Remote Communication Engineering Experiments Through Internet

Sanjib Das, L. N. Sharma and A. K. Gogoi
Department of Electronics & Communication Engineering,
Indian Institute of Technology Guwahati, Guwahati-781039, India.

Abstract—**In technical education, laboratory components comprise an essential and integral part without which engineering education remains incomplete. Experiments conducted on laboratory equipments lend a practical touch to the theoretical knowledge acquired by the students. However, setting up a specialized laboratory consisting of sophisticated and expensive equipments such as Digital Storage Oscilloscope, Signal Generator, Spectrum Analyzer and Network Analyzer is an expensive and unaffordable proposition for many universities and engineering colleges. Sophisticated technologies incorporated in recent models of such high-end equipments enable remote access through Internet to the instruments. This concept of accessing these expensive instruments over the Internet can be exploited by setting up a *Remote Laboratory*. This remote laboratory system aims at not only providing an opportunity to students from distant places to conduct hardware experiments but also to take the corresponding measurements. In this work, real-time hardware experiments have been designed and implemented. These are based on modulation techniques widely employed in Communication Engineering. An interactive Graphical User Interface (GUI) environment has also been developed using Microsoft Visual Basic. This GUI is provided at the user end to facilitate the remote control and access of various instruments and experiment setups. It has been specifically designed and optimized for a low-bandwidth remote access link. The above mentioned system, as a whole, uses real-time capture of images and data from the instruments to perform experiment-related measurements.**

Index Terms— **Communication Engineering, Modulation, Remote Laboratory.**

I. INTRODUCTION

The advancements in Internet Technology have enhanced remote learning process in various ways. Remote learning, a current trend in the academic community, has led us to think how the teaching material can reach the remote users in easy, efficient and cost effective ways. This form of pedagogy typically includes providing a digital library of required documents, class notes and pre-recorded classroom lectures in the form of multimedia gadgets. The major disadvantage of this form of teaching is the lack of an interactive session between the student and the instructor. Recent trends indicate that the universities are trying to provide a more meaningful learning experience by conducting interactive online teaching sessions to the students hailing from remote geographical locations.

Remote distance education may include the laboratory component of an academic curriculum. Hands-on training is essential, since it renders completeness to the learning process as a whole. Advancements in Instrumentation have made it possible to setup remote laboratory experiments by enabling remote access to various laboratory instruments over the Internet. With the ability to configure instruments and data acquisition systems remotely via software, it has become possible to share expensive instruments and equipment to conduct real-time experiments remotely. Thus remote laboratory experiments and remote laboratory instrument control through Internet is fast becoming a reality.

Setting up a remote laboratory reduces maintenance and laboratory supervision costs of a university. It also reduces space requirements since only one experimental setup is used to facilitate an entire class. Students could be familiarized about the experiment setups before the remote laboratory class (pre-lab session or laboratory tutorial class) during the lecture session. However, in reality, there are certain constraints such as bandwidth requirement, Internet time lag during data transfer, Internet traffic management and technical architectural needs, which need to be addressed.

There are various approaches found in literature that have been followed by different universities and researches for setting up remote laboratory in many application areas. The works in [1, 2] describe web-based virtual electronics laboratories, one on frequency modulation experiment and the other on coupled tank apparatus, a multi-input-multi-output (MIMO) system, developed at National University of Singapore using LabView and Java applet programming. An online laboratory for Microelectronics test circuit utilizing Java applet has also been developed [3]. The work in [4] describes a web-based laboratory for remote control of an inverted pendulum using Matlab and Java programming. A remote laboratory based on experiments for control engineering course has been developed at University of Texas, Arlington using Microsoft Netmeeting and Matlab's Simulink environment [5]. An ongoing work for the development of remote lecture demonstration and experiments for communication engineering using LabView through web browser has been described in [6].

It has been found that most of the real-time remote laboratory systems use either LabView software from National Instruments or Matlab software from Mathwork Corporation or Agilent Technologies' VEE pro for instrument control along with Java Applet programming for client and server side web interfaces. In order to use this software, the demand for bandwidth is very high and it takes a considerable amount of time to load the Java applet at the client side over a low bandwidth network link.

In our approach, we have used Microsoft Visual Basic to develop our own custom interface panels for all instruments for remote control operation from client side. We have incorporated all the major features of an actual instrument in our interface to be controlled from client side. We have designed six experiments based on practically used modulation techniques that are theoretically taught in most of the Communication Engineering courses. The user interfaces for all the experiments have been designed in a manner to give the remote client a basic understanding of a particular modulation technique. The facility for viewing waveform and spectrum at different stages of an experiment has also been provided using hardware multiplexer and analog / digital switches that can be controlled from the client GUI.

II. SYSTEM ARCHITECTURE

The overall architecture of the system consists of designing six hardware experiments based on basic modulation techniques along with the GUI development in software environment for each of the experiments and instruments required for remote operation. It also includes designing of switching matrices using hardware multiplexers, relays and analog switches to control various input/output parameters for each experiment.

A. *Hardware Architecture*

A general scheme of the system that has been designed using client-server architecture is shown in Fig. 1. The remote users can logon to the remote server using TCP/IP link over the Internet and can select and perform a particular experiment of his/her choice from the available list.

At the server end, hardware setup of all the laboratory instruments and the experiments are attached to the server, which we define as an *Application Server*. The application server is connected to the Internet using an Ethernet card. It is also connected with the programmable instruments through a PCI General Purpose Interface Bus (GPIB) card and GPIB cables. An experimental board containing different experiments is connected to the parallel port of the application server. For selecting a particular experiment and its corresponding input/output signals, hardware multiplexers have been integrated with the experiment board. A web cam (Logitech Quick Cam pro 3000) has been attached to the application server to enable the users to view real-time capture of Oscilloscope and Spectrum Analyzer's display as well as the experimental setup.

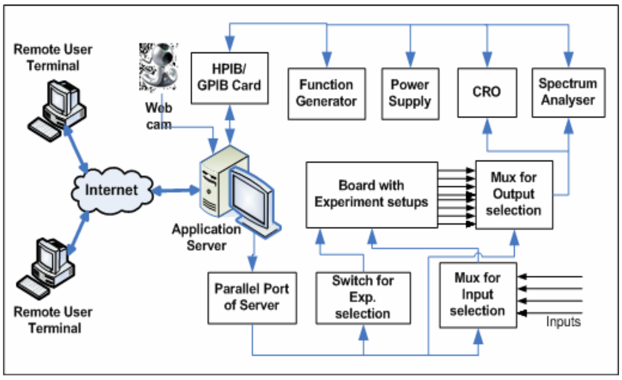

Figure 1. Hardware architecture of the laboratory

B. *Software Architecture*

The software design for the remote laboratory system focuses on a client-server type software model whose primary functionality is to interface the remote users with the application server and controlling of the actual laboratory experiments/instruments. Fig. 2 shows the software architecture of the remote laboratory system.

The GUIs for both the client and the server side have been developed using Microsoft Visual Basic (VB) 6.0. The underlying protocol for communication between server and client side is TCP/IP for which socket programming has been used within VB.

The server side GUI allows the instructor to monitor and control the server process as well as to modify the configurations of the instruments. It has three main components:

- The Instrument Interface Layer (IIL) handles all the tasks of communication to and fro from the instruments using the instrument controller (a PCI GPIB card). Standard Command for Programmable Instruments (SCPI) language library has been used in VB environment to send commands to and receive data from the instrument driver, which uses the GPIB IEEE 488.2 standard protocol to drive the instruments.
- The experiment interface layer sends command to the parallel port for selection of different experiments as well as changing different input/output parameters related to the experiment.
- The web cam interface sends video frame feedback of the experiment setup to client side.

Fig. 3 shows a screenshot of server side GUI using which the instructor can monitor all the commands generated from the client side.

The client side GUI allows the user to select different experiments and to control various functions of the instruments associated with the experiment in real-time. For each instrument as well as experiment at the server-end, corresponding GUI has been designed at the client-end using VB control toolbox. The client's command generator for instrument issues commands according to the parameter set specified by the user for a particular instrument and transmits them via the TCP/IP socket to the server. The experiment results sent back by the server are then handled and displayed in the client GUI.

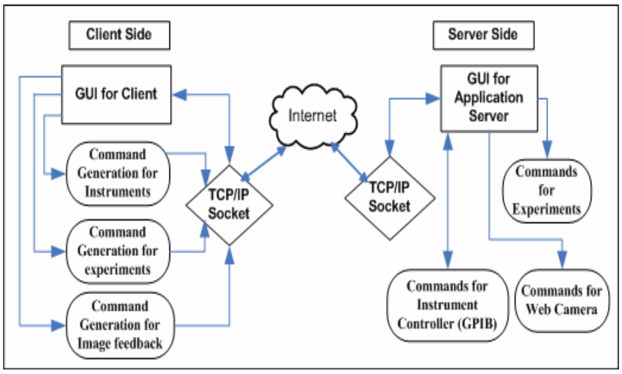

Figure 2. Software architecture of the laboratory

The client's experiment command generator issues command requests to the remote server for selecting a specific experiment and its input/output parameters. The client's GUI can also request for image feedback from the server-end using its command generation for image feedback. Instead of transmitting continuous video frames from the remote server, which would need large bandwidth, only one frame per video request would be sent from the server side and displayed in the client's GUI. Fig. 4 shows the screenshot of the client side window along with the GUI designed for each instrument for remote control operation through Internet.

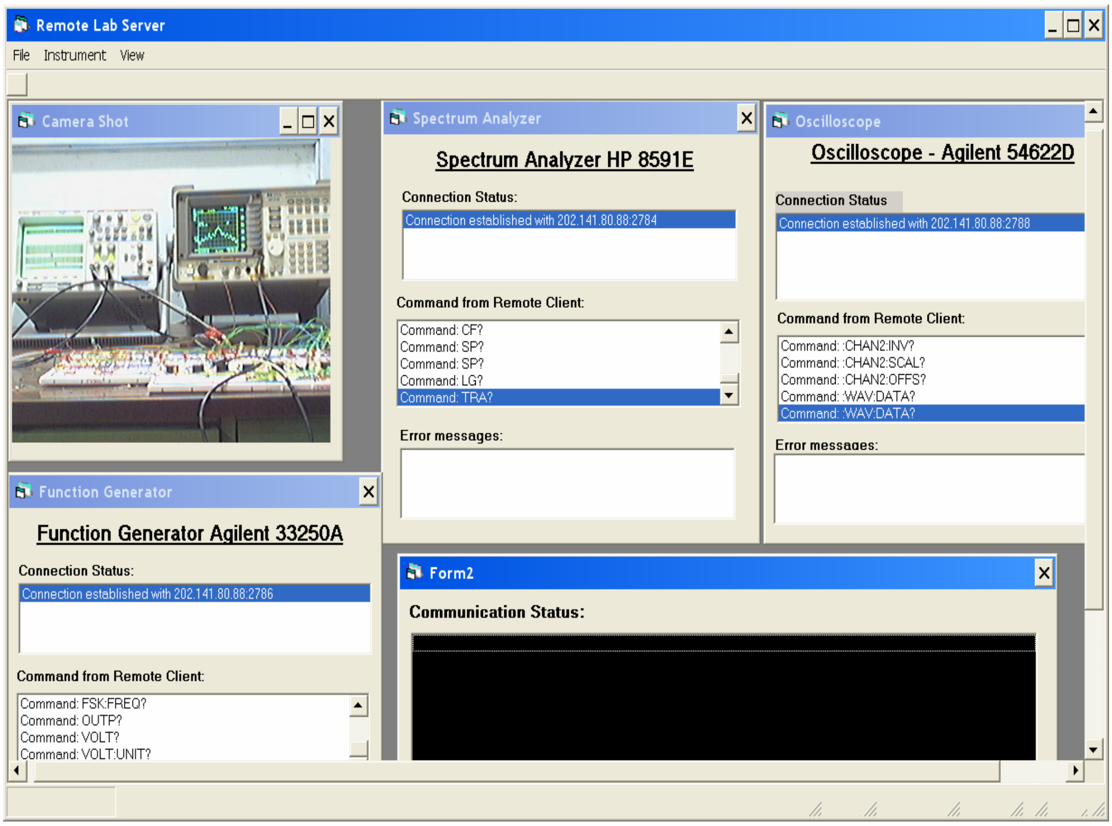

Figure 3. Server-side graphical user interface

Figure 4. Client-side graphical user interface

C. *Selection of experiments*

The parallel port of the application server has been used to control switching matrices consisting of various relays, analog/digital switches and multiplexers. These switching matrices are required for selection of experiments and different input/output signals related to the experiment. They also help in sharing the same set of laboratory equipment among the different experiments. Fig. 5 below depicts how the eight data lines of the parallel port have been used for controlling various switching matrixes.

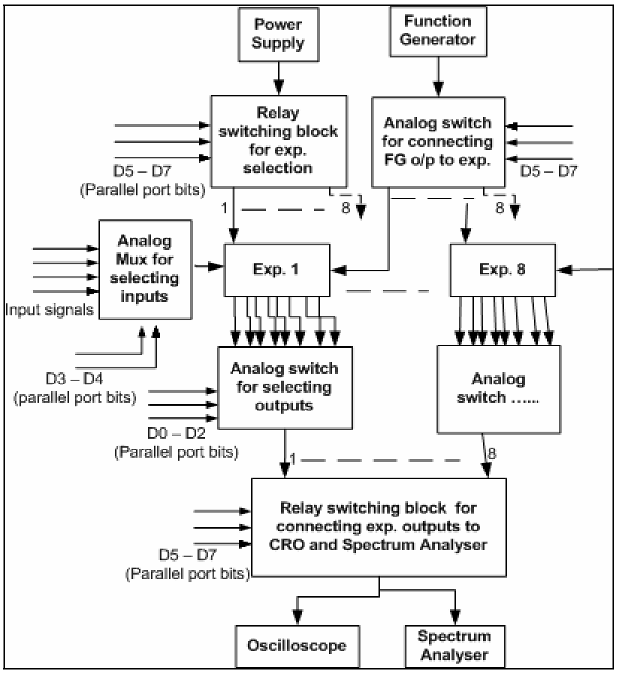

Figure 5. Switching matrix using parallel port data bits

The three Most Significant Bits (MSB) of the parallel port, D7 through D5, have been used to control relays and switches for selection of experiments and instruments. Bits D4 and D3 have been used to select any one of the four possible input signals required for an experiment. Similarly, bits D2 through D0 of the parallel port have been used to view various outputs of an experiment at different test points.

III. DESIGN OF EXPERIMENTS

Six experiments have been designed based on different modulation techniques and their circuitry has been mounted on a single experiment board. Out of these six experiments, four experiments are based on digital modulation schemes (BPSK, QPSK, 8-PSK, binary FSK) and the remaining two are based on analog modulation schemes (AM and FM). All these experiments are connected to the application server along with the required laboratory instruments.

A. *Binary Phase Shift Keying (BPSK)*

BPSK is the fundamental modulation scheme amongst PSK modulation techniques. In PSK modulation, the phase of the carrier waveform is shifted to different angles depending upon incoming data streams. The usual techniques for generation of PSK signals are mainly based on either using continuous-time analog signals or stored waveform, by digital means. Our method uses a simpler method of generation of PSK signals using logic gates. This method of generation of PSK modulated signal is easy and it reduces overall system complexity. Fig. 6 below shows the user interface depicting the hardware block diagram of the BPSK modulator that has been designed using simple logic gates.

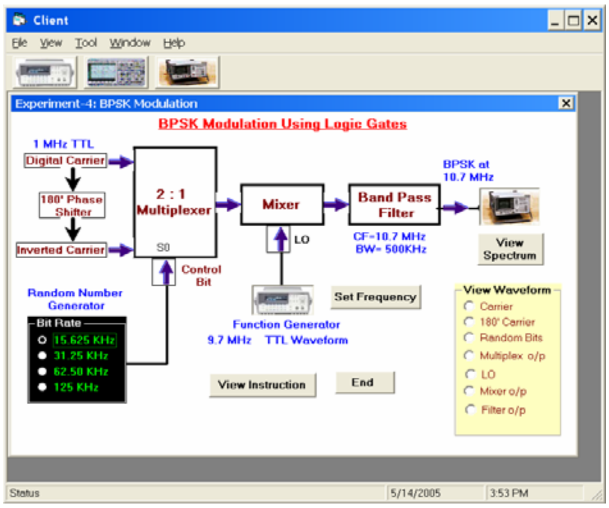

Figure 6. BPSK generation using logic gates

A 2:1 multiplexer selects one of the waveforms: Carrier or its inverted form, depending on the current value of the data bit. A user selectable option for specifying the bit rate for the generation of pseudo random data is provided in the GUI, which automatically selects the required bit rate using hardware multiplexer. The output of the multiplexer would be a BPSK modulated TTL signal. This signal is up-converted using a logical AND gate. The mixer output is passed through a 10.7MHz band-pass filter to obtain a BPSK modulated analog waveform at 10.7MHz. User selectable option for test outputs at different stages of the experiment has also been provided in the GUI.

Fig. 7 shows the client side screen performing BPSK modulation experiment for pseudo random input data sequence at a bit rate of 31.25 Kbps.

B. *Quadrature Phase Shift Keying (QPSK)*

QPSK modulator experiment has also been designed using simple logic gates. A 4:1 multiplexer selects one of the four waveforms: Carrier (0 rad), $\pi/2$-shifted carrier, π-shifted carrier, $3\pi/2$- shifted carrier; depending on the current value of the di-bit. A user selectable option for specifying the bit rate for the generation of pseudo random data is provided in the GUI. A serial to parallel converter separates I and Q channels required for the generation of QPSK waveform. These two channels form the *select* signals for the multiplexer. The output of the multiplexer would be a QPSK modulated TTL signal. This signal is up-converted using a logical AND gate. The mixer output is passed through a 10.7MHz band-pass filter to obtain a QPSK modulated analog waveform at 10.7MHz.

Fig. 8 shows the client side screen performing QPSK modulation experiment for pseudo random input data sequence at a bit rate of 31.25 Kbps.

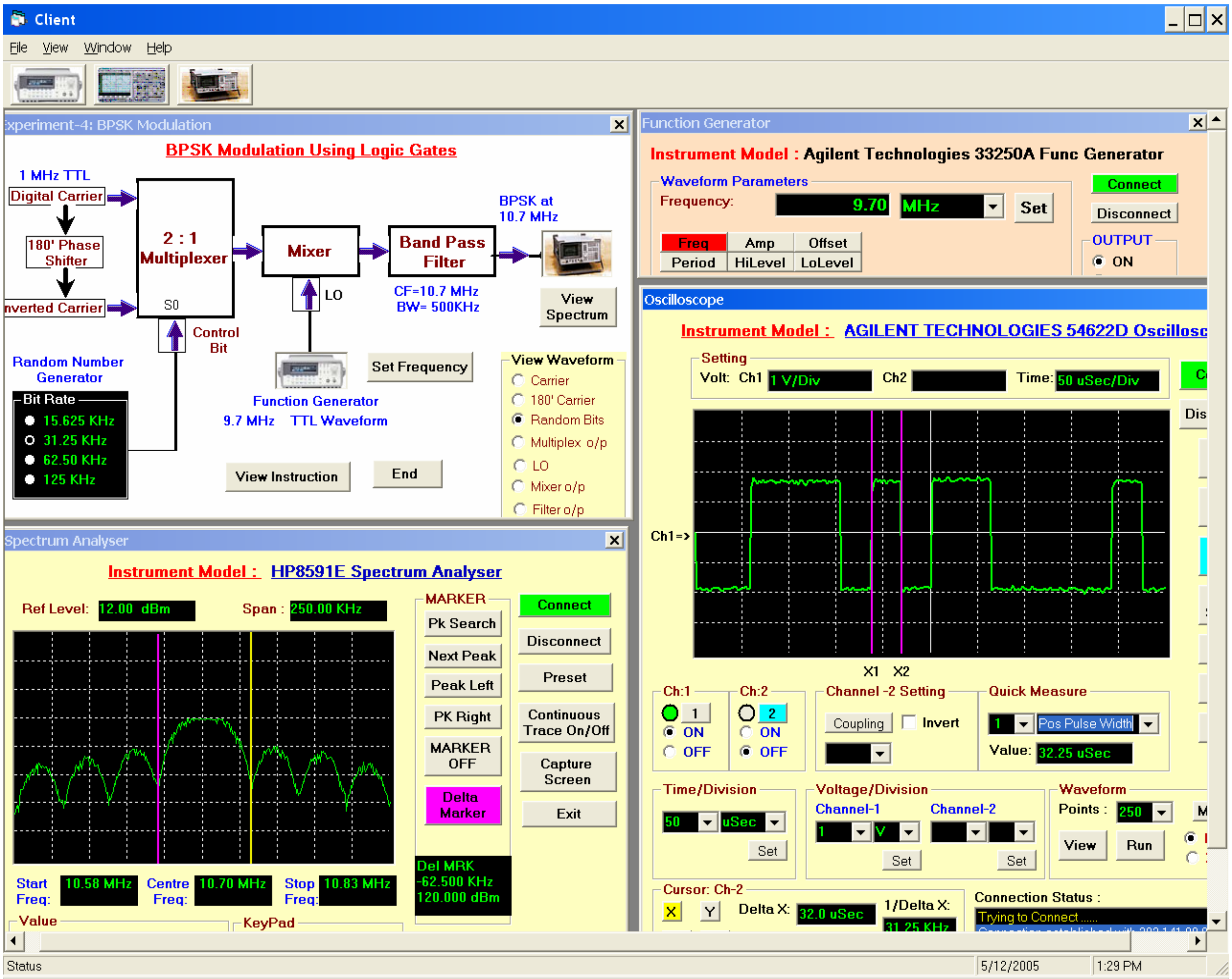

Figure 7. Client performing BPSK modulation experiment

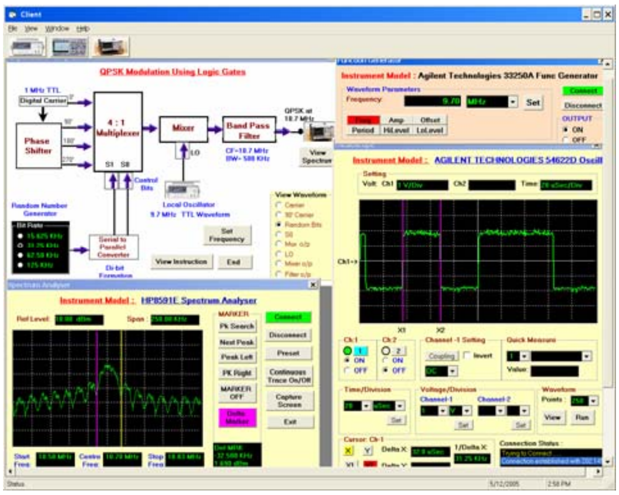

Figure 8. Client performing QPSK modulation experiment

C. *Eight Phase Shift Keying (8PSK)*

8PSK modulator experiment has been designed on similar principles using simple logic gates. An 8:1 multiplexer selects one of the eight waveforms: Carrier (0 rad), $\pi/4$-shifted carrier, $\pi/2$-shifted carrier, $3\pi/4$-shifted carrier, π-shifted carrier, $5\pi/4$-shifted carrier, $3\pi/2$- shifted carrier and $7\pi/4$-shifted carrier; depending on the current value of the tri-bit. A user selectable option for specifying the bit rate for the generation of pseudo random data is provided in the GUI. A serial to parallel converter provides tri-bit information, required for the generation of 8PSK waveform. These three channels form the *select* signals for the multiplexer. The output of the multiplexer would be an 8PSK modulated TTL signal. This signal is again up-converted using an AND gate. The mixer output is passed through a 10.7MHz band-pass filter, to obtain an 8PSK modulated analog waveform at 10.7MHz.

Fig. 9 shows the client side window performing 8PSK modulation experiment for pseudo random input data sequence at 62.5 Kbps bit rate.

D. *Frequency Shift Keying (FSK)*

FSK modulator circuit has been designed using the XR-2206 IC. A user selectable option for specifying the bit rate for the generation of pseudo random data is provided via GUI. Depending on the voltage level of the digital signal, the circuit switch between two frequencies (f1=270 KHz for logic '0' and f2=430 kHz for logic '1') to generate FSK modulated signal.

Fig. 10 shows the client side screen performing FSK modulation experiment for input bit rate of 62.5 Kbps.

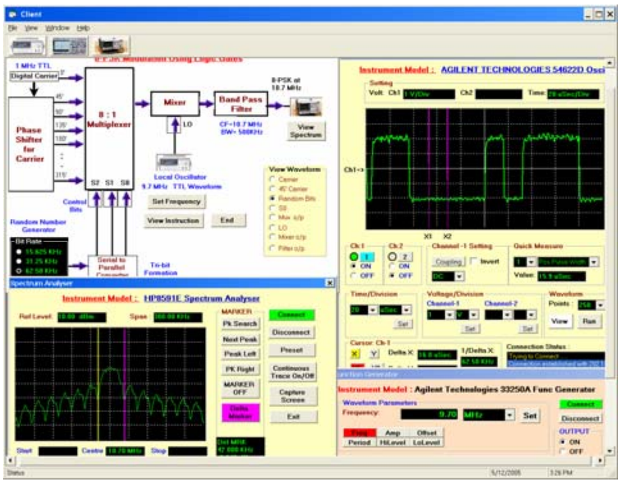

Figure 9. Client performing 8PSK modulation experiment

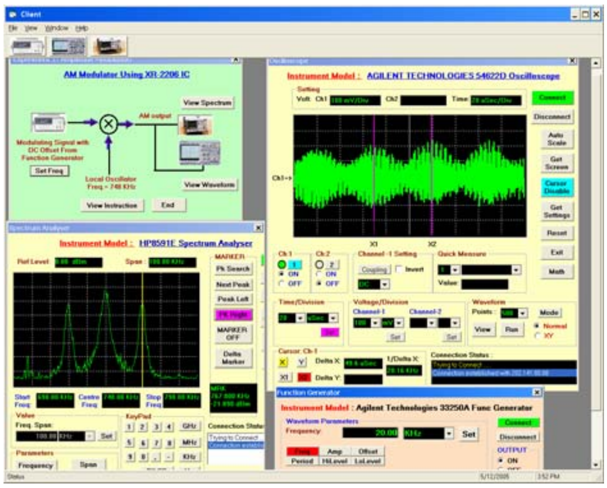

Figure 11. Client performing amplitude modulation experiment

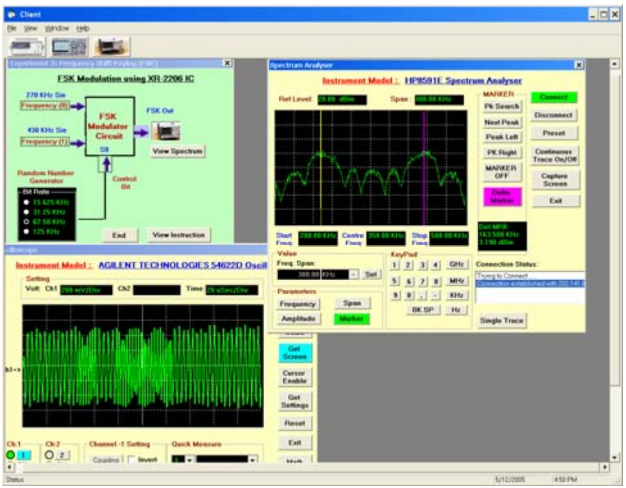

Figure 10. Client performing FSK modulation experiment

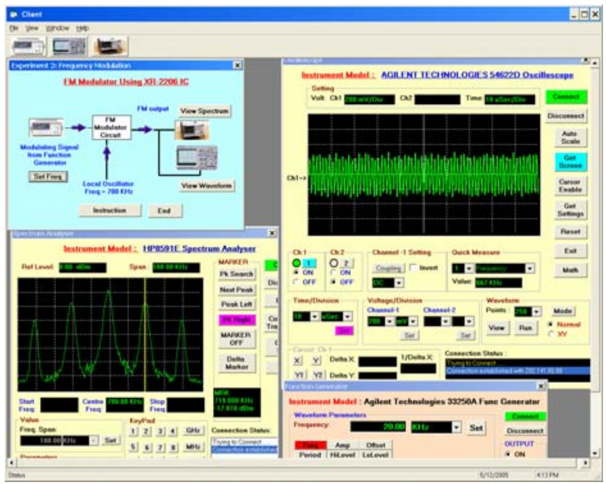

Figure 12. Client performing frequency modulation experiment

E. Amplitude Modulation (AM)

AM modulator circuit has also been designed using XR-2206 function generator IC. The modulating signal is given from the function generator which amplitude modulates the fixed carrier signal of 748 KHz to generate AM signal. The users can change the frequency, amplitude as well as shape of the modulating signal by sending appropriate commands to the function generator connected to the application server. The remote user can also view the time domain plot of the modulated signal and its spectrum in its own window.

Fig. 11 shows the client side window performing amplitude modulation experiment for modulating frequency of 30 KHz.

F. Frequency Modulation (FM)

FM modulator has been designed using general-purpose XR-2206 function generator IC. The modulating signal is given from function generator which modulates the frequency of the carrier signal of 700 KHz to generate the FM signal.

Fig. 12 depicts the client side screen performing frequency modulation experiment for modulating frequency of 20 KHz, 20mVp-p.

IV. DISCUSSION

The overall system described above has been designed for practical implementation and a real setup has been developed in the department of Electronics & Communication Engineering (ECE), IIT Guwahati for demonstration to the undergraduate students studying Communication Engineering course. Fig. 13 shows the pictorial view of the remote laboratory that has been set up. At present, we have designed six experiments as mentioned above, for demonstration purpose. Integration of more experiments to the current system is under progress. The single experiment board containing the six different modulation experiments can handle single client connection at a time, on timesharing basis. However, a single client can select any one of the six different experiments according to his/her own choice.

The bandwidth requirement for the entire system has been optimised by sending only the necessary information and commands over the TCP/IP link. Instead of sending continuous frames using webcam, only a single frame would be transmitted at a given point of time, based on the client's request. The system has been tested on different network environments to validate its consistent performance. The system works smoothly over the LAN as well as the wireless network environment without introducing any appreciable delay while executing a

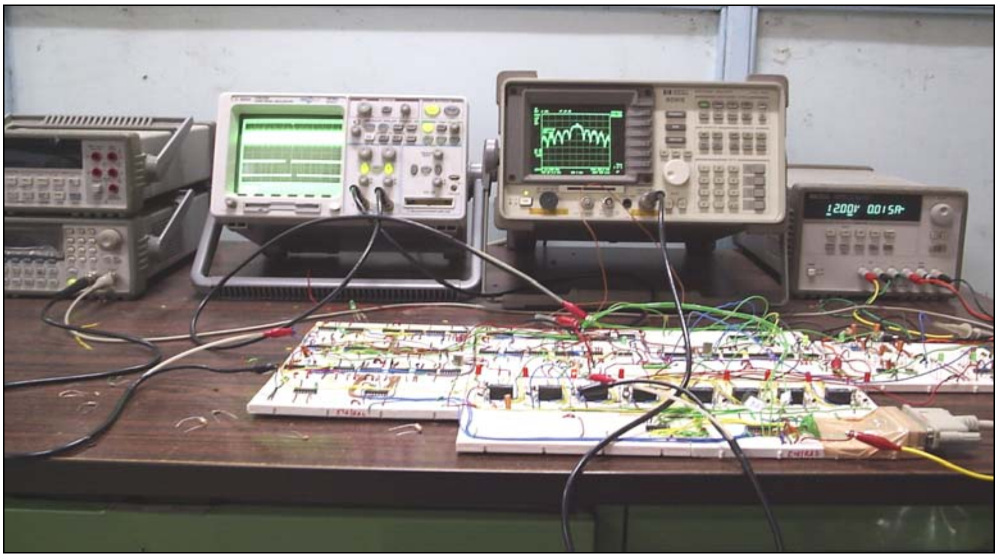

Figure 13. Actual setup of the remote laboratory at ECE department, IIT Guwahati

client's request. But over a dial-up link using a commercial 56-kbps modem, the time needed to access remote laboratory site and start the client window is about 30 seconds. It takes less than 10 seconds for the system to complete a GPIB command execution including transmission of the commands, receiving and plotting the data over the dial-up link.

The system is quite scalable in the sense that new experiments sets can be easily integrated into it without major hardware changes. Only the GUIs for new experiments have to be designed and developed in software.

The overall cost of software development has been minimized, by designing our own custom interfaces without using additional third party software. This helps in essay distribution of the software among different clients.

V. CONCLUSION

This paper describes the development of a remote laboratory with some hardware experiments based on modulation techniques used in Communication Engineering course. The GUIs for all the experiments have been custom designed within VB in block diagram manner to give students a basic idea about the different modulation schemes, their time-domain waveforms and spectrums. All the instruments required for this laboratory have also been programmed and their GUIs have been custom designed using VB for remote control operation. Students just need to download one executable file to logon to the remote laboratory and perform the experiments.

Although setting up of a remote laboratory can not replicate the real laboratory as a whole, still it has been tried to provide students a way of alternate learning to verify their theoretical knowledge. Integration of online lectures can be made to the existing system to make the experiments more interactive for students.

REFERENCES

[1] Chi Chung Ko et. al., "A Web-Based Virtual Laboratory on a Frequency Modulation Experiment," *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, vol. 31, No. 3, pp. 295-303, August 2001.

[2] C.C Ko, Ben M. Chen, Jianping Chen, Yuan Zhuang and Kay Chen Tan, "Development of a web-based Laboratory for Control Experiments on a Coupled Tank Apparatus," *IEEE Trans. on Educ.*, vol. 44, No. 1, pp. 76-86, February 2001.

[3] Ralph M. Ford, Jonathan Bondzie, and Paul Kitcho, "Java Applets for Microelectronics Education," *IEEE Trans. on Educ.*, vol. 44, No. 2, May 2001.

[4] J.Sanchez, S.Dormido, R.Pastor, F. Morilla, "A Java/Matlab-Based Environment for Remote Controlled System Laboratories: Illustrated with an Inverted Pendulum," *IEEE Trans. on Educ.*, vol.47, No.3, pp. 321-329, August 2004.

[5] Nitin Swamy, Ognjen Kuljaca, and Frank L. Lewis, "Internet-Based Educational Control Systems Lab Using NetMeeting," *IEEE Trans. on Educ.*, vol. 45, no. 2, pp. 145-151, May 2002.

[6] Ladimer Nagurney, "Work in Progress - Remote Lecture Demonstrations and Experiments for Communication Engineering," *34th ASEE/IEEE Frontiers in Education Conference*, T3C, October 20-23, 2004, Savannah, GA.

[7] "Agilent 54622D Mixed Signal Oscilloscope Programmer's guide", Agilent Technologies., Dec. 2000.

[8] "Agilent SICL User's Guide for Windows", Agilent Technologies, 6th edition, April 2003.

[9] "Agilent 33250 80MHz Function/Arbitrary Waveform Generator User's Guide", Agilent Technologies, April 2000.

[10] "HP 8590 E-Series and L-Series Spectrum Analyser and HP 8591C Cable TV Analyser Programmmer's Guide", Hewlett Packard, August 1995.

AUTHORS

Sanjib Das is working as a Research Engineer with the Dept. of Electronics and Communication Engineering, Indian Institute of Technology Guwahati, Assam, India. (e-mail: sanjib@iitg.ernet.in).

L. N. Sharma is working as a Research Engineer with the Dept. of Electronics and Communication Engineering, Indian Institute of Technology Guwahati, India. (e-mail: lns@iitg.ernet.in).

A. K. Gogoi is a Professor in the Dept. of Electronics and Communication Engineering, Indian Institute of Technology Guwahati, India. (e-mail: akg@iitg.ernet.in).

Manuscript received February 4, 2006.