Remote Laboratories for Control Education [1]

Henry H. Hahn and Mark W. Spong
Dept. of General Engineering
University of Illinois at Urbana-Champaign
104 S. Mathews Avenue
Urbana, Ill. 61801

Abstract

This paper describes progress in the development of an integrated network of internet based control laboratories at the University of Illinois at Urbana-Champaign. A key feature of our project is the use of real experiments in addition to simulation and animation. Our first set of experiments includes a DC-motor, an inverted pendulum, an inertia wheel pendulum, and a two-link, direct drive robot arm. These experiments were first tested online in the summer of 2000 and we are class testing them in the fall of 2000 and spring of 2001.

1 Introduction

In the College of Engineering at the University of Illinois we have developed a network of College level laboratories for instruction in controls and related subjects. By raising laboratory instruction from the Department level to the College level, we have succeeded both in the elimination of duplicate facilities and in the integration of laboratory instruction across Departments. Individual Departments retain their existing controls courses while scheduling lab sections in the Collegewide and centrally maintained laboratory facilities. Our laboratory network consists of a central controls lab, the College of Engineering Control Systems Laboratory (CO-ECSL), which has eighteen (18) lab benches containing state-of-the-art equipment for instruction in feedback control systems, and several satellite laboratories, including the Robotics and Automation Laboratory, the Mechatronics Laboratory, and the Fluid Power Systems Laboratory. Our laboratory network currently services nearly a dozen courses in the Electrical, Mechanical, Aerospace, and General Engineering Departments with a cummulative enrollment of well over 600 students per year.

We are now developing an integrated set of remote control experiments, accessible over the internet, to provide full integration of our laboratory network and experiments, to enhance their functionality and to facilitate broader access to our laboratory facilities. This paper describes our initial efforts in this development. The labs themselves can be found at http://weblab.ge.uiuc.edu.

2 Features of the Remote Laboratories

Our potential remote laboratory experiments include the following:

1. In-house developed plants. Our current list is extensive and includes: DC-motors, water tanks, active noise cancellation system, various inverted pendulum type systems, including rotary pendula, and pendubots, ball and beam plants, active vision systems, direct-drive robots, industrial manipulators, pneumatic and hydraulic plants, and more. Our initial focus, reported here, is on a DC-motor experiment, an inverted pendulum, and a two-link direct drive robot.

2. A common control interface and architecture that can be run in a browser. Our remote laboratories are being developed with a common "look and feel," i.e., based on a common user interface and control architecture. This means that new experiments can be quickly added as they are developed once the common "shell" is created and also that students will have to learn only one interface in order to access the entire library of experiments. Our experiments can also be run in a standard browser, (Netscape or Internet Explorer) and do not require the user to download and/or install software locally. We find that systems that require the local installation of special software often encounter compatibility problems and discourage the user from making the effort necessary to get the experiments working.

3. Streaming video so that the remote user gets the sense of "telepresence" in the laboratory. We have a controllable camera that the user can pan in order to see the entire laboratory or to view a particular plant being controlled.

4. A Matlab/Simulink interface. Matlab has become the de-facto standard tool for control system de-

[1] This research was partially supported by the National Science Foundation grant CMS-9840985 and by the Sloan Center for Asynchronous Learning Environments (SCALE) at the University of Illinois.

sign in most universities. Our system allows data capture for plotting and analysis in Matlab. We also provide mathematical models of our plants in Matlab/Simulink that the user can download and simulate locally. He/she can then design and simulate controllers for these plant models prior to controlling the actual hardware. Eventually, we will allow the user to upload his/her own controller code (Simulink blocks) to control the physical plant. The security issues involved in this feature will be discussed in a future paper.

We have developed a Java based Graphical User Interface that performs several functions. It is an icon based interface to the laboratory experiments that allows Matlab/Simulink simulation code and OpenGL animation code to be downloaded to the remote user. This is important to allow the user to test, simulate, and animate his/her controller design before it is implemented on the actual hardware in the lab.

The Java GUI will download the control module to the laboratory plant and bring back the video data for presentation to the user, as well as bringing back the Matlab data for plotting and analysis by the remote user.

3 Motivation

There are several important reasons that we are interested in developing these internet laboratories.

1. Integration and Increased Exposure to Control Systems. We are integrating our various laboratories into a coherent network that provides horizontal and vertical integration of control concepts within the engineering curricula. Having the laboratory experiments available over the web means that students from any of the laboratories can access experiments in all other laboratories. Thus instructors are more likely to assign experiments from multiple labs which gives students increased exposure to a broad range of control applications.

2. Increased Efficiency and Student Convenience. Allowing students access to the laboratory at their leisure, even from their homes and dormitories, provides increased opportunity for students and greater utilization of equipment. Students will have more time to better understand the concepts being presented, and they can complete prelab assignments before actually coming to the lab.

3. Greater Exposure to Control Systems. Increasing the amount of time that students can interact with controls is one of the major benefits of the web-lab initiative. It may also be a time for observing the meshing of control theory and real systems, for observing how real systems respond as opposed to modeled or simulated data.

4. Distance Education. The internet control labs will greatly benefit many of our off-campus courses which previously could not offer students any laboratory instruction. Although not a perfect replacement for on-site laboratory instruction, remote labs can provide a nearly realistic experience for students unable to come to campus.

4 Physical Setup

The following is a description of the general physical setup and components of each of the weblab components.

(1) Queued Client

Only one user can control the physical system at a time. The Queued Client can access the experiment applet while the experiment is being run but is locked out from actually controlling the hardware. When the Queued Client first attempts a log on, he or she receives a message indicating the status of the experiment and his or her position in the queue. As the preceding users logoff, the queued client is notified of its progression in the queue. Finally, when the last preceding user logs off, the queued client gains access to the experiment.

While waiting for access to the experiment, the Queued Client can watch the video of the experiment and can design his/her own controllers and implement them on a simulation model.

(2) Controlling Client

As stated previously, only one client can control the experiment at a time, the Controlling Client. The controlling client also has access to the web server. The connection from the controlling client to the control server is done over the Internet. A Java applet loaded from the web server creates a connection with the software residing on the control server. The controlling client can both send and receive data from the control server, as can the control server to the controlling client via the Java applet. The port on which the experiment runs must be open for both incoming and outgoing data transmissions. With Java, both TCP/IP and UDP protocols can be used. TCP/IP is the protocol used currently in the UIUC web-labs.

(3) Web Server The connection from the client to the web server is essentially unidirectional; most of the content is delivered from the web server to the clients. The web server directs the clients to the appropriate experiments. We have tested three different web server

software packages, Microsoft Personal Web Server for Windows 95, Apache, and Microsoft Internet Information Server 4.0. Currently, we find Microsoft Internet Information Server 4.0 the best suited to our weblab needs. For example, adding restricted access to certain portions of the website is easy to implement with IIS (it is possible with Apache but quite a bit more work.) User accounts are also much easier to handle as they are integrated into Windows NT's user account system. IIS 4.0 also has an easy to configure FTP server. The free version of IIS 4.0 we are using, though, only allows up to 10 connections at one time.

(4) Control Server

The control server is the center of the experiment. It handles the actuall feedback control signals (input and output) and also all communication with the controlling client. Currently, a real-time operating system, namely VenturCom's Real-Time Extensions for WindowsNT, has been installed on the control server for execution of the real time control tasks without possibility of processor interruption from outside sources.

5 Experiments

5.1 Motor Control

The first experiment that we developed was a PI speed controller for a DC motor. The user has the power to change the proportional and integral gains while the speed controller is running. While the experiment is running, the user can view the response with the graphing portion of the Java applet. The data that are viewable include the reference signal, response, and error. Figure 4, shows is an actual picture of the graphing window. The plotter scrolls data across the screen from right to left similar to that of an oscilloscope display. In Figure 4, it can be noted that there are several menu bar items. These items are File, Zoom, Axis, Plotter Refresh Rate, Data Series, Controller, and Help. The File menu currently has the option to save the plotted data to the server from which the data is sent. The actual transmission of the data is subject to bandwidth available but the actual experimental data can be saved to the server at resolutions higher than available for Internet transmission. The server can record data from sample rates as high as 10kHz; this is the threshold for the real-time operating system that is being used. Currently, the controller can be adjusted server-side to run at various rates (which is also the output sample rate in this case.) The transmission rate is also variable (server-side) and can be set as high as about 100Hz with decent timing accuracy.

The PI speed controller server is another crucial part of the experiment. The server handles communication to and from the Java applet, transmits the data, and handles all configurable features. The server implements TCP connections and has a communication/handshaking protocol specific to the experiment. Some of the features of the server include data transmission rate settings, experiment sample rate settings, and data file saving. It also handles all the incoming commands from the Java applet.

5.2 Direct Drive Robot

The second experiment described here is a control lab for a two link direct drive robot. This laboratory exercise is given in the course GE 389, Robot Dynamics and Control, after the students have been exposed in lecture to concepts of Lagrangian dynamics of robots, PID control, trajectory generation, and computed torque/inverse dynamics control. The robot arm is shown in Figure 1 below.

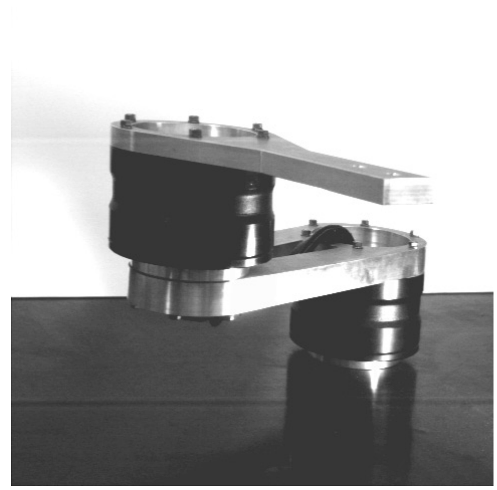

Figure 1: D2R2: The Direct Drive Robot

In previous labs the students would have already derived the dynamic equations of motion of this arm and simulated the free response. Then they would compare in simulation

- A PID control for a set point reference command
- A PID control for a cubic polynomial trajectory
- An inverse dynamics control with a PID outer loop control for the same cubic polynomial trajectory

Following these simulation laboratories the students repeat the above tests on the actual hardware. Shown in Figure 5 is the GUI applet for this laboratory. The students can enter PID gains, the dynamic parameters of the robot, and the coefficients of the cubic polynomial reference trajectory. The low level details of the controller implementation, such as sampling rate, encoder and A/D interface are transparent to the students. These issues are covered in other courses so that the students in this lab are free to consider only the higher level issues of nonlinear robot dynamics and control.

5.3 Inverted Pendula

The final two experiments reported here are an inverted pendulum and an inertia wheel pendulum. The inverted pendulum is the traditional cart-pole systems while the inertia wheel pendulum is a novel device consisting of a physical pendulum with a motor/flywheel assembly attached to the free end of the pendulum. The coupling torque generated by the angular acceleration of the flywheel is thus the control input to the system. The Java GUI's for both systems are shown below.

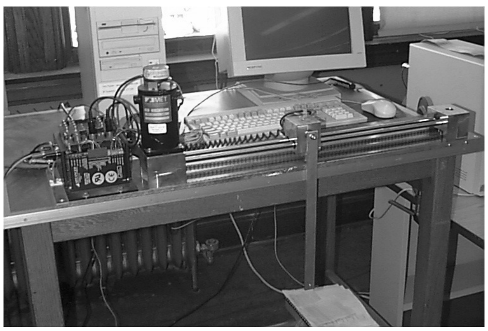

Figure 2: Inverted Pendulum

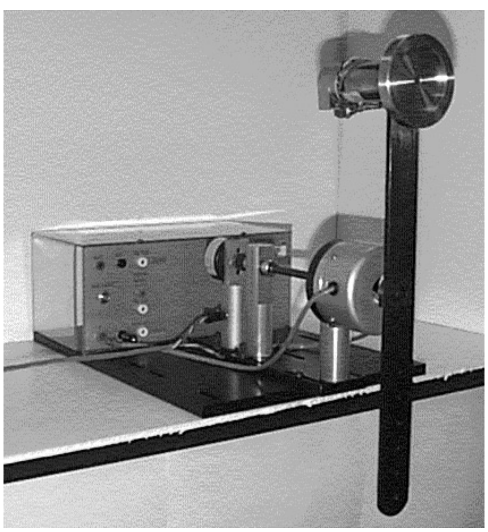

Figure 3: Inertia Wheel Pendulum

With these systems the user may investigate introductory linear state feedback control concepts for stabilization of the pendula about the vertical position or he/she may investigate nonlinear techniques for swinging the pendulum up to the inverted position from the vertically downward position. This problem is also suitable for investigation of control mode switching as well as concepts such as controller saturation [9].

6 Extensions

All of the experiments described above allow the user to adjust gains, set points, and other parameters in real time but do not allow additional flexibility in controller design. Our next extension of the web lab will be a feature to enable the user to upload his/her own controller code. This requires the investigation of robust, fault tolerant control architectures, such as the Simplex Architecture[7], that can accept downloads of control modules over the internet, link them to the local real-time system and run them while guaranteeing stable operation. This guarantee of stability is important in order to allow maximum flexibility and access to unknown and possibly inexperienced users to test their controller designs. This second phase of our project will be reported in a future paper.

References

[1] Alleyne, et. al., "A Collegewide Laboratory–Based Program in Control Systems Technology at The University of Illinois at Urbana-Champaign," 35th IEEE CDC, Kobe, Japan, Dec. 1996.

[2] Årzén, Karl-Erik, "A Real-Time Systems Course Focused on Control Systems," *IEEE Real Time Systems Symposium*, Pozan, Poland, Dec. 1998.

[3] de Queiroz, et. al., "A Robotics and Computer Vision Virtual Laboratory," AMC'98, pp. 694-699, Coimbra.

[4] Jochheim, A., and Röhrig, C., "The Virtual Lab for Teleoperated Control of Real Experiments," *Proc. 38th IEEE CDC*, pp. 819-824, Phoenix, AZ, Dec. 1999.

[5] Overstreet, Jamahl and Tzes, Anthony. Internet-Based Client/Server Virtual Instrument Designs for Real-Time Remote-Access Control Engineering Laboratory. Proc. Of the American Control Conference. San Diego, CA 1999: 1472-1471.

[6] Overstreet, Jamahl and Tzes, Anthony. "An Internet-Based Real-Time Control Engineering Laboratory." IEEE Control Systems October 1999: 19-34.

[7] Sha, L., "Dependable System Upgrades," *IEEE Real Time Systems Symposium*, Pozan, Poland, Dec. 1998.

[8] Spong, M.W., "Control Education Crossing Department Boundaries," *Proc. American Control Conference*, San Diego, June, 1999.

[9] Spong, M.W., Corke, P., and Lozano, R., "Nonlinear Control of the Inertia Wheel Pendulum," *Automatica*, submitted, September, 1999.

[10] Spong, M.W., and Tsao, T.-C., "Mechatronics Education at the University of Illinois," *IFAC Triennial World Congress*, Beijing, China, July, 1999.

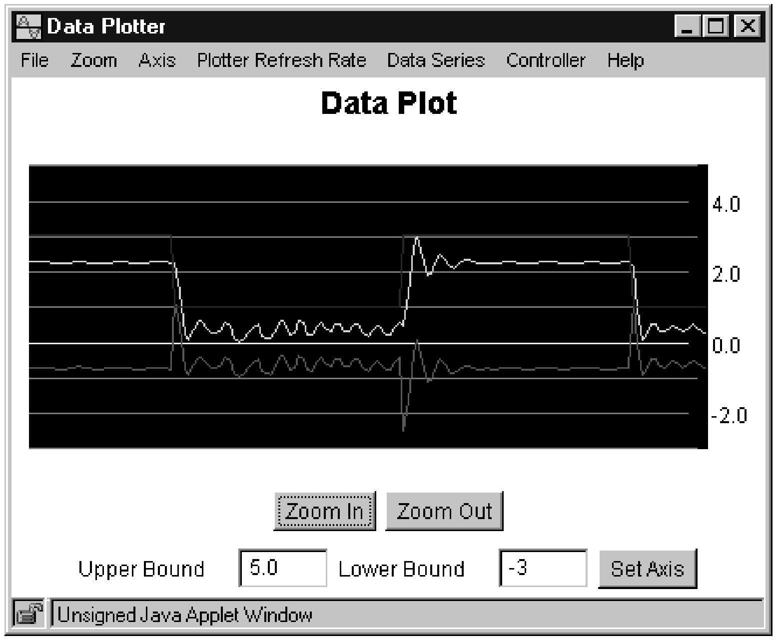

Figure 4: Java Data Plotting Applet

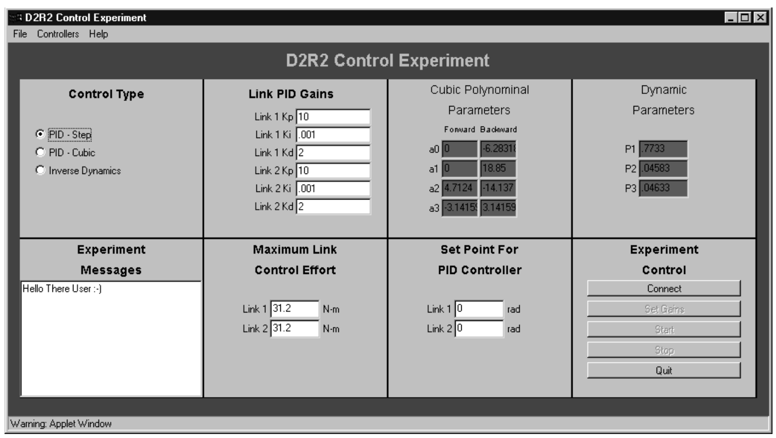

Figure 5: D2R2 Control Lab GUI

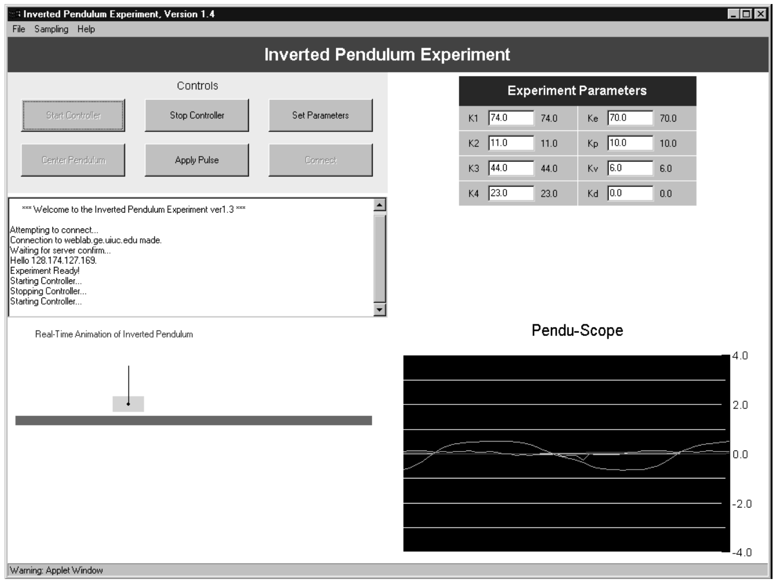

Figure 6: Inverted Pendulum GUI

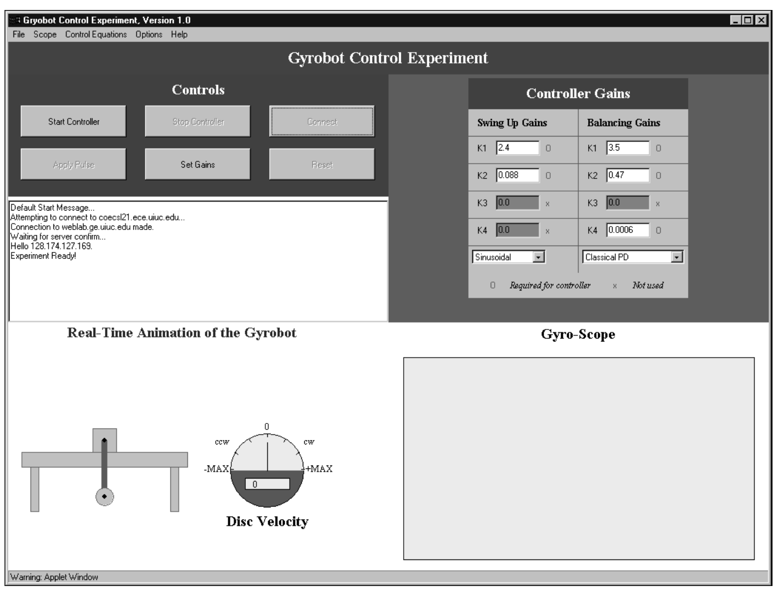

Figure 7: Inertia Wheel Pendulum GUI