

Web-Based Flow Control of a Three-Tank System

Lei Wu, David Cartes, and Chiang Shih

Department of Mechanical Engineering, FAMU-FSU College of Engineering, Florida State University
Tallahassee, FL, 32310, USA

ABSTRACT

In this paper, three different web-based LabVIEW control structures are designed, LabVIEW to LabVIEW control, LabVIEW to web browser control through CGI, and LabVIEW to web browser control through ActiveX + DataSocket. These three different web-based control structures are implemented and compared on an actual three-tank system. The objective is to enable the remote users to run the two pre-designed experiments through Internet in real time. The remote users should also be able to specify the control parameters for the experiments, watching the live video of the water height during the experiment, and getting data after the experiment to study control subjects. All three structures realize the web-based control concept under certain conditions. The live video broadcasting setup is the same for these three structures, and therefore they deliver the similar live video playback performance. Except the live video, these three structures differ very much. In terms of the data acquisition and control performance, the CGI method delivers the best performance with the shortest data acquisition period. In terms of the data communication, both LabVIEW to LabVIEW and ActiveX + DataSocket structures enable live data transfer. While LabVIEW to web browser through CGI structure can only enable the users to download the data file after experiment finishes. In terms of the implementation in real life, CGI has the widest user group. Everyone who has a web browser can have access to the web-based experiments through CGI. LabVIEW to LabVIEW control requires that the remote users have some knowledge about LabVIEW and have LabVIEW version 4.1 or above installed in his/her computer. The application area of LabVIEW to web browser control through ActiveX + DataSocket is very limited because ActiveX is a double-edged technology on Internet. In short, LabVIEW to web browser control through CGI delivers the best performance overall among these three different web-based control structures.

Keywords: web-based control, flow control, three-tank system, LabVIEW, CGI, ActiveX.

1. INTRODUCTION

In 1990s, Internet technology has undergone a rapid growth and injected its influence into many engineering fields. The idea of operating engineering plants remotely through Internet is becoming both theoretically and commercially possible. In the academic area, the Internet provides a perfect platform for implementing distance learning and distance laboratory experience. In recent years, many universities started to develop web-based laboratories. In 1996, Aktan and etc developed a real-time remote-access control engineering teaching laboratory at the College of Engineering of Oregon State University [3]. This remote laboratory used the Internet to provide controller implementation on a three degree of freedom robot arm. From 1999, National University of Singapore has developed a series of web-based control experiments, including an oscilloscope control experiment, a frequency modulation experiment, a

coupled tank control experiment, and a two-degree-of-freedom helicopter model control experiment [4-8]. These web-based laboratories are currently utilized in both undergraduate and graduate level courses. These web-based laboratories benefit the students by bestowing upon them the flexibility to arrange their own lab times from any place. Generally, the feedback from students is very positive. The above web-based applications are all developed in the LabVIEW environment. LabVIEW has several different structures to develop a web-based laboratory. To study and compare the different web-based control techniques, a three-tank system is designed and developed as the web-based application in this project. The three-tank system is a simplified model of various different water level control applications in the industry field, such as steam generators in power generating processes, reactors in many chemical plants, and storage tanks in oil/gas production industry. The system was first studied and controlled locally. Then different web-based control techniques were implemented on the system. Besides serving as the research basis of the web-based control techniques, the web-based three-tank control system is also a convenient distance laboratory facility to study controls subjects for students in FAMU-FSU College of Engineering.

2. DIFFERENT WEB-BASED CONTROL STRUCTURES

The web-based communication between LabVIEW programs and other programs (including LabVIEW programs) is usually realized through client-server architecture. In the client-server model, clients request services from servers and servers usually wait for clients to initiate a connection. After the connection has been setup, the server and the client can transfer data back and forth using some network protocol. When the current client closes the connection, the server will continue to wait for other clients' request. A server may serve multiple clients at a time as well.

Depends on the clients' application types, there are several ways to connect a LabVIEW server program to the Internet. Figure 1 shows an overview of some different technologies to communicate with a LabVIEW server program over the Internet.

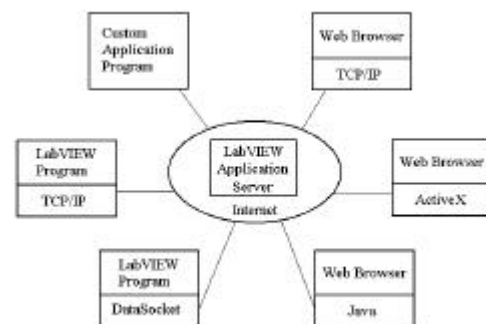


Figure 1: Different Technologies for Communication with LabVIEW Over Internet

In general, to communicate with a LabVIEW server program, the remote clients application interface should be one of the following three types:

- A special custom application program written in another language (e.g., Visual Basic, C++ and etc).
- Another LabVIEW program.
- A web browser.

Because it is difficult to write special custom applications and for the clients to install and learn to use new applications, method (a) has become outdated and rare nowadays. Therefore, only method (b) and two different types of method (c), through CGI and through DataSocket + ActiveX, are implemented and compared in this paper.

2.1. Web-based LabVIEW to LabVIEW Control Structure

Figure 2 shows the schematic setup of the LabVIEW to LabVIEW control in this project. On the server side, there are two computers, one for broadcasting video, and one for data acquisition and control (DAQ server). The video server is a Gateway G6450 computer which uses RealProducer Basic 8.51 to create the live video streaming and broadcast the stream to a RealServer provided by FAMU-FSU College of Engineering, which is the same server for broadcasting various live lecture videos from the College of Engineering. The DAQ server is a Dell Dimension XPS T500 computer with LabVIEW version 6i. The remote clients' computer should have LabVIEW version 4.1 and above to conduct the web-based experiments. Prior to the experiment, the user needs to download the corresponding client LabVIEW program from the DAQ server.

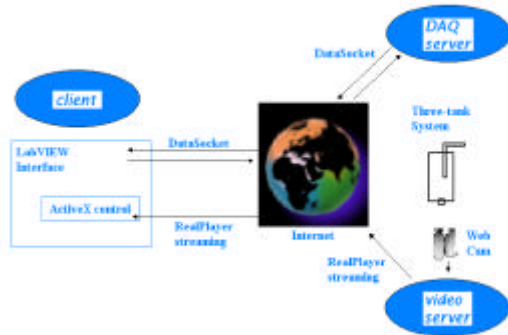


Figure 2: LabVIEW to LabVIEW Control Schematic

As shown in Figure 2, there are two kinds of communication between the client and the server. One is data communication through DataSocket. The client LabVIEW program sends control parameters and receives experimental data through different DataSocket channels from the DAQ server LabVIEW program. The other is the video stream communication through RealPlayer. In the client LabVIEW program, there is an embedded ActiveX control to display the video stream. The embedded ActiveX object is a Microsoft web browser which is again embedded by another ActiveX control, a RealPlayer object. The RealPlayer object receives and plays back the live experiment video stream from the RealServer at the FAMU-FSU College of Engineering.

A normal sequence of the server and the client LabVIEW programs is shown in Figure 3. In this sequence, a DataSocket channel "server" is used to denote experiment status, which can have the following four different values,
 "0": server program is off.
 "1": experiment busy. Experiment 1 running, controlling valve 1 to track a step input.

"2": experiment busy. Experiment 2 running, controlling valve 2 to track a sinusoidal input.
 "3": experiment idle, waiting for connections.

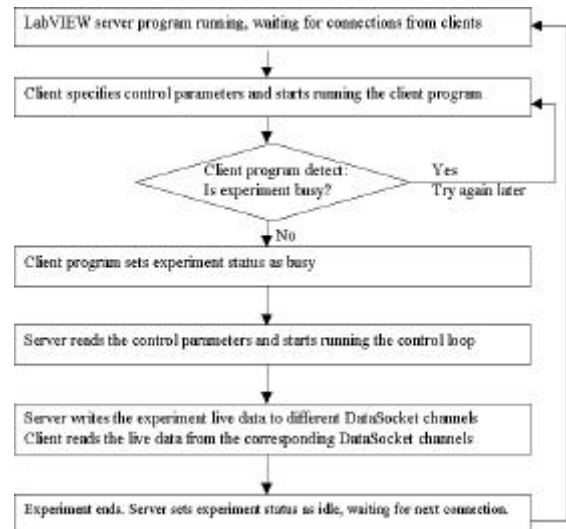


Figure 3: Data Communication between LabVIEW Server and Client Program

In a normal sequence, the client program sets the experiment status as busy by writing "1" or "2" to the DataSocket channel "server". This action will also trigger the server program to call another program, the control loop, to perform the control experiment. The server program will pass the experiment parameters specified by client to the control loop program. During the control loop, the server will continuously write experiment data, the water height and the control voltage to valve 1, to corresponding DataSocket channels and save the data on server's harddisk. And the client program will continuously read data and display them on its front panel, and also save the data to its own harddisk. In the meantime, client can monitor the real time experiment video in the front panel. After the experiment finishes, both the client program and the server program write a "3" to DataSocket channel "server" to set the experiment status as idle. To avoid the excess running of one experiment, the experiment running time is limited within 20 minutes in the server program.

Besides the normal sequence discussed above, there is also some possibility that the client will abort the experiment before its end. If this happens, it will be in vain to run the control loop program any longer on the server. Because the last sequence of the client program is writing a "3" to channel "server", to avoid the unnecessary program running, the control loop program will continuously read the channel "server" to see if it is "3". Once it is "3", the control loop will stop and the experiment will become idle and wait for the next connection.

2.2. Web-based LabVIEW to Web Browser Control through CGI

Nowadays, nearly all computers are installed with web browser program, mainly Internet Explore, while very few computers have LabVIEW program. If a web-based LabVIEW application is developed for remote web browser users, it will have much more potential users than developed for LabVIEW users. It is also an extra burden, and not necessary mostly, for students to learn LabVIEW program prior to the lab. Therefore the architecture of LabVIEW to web browser control has a wider application area and is more convenient in some area than LabVIEW to LabVIEW control architecture.

CGI stands for *Common Gateway Interface*. It is a communication protocol by which a web server communicates with other application programs and databases. It is the most widely used standard to call other programs in the server from web browser. CGI just defines an interface. It isn't a programming language or an application, but it can be written in almost any language and LabVIEW 6i fully supports the CGI standard through its Internet Toolkit package.

Figure 4 shows the schematic setup of the LabVIEW to web browser control through CGI. The two servers are the same two computers used in the LabVIEW to LabVIEW control. Different from LabVIEW to LabVIEW control, the remote users don't need to download any special program in order to perform the remote control.

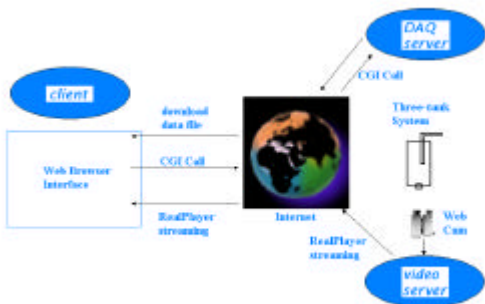


Figure 4: Schematic of LabVIEW to Web Browser Control through CGI

As shown in Figure 4, there are also two kinds of communication between the client and the server, the data communication and the live video communication. Since the user interface is a web browser, the RealPlayer is easily embedded in it to play the experimental live video. Different from the live video, there is no live data communication between the server and the client. CGI programs can only read data and generate result once during one running. The client first sends out the experiment parameters and starts the experiment. Once the client starts the experiment, he/she can't take any action to stop it unless server stops it. The experiment will run for the pre-specified time and then the server will generate and publish a web page to the client's web browser. The web page contains a link to the experiment data file, an excel file, for the client to download. Since there are lots of potential users, there is some possibility that some user may abuse the experiment. Therefore, different from the LabVIEW to LabVIEW control, the web browser users do not have the right to specify the experiment running time. The running time is pre-specified in the server program. Figure 5 shows the data communication sequence between the server and the client.

2.3. Web-based LabVIEW to Web Browser Control through ActiveX + DataSocket

As stated in section 2.2, CGI can't enable live data communication between the LabVIEW server program and the clients' web browser. While another structure, ActiveX + DataSocket can realize this goal. The live video broadcasting implementation is the same as the two previous structures. The data communication is similar to the one shown in Figure 2. There is an ActiveX object (programmed with Visual Basic) embedded within the client's web browser. The control parameters and the measurement data are transferred lively between this ActiveX object and the DAQ server LabVIEW program through different DataSocket channels. The data

communication sequence is similar to the one shown in Figure 3. But since it is targeted at the general web browser users, the experiment running time is pre-specified in the server program.

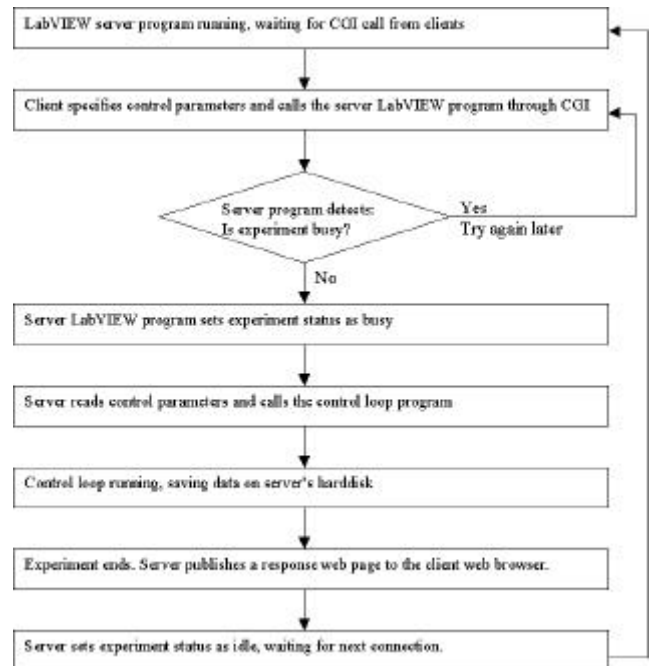


Figure 5: Data Communication between LabVIEW and Web Browser through CGI

3. THREE-TANK SYSTEM DESCRIPTION AND LOCAL PID CONTROL

As a simple water level control plant, the three-tank system, schematically shown in Figure 6, consists of three rectangular Plexiglas tanks sitting respectively on the three levels of a three-story steel frame. The tanks are named tank 1, tank 2 and tank 3 respectively from top to bottom. Tank 3 serves merely as a reservoir. Two proportional control valves are used to regulate the flow between each tank. Valve 1 is located between tank 1 and tank 2, which controls the inlet mass flow to tank 2. Valve 2 is located between tank 2 and tank 3, which controls the outlet mass flow from tank 2. A pressure sensor is mounted at the bottom of tank 2 to measure the water height inside the tank. There is a pump between tank 1 and tank 3. When the power is turned on, water is continuously pumped from reservoir tank 3 to tank 1, and then flows through the regulated valve 1 to tank 2, and in turn through regulated valve 2 back to the reservoir tank 3. Since the pump pumps water at a much faster rate than the maximum flow rate from tank 1 to tank 2, there is also a bypass flow path from tank 1 to tank 3 through the pipe connecting tank 1 and tank 3 at the other side. There are two Plexiglas partitions inside tank 1. The small partition prevents the water jet coming from the pipe from interfering with the outlet flow through valve 1. The large partition maintains the water level in tank 1 at the same height as the partition. The extra amount of water coming from the pump will overflow the top of the large partition and flow back to tank 3 through the bypass pipe. A picture of the real system is shown in Figure 7.

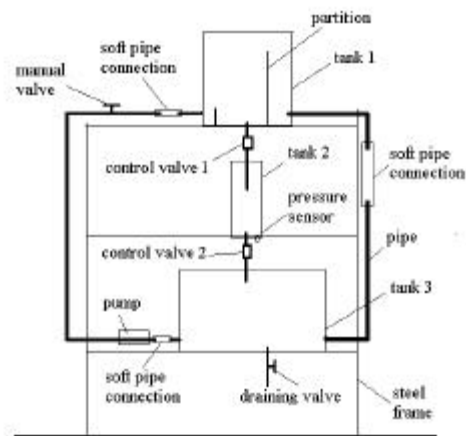


Figure 6: Three-Tank System Schematic Drawing

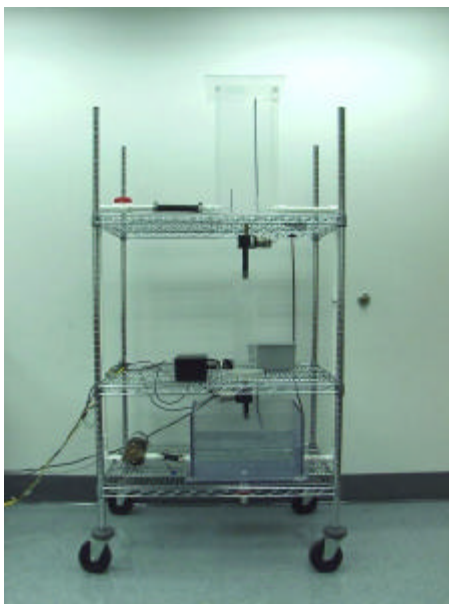


Figure 7: Three-Tank System Picture

The objective is to control the water height in tank 2 to follow some reference input. Two experiments were developed in this project. One is to control the water level in tank 2 to follow a step input via controlling valve 1 (the opening of valve 2 is maintained constant at 0.60V). The other is to control the water level to follow a sinusoidal input via controlling valve 2 (the opening of valve 1 is maintained constant at 0.32V). Since the web-based control structures are the same for these two experiments, only the first experiment will be discussed in this paper.

From the open loop step response, the plant transfer function has been identified as $G(s) = \frac{0.75}{s + 0.009795}$ using Least Square

Identification method. Since the system is a first-order system, a PID controller will be sufficient for tracking a step input. Use root locus design method, a PID controller of the form $D(s) = \frac{0.987s^2 + 0.5s + 0.06}{s}$ was designed. The

corresponding simulated close loop step response is shown in Figure 8. The same PID control law is implemented in all the different web-based control structures in this paper.

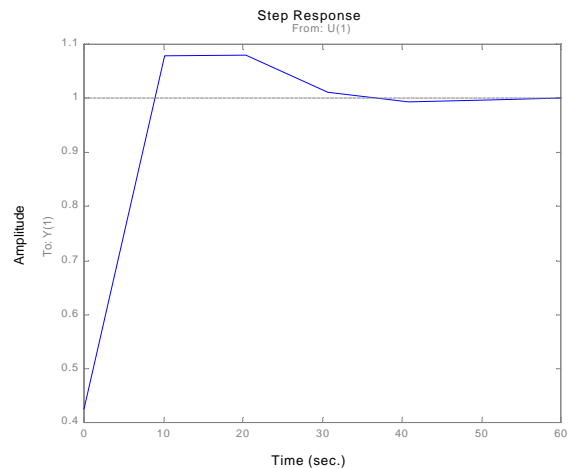


Figure 8: Close Loop Step Response of the System

4. RESULTS AND DISCUSSION

Different web-based control structures were tested on the three-tank system within the FAMU-FSU College of Engineering network.

4.1. Web-based Implementation of LabVIEW to LabVIEW control

Figure 9 shows the front panel of the client's LabVIEW program.

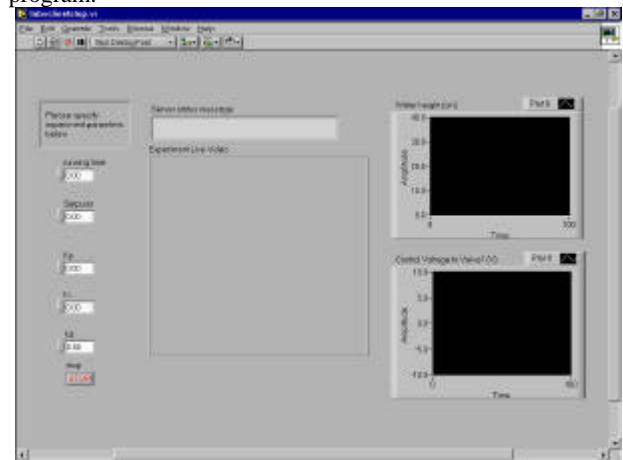


Figure 9: Front Panel of the LabVIEW Client Program

During the web-based control test, the live video playback has a delay of 1~3 seconds between each images, which is acceptable. While control loop period has been significantly elongated from 0.3 sec (for the local control) to about 4 sec. This is because the live data transfer through DataSocket requires much computer resources and there are three live DataSocket channels, the experiment status, the water height, and the control voltage, in the LabVIEW control loop. The live data transfer between the LabVIEW server and client program takes much time and therefore deteriorates the control performance. However, the client computer used only has a PII 450MHz CPU. If the server and client computer have much faster CPUs, the data acquisition delay should be decreased drastically.

4.2. Web-based Implementation of LabVIEW to Web Browser Control through CGI

Figure 10 shows the client web browser interface for the experiment.

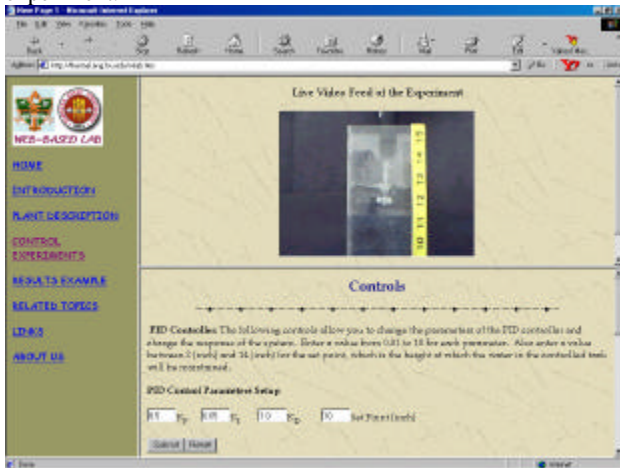


Figure 10: Client Web Browser Interface

A video image delay of 1~2 sec is experienced during the web-based implementation for the LabVIEW to web browser control through CGI. Since CGI doesn't support live data transfer, the control performance of the LabVIEW to web browser control through CGI should be close to the local control performance. Figure 11 shows the web-based control performance for this structure. A performance comparison between the web-based implementation and the local control simulation is shown in Figure 12.

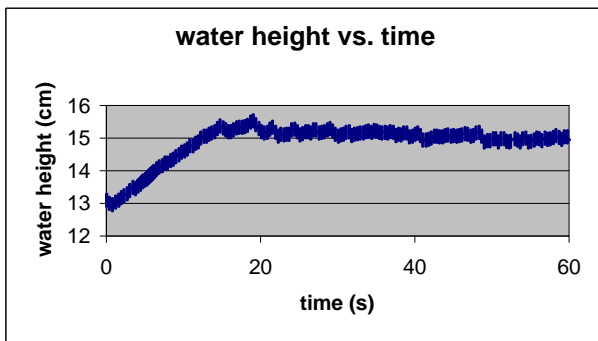


Figure 11: Web-based Controller Implementation Using CGI

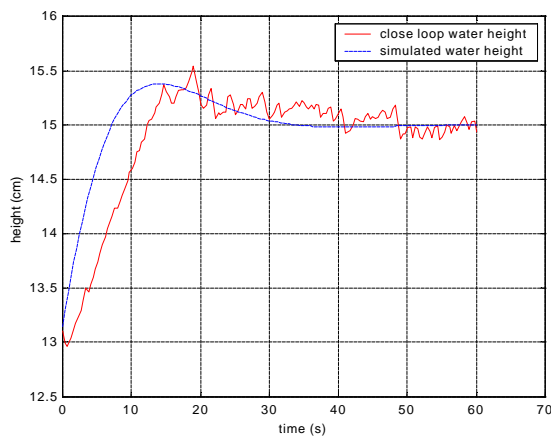


Figure 12: Comparison of Local Response Simulation and Web-based Response

It can be seen that there is some delay of the water height response compare to the simulation results. This is because the real control effort, the opening of the valve, is limited. Initially the tracking error is very large, which requires more control

effort than the valve can afford. Therefore the valve fully opens and in turn determines the rise slope of the water height. After the transit part, the water height response behaves like the simulated response. There is nearly no control performance deterioration when implementing controller through Internet.

4.3. Web-based Implementation of LabVIEW to Web Browser Control through ActiveX + DataSocket

Several computers in the public computer laboratory and the graduate student laboratory at FAMU-FSU College of Engineering were used as the remote clients to test the ActiveX + DataSocket structure. The results were frustrated. Nearly all the computers can not conduct the web-based control experiments. The embedded ActiveX control was not running on these computers, which led to the failure of the data communication between server and client. This is because that most universities and institutions prohibit the use of ActiveX applications in their computers. There are various ActiveX applications developed by all kinds of people on the Internet for downloading. Some of them may cause computer malfunctioning, even erase information saved in the hard disk, if you run them. In order to prevent the possible damages caused by running unknown ActiveX program, most universities and institutions prevent the running of ActiveX on their networks. This is the biggest problem with the real life implementation of the ActiveX + DataSocket structure. During the test, only one computer, another graduate student's personal computer can run the ActiveX control and have data communication with the LabVIEW server program. The live video delay is about 1~2 sec, which is about the same as the one with CGI. Because of the impracticability of the ActiveX + DataSocket structure, no more test was conducted with this kind of web-based control structure.

Table 1 compares the performance of these three different web-based control structures discussed above.

Table 1: Performance Comparison of the Three Web-based Control Structures

		Video	Data	Implementation
LabVIEW to LabVIEW		1~3 sec delay	Live data. 4 sec data acquisition period	Clients need LabVIEW software.
LabVIEW to Web Browser	CGI	1~2 sec delay	No live data. 0.4 sec data acquisition period	Easy to implement. Easy for clients to use.
	ActiveX + Data Socket	1~2 sec delay	N/A	Strong implementation restrictions.

5. CONCLUSION

In this paper, three different web-based LabVIEW control structures are designed, LabVIEW to LabVIEW control, LabVIEW to web browser control through CGI, and LabVIEW to web browser control through ActiveX + DataSocket. These three different web-based control structures are implemented and compared on an actual three-tank system. The objective is to enable the remote users to run the two pre-designed experiments through Internet in real time. The remote users should also be able to specify the control parameters for the experiments, watching the live video of the water height during the experiment, and getting data after the experiment to study

control subjects. All three structures realize the web-based control concept under certain conditions. The live video broadcasting setup is the same for these three structures, and therefore they deliver the similar live video playback performance. Except the live video, these three structures differ very much. In terms of the data acquisition and control performance, the CGI method delivers the best performance with the shortest data acquisition period. In terms of the data communication, both LabVIEW to LabVIEW and ActiveX + DataSocket structures enable live data transfer. While LabVIEW to web browser through CGI structure can only enable the users to download the data file after experiment finishes. In terms of the implementation in real life, CGI has the widest user group. Everyone who has a web browser can have access to the web-based experiments through CGI. LabVIEW to LabVIEW control requires that the remote users have some knowledge about LabVIEW and have LabVIEW version 4.1 or above installed in his/her computer. The application area of LabVIEW to web browser control through ActiveX + DataSocket is very limited because ActiveX is a double-edged technology on Internet. In short, LabVIEW to web browser control through CGI delivers the best performance overall among these three different web-based control structures.

6. ACKNOWLEDGEMENT

This project is partially supported by NSF under the grant DUE-9980775.

7. REFERENCE

- [1] D. Cartes, L. Wu, T. Curry and D. Chabukswar, "Adaptive Liquid Level Control of a Three Tank System", **Proceedings of the 45th Annual ISA POWID Conference**, June 2002.
- [2] D. Cartes and L. Wu, "Experimental Evaluation of Adaptive Three-Tank Level Control", **ISA Transactions**, *in review*.
- [3] Aktan B, Bohus CA, Crowl LA, and Shor MH, "Distance learning applied to control engineering laboratories" **IEEE Transactions on Education**, 39 (3): 320-326 AUG 1996
- [4] S. H. Chen, R. Chen, V. Ramakrishnan, S. Y. Hu, Y. Zhuang, C. C. Ko and B. M. Chen, "Development of remote laboratory experimentation through Internet," **Proceedings of the 1999 IEEE Hong Kong Symposium on Robotics and Control**, Hong Kong, pp. 756-760, July 1999.
- [5] C. C. Ko, B. M. Chen, S. H. Chen, V. Ramakrishnan, R. Chen, S. Y. Hu and Y. Zhuang, "A large scale web-based virtual oscilloscope laboratory experiment", **IEE Engineering Science and Education Journal**, Vol. 9, No. 2, pp. 69-76, April 2000.
- [6] C. C. Ko, B. M. Chen, J. Chen, Y. Zhuang and K. C. Tan, "Development of a Web-based laboratory for control experiments on a coupled tank apparatus", **IEEE Transactions on Education**, Vol. 44, No. 1, pp. 76-86, February 2001.
- [7] Jing Zhang, Jianping Chen, C. C. Ko, Ben M. Chen . and S. S. Ge, "A Web-Based Laboratory on Control of a Two-Degree-of-Freedom Helicopter," **IEEE Conference on Decision and Control**, Orlando, FL, December 4-7, 2001.
- [8] C. C. Ko, B. M. Chen, S. Y. Hu, V. Ramakrishnan, C. D. Cheng, Y. Zhuang and J. Chen, "A web-based virtual laboratory

on a frequency modulation experiment", **IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews**, Vol. 31, No. 3, pp. 295-303, August 2001.

[9] Jeffrey Travis, **Internet Applications in LabVIEW**, NJ: Prentice Hall Inc. Pub., 2000.