Int. J. Engng Ed. Vol. 18, No. 5, pp. 557–565, 2002
Printed in Great Britain.

0949-149X/91 $3.00+0.00
© 2002 TEMPUS Publications.

Integrating Nanotechnology into Undergraduate Experience: A Web-based Approach*

TIMOTHY CHANG, PUTTIPHONG JAROONSIRIPHAN
Department of Electrical & Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102, USA. E-mail: changtn@adm.njit.edu

XUEMEI SUN
Software Center, Motorola Inc., 37101 Corporate Drive, Farmington Hills, MI 48331, USA

This paper describes a web-based distance experiment on the control of a two-degrees-of-freedom (DOF) monolithic piezoelectric actuator. This actuator is part of a six-DOF manipulator [1] capable of linear resolution to 2 nanometers and angular resolution to 1 arc-second [2]. The LabVIEW/DataSocket approach has been adopted in this implementation to enhance streaming of live data as well as to facilitate the interaction among the various system modules such as PC platform, digital signal processor board, client/server interface, and the nanopositioner. Besides a LabVIEW-based client program which requires the remote PC to have LabVIEW installation, a Web browser-based client plug-in has also been generated for students/users who do not have programming experience with LabVIEW. Test results for multi-client access have been very favorable. Reliable real-time control bandwidth of the two-DOF nanopositioner was detected at about 3 KHz, adequate for undergraduate control experiments. The same client/server interface can also be applied to a wide range of experiments for 24/7 and global access.

INTRODUCTION

IT IS WELL RECOGNIZED that nanotechnology is a strategically important area with significant growth potential. Therefore, it is of educational interest to expand the study of nanotechnology from advanced graduate level to senior undergraduate level. The challenge with offering nanotechnology to undergraduate students is twofold:

- the course content/curriculum design;
- experimental issues.

It has been observed that an additional experimental component in a senior/graduate engineering lecture course is highly beneficial to promoting learning and student interest [3]. However, nanotechnology experiments are delicate, limited in availability, and expensive to set up and maintain. The use of a web-based approach circumvents these drawbacks and enables the experiment to be run securely, safely, and on a 24/7 basis.

In this paper, the experience in offering a nanopositioner control experiment to a senior undergraduate class is reported. The nanopositioner is a monolithic piezoelectric device capable of six degrees-of-freedom and resolution to about 1 nm. The experimental unit has two DOF (x and y) and is driven by two high voltage amplifiers (± 550 Volts) from a TMS320C31-based digital signal processor board. Two capacitance sensors are used to detect the positioner displacement for frequencies up to 3 kHz. Concerns for safety of the students (since high voltage drives are used) and the hardware (delicate and expensive) have been the main reasons for restricting direct student access. In this project, the students access the experiment through LabVIEW DataSocket. The server computer houses the TMS320C31 digital signal processor board and is responsible for client interface and running of the DSP functions such as code compilation, execution, and data transfer. Since the client access is restricted to code operation in the DSP, security of the PC is increased. Furthermore, a number of safeguards such as actuation level, slew rate, etc. have been set in to minimize the possible damage to the experiment.

DESCRIPTION OF THE 2-DOF ACTUATOR

The nanopositioner described in this paper is a monolithic design with very low component count and therefore, can be constructed with significantly lower cost. Applications of this actuator include: optoelectronics assembly, optical fiber alignment, and nanomaterial processing. Preliminary test results from the 2-DOF prototypes indicate the following performance values:

* Accepted 1 March 2002.

Resolution	2 nm
Rise time	1.5 μm/ms
Range	50 μm
Load	1 kg
Input voltage	500 V
Angular coupling between x, y axes	$< 1^\circ$
Resonance	> 500 Hz

Referring to Fig. 1, the 2-DOF actuator consists of an interlaced stack of monolithic PZT cruciforms and inert baseplates. Motions in the x and y-plane are generated by applying voltages to the electrodes on the PZT cruciforms. The actuator is constructed from piezoelectric ceramic such as lead zirconate/lead titanate (PZT or LZT) which exhibits microstrain upon proper poling and electrical excitation. With its monolithic structure, a total of three DOF can be designed into a single cruciform actuator (x, y, and theta axes).

The static displacement of an actuator with length L, width W and thickness T varies according to the following formulas:

$$\Delta x = \frac{d_{31} L}{T} n_b V \tag{1}$$

$$\Delta y = \frac{d_{31} W}{T} n_b V \tag{2}$$

where the piezoelectric constant d_{31} is expressed in m/V. Δx, Δy are the relative displacement in the x, y directions and V is the applied voltage. n_b is the number of PZT cruciforms. It should be mentioned that the cruciform operates in the d_{31} mode so stacking of elements must be interlaced with an inert baseplate to enable addition of the individual displacements. The theta angular motion is realized by etching proper electrode patterns on the cruciform [1]. The pressure produced by the deformation is given by:

$$P_x = P_y = \frac{d_{31} Y_{11}^E}{T} V \tag{3}$$

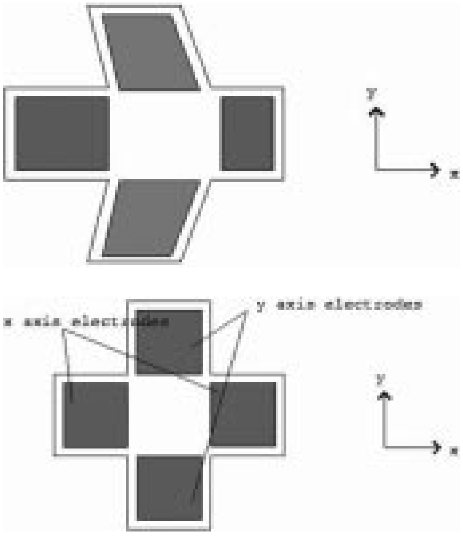


Fig. 1. Cruciform baseplate and its motion.

where Y_{11}^E is the Young's modulus of the material. With soft ceramic, typical values of d_{31} and Y_{11}^E are: $-250 \times 10^{-12}\,\mathrm{m/V}$ and $6 \times 10^{10}\,\mathrm{N/m^2}$. Maximum applied voltage is about 1 kV per millimeter thickness. An actuator constructed from ten 1 cm side and 0.5 mm thick baseplates produces a range of 25 microns and a force of 75 N under 500 V peak excitation in the x, y plane. Figure 1 illustrates the geometry of a single piezoelectric baseplate and its corresponding x-axis motion under differential voltage excitation at its electrodes. Similar motion can be generated for y-axis.

The corresponding minimal order analytic realization, modeling dynamics up to the first resonance, is given by Equations (4) and (5):

$$\begin{aligned} &\frac{d^2x}{dt^2} + 1.0502 \times 10^3 \frac{dx}{dt} + (1.3511 \times 10^7)x \\ &\qquad - (9.4572 \times 10^5)x^2 + (4.4607 \times 10^4)x^3 \\ &\quad = (1.3984 \times 10^5)u_x \end{aligned} \tag{4}$$

$$\begin{aligned} &\frac{d^2y}{dt^2} + 7.742 \times 10^2 \frac{dy}{dt} + (1.0572 \times 10^7)y \\ &\qquad - (7.4200 \times 10^5)y^2 + (2.9927 \times 10^4)y^3 \\ &\quad = (1.4244 \times 10^5)u_y \end{aligned} \tag{5}$$

where x, y are displacements expressed in microns and u_x, u_y are drive-voltages on the PZT. It is observed that the first resonances for the x and y-axes occur at 585 Hz and 518 Hz, respectively. These resonances have a Q factor of about 5–7. Characteristics of the two axes are quite similar with low mutual interaction due to the orthogonal structure. Test results indicate that as each axis is traversed through its full range of motion, the orthogonal member undergoes less than 1° of misalignment. The difference in the resonant frequencies is caused by fabrication tolerance and the way the capacitance probe target is mounted.

Experimental setup of the test system is shown in Fig. 2. It consists of the following: two TREK high voltage amplifiers, two ADP capacitance sensors, a TMS320C31-based digital signal processor board with four, 14-bit input channels and two 12-bit output channels.

WEB ACCESS WITH DATASOCKET

Prior to using the DataSocket platform, a number of web access software solutions were considered and implemented: CGI/Perl, Java, TCP/IP. For example, a server applet has been generated to handle upload and setting experimental parameters. A client applet handles client-side user interface and deciphers input settings [4]. This Java-based approach requires a great deal of customization and software development and therefore, is quite unsuitable for undergraduate education for which ease of use and attractiveness of the GUI are important. Hence, a new software

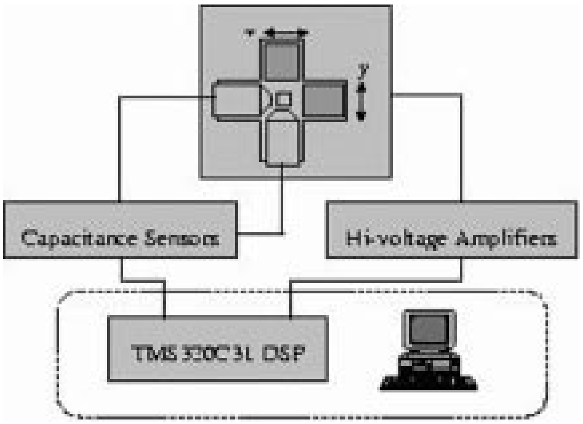


Fig. 2. Experiment setup.

platform is necessary and after extensive search, DataSocket is chosen.

The National Instruments' LabVIEW (Laboratory Virtual Instrument Engineering Workbench) has been a popular item of software for academic and industrial control experiments. It features a graphical user interface and drivers for controlling real-time processes. DataSocket, included with LabVIEW, is a new Internet programming technology that simplifies live data exchange among computers connected through a network [11–13]. DataSocket is designed specifically for sharing, subscribing, and publishing live data in measurement and automation applications. In particular, DataSocket is designed for live data transfer to multiple clients where a URL is used by users to connect to a data source location in the DataSocket server so as to control or share data with DataSocket applications. Although DataSocket handles connections to various types of data sources and existing schemes such as http, ftp, file, and opc, server communicates and exchanges data between two applications using the **dstp** (DataSocket Transfer Protocol).

The three components in a DataSocket subsystem are:

1. The DataSocket Server.
2. The writer (the digital signal processing board for this application).
3. The reader (remote client).

The advantages of maintaining the DataSocket Server on a separate machine other than on the writer or reader are increased robustness, scalability, and security. According to National Instruments [12], DataSocket simplifies network (TCP/IP) programming by automatically managing connections to clients and automatically converting measurement data, regardless of the types of data used or the source of the data, to and from the stream of bytes sent across the network. This automatic conversion eliminates the need to write complicated parsing code and network complexity. Therefore, users do not have to write the parsing code. Also, users can switch between data sources just by changing a simple URL. DataSocket is an ActiveX control; thus, users can use it to develop data-sharing applications in ActiveX containers such as Visual Basic, and Visual C++. This aspect simplifies the use of DataSocket to broadcast live data.

There are two modes of client operations under the DataSocket setup:

1. LabVIEW-based client program where a VI program runs within the LabVIEW environment in the client computer. This is the preferred mode for research activities as the user can interactively expand on the data processing capability by adding more VIs to the system block diagram. However, this mode of operation necessitates the client to have LabVIEW software and G-programming experience.
2. Web browser-based client program where a plugin to the web browser can be download by the user and the virtual instrument panels are activated within the web browser. This way, the user can operate the experiment in real time with minimum learning time. However, the interface configuration is fixed and cannot be modified. This is the preferred mode for undergraduate students who are more limited in time, resource, and programming experiences. This mode requires the experiment designer to create a DataSocket reader component (ActiveX control), using ComponentWorks and Visual Basic codes to manipulate those controls, and then insert this component into a web page.

Finally, to protect from undesirable users that could misuse this application, DataSocket server manager can handle this problem by configuring the DataSocket server setting. The server manager can specify which machines can create items, write items, or read items; including access permission, number of connections and predefined data items stored in the server. Users can connect a total of 1000 readers and writers to one DataSocket server. The overall system block diagram is given in Fig. 3.

SYNTHESIS OF DATASOCKET VI AND PLUGIN

To create DataSocket client web page, users have to create a DataSocket reader component with which users connect to the server. Users create that component with ActiveX controls and Visual Basic codes to manipulate those controls and save this component as an ActiveX control. After making the ActiveX control, it is plugged into a Web page.

The necessary software tools to build ActiveX control include:

- ComponentWorks 2.0 DataSocket and user interface (UI) controls.
- Visual Basic 5.0 or higher (Version 6.0 for this application).
- Recent versions of Netscape or Internet Explorer.

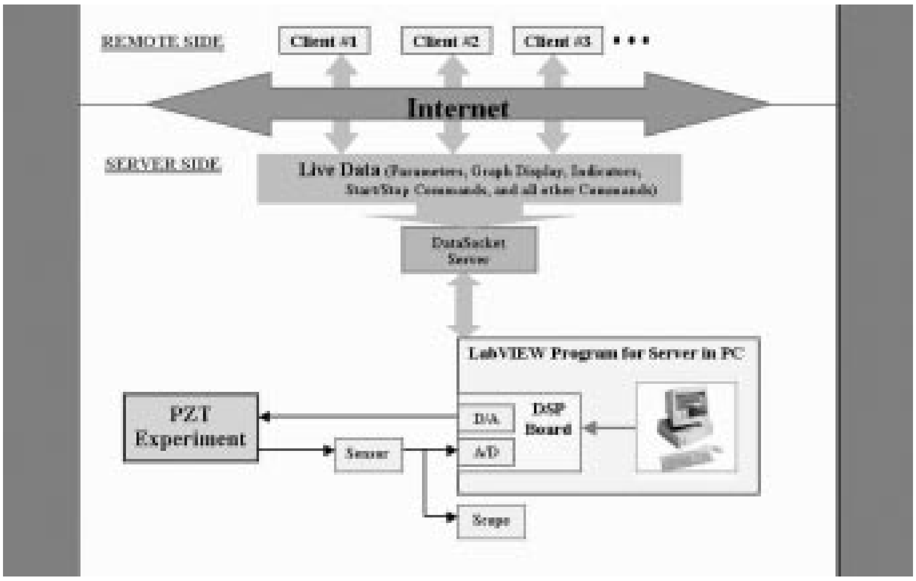

REMOTE SIDE
Client #1
Client #2
Client #3
• • •
Internet
SERVER SIDE
Live Data (Parameters, Graph Display, Indicators, Start/Stop Commands, and all other Commands)
DataSocket Server
LabVIEW Program for Server in PC
PZT Experiment
D/A
A/D
DSP Board
Sensor
Scope

Fig. 3. System block diagram.

The goal is to create the DSReader component using ActiveX controls in Visual Basic, and design a component that connects to the DataSocket Server, which reads live data from the server, and displays that data on a graph.

The procedures are listed below:

1. Loading ActiveX controls by opening ActiveX control, then selecting National Instruments CW DataSocket and National Instruments CW UI.
2. Place all control components including Graph, Control buttons, CWDataSockets, Textbox, etc. onto the control form.
3. Turn on the server station and load program to DSP board.
4. Disable all parameter-control buttons until the client presses 'Run Client'.
5. After a client presses 'Run Client' button, the CWDataSockets are automatically connected to DataSocket Server specified by IP address and item names, then the program does as the following:
   - CWDataSocket 1 is in the 'Write Mode' for writing and sending data (all data in attributes) to Server.
   - CWDataSocket 2 is in the 'ReadAutoUpdate Mode' for reading data (parameters) from Server.
   - CWDataSocket 3 is in the 'ReadAutoUpdate Mode' for reading data (graph data and all data in attributes) that are generated from server, and then these data are displayed on the client side.
   - CWDataSocket 4 is in the 'Write Mode' for sending Start/Stop command to control the server.
6. After all parameter-control buttons are enabled, clients can control the server with indicators and digital display shown during run time.
7. In case that many clients are logging on to the server at the same time, the first client will be absolutely the first priority to take control until he/she presses 'Stop Client'
8. As soon as, the first client presses 'Stop Client', all CWDataSockets are disconnected. This action causes the server to stop running and all the rest of clients become standby.
9. In case that, clients need to set priority to control server, all clients have to stop program and connect to the DataSocket Server again. Therefore, whoever logs on first will be the first priority to control the server.
10. In ComponentWorks application with Visual Basic, it is possible to publish or subscribe data through DataSocket Server directly without creating any Visual Basic codes. This property is called 'Bindings', which a programmer just identifies URL/item name and values/data to publish or subscribe for each component. The other component(s) communicating with that component have to be specified the same URL and item name. After the program is running, this handshaking will take effect automatically. Therefore, some procedures that publish or subscribe data to the server are not shown in the Visual Basic Code.
11. After creating Visual Basic codes, programmer has to generate the ActiveX control, whose file extension is .ocx
12. Creating the HTML file by using the Package and Deployment Wizard (Visual Basic 6.0) will generate an Internet distribution package, which contains a cabinet (.cab) file and the HTML file. The cabinet file contains information about the DSReader component, and it is automatically downloaded, expanded, and installed by Internet Explorer so that clients on other computers can view the program component with their Web browser.

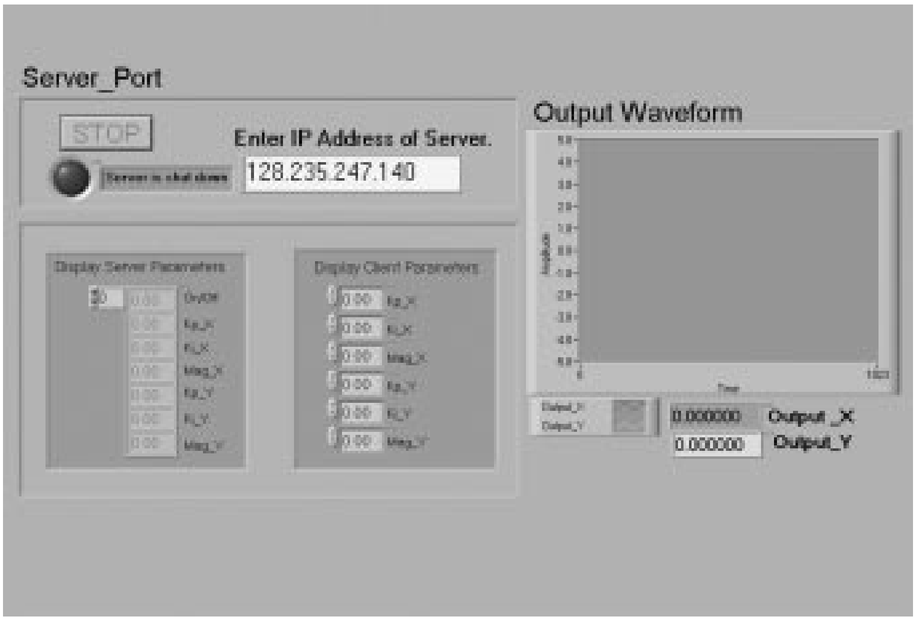


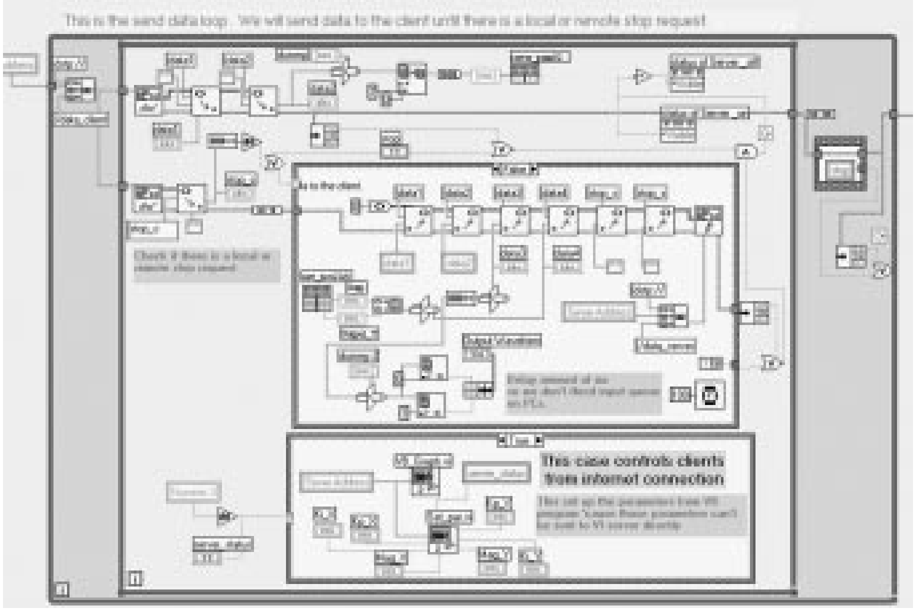
Fig. 4. Server socket panel.

13. There are three files associated with the Visual Basic project: .vbp (project file), .ctl (DSReader component), and .vbw (workspace file). The programmer needs these files to modify the program component in Visual Basic. After modifying the Visual Basic project, the programmer needs to remake the .ocx file, which saves the program component as an ActiveX control that can be embedded in an HTML (.htm) file.
14. After the programmer makes the .ocx file, he/she can delete previous .cab and .htm files and use either one of the Applications Setup Wizard or the Package and Deployment.

RESULTS AND FUTURE PLANS

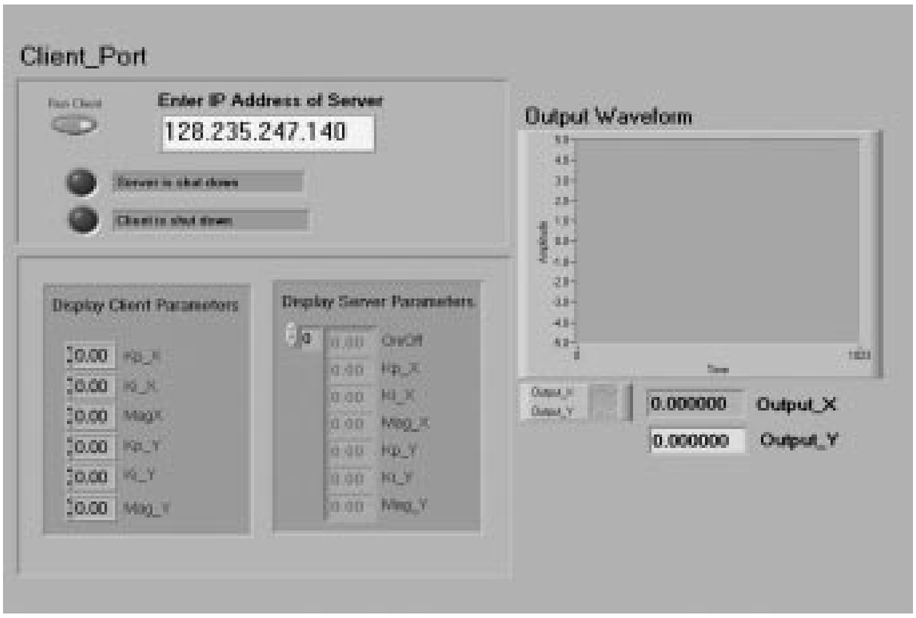
Some typical screens of the server socket panel and server socket block diagram are shown in Figs 4 and 5, respectively. The screen design uses standard VIs from LabVIEW and is completely reconfigurable. Even though the block diagram seems complex, it is capable of running at sufficient bandwidth. The client DataSocket panel (for LabVIEW users) and client DataSocket diagram are shown in Figs 6 and 7. This system is also completely reconfigurable as long as the interface structure is not altered.

Finally, for the non-LabVIEW users, the web



This is the send data loop. We will send data to the client until there is a local or remote stop request
This case controls clients from internet connection

Fig. 5. Server socket block diagram.



Client_Port

Run Client

Enter IP Address of Server

128.235.247.140

Server is shut down

Client is shut down

Display Client Parameters

0.00 Kp_X
0.00 Ki_X
0.00 MagX
0.00 Kp_Y
0.00 Ki_Y
0.00 Mag_Y

Display Server Parameters

0

0.00 On/Off
0.00 Kp_X
0.00 Ki_X
0.00 Mag_X
0.00 Kp_Y
0.00 Ki_Y
0.00 Mag_Y

Output Waveform

0.000000 Output_X

0.000000 Output_Y

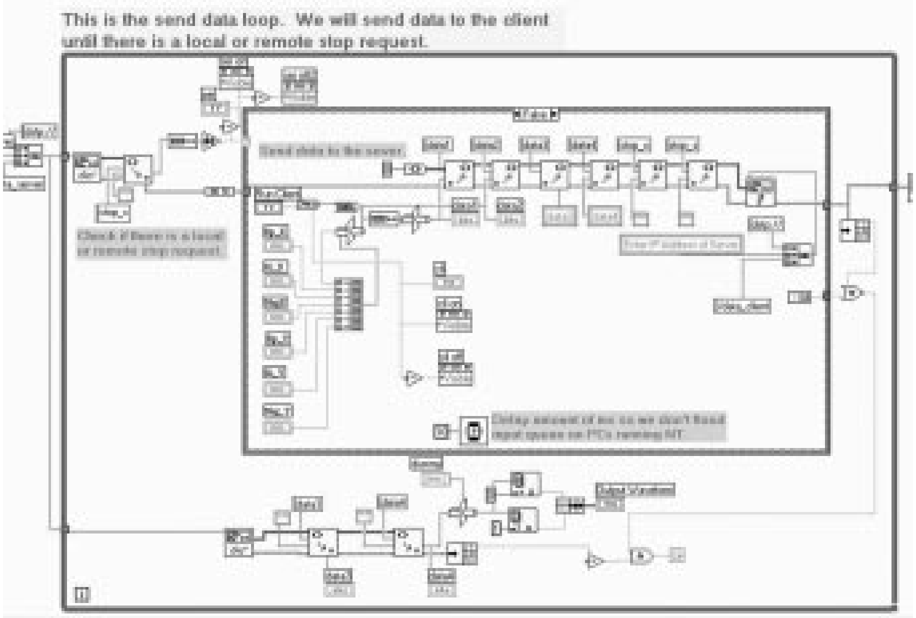
Fig. 6. Client socket panel.

browser based client is shown in Fig. 8. This interface is fixed and cannot be easily modified. This restriction is actually advantageous for undergraduate applications due to the large number of student-access and the high potential for 'hacking'.

Among the experiments performed on this system, the hysteresis compensation and resolution tests were the most successful and pedagogically effective. For the hysteresis test, the students experienced that in a nanoscale arena, nonlinearity such as hysteresis had a significant effect on the precision motion of the actuator. The uncompensated response is shown in Fig. 9 where it is noted that an inherent hysteretic error of a few microns is present. As shown in Fig. 10, proper feedback compensation had virtually eliminated the error.

For the resolution test, fine motion of the actuator under closed loop control is assessed by a series of three linearity tests where the actuator steady-state outputs are measured under different excitation ranges and step sizes. The ranges of excitation are 0.25 μm, 0.1 μm, and 0.025 μm for the three tests. In each test, the actuator is commanded to move between the maximum and minimum limits of the range with a predefined step size (0.025, 0.01, and 0.0025 μm for the three tests). The data are averaged to yield the DC bias.

Typical test data corresponding to the



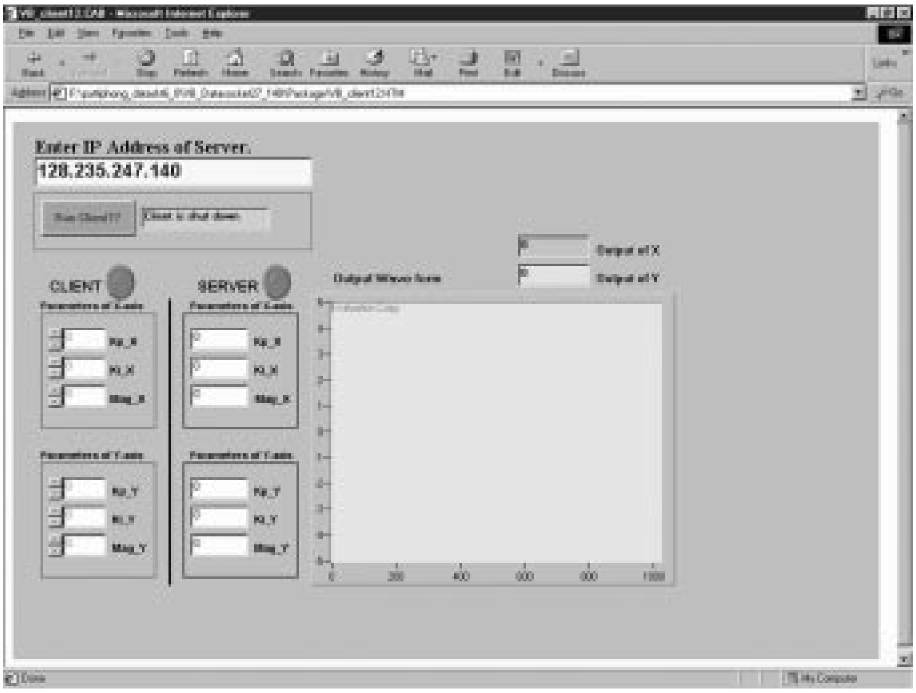
This is the send data loop. We will send data to the client until there is a local or remote stop request.

Check if there is a local or remote stop request.

Send data to the server.

Delay amount of ms so we don't flood input queues on PCs running NT.

Fig. 7. Client socket block diagram.



Enter IP Address of Server.
128.235.247.140

Run Client??
Client is shut down

Fig. 8. Client web panel.

0.0025 μm step size are plotted in Fig. 11 where the left and right plots represent the response of the x and y axes, respectively. It is observed that the controlled output is highly linear, with no trace of hysteresis. In Fig. 11, slight distortion of the scale factor is noticed. This distortion is caused by ambient noise and sensor resolution.

The web experiments are now a standard component of the undergraduate course 'EE482–Instrumentation and Control'. This course deals with analysis of dynamic systems using frequency and time domain methods. Both linear and nonlinear characteristics are introduced. Until the recent revisions, this course tends to be highly mathematical and lacks 'real-life experience'. The use of web experiments has significantly improved interest level and learning of key concepts. The nanopositioner, with its resonant and hysteretic characteristics, serves as an excellent experiment to illustrate the macroscale dynamic properties as well as the nanoscale phenomena.

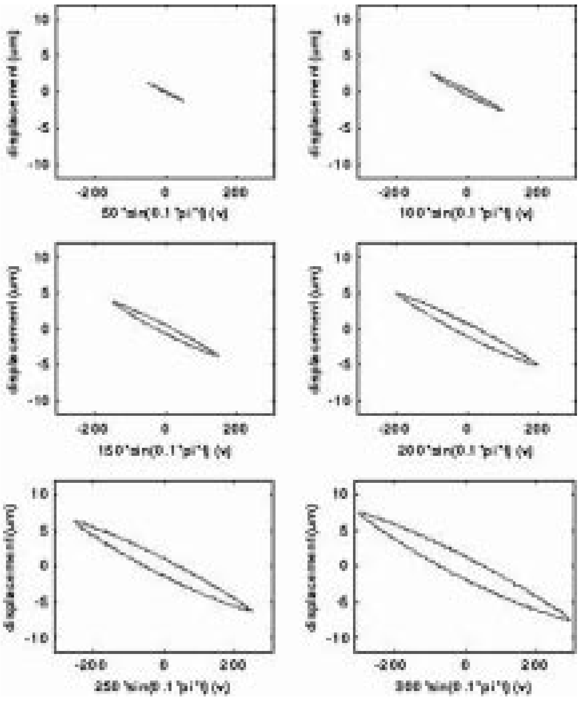


Fig. 9. Hysteresis characteristics of the uncompensated actuator.

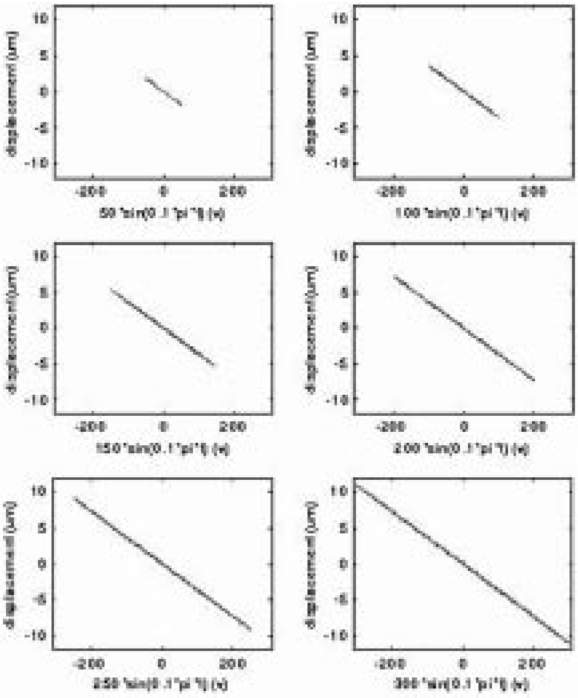


Fig. 10. Hysteresis characteristics of the compensated actuator.

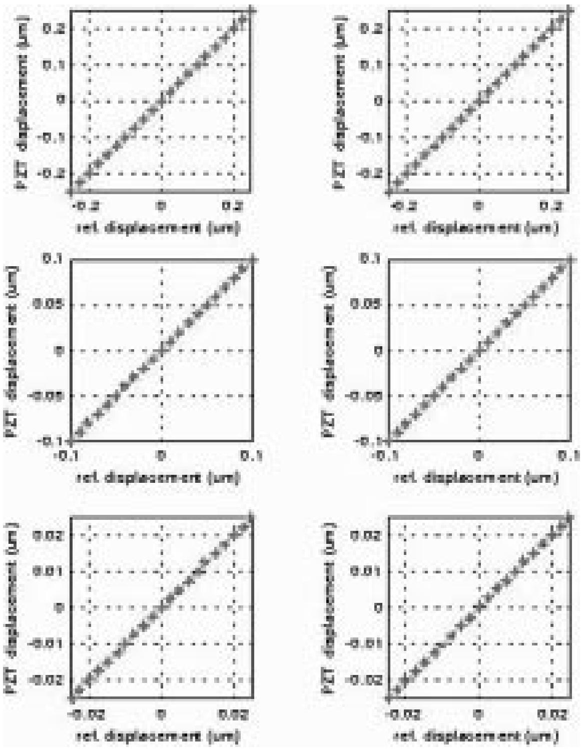


Fig. 11. Resolution test with x-axis (left) and y-axis (right).

An outline of the course is shown below:

- Introduction to Systems, Review of LaPlace Transform.
- Transfer Functions, Signal Flow Graphs, Stability.
- Frequency Response of Linear Systems: Bode and Nyquist.
- Identification of Dynamic Systems.
- Performance Characteristics and Feedback; Case study: positioners, actuators, and sensors.
- Properties and Solution of State-Space Systems.
- Modeling of Physical/Biological/Nano Processes, Linearization.
- Stability Assessment: Routh-Hurwitz.
- Root Locus Method.
- The Nyquist Stability Criterion.
- Describing Functions and Limit Cycles.
- Application of Describing Functions.

The nanopositioner should be introduced towards the end of the semester so that the students can be fully prepared to handle the analytic portion of the experiment.

Other plans to utilize the web based nanopositioner include:

- Engage national/international collaboration on nanopositioner research and applications. For example, application to DNA microarray expression analysis has been initiated with the Center For Applied Genomics, Public Health and Research Institute.
- Introduce the nanopositioner experiment along with other robotics experiments to high school students under the NJ I-Tower grant supported by the New Jersey Department of Education.

Acknowledgment—This work is supported, in part, by an NIST ATP Contract No. 70NANB5H1092 Precision Optoelectronics Assembly Consortium and New Jersey Information-Technology Opportunities for the Workforce, Education, and Research (NJ I-TOWER).

CONCLUSIONS

In this paper, web-based control of a two-degrees-of-freedom nanopositioner is described. The device has a monolithic construction—both the x and y-axes reside on the same PZT ceramic so that a number of advantages can be achieved: high bandwidth, high resolution, and load capacity while minimizing production cost. However, this experiment is delicate and the high voltage drives pose a potential safety problem. For these reasons, undergraduate access has been excluded in the past. The use of web-based control has circumvented these drawbacks and enabled the experiment to be run securely, safely, and on a 24/7 basis. The key enabling technology is the LabVIEW/DataSocket package which provides the standardized interface for live data streaming and multi-user access. Furthermore, along with ComponentWorks, a web browser plugin has been generated, allowing non-LabVIEW users to run the nanopositioner experiment remotely. Test results indicate that sufficient control and data exchange bandwidth can be achieved for multi-user access. From the class survey, student response to the experiments had been highly favorable and therefore more experiments will be deployed for the 2001 Fall semester.

REFERENCES

1. T. Chang, Multiple degree of freedom piezoelectric actuator, U.S. Patent pending, 1998.
2. T. N. Chang and X. Sun, Analysis and control of monolithic piezoelectric nano-actuator, *IEEE Trans. Control Systems Technology, Special Issue on Smart Materials*, January, 2001.
3. T. N. Chang and D. Chang, Graduate engineering student performance assessment: how learning pattern affects test scores, *Proc. ASEE2000*, St. Louis, MO, June 18–21, 2000.
4. T. N Chang and D. C. Hung, Web-based distance experiments: design and implementation, *Proc. Int. Conf. Engineering Education*, Taiwan, August 14–18, 2000.
5. F. Charette, A. Berry, and C. Guigou, Dynamic effects of piezoelectric actuators on the vibrational response of a plate, *J. Intelligent Material Systems and Structures*, pp. 513–524, 1997.
6. G. J. Knowles and J. Murray, Foundations of piezoelectronic theory for distributed parameter systems, *Proc. SPIE—The International Society for Optical Engineering Smart Structures and Materials 1997: Mathematics and Control in Smart Structures*, pp. 324–334, 1997.

7. M. Goldfarb and N. Celanovic, Modeling piezoelectric stack actuators for control of micromanipulation, *IEEE Control Systems Magazine*, pp. 69–79, 1997.
8. T. Alberts, T. DuBois, and H. Pota, Experimental verification of transfer functions for a slewing piezoelectric laminate beam, *Control Engineering Practice*, pp. 163–170, 1995.
9. S. Agrawal, D. Tong, and K. Nagaraja, Modeling and shape control of piezoelectric actuator embedded elastic plates, *J. Intelligent Material Systems and Structures*, pp. 514–521, 1994.
10. C. Won, D. Sparks, K. Belvin, and J. Sulla, Application of piezoelectric devices to vibration suppression: from modeling and controller designs to implementation, *AIAA Guidance, Navigation and Control Conference*, p. 1381, 1992.
11. H. Edwards, Building an interactive web page with datasocket, *Application Note 127*, June 1999, Part Number 341572B-01, National Instruments.
12. H. Edwards, Connecting measurement studio user interface ActiveX controls to remote data, *Application Note 151*, November 2000, Part Number 341972C-01, National Instruments.
13. Integration of the Internet into your measurement system, *DataSocket Technical Overview*, March 1999, Part Number 341680A-01, National Instruments.

Timothy Chang received his Bachelor of Engineering degree from McGill University with distinction. He received his M.A.Sc. and Ph.D. degrees in Electrical Engineering from University of Toronto. He is currently an Associate Professor of Electrical & Computer Engineering at New Jersey Institute of Technology (NJIT) and the Coordinator of the Intelligent Systems Area. Prior to joining NJIT in 1991, Dr. Chang was a Senior Research Specialist and Program Manager at Kearfott Guidance & Navigation Corp. He has also served as an engineering consultant for a number of companies. Dr. Chang's current research interests include: nano-scale precision systems, mechatronics, decentralized and distributed systems, embedded systems, robotics, and education research methods.

Puttiphong Jaroonsiriphan is currently a Ph.D. candidate at the Department of Electrical & Computer Engineering at New Jersey Institute of Technology (NJIT). He received his BS. Eng degree from Kasetsart University, Bangkok, Thailand and MS Eng. degree from NJIT, in 1994 and 1998 respectively. After graduation, he worked for Siam Iron and Steel Co., Ltd, Thailand as a Design and Control engineer, in charge of the revamp of a ladle car project for increasing efficiency of plant line. After he received his MS Eng. degree from NJIT, he worked for Phoenix Lighting Inc., which collaborated with other lighting companies to upgrade lighting for shopping malls and government buildings.

Xuemei Sun obtained Master of Science and Bachelor of Engineering degrees in Electrical and Computer Engineering from Chengdu University of Science and Technology in 1992 and 1989, and her Ph.D. degree in Electrical and Computer Engineering from New Jersey Institute of Technology in January 2001. At present, she is a software engineer at Motorola Inc. in Michigan, and specializes on embedded system control.