

Web-Based Distance Experiments for Real Time Control

Timothy Chang and Puttiphong Jaroonsiriphan

Department of Electrical & Computer Engineering

New Jersey Institute of Technology, 07102

Keywords: Web Experiment, Real Time Control, Telepresence

Abstract: This work focuses on the design and implementation of web-experiments based on the LabVIEW/Datasocket approach 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 experimental hardware. Besides the LabVIEW based client program which requires the remote PC to have LabVIEW installation, a web-browser based client plugin has also been generated for students/users who do not have programming experience with LabVIEW. Test results for multi-client access have been favorable. Reliable real time control bandwidth was detected at about 3KHz, adequate for undergraduate control experiments. The same client/server interface can also be applied to a wide range of experiments for 24/7 and multi-users global access. Other advantages of web-based experiment include: improved safety conditions, reduced cost, and broad base access for all students (full/part time, handicapped). Examples include identification of dynamic systems and DC motor control. Student response and implementation experience are also described.

(I) Introduction

It is well established that hands-on experience can significantly improve student learning and interest level in the course materials [1]. This is especially important in courses that are theoretical with a high degree of mathematical analysis. However, a number of constraints prevent the broad base integration of experiments into engineering courses. For examples, laboratory facilities/hours, safety issues, and software development effort are a few of the limiting factors that most instructors would encounter. With the increase in bandwidth through high speed and ISDN data lines, web-based distance experiments can play a significant role in supporting the learning process by providing a 24/7 laboratory with real time experiments so that the students can draw realistic conclusions and gain insight for real-life problem solving.

A number of web-based distance experiments are already in place. Earlier systems [2], [3] rely on custom software development e.g. cgi/Perl, Java/C++, etc. Maintenance and upgrade of the codes become problematic. Furthermore, most earlier system lack real-time streaming capabilities due to the client side control privilege. The more recent implementations of the web experiments are based on existing software platforms such as LabVIEW or Matlab. In [4], [5], and [6], various web experiments are supported by the

Wincon/Simulink combination package through TCP/IP protocol. In [7], distance experiments in process control and dynamics are made available via LabVIEW interface. In [9], monitoring of distributed processes are carried out on LabVIEW TCP/IP VI (Virtual Instrumentation) using the HTTP web server.

While these approaches are meritorious and work well for the intended applications, it is the objective of this work to devise a general-purpose, distance experimental system that possesses the following characteristics:

1. Efficient interactive/multi-user operations.
2. Machine independent (currently developed for the PC platform).
3. Secure operations.
4. Graphics User Interface compatible.
5. High processing bandwidth with true real-time data streaming capabilities.
6. Low cost, user friendly interface.
7. Global resource sharing:
 - Intra/inter-campus
 - Inter-university
 - K-12/university
 - International

Hence, a new software package is necessary and LabVIEW/ Datasocket is chosen to support the implementation of web experiments. A description of Datasocket is given in the following section.

(II) Web Access with Datasocket

The National Instruments' LabVIEW (Laboratory Virtual Instrument Engineering Workbench) has been a popular software for academic and industrial control experiments. It features graphics 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], [12]. It 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, the 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 analog I/O board and the digital signal processing board), and 3. the reader (remote client). The advantages of being able to maintain the DataSocket Server on a separate machine other than on the writer or reader are increased robustness, scalability, and security. According to National Instruments [11], [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 Virtual Instrumentation (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 downloaded 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. Users can specify which machines can create items, write items, or read items; including access permission, number of connection and predefined data items stored in the server. Users can connect a total of 1000 readers and writers to one DataSocket Server.

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:

1. ComponentWorks 2.0 DataSocket and User Interface (UI) controls
2. Visual Basic 5.0 or higher (I use version 6.0)
3. Recent versions of Netscape or Internet Explorer.

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 procedure is given in [10], [11], [12].

In summary, advantages of the Datasocket approach include:

- Increased robustness, scalability, and security.
- Handles connections to various types of data sources and existing schemes such as http, ftp, file, and opc.
- Automatically managing connections to clients and automatically converting measurement data.
- Users can switch between data sources by changing a simple URL.
- Users can develop data-sharing applications in ActiveX containers such as Visual Basic, and Visual C++.
- Accommodates three modes of operations:
 - Virtual experiments: simulated experiments on server
 - Direct data acquisition and control: PC runs signal processing programs and processes data
 - Indirect data acquisition and control: PC serves as a local host to a digital signal processing system which crunches the data at very high speed.

(III) Course Description

Web experiments were introduced into the senior level course “EE482 Instrumentation and Control” in the 2001 academic year and has since been an integral part of this required course. The old curriculum focuses extensively on classical frequency response methods such as complex variables, frequency response methods (Bode, Nyquist, Nichols), stability assessment techniques (Routh-Hurwitz, root locus), performance criteria (sensitivity, steady accuracy, transient response), and compensation (lag, lead). Although it may be argued that the course contents possess educational values, it is generally agreed that significant revision is necessary to reflect the change in technologies and modern engineering career challenges. In particular, computer-aided analysis, state space methods, and nonlinear systems are introduced into the curriculum, replacing lag, lead compensation, Nichols chart, and parts of the performance criteria. That is, the emphasis of the course is on analysis rather than control design which is relegated to a second course “EE486 Control Systems Electives”. An outline of the lecture 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
- Review and Project Presentation

The revised contents can potentially be highly mathematical and run into the same problems of being disconnected from the real world. Therefore, a series of web experiments are introduced to complement the lecture materials. The students form teams of two to tackle the experiments. For the “prime time” period (9am-9pm), the students are required to sign up for a weekly schedule while the off hours (9pm-9am) are available on a first come first serve basis with a maximum 60 minutes time allocation. Each team has a unique ID and password for access control purpose.

(IV) Web Experiments

To participate in the web experiments, the students connect their PCs to the server located in the NJIT Real-time Control Lab by using Web Page Plug-In and applying IP address of the server. The students (client) connect to the server containing DataSocket reader and writer components that automatically control and display the real time data. This way, the students can control and monitor experiment in real-time and enjoy the full benefits of telepresence. In terms of connectivity requirements, the students need to have Internet access either through LAN or WAN. For off-campus access (e.g. dial-up or DSL) the students must have a valid NJIT account and a VPN (Virtual Private Network) program to get across the firewall.

A description of the web experiments is now given:

1. **First and Second Order system identification:** This experiment deals with the characteristics of first and second order systems. As such, the experiment is a simulation to focus on the linear dynamics and to get the students ready for the second web experiment on DC motor control. A simulation loop is set up on the server and the students are to identify the open loop transfer functions (first and second order) by means of the input-output data. The system block diagram is shown in Figure 1. The client (student) and server screen captures are shown in Figures 2 and 3, respectively. After choosing the first or the second order system, the students then specify the type of inputs to the system as well as the loop gain in order to establish the closed loop characteristics of the system. When students are satisfied with the results, they can save the data by selecting the files to save in the local drive. The results are then analyzed to derive the open loop transfer functions. This experiment is introduced around the fifth week of the course, after coverage of the topic “Identification of dynamic systems”.

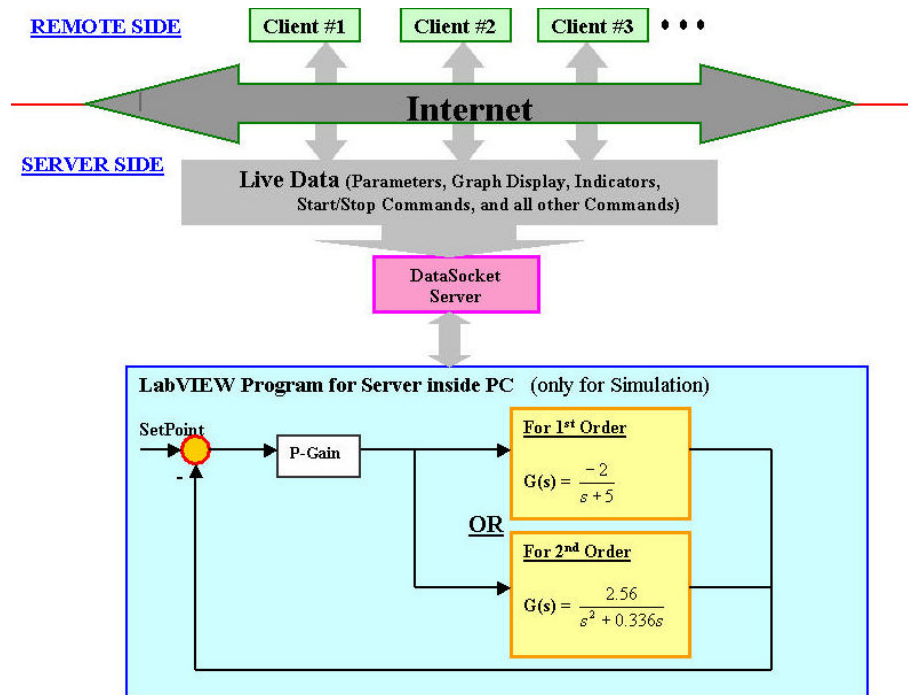


Figure 1 Web experiment 1: first and second order systems identification

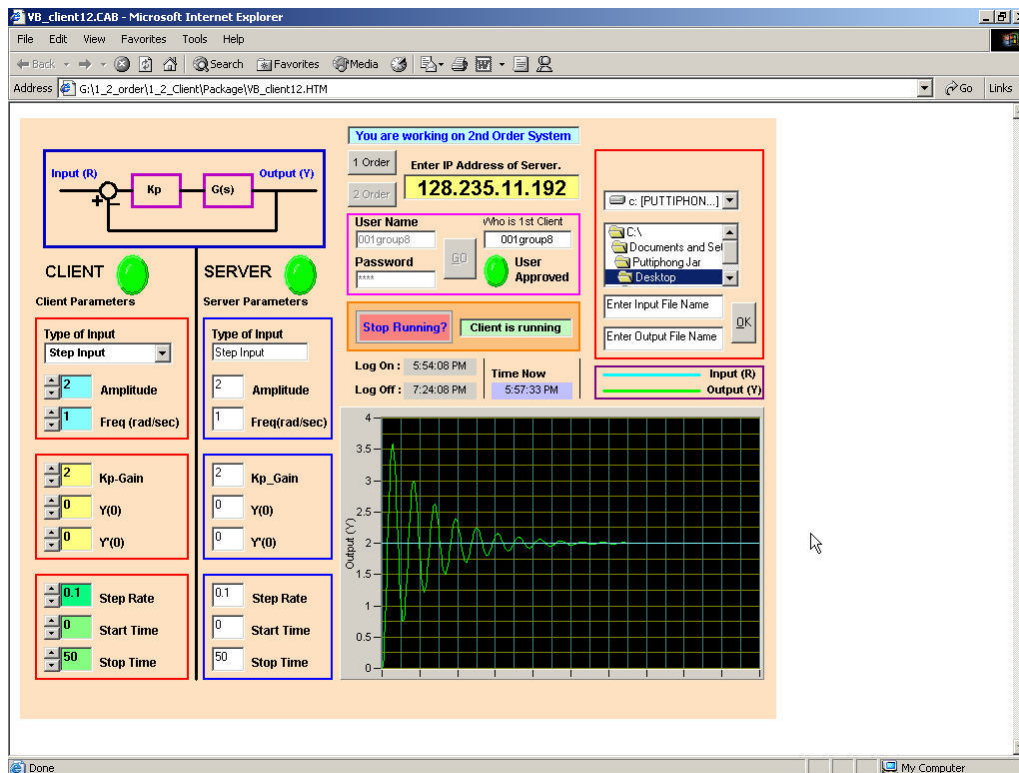


Figure 2 Web experiment 1 client screen capture

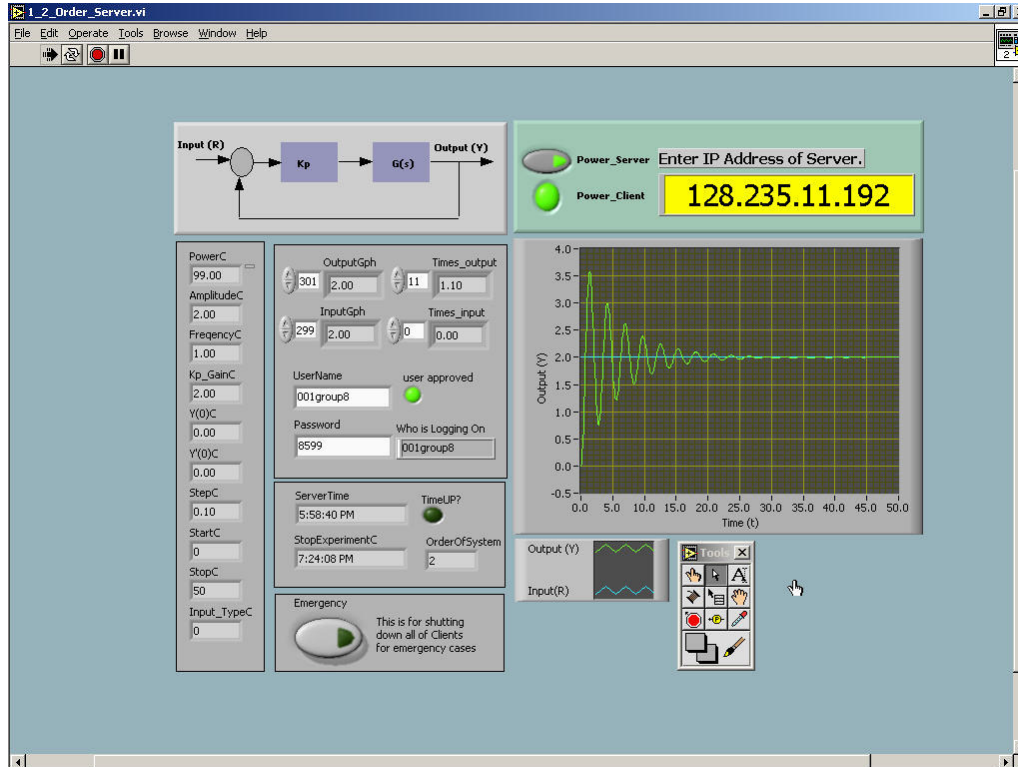


Figure 3 Web experiment 1 server screen capture

2. **Position Control for DC Motor.** This experiment demonstrates motor control and the effects of the non-linearity present in a mechanical system. The set up consist of a standard DC motor/potentiometer/amplifier combination as shown in Figure 4. Unlike the ideal linear systems of experiment 1, the responses of non-linear system under proportional control does possess non-zero steady state error and exhibits limit cycles resulting from the dead zone and parasitic dynamics. The dead zone represents frictions in the motor bearings and has been broadened to increase the observable effects. The students are also provided with the corresponding simulation block diagram as shown in Figure 5. The goals of this experiment are (1) to determine the motor parameters and dead zone characteristics through a combination of analysis and simulation and (2) compare the response of the nonlinear system with a linear plant. Screen capture of the client program is shown in Figure 6. This experiment is introduced around the tenth week of the course, after topics such as performance criteria and nonlinearities are introduced. It should be noted again that the primary goal of this experiment is to introduce the students to the nonlinear characteristics and the process of identifying the nonlinear plant. Control of the motor through more advanced control will be discussed in a follow-up course EE486 Control Systems Electives.

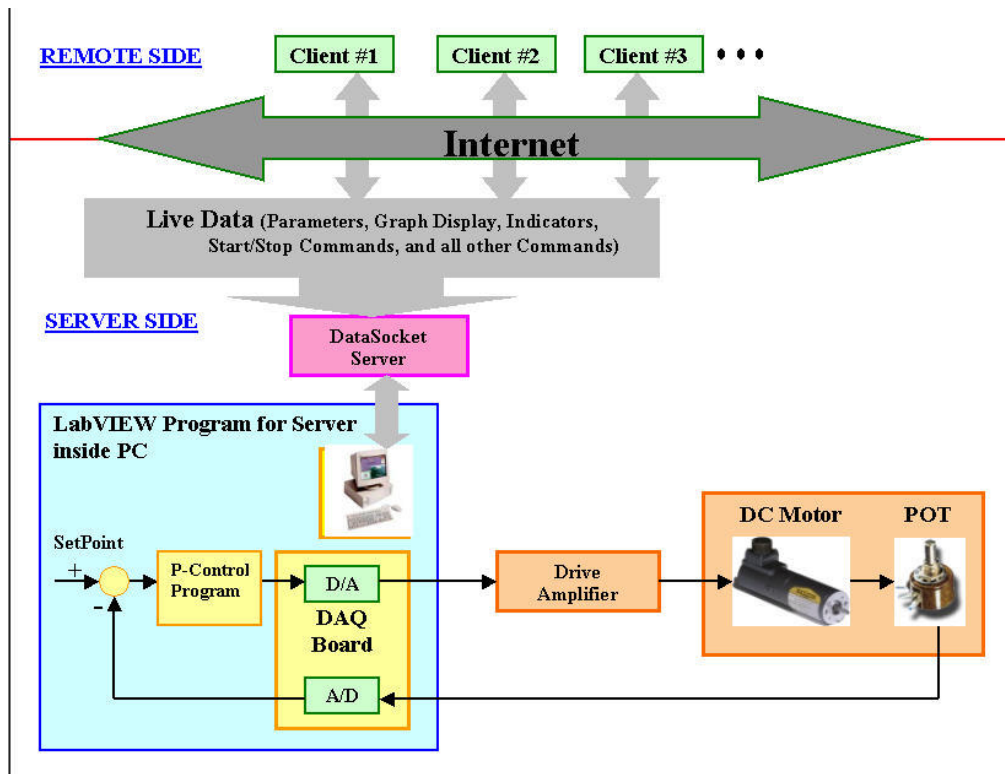


Figure 4 Web experiment 2: DC motor control

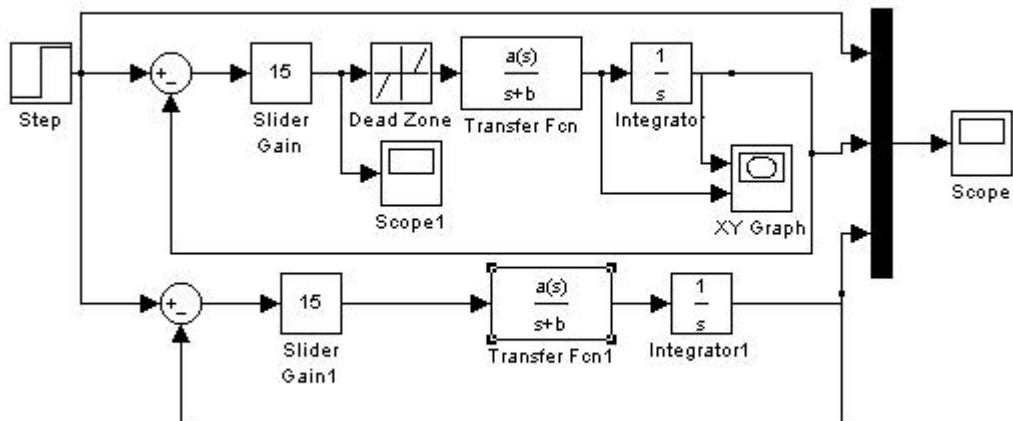


Figure 5 Simulation block diagram for the DC motor experiment

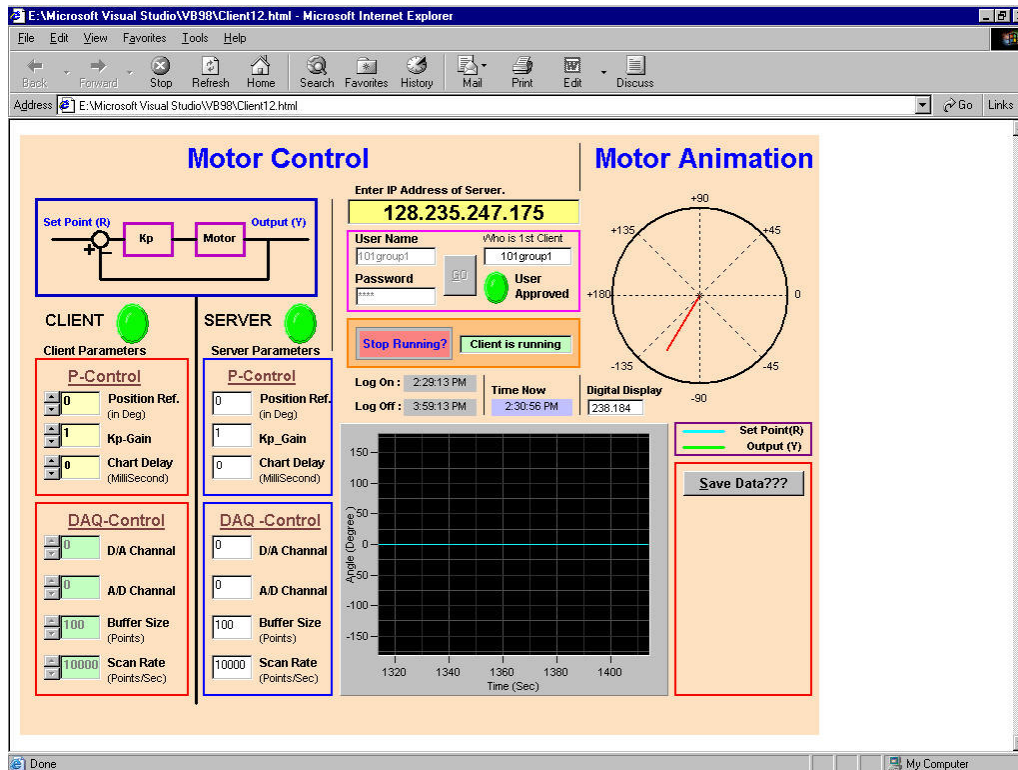


Figure 6 Web experiment 2: Client screen capture

(V) Results and Future Plans

For Experiment 1, the concept is relatively simple and there are relatively few surprises. Most students are able to come up with the open loop transfer functions using either step response (most popular) or frequency response methods. The identification experiment serves as a summary of linear systems characteristics. However, unlike the traditional approach of synthesizing the output according to the plant model, identification requires a higher level of understanding of system properties and the effects of feedback on input-output relationship.

For Experiment 2, the DC motor control system, the situation is different. Most students have difficulties initially in dealing with the system because of a number of factors: (1) the experiment is not 100% time-invariant/repeatable, (2) the response curves exhibit severe nonlinear behavior such as limit cycle and “uncertain” steady state values. Nevertheless, the lecture materials and simulation block diagram were helpful in helping the students to “sort out” the nonlinear dynamics.

Overall, the student experience has been very positive and the stated goals of strengthening learning and enhancing interest level have been achieved. However, further development is necessary to fully optimize the experiments.

- Introduce more web experiment modules
- Improve user communication e.g. via Whiteboard and/or ALN.
- Add streaming video, audio, and more beta sites

(VI) Conclusions

In this paper, web-based distance experiments for real time control and signal processing is described. The experiments are introduced to complement the lectures of a senior level course on instrumentation and control. The use of web-based control has enabled the experiments to be run securely, safely, and on a 24/7 basis. The key enabling technology is the LabVIEW/Datasocket package that enables the experiment designers to generate a web browser plugin for non-LabVIEW users to run the experiments remotely. Student feedback and observation confirm the effectiveness of the web experiments in promoting learning of key concepts and interest level in the subject materials.

(VII) References

- [1] Chang, T.N., Chang, D., "Graduate Engineering Student Performance Assessment: How learning pattern affects test scores," Proceedings of the ASEE2000, St. Louis, Mo, June 18-21, 2000.
- [2] Aktan, B., Bohus, C.A., Crawl, L.A., Shor, M.H, "Distance learning applied to control engineering laboratories," IEEE Transactions on Education, Vol. 39, pp 320 – 326, Aug. 1996.
- [3] Shen, H. Xu, Z., Kristiansen, V., Strom, O, and Shur, M, "Conducting Laboratory Experiments over the Internet," IEEE Transaction on Education, Vol42, pp. 180-185, August, 1999.
- [4] Exel, M., Gentil, S., Michau, F., Rey, D., "Simulation workshop and remote laboratory: two web-based training approaches for control," Proceedings of the American Control Conference, v5, 2000, 2000 American Control Conference, Jun 28-Jun 30 2000, Chicago, IL, USA, p 3468-3472.
- [5] Junge, T.F., Schmid, C., "Web-based remote experimentation using a laboratory-scale optical tracker," Proceedings of the American Control Conference, v4, 2000, 2000 American Control Conference, Jun 28-Jun 30, 2000, Chicago, IL, USA, p 2951-2954
- [6] Schmid, C., Ali, A, "Web-based system for control engineering education," Proceedings of the American Control Conference, v5, 2000, 2000 American Control Conference, Jun 28-Jun 30 2000, Chicago, IL, USA, p 3463-3467.
- [7] Henry, J., "Running Laboratory Experiments via the World Wide Web," Session 3513 at the 1998 National Meeting of ASEE, Seattle, Washington.

[8] Chang T.N and Hung, D.C. "Web-based Distance Experiments: Design and Implementation," Proceedings of the International Conference on Engineering Education, Taiwan, August 14-18, 2000.

[9] Tan, K.K., Lee, T.H., Soh, C.Y., "Remotely operated experiment for mechatronics: Monitoring of DCS on the internet," IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM, v 2, 2001, 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics Proceedings, Jul 8-12 2001, Como, p 1106-1111.

[10] Chang, T.N., Jaroonsiriphan, P., Sun, X. "Integrating nanotechnology into undergraduate experience: a web-based approach," to appear at the International Journal of Engineering Education, Special Issue on Engineering Education in Nanotechnology.

[11] Edwards, H., "Building an Interactive Web Page with DataSocket," Application Note 127, June 1999 Edition, Part Number 341572B-01, National Instruments.

[12] Edwards, H., "Connecting Measurement Studio User Interface ActiveX Controls to Remote Data," Application Note 151, November 2000 Edition, Part Number 341972C-01, National Instruments.

Acknowledgment

This work is supported, in part, by a New Jersey Information-Technology Opportunities for the Workforce, Education, and Research (NJ I-TOWER).