

Internet Based Experiments for Physical Laboratory Set-up

Abul K M Azad¹, Andrew Otieno¹, Omar Ghrayeb² and Navin Anand³

¹Department of Technology, Northern Illinois University, Illinois, IL-60115.

²Department of Industrial Engineering, Northern Illinois University, IL-60115.

³ Graduate student, Department of Technology, NIU, Illinois, IL-60115.

Email: azad@ceet.niu.edu

Abstract

This paper presents the development of an Internet based experimental laboratory facility where students and faculty can perform experiments through a web-assisted physical laboratory from a remote location. The equipment is housed within a laboratory, while the only requirement for the remote site is a PC with Internet access and a browser. In order to develop this system, a suitable experiment needs to be identified. Once the experiment has been selected, the required tasks can be divided into three major steps. The first step involves identifying the accuracy of the experiments, number of inputs and outputs and the system's speed along with the selection of a suitable interfacing card and software. The second step deals with the development of software for experiment control and data acquisition along with a user-friendly graphical user interface. The third and final step involves the accessing of the experiments via the Internet. In this project, four experiments covering various fields of engineering and engineering technology were tested successfully.

1 Introduction

With the rapid 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^{1,2}. Especially, distance education and non-traditional classrooms have the capability to reach more students using specialized instruction and self-paced learning. Integration of the Internet with educational systems is well described in Poindexter and Heck³. In almost all the cases, web courses are only based on theoretical/simulation materials. While a good learning experience can be obtained

from such systems in many situations, effective and complete learning, especially in applied engineering, requires a mixture of theoretical and practical sessions.

To address this important issue, a number of attempts have been made to provide students with practical exercises or experimentation experience through the web⁴⁻⁶. In most of the cases, these systems require custom designed hardware and software to interface with a PC system. Once the experiment has been interfaced with a computer, then the question of developing a suitable graphical user interface and also accessing the experiment through the Internet arises. All these require a considerable amount of time, effort and resources. This is one of the main reasons why a complete laboratory course has not been delivered through the Internet.

The only way to make the Internet based laboratory system viable is to move toward a modularity concept. Under this strategy there needs to be a few modular hardware and software blocks that can be made easily adaptable to a range of engineering and engineering technology laboratory experiments. In this project, while developing the experiments for Internet delivery, a modularity concept has also been addressed.

2 Experimental Facilities

In this project four experiments were conducted, and all of them can be accessed and controlled through a remote PC over the Internet. The experiments were developed through a project that involved collaboration between two engineering departments: Industrial Engineering and Engineering Technology. The choice of experiments has been made to cover a range of areas within the two collaborating departments. The general description for each of these experiments is given below.

2.1 Motor health monitoring system

Monitoring the condition/health of a motor is vital for timely maintenance of any motion system. This is especially critical for a industrial environment where delivery of goods is of prime importance. The purpose of the experiment is to monitor the condition of a motor by analyzing the frequency spectrum of the vibration from the motor body. Both the time and frequency domain information can be accessed from a control room via the Internet, which may be located far away from the shop floor. The early detection of any fault can then be passed to the maintenance people for their timely action.

The experimental system used for this purpose is shown in Figures 1 and 2. The motor under test is a 24V dc motor operated through a relay. The relay can be activated from a computer through an input/output (I/O) card housed in the PC. A tiny accelerometer is glued onto the motor body (Figure 2). After suitable amplification the accelerometer output is fed to the PC through a I/O card. The output is then displayed in both time and frequency domains through LabVIEW Virtual Instrument (VI). The frequency domain characteristics can then be analyzed to ascertain the condition of the motor. The experiment can be triggered from a remote PC. Additionally, data can be stored on the hard disk of a remote PC via the Internet.

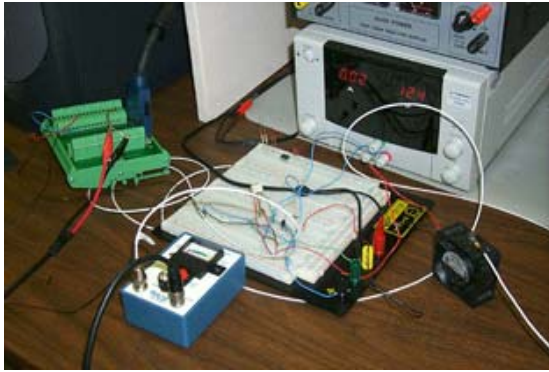


Figure 1: Complete experimental system for motor health monitoring.



Figure 2: Motor accelerometer set-up.

2.1 Digital signal processing simulation

Unlike the above experiment, this is purely a simulation experiment. The user can choose various types of signals, which can be generated within a software environment. The time domain signal along with its unfiltered frequency spectrum and filtered spectrum are displayed. The user has the freedom to choose from a range of signals, such as sine, square, sawtooth, and triangular wave. This kind of experiment is essential for the students to study the shape and frequency content of standard waveforms.

2.3 Acoustic emission monitoring of tool wear

Machining operations are a crucial part of the metal fabrication industry enabling parts to be produced with high precision and consistency. In machining, one of the main reasons for poor quality is the inability of worn tools to maintain good surface finish and dimensional control. Thus, in order to maintain machining quality and to prevent damage to the work-piece, tool condition ought to be monitored accurately or predicted early to prevent the rapid wear stage.

The aim of this experiment is to develop a cutting tool condition monitoring system that can be remotely accessed from another location. This system utilizes Acoustic Emission (AE) techniques to monitor tool wear during machining. In effect, it is intended to reflect an unattended machining situation where a worn out tool can be detected remotely from any convenient location, thus freeing up an operator to physically stay at the station and continuously monitor the machining process.

Several techniques have been developed for monitoring tool condition during machining. AE is an indirect technique in which waveforms can be analyzed and characterized to correlate with the amount of wear on the tool. In this experiment, a Kistler 8152B sensor was employed to pick up the AE from the cutting interface (Figure 3). This sensor was attached to the tool shank by an M6X10 hex screw (Figure 4). To reduce the effects of signal attenuation due to the high acoustic impedance of air, a couplant (grease) was applied between the sensor and the shank.

Data acquisition is performed in two stages. First the raw signals are conditioned using a Kistler AE-Peizotron coupler 5125B. In the second stage, the local host computer captures filtered AE and RMS (root mean square) signals. The host computer is connected to the coupler through a National Instruments (NI) data acquisition card housed in the PC. To store and display the collected data, a VI was created using LabVIEW. Data acquisition in this system can also be triggered, collected and stored remotely via the Internet.



Figure 3: Complete experimental system for AE data collection.



Figure 4: Attached AE sensor.

2.4 Assembly line product counter

Keeping track of raw materials or finished goods in an assembly line is essential for production records. Although this can be a manual process involving physical counting, the problem can be very challenging for high-speed conveyor systems. The aim of this experiment is to utilize simple non-contact proximity sensors to count assembled finished products on a conveyor loop in an automation laboratory. The counter output is passed to a computer through suitable interfacing tools for graphical and numerical display of the number of products on an appropriate VI. Finally, the VI is made accessible through a remote PC over an Internet connection. Figures 5 and 6 show the experimental set-up for this product counter system.



Figure 5: Complete experimental system for product counter.

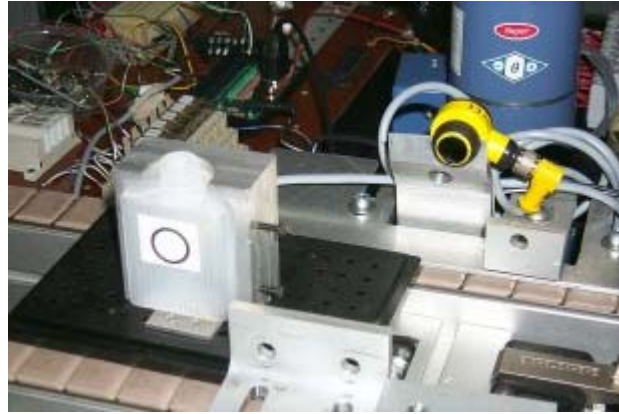


Figure 6: Product sensor set-up.

3 Interfacing Hardware and Software

Initially, National Instruments I/O cards were chosen as the system hardware for data acquisition and control, while Matlab was chosen as the software. Operational details of these two can be found in References 7 and 8. Subsequently, it was realized that National Instrument's LabVIEW software has much more flexibility for data acquisition and control over the Internet⁹.

LabVIEW can also be used along with other third party software, making it more attractive for this application. It can also incorporate Matlab functions^{10,11}. Apart from this, LabVIEW has an inbuilt server facility, which can be utilized for Internet access of experiments¹¹⁻¹². The I/O interface card used for all these experiments is NI 6040E¹⁴. This card has 12-bit accuracy with 16 single ended, 8 differential channels, 2 channels analog output of 12-bit resolution, 8 digital I/O lines, and 2 up/down counters of 24-bit resolution. These specifications are enough to meet the requirements for all the experiments.

4 Development of Graphical User Interface

It is very important to provide a very good graphical user interface (GUI) for an Internet based experimental facility. This is not only needed for providing a user friendly facility but also to attract the student(s) to perform experiments without any supervision and traditional help provided during a laboratory class¹⁵. LabVIEW provides a facility to develop a GUI called virtual instrument (VI), which can serve both of the above purposes^{16,17}. The concept of VIs is to create more powerful, flexible, and cost-effective instrumentation systems around a PC using software as the engine and interface. A VI can easily export and share its data and information with other software applications.

In this project, apart from the digital signal processing simulation Vis, all other experiments are developed from two main VI blocks. These blocks served as the basic modules for all other VIs. The first one is an *Analog Data Acquisition VI*, which acquires data continuously from one or more analog channels. The second one is a *Software Digital Trigger VI*, which triggers a digital output channel and holds the trigger until the user prompts. Two such VI's together with their respective wiring diagrams are shown in Figures 7 to 10.

The front end for the motor health condition monitoring system VI is shown in Figure 7, along with the wiring diagram in Figure 8. This particular experiment involved a continuous trigger from LabVIEW to the data acquisition (DAQ) card, which is internally configured to activate a start pulse on PFI0/TRIG1. This continuous pulse activates a relay in the driver circuit of the motor. The vibration of the motor is picked up by an accelerometer and its output is amplified through an instrumentation amplifier. The amplifier output is connected to one of the channels in the DAQ board and displayed in real-time using a waveform graph (Figure 7). The frequency spectrum of the input time signal is also computed simultaneously and displayed. The user has the options to enter the device number, channel(s), scan rate, buffer size, and the scans to read at a time. For this particular case, the analog input channel 0, channel ACH0, has been utilized.

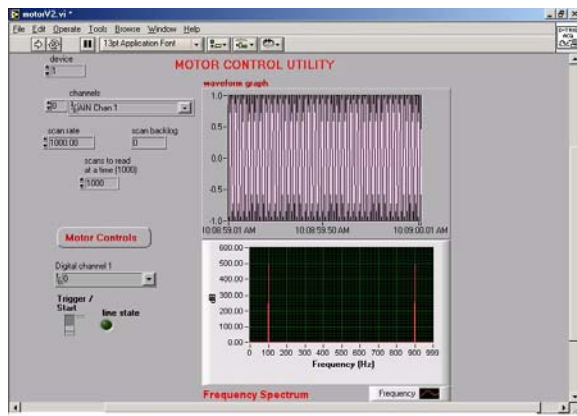


Figure 7: Front end for motor experiment.

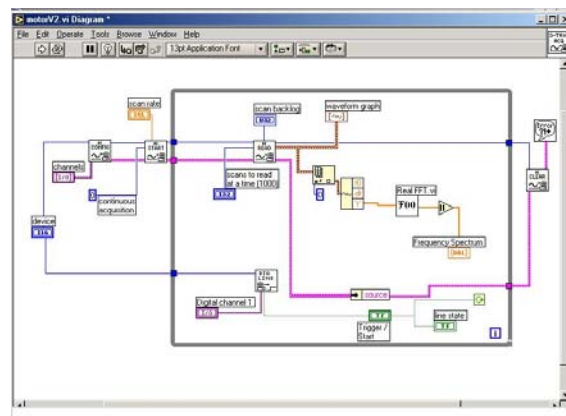


Figure 8: Wiring diagram for motor experiment.

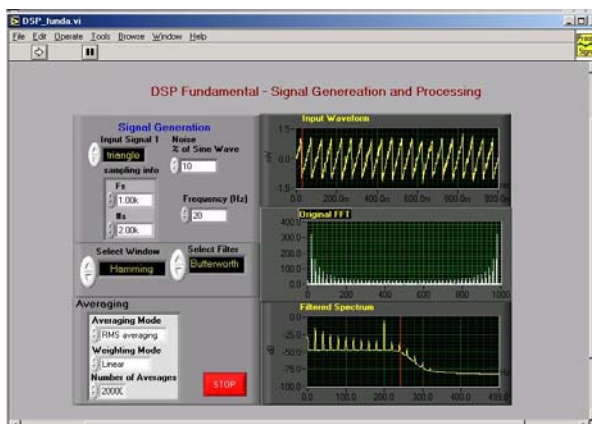


Figure 9: Front end for digital signal processing simulation.

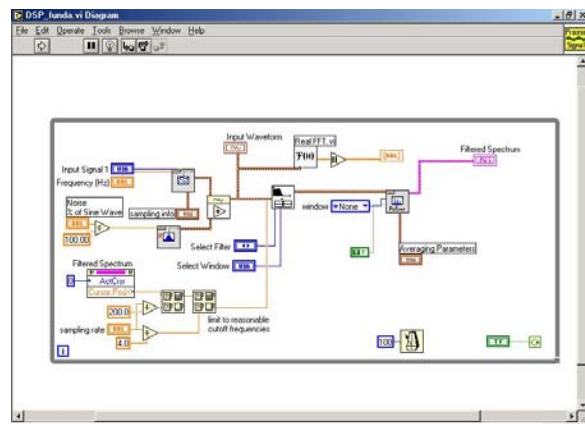


Figure 10: Wiring diagram for digital signal processing simulation.

The VI developed for digital signal processing simulation explains the DSP basics. Here the objective is to display the fast fourier transform (FFT) of some commonly used signals and the effect of noise on them. This VI also has three graphs to display (Figure 9). The first one is for

a simulated signal, the second is FFT of the signal, and the third one is for a filtered power spectrum. This particular VI utilizes some of the inbuilt features of LabVIEW.

The sub-VIs, which are used with this VI are real FFT, FFT Power Spectrum VI and Filter and Window VI. The real FFT computes the discrete Fourier transform of the input time sequence. FFT Power Spectrum VI computes the average power spectrum of the input sequence based on the various averaging parameters, and also the type of filter to be utilized from Filter and Window VI. The noise is a Gaussian White noise, which is mixed with the simulated input waveform types.

The frequency range of operation for this VI is up to 499 Hz. In this VI, the user can enter the percentage of noise that is to be added and the type of signal and also indicate the various filtering parameters. This VI can be used as an excellent tool in the demonstration of certain DSP basics. The wiring diagram for the VI is given in Figure 10.

5 Performing Experiments via Internet

Each experiment is physically connected to a dedicated PC, which is called a *server PC* and the PC's at the remote location used to access the experiment through the Internet are generally known as *client PCs*. The operation and control of an experiment over the Internet and making the data available within a client PC is important parts for this project.

LabVIEW has undergone several revisions and the newer versions now have utilities such as TCP/IP, Internet Toolkit, VI Server, Front Panel Web Publishing, Remote Data Acquisition (RDA), DataSocket, and so on. These utilities have enhanced its ability to enable the development of distributed applications. In addition, several third party toolkits have enabled Internet-based VI control. These are LabVNC, AppletVIEW, LabSQL and LabPerl. A comparative study of various methods is shown in Table 1^{18,19}. Considering the time and resource limitation for this project, only two methods were tested for implementation. One is the DataSocket technology (within the LabVIEW environment) and the other is LabVNC.

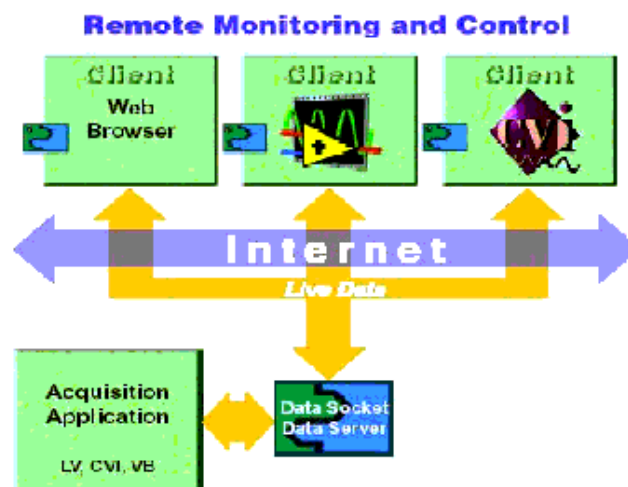


Figure 11: Block diagram for data socket technology.

	Application Development	Range of Applications	User Interaction	Programming Knowledge	Security	Cost (additional software)
Remote Viewing*	Very simple	Limited (no live data)	None	Only LabVIEW	Good	None
CGI Technology*	Relatively difficult	Moderate (no live data)	Moderate	LabVIEW +CGI +HTML	Good	Adds-on Internet toolkit
Data Socket and ActiveX*	Moderately difficult	Good; (supports live data)	Good (only on Explorer)	LabVIEW + Visual Basic + HTML	Could be problematic	Component Works +VB
Data Socket and Java Applet*	Difficult	Good (supports live data)	Very good	LabVIEW+JAVA+HTML	Good	Visual Java + VIB
Applet View*	Relatively simple	Good (supports live data)	Very good	LabVIEW +AppletView	Good	AppletView
Client-Server (TCP/IP)**	Difficult	Very good (supports live data)	Excellent	LabVIEW	Very Good	None
VI Server**	Relatively difficult	Good (supports live data)	Good	LabVIEW	Very Good	None
Data Socket **	Minimally difficult	Good (supports live data)	Good	LabVIEW	Good	None

Table 1: A comparison between the available methods for Internet access to an experiment.

Data socket technology in LabVIEW simplifies the live data exchange over a network between computers. The underlying protocol that has been used in these applications is data socket transfer protocol (DSTP). This protocol is used to access a laboratory application from a remote site over some communication medium. The access could primarily involve acquisition, data logging, analysis and presentation.

DataSocket technology is based on industry-standard TCP/IP and is targeted for fast and live data transfer. DataSocket implements an easy-to-use, high-performance programming interface designed for sharing and publishing live data in measurement and Automation applications. DataSocket mainly consists of two pieces – the DataSocket API and the DataSocket Server (Figure 11). The DataSocket API is linked with the Internet and presents a single interface for communicating with multiple data types from multiple languages.

LabVNC in essence is a remote display system to view a computing desktop environment not only on the machine where it is running but also from a remote client PC through an Internet browser. The operation of LabVNC is relatively simple. When LabVNC is activated, a window called “LabVNC.vi” (Figure 12) will pop-up. The user needs to select the options on this window according to the level of access allowed for the client PC. This level of access can be from just viewing a VI to doing almost everything that can be done within the server PC itself. After selecting the options, the user then turns the server “on” by moving the slide button under LabVNC server switch.

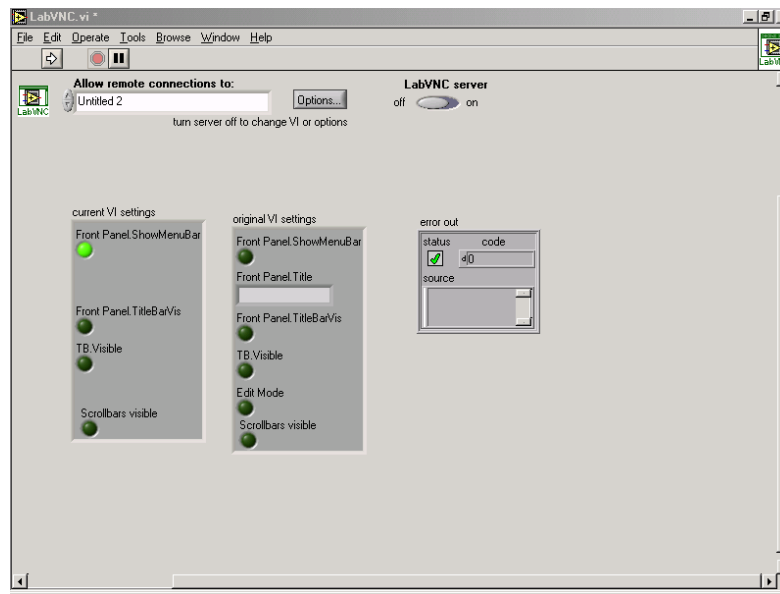


Figure 12: LabVNC option window when activated.

Although Data Socket Technology provides an interactive means for performing an experiment over the Internet, it has some limitations. The client PC should also have the LabVIEW software along with the VIs used for an experiment. Any change in VI in the server PC needs to be manually transported to the client PC as well. On the other hand, LabVNC is a cross-platform, open source utility that uses the open-source VNC (virtual network computing) protocol. LabVNC can instantly enable web control over a VI running on a server. LabVNC will create a Java Applet on the fly that is an exact representation of any VI. This Java Applet will run in a remote client browser and give full control over the VI.

6 Conclusion

Through this project, a facility has been developed for performing hardware experiments through the Internet. A range of experiments has been considered and integrated in this facility from two departments in the College of Engineering and Engineering Technology. The steps followed for each of these cases are: identification of suitable interfacing card and software, development of software for experimental control and data acquisition along with a GUI, and finally accessing

an experiment via the Internet. Commercially available hardware and software are used to keep the system adaptable with other existing platforms.

Along with providing Internet access to an experiment the project was also focused on developing a modular concept, so that this facility can be extended to other experiments without much time, effort and resources. This kind of system can open a new door towards offering a complete Internet based course, even with an experimental part, with limited resources. Apart from an educational application, this facility can also be used for Internet based industrial floor management and quality control purposes.

The work carried out in this project has demonstrated significant potential for further development of Internet based laboratory course delivery. This project is in progress for the development of full-fledged labs for course offerings and external funding is being sought for this cause. As the development progresses, some of the issues we are addressing include safety, security and access, and scheduling. The findings of the new developments and results of student interaction will be reported at a later suitable time after the students have had a chance to perform the experiments in their respective courses.

7. Acknowledgement

The investigators wish to express their gratitude to the following: Omron Electronics Foundation for funding this work; Dr. Romualdas Kasuba, Dean of College of Engineering and Engineering Technology (CEET), and Technology Department Chair Dr. Cliff Mirman for their continuous support throughout the project; Computer Support Service of CEET, who also played an important role by providing timely support whenever it was needed.

The investigators would also like to thank the student workers Mr Navin Anand and Mr Bharath Thota, for their contribution towards this project.

References

1. Mott, R.L. et al. (2002). Future directions for mechanical manufacturing, and industrial engineering technology programs, *Journal of Engineering Technology*, **19**(1), pp.8-15.
2. Machine Vision Association of the Society of Manufacturing Engineers. *2000 Machine Vision Industry Trends Report, 2000*. Available: www.sme.org/cgi-bin/membhtml.pl?/mva/00mvatrends.htm&&&SME&>.
3. Poindexter, S. E. and Heck, B. S. (1991). Using the web in your courses: What can you do ?, What should you do ?, *IEEE Control Systems Magazine*, **19**, pp83-92.
4. Catlin, A. C., et al., *The SoftLab experience: Building virtual laboratories for computational science*, Available: http://www.cs.purdue.edu/research/cse/softlab/softlab-vlabs/softlab-framework/softlab_report/report.html.
5. Hites, M. (2002). Creating and running undergraduate experiments controlled through the internet, *Proceeding of the IL/IN Sectional Conference of American Society of Engineering Education*, April 12, pp.192-198.
6. Distance Learning Remote Laboratories using LabVIEW, National Instruments developer zone.
7. The MathWorks: Developers of MATLAB and Simulink for Technical Computing, <http://www.mathworks.com/>.
8. National Instruments – Measurement and Automation, <http://www.mathworks.com/>.
9. Special Issue: LabVIEW Applications in Engineering Education, (2000) *The International Journal of Engineering Education*, **16**(3).
10. Johnson, G.W. and Jennings, R. (2001) *LabVIEW: Graphical Programming*, McGraw-Hill.
11. Essick, J. (1999). *Advanced LabVIEW Labs*, Prentice-Hall.
12. Travis, J. (2000). *Internet Applications in LabVIEW*, Prentice-Hall.
13. Robert H. Bishop, “Learning with LabVIEW 6i”, Prentice Hall Publishers.

14. NI 604xE – Products and Services – National Instruments – Netscape
<http://sine.ni.com/apps/we/nioc.vp?cid=10159&lang=US>
15. Wang, A. Y. et al. (2002). Predictors of Performance in the Virtual Classroom: Identifying and Helping At-Risk Cyber-Students, *Technological Horizons in Education*, **29**(10) pp.21-28.
16. Gary W. Johnson and Richard Jennings, “ LabVIEW Graphical Programming”, McGraw-Hill, Third Edition.
17. DataSocket TM Simplifies Live Data Transfer from LabVIEW, National Instruments Technical information white paper.
18. Building an Interactive Web Page with DataSocket, *Application Note 127*, National Instruments, June 1999.
19. VNC–Virtual Network Computing from AT&T Laboratories Cambridge,
wysiwyg://437/http://www.uk.research.att.com/vnc/.

BIOGRAPHICAL INFORMATION

ABUL K M AZAD

Obtained PhD (control engineering) from the University of Sheffield (UK) in 1994. He is now an Assistant Professor with the Engineering Technology Department of NIU. Dr Azad has over 50 papers in this area and is active with professional bodies. His current teaching and research interest includes digital electronics, mechatronics, intelligent control and real-time computer control of engineering systems.

ANDREW W. OTIENO

Received his Ph.D. from Leeds University, UK in 1994 and has been at Northern Illinois University since August 2000. His research is in the area of finite element modelling, machining processes, tool wear monitoring and structural health monitoring. He has experience in hardware/software interfacing with special applications in machine vision. He is a member of the ASEE and the SME.

OMAR GHRAYEB

Received his Ph.D. in Industrial Engineering from New Mexico State University, NM in 2000. He is an Assistant Professor in the Department of Industrial Engineering at Northern Illinois University. His research interests are in Production Planning and Scheduling, Design and Analysis of Manufacturing Systems, and Soft Computing applied to manufacturing systems.

NAVIN ANAND

Navin is presently a graduate student in the Department of Electrical Engineering at Northern Illinois University. He graduated from the Department of Technology at Northern Illinois University with an MS in Industrial Management in December 2002. His scholarly interests are in communications and software development.