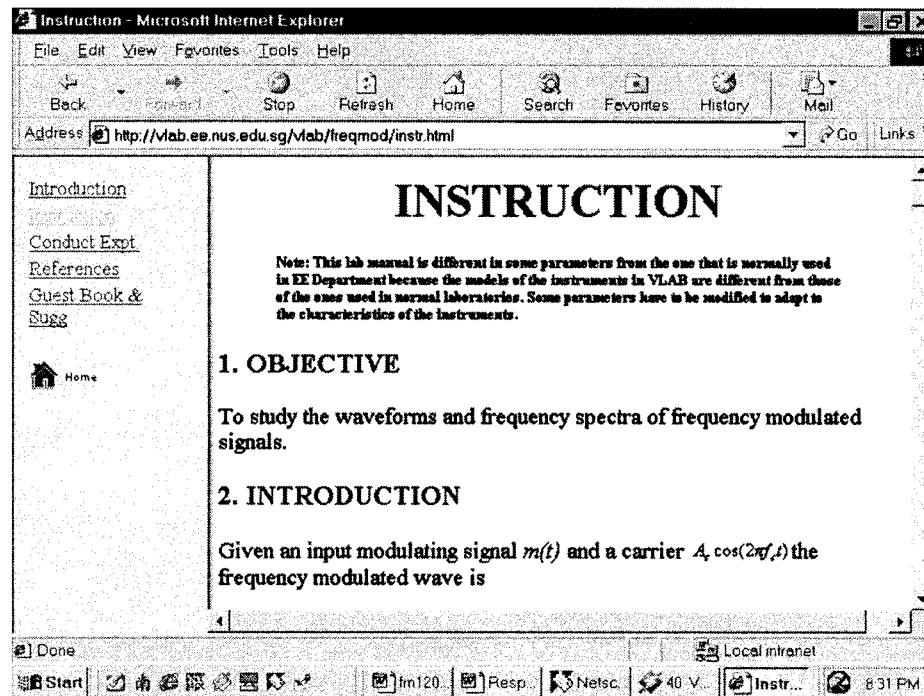


Fig. 5. Instruction page.

experiment. After logging off, the window containing the instruments will be shut down and the user will be prompted that the current session is closed.

B. Typical Session

To conduct the remote experiment using the interface described, the user needs to have a web browser supporting Java installed on the client terminal. A typical experimental 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 conducting 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 VLAB, the experiment, and the various instruments involved. Fig. 5 shows an instruction page containing background information and the procedure of the experiment.

After getting familiar with the experiment, the user can click “Conduct Expt” to conduct the actual experiment. Username and password for authentication will be requested from the user. If the user 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.

After authentication, an internal session ID is granted, and the main user interface for controlling the real instruments will appear. The user can now conduct the actual experiment and obtain results through real-time feedback. For the experiment on frequency modulation, some possible operations by a typical user include the following.

- 1) Vary the variable resistor to set the voltage at pin five (the test point in Fig. 4) of the voltage controlled oscillator (VCO).
- 2) Measure the frequency and voltage accurately with the frequency counter and voltmeter to find the frequency sensitivity of the modulator.
- 3) Direct the signal generator to produce an appropriate output and adjust its amplitude and frequency to make it satisfy certain conditions on modulation frequency and modulation index.
- 4) Use the spectrum analyzer to measure the spectrum and other characteristics of the frequency modulated signal.

The session can be terminated by the user anytime or after a default time limit of 30 min. The time remaining is updated once every one minute and shown on the main user interface. Once the user logs off or the session is terminated, all connections are released.

III. SYSTEM ARCHITECTURE

A. Hardware System

Fig. 6 gives a block diagram of the main hardware structure and components in the system which can be divided into the following four subsystems.

- 1) A PC with an Ethernet card works as the main controller equipped with a GPIB card and a data acquisition (DAQ) card. It is connected to the Internet through the NUSNET-III network at NUS. (NUSNET-III is a campus-wide network which interconnects 104 departments in 90 buildings and covers a campus area of 150 hectares. It serves a population of 24 000 students and 2700 staff.) The main controller receives command strings from the WWW server through a TCP/IP channel for instrument control and analog input/output (I/O).

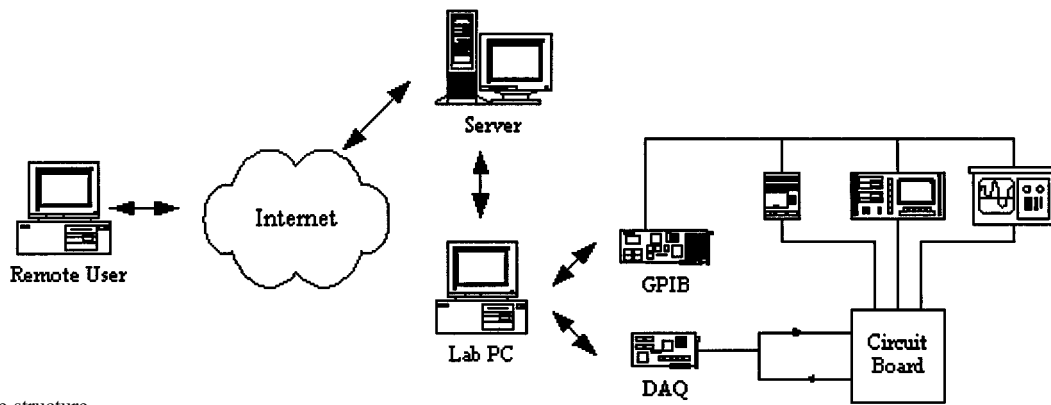


Fig. 6. Hardware structure.

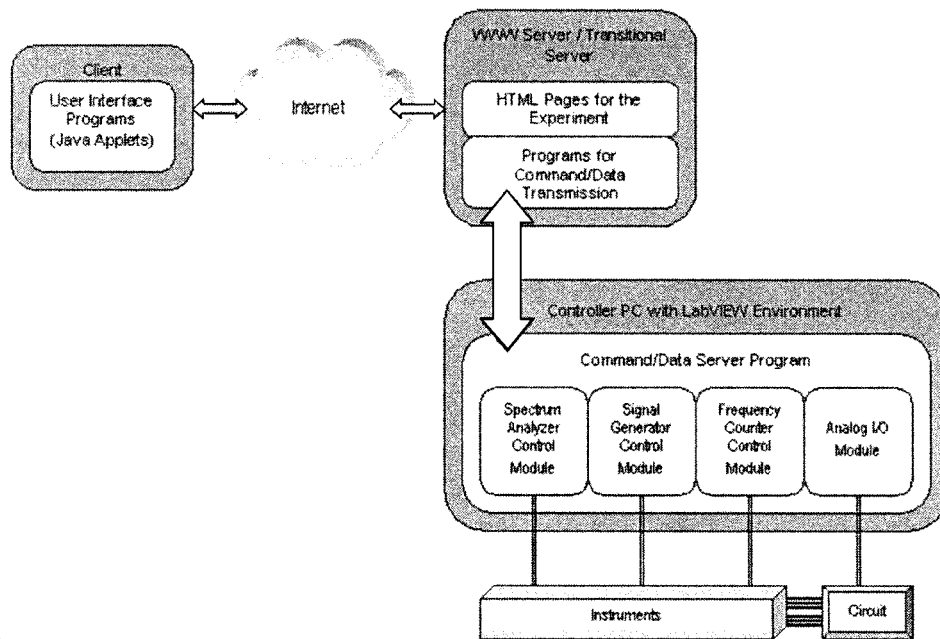


Fig. 7. Software structure.

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).

- 2) 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 standard commands for programmable instruments (SCPI) commands defined in IEEE488.2. Any data resulting from the execution of the command will also be sent back to the controller PC through the GPIB.
- 3) The DAQ card installed on the controller PC carries out analog I/O. Specifically, it sets or measures the voltage at a specific test point (see Fig. 4) on the circuit board.
- 4) A WWW server hosts the web site of the virtual laboratory.

B. Software System

Fig. 7 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 submodule to process the commands

from the user while analog I/O is also handled under a submodule within the main program. A WWW server with Red Hat Linux 5.2 and Apache HTTP Server hosts the web pages for the experiment. An mSQL database system is installed to manage user authentication. The GNU C program that transmits command strings from the client side to the controller PC and passes sampled data in the reverse direction also runs on the Linux server. Java Applets embedded in HTML files are downloaded for running on the client machine to provide a user-friendly interface. Regardless of the operating system of the client terminal, a web browser supporting Java is sufficient to carry out the experiment.

C. Local Controls and Analog I/O

As described, LabVIEW is adopted to implement local instrument control. It supports RS-232, GPIB, Virtual Instrument Software Architecture (VISA) functions, and plug-in DAQ boards. Commands for modifying settings or initiating specific actions are sent to programmable instruments through the GPIB interface. On the basis of the GPIB port number assigned to the instrument, the commands are transferred through the GPIB card and cable to the corresponding instrument, which

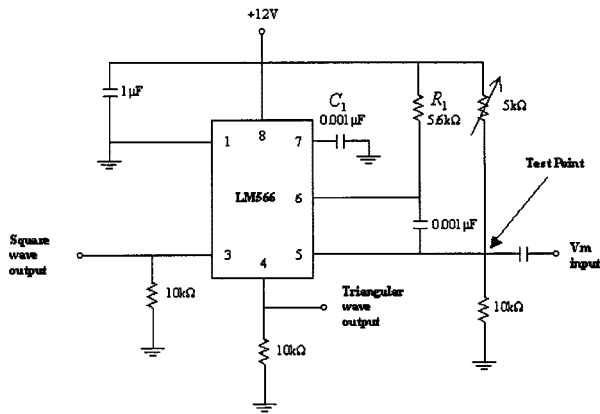


Fig. 8. Voltage controlled oscillator circuit.

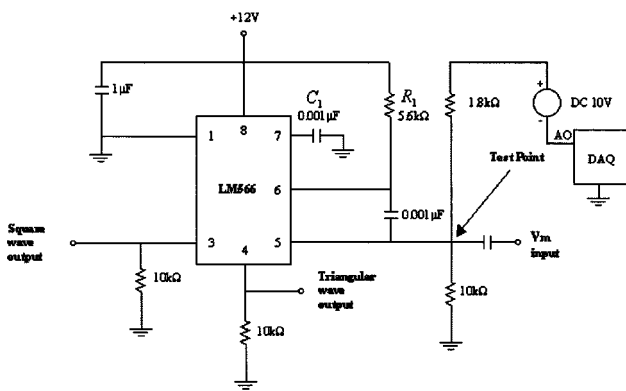


Fig. 9. Modified VCO circuit.

interprets the commands and takes appropriate actions. The system checks all parameters entered and will only accept and send those commands that are within a set that will not cause malfunction or risk of any form. The set of instrument is located in a standard laboratory environment, and the overall risk of unattended operation may even be lower than when inexperienced students are physically conducting the experiments and, in the process, subjecting themselves to the possibility of touching line voltages.

One reason for using the DAQ board is that the voltage at the test point on the circuit board has to be measured accurately so that the frequency sensitivity of the modulator can be derived. Since this voltage is measured by a voltmeter when conducting the actual experiment, the voltage measured by the board is displayed within the panel of a voltmeter in the virtual laboratory.

Another reason is that there is actually a potentiometer in the VCO circuit of Fig. 8 on the circuit board and the user is supposed to change the voltage over the range of 8.5 V to 11.5 V at the test point by adjusting the resistance of the potentiometer. As it is not possible to physically turn the potentiometer when conducting the experiment remotely, the VCO circuit is modified as shown in Figs. 9 and 10 so that a 10 V DC source is combined with the DAQ board analog output and applied to the test point of the circuit. Thus, as the remote user turns the potentiometer in the virtual laboratory, the system will change the output voltage level at the test point appropriately.

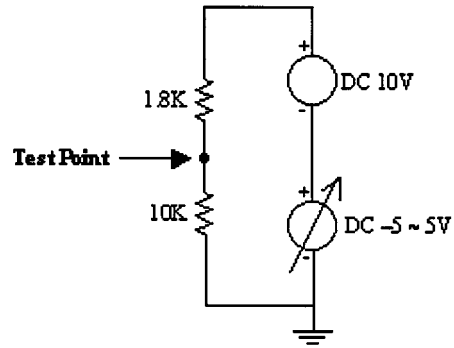


Fig. 10. Simplified circuit for tuning the voltage at the test point.

D. Double Client–Server Structure

VLAB makes use of the DCSS as shown in Fig. 11. There are two pairs of client–server in the whole system. The first one is the client–web server pair and the second one is the web server–controller pair. The two pairs are connected by a C program for the convenience of creating sockets and high efficiency. It is felt that this double client–server architecture is better than the single round client–server adopted in [12] as the system becomes more flexible and extensible. Specifically, it will be easier to add new experiments by just changing the controller programs.

The structure is implemented using TCP sockets. Since a TCP connection established by a client with a server remains in place till the client or the server closes the connection, it is the ideal communication protocol for implementing web-based control of instruments that require frequent parameter adjustments. Once a connection is established, there is no need to keep disconnecting and connecting over and over again for every change in parameters. This is in contrast to systems that use the UDP as the communication protocol [14]. Specifically, since the latter is a connectionless protocol and there is no guarantee that UDP datagrams will reach their destination, it may not be appropriate for high quality communication.

Fig. 12 depicts the interaction between the client and server under TCP, where individual C programs running on the web server transmit both the command strings from the client to controller and the data from the controller to the client. Since the programs run in Linux, GNU C is selected as the programming language for its convenience in creating sockets and high efficiency. On the controller, LabVIEW supporting TCP sockets in the form of subvirtual instruments (VIs), “TCP Listen.vi,” “TCP Read.vi,” “TCP Write.vi,” and “TCP Close Connection.vi” is used.

In more details, Fig. 13 illustrates the communication scheme implemented for the virtual laboratory. The Java Applet running on the client side sends a request for connection to the program, “transit.c,” running on the web server. The program in turn sends a connection request to the controller as well as accepts the request from the client. The “TCP Listen.vi” on the controller then accepts the request to result in a link from the client to the controller being established to transfer commands and data. When the session is terminated, the link will be released by closing all the sockets.

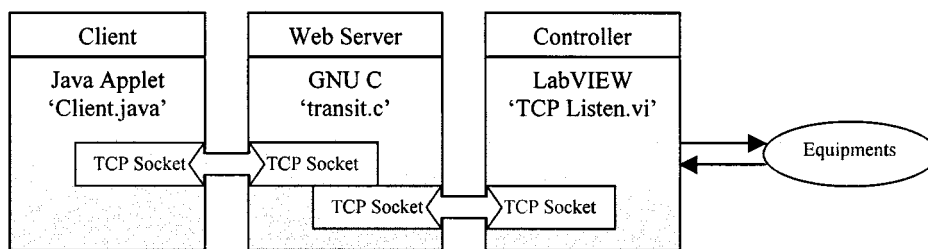


Fig. 11. Double client-server structure (DCSS).

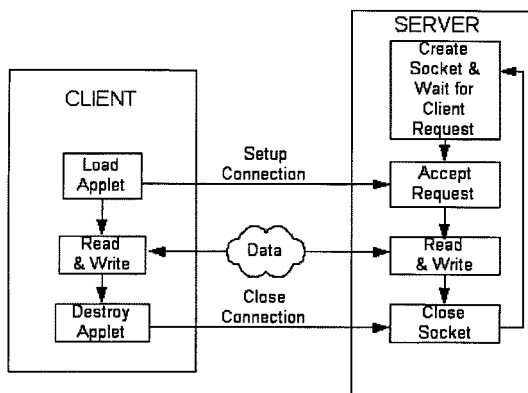


Fig. 12. Client-server interaction using TCP.

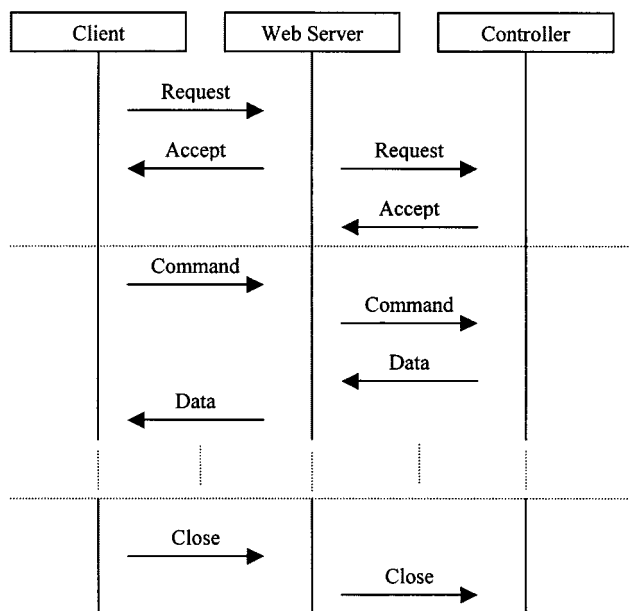


Fig. 13. Communication scheme for double client-server structure.

The above architecture is chosen for security reasons. Only an authorized request is supposed to be accepted by VLAB and, before the request reaches the controller machine, it has to be authorized by the system. The authentication is done at the web server using CGI programs written in PERL 5 to filter out malicious connection requests and commands to both the controller and the instruments. This also prevents the chaotic situation of having more than one authorized user controlling the real instruments at the same time. So long as communication is through the web server, conflicts will be detected by the CGI program for authentication. This method of remote instrument control can be

generalized to be adapted to any web-based control instrument systems.

E. Real-Time Transfer of Spectrum Analyzer Display

To enable users to see the frequency spectrum of the signals, the display of the spectrum analyzer must be transferred to the client side. The spectrum and other parameters to be displayed are sampled by GPIB from the spectrum analyzer and transferred to the client program. The client program, in turn, reconstructs the display in the user interface. To reduce data traffic, a new spectrum is sent only when the server program detects that there is a significant change in the spectrum. As illustrated in Fig. 13, the transfer of data is also done through a double-client scheme. Once the client sends a request and the link is set up, the controller keeps sampling the instrument and pumping the sampled data with the web server working like a pipe, passing exactly what it receives.

F. User Authentication

Unlike those that are based on simulation, conducting the experiment in VLAB requires using a set of real physical instruments. Thus, as with carrying out an actual experiment where only one person can assume control, the apparatus and virtual experiment should be accessible in an experimental session by only one Internet client, even though it is possible that a few students may be discussing what to do on the client side. At present, if a client is conducting the experiment and is in control of the set of instruments, access to those parts of the system that control the instrument by another client will be blocked through an authentication process. Note, however, that access to the help and documentation pages are not controlled and users can freely access these pages any time.

The program for user authentication is coded in Perl and runs on the web server. The structure of the authentication system is depicted in Fig. 14. The parameter interpretation module extracts the parameters from the CGI request and determines which submodule should be called. If it is a logon request, the logon module will search the user database for the user ID and password given by the user on the client side. Once the user ID and password are proved legal, a session ID is generated and written into a data file and referred to as a session database, to indicate that a user has logged on. At the same time, an access grant is returned. Conversely, if the user ID or password is not correct, the logon module returns an access denial.

When a page request is submitted, the page request module will be called. It first checks the access permission of the current user by querying the session database. If the user has logged on

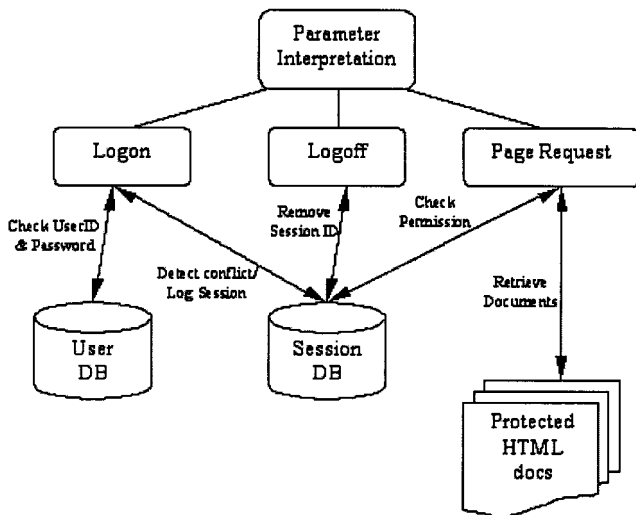


Fig. 14. User authentication.

and the session has not expired yet, the required HTML document, which is protected from being accessed by other means, will be sent to the client's browser.

The logoff module is called when the user requests for logoff. It removes the current active session ID from the session database to allow other users to access the protected web pages.

IV. IMPACTS OF VIRTUAL LABORATORIES

The experiment described in this paper is currently used to support the teaching of an undergraduate course on Communication Principles in the Department of Electrical and Computer Engineering, National University of Singapore (NUS). 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. In fact, all experimental sessions involve conducting the experiment in groups of a few students, and this sharing of equipment degrades the learning experience significantly.

With the launch of the VLAB, the above problems are resolved to a certain extent. Students are now able to carry out experimental work and gain hands-on experience of using an expensive spectrum analyzer according to their own schedule and pace on an individual basis. In addition, this can be done even if the physical laboratory is not open for student access or when students are off campus.

Lastly, access of VLAB by other part-time students or users at other locations is also possible, and the virtual laboratory is fast becoming a cornerstone to complement the experience of engineering experimentation for part-time students and those on distant learning programs.

V. CONCLUSION

A general methodology to create a web-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 the National University of Singapore (NUS). The laboratory requires only a common web browser to access, and incorporates schemes for reducing data traffic and authenticating users. It enables students to have a natural hands-on experience of using an expensive spectrum analyzer on a one-to-one basis, and provides a solution for distant engineering education. The system uses a double client-server structure (DCSS), where access to the experiment is via two rounds of client-server processing. This design methodology can be extended to other experiments. The virtual laboratory can be accessed at the web site <http://vlab.ee.nus.edu.sg/vlab/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 Contr. Syst. Mag.*, vol. 19, pp. 83–92, Jan. 1999.
- [2] A. Simione and J. Tuttle, "Designing and implementing college-wide Web-based course materials: A case study," in *Proc. North American Web Conf.*, Fredericton, NB, Canada, 1997, [Online]. Available: <http://www.uvm.edu/~hag/naweb97/papers/tuttle.html>.
- [3] R. Pascoe, "Introducing WWW technology into tertiary teaching: A personal perspective," in *Proc. North American Web Conf.*, Fredericton, NB, Canada, 1997, [Online]. Available: <http://www.uvm.edu/~hag/naweb97/papers/pascoe.html>.
- [4] J. Rosenblum and E. Healy, "Developing an educational intranet," in *Proc. North American Web Conf.*, Fredericton, NB, Canada, Oct. 1996, [Online]. Available: <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," in *Proc. Amer. Contr. Conf.*, Philadelphia, PA, 1998, pp. 1304–1309.
- [6] S. G. Crutchfield and W. J. Rugh, "Interactive exercises and demonstrations on the Web for basic signals, systems, and controls," in *Proc. 36th Conf. Decision Contr.*, San Diego, CA, 1997, pp. 3811–3815.
- [7] V. L. Stonick, "Teaching signals and systems using the virtual laboratory environment in ECE at CMU," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 1993, pp. 36–39.
- [8] J. Pasquatette, "Activating the Internet for virtual instrumentation?" [Online]. Available: <http://www.evaluationengineering.com/pctest/articles/e702pcni.htm>.
- [9] P. Antsaklis *et al.*, "Report on the NSF/CSS workshop on new directions in control engineering education," *IEEE Contr. Syst.*, vol. 19, pp. 53–58, May 1999.
- [10] B. Foss, T. Eikass, and M. Hovd, "Merging physical experiments back into the learning arena," in *Proc. Amer. Contr. Conf.*, Chicago, IL, 2000, pp. 2944–2948.
- [11] A. C. Catlin *et al.*, "The SoftLab experience: Building virtual laboratories for computational science," [Online]. Available: http://www.cs.purdue.edu/research/cse/softlab/softlab-vlabs/softlab-framework/softlab_report/report.html.
- [12] B. Aktan *et al.*, "Distance learning applied to control engineering laboratories," *IEEE Trans. Educ.*, vol. 39, pp. 320–326, Aug. 1996.
- [13] S. Werges and D. Naylor, "A networked instructional instrumentation facility," [Online]. Available: <http://www.mal.uic.edu/marble/brief.htm>.
- [14] D. Gillet *et al.*, "Recent advances in remote experimentation," in *Proc. Amer. Contr. Conf.*, Chicago, IL, 2000, pp. 2951–2954.
- [15] National Instruments Corporation, "Academic solutions: Remote experiment control with LabVIEW," [Online]. Available: <http://www.ni.com/academic/remote.htm>.
- [16] Environmental Research Institute on Michigan (ERIM), "The remote scientist," [Online]. Available: 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," in *Proc. Amer. Contr. Conf.*, Chicago, IL, 2000, pp. 2951–2954.
- [18] C. C. Ko, "A large scale web-based virtual oscilloscope laboratory experiment," *Proc. IEE Eng. Sci. Educ. J.*, vol. 9, no. 2, pp. 69–76, 2000.
- [19] C. C. Ko *et al.*, "Development of a Web-based laboratory for control experiments on a coupled tank apparatus," *IEEE Trans. Educ.*, vol. 44, pp. 76–86, Feb. 2001.

- [20] C. Krause. From a distance: Remote operation of research equipment. [Online]. Available: <http://www.ornl.gov/ORNLReview/v30n3-4/remote.htm>.
- [21] J. Travis, *Internet Applications in LabVIEW*, (Apr. 2000). [Online]. Available: http://www.phptr.com/ptrbooks/ptr_0130141445.html.
- [22] L. Wells and J. Travis, *LabVIEW for Everyone: Graphical Programming Made Even Easier*. Englewood Cliffs, NJ: Prentice-Hall, 2000. [Online]. Available: <http://www.ni.com/reference/books/lv4every.htm>.
- [23] A. Rocke. Modeling virtual instruments using LabVIEW. [Online]. Available: <http://www.ece2.engr.ucf.edu/~aro/labview/LabVIEW.html>.
- [24] VI Control Systems. Computer control and data acquisition systems. [Online]. Available: <http://www.vicsys.com/cntrlDAQ/cntrlDAQ.htm>.
- [25] Comprehensive Solutions. LabVIEW software development. [Online]. Available: <http://www.comprehensivesolutions.com/labview.htm>.



Chi Chung Ko (SM'93) received the B.Sc. (1st class honors) and Ph.D. degrees in electrical engineering from Loughborough University of Technology, Leicestershire, U.K., in 1978 and 1981, respectively.

Currently, he is with the Department of Electrical and Computer Engineering, National University of Singapore (NUS). His current research interests include digital signal processing, adaptive arrays, communications and networks. He has written over 150 technical publications in these areas.

Dr. Ko has served as an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING from 1997 to 1999, and is currently serving as an Associate Editor of the IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION.



Ben M. Chen was born on November 25, 1963, in Fuqing, China. He received the B.S. degree in mathematics and computer science from Amoy University, Xiamen, China, the M.S. degree in electrical engineering from Gonzaga University, Spokane, WA, and the Ph.D. degree in electrical and computer engineering from Washington State University, Pullman, in 1983, 1988, and 1991, respectively.

From 1983 to 1986, he was a Software Engineer at the South China Computer Corporation. From 1992 to 1993, he was an Assistant Professor in the Electrical Engineering Department, State University of New York at Stony Brook.

Since 1993, he has been with the Department of Electrical and Computer Engineering, National University of Singapore (NUS), where he is currently an Associate Professor. His current research interests are in linear control and system theory, control applications, development of Internet-based virtual laboratories, and Internet security systems. He is an author or co-author of five monographs: *Hard Disk Drive Servo Systems* (New York: Springer, in press), *Robust and H_∞ Control* (New York: Springer, 2000), *H_∞ Control and Its Applications* (New York: Springer, 1998), *H_2 Optimal Control* (Englewood Cliffs, NJ: Prentice-Hall, 1995), *Loop Transfer Recovery: Analysis and Design* (New York: Springer, 1993), and one textbook, *Basic Circuit Analysis* (Englewood Cliffs, NJ: Prentice-Hall, 1998).

Dr. Chen was an Associate Editor on the Conference Editorial Board of the IEEE Control Systems Society from 1997 to 1998. He is currently an Associate Editor of IEEE TRANSACTIONS ON AUTOMATIC CONTROL.

Shaoyan Hu was born in Shanghai, China, in 1974. He received the B.Eng. degree in communication engineering from Shanghai Jiaotong University and the M.Eng. degree from the National University of Singapore (NUS), in 1996 and 2000, respectively.

His current research interests include remote instrument control and Internet application development.

Vikram Ramakrishnan was born in New Delhi, India, in 1974. He received the B.E. degree in electrical and electronics engineering from Anna University, Madras, Spain, and the M.Eng. degree in electrical and computer engineering from the National University of Singapore (NUS) in 1997 and 2000, respectively.

His current research interests include web-based remote instrument control for distance education and distributed application and server-side component development.

Chang Dong Cheng was born in Jiangxi, China, in 1968. He received the B.S.E.E. and M.S.E.E. degrees from Harbin Institute of Technology, Shanghai, China, and the Ph.D. degree from Beijing University of Aeronautics and Astronautics in 1990, 1995, and 1999, respectively.

From 1998 to 1999, he was a Senior Research Engineer at the China Academy of Telecommunications Technology. Since 2000, he has been a Research Fellow in the Department of Electrical and Computer Engineering at the National University of Singapore (NUS). His research interests include Internet applications, virtual reality, and distance education.

Yuan Zhuang was born in Anhui, China, in 1975. He received the B.Eng. degree in electrical engineering from Zhejiang University, Hangzhou, China, and the M.Eng. degree from the National University of Singapore (NUS), in 1997 and 2000, respectively.

His current research interests include remote instrument control, distributed system application, and web technology development.

Jianping Chen was born in Fujian, China. He received the B.Eng. degree in control theory from Shanghai Jiaotong University, Shanghai, China and the M.Eng. degree in control theory from Xiamen University, Fujian, China, in 1993 and 1999, respectively.

From 1993 to 1996, he conducted research on communication engineering at the South-Cube High-Tech Co. Ltd., Xiamen. His current research interests include network security and intelligent control.