

# Development of a Web-Based Laboratory for Control Experiments on a Coupled Tank Apparatus

C. C. Ko, *Senior Member, IEEE*, Ben M. Chen, *Member, IEEE*, Jianping Chen, Yuan Zhuang, and Kay Chen Tan, *Member, IEEE*

**Abstract**—The Internet provides a new environment for developing a variety of applications for educational and research purposes. This paper presents the implementation of a web-based laboratory experiment on a coupled tank apparatus, a Multi-input–Multioutput (MIMO) system. The web-based laboratory has been developed to serve students and staff in the Department of Electrical Engineering at the National University of Singapore (NUS). The laboratory is an educational tool for teaching students the basic principles and methodology in performing a series of experiments on a coupled tank apparatus at any time and from any location through the Internet. With the capability to implement strategies for manual, proportional integral derivative (PID), general state-space, and fuzzy logic control, the laboratory also provides a platform for research staff to test control algorithms. Video conferencing has been used to provide audio and video feedback, with a camera mounted on a movable platform so that the user can control both the zooming and viewing angle. The laboratory can be accessed at <http://vlab.ee.nus.edu.sg/vlab/control/>.

**Index Terms**—Internet, virtual laboratory, web-based control.

## I. INTRODUCTION

THE CONCEPT of a web-based laboratory is not new. Shor and Bhandari [2] developed a distance learning application that allowed a user to remotely conduct experiments in the Control Engineering Laboratory at Oregon State University. The application permitted the experiments to be run via the World Wide Web (WWW). Only a basic web browser that ran Java is needed to gain access and a robot experiment was developed for demonstration. Bytronic Process Control unit [3], referred to as the process rig in the Process Control and Automation Laboratory at Case Western Reserve University, can be accessed remotely via the Internet. Using a web browser, the user can log on and post the parameters from a remote client to a Laboratory Virtual Instrument Engineering Workbench (LabVIEW, from National Instruments [6]) web server that is connected to a process rig via a programmable logic control (PLC) module. The image is generated by a program and refreshed on the client side using server push technology. A remote laboratory called VLAB [1] on an oscilloscope experiment was recently set up in the Department of Electrical Engineering, National University of Singapore (NUS). These remote laboratories are actual laboratory experiments that are run remotely via a web interface, and are well suited to distance learning courses where students need not be physically present on campus. They also facilitate



Fig. 1. Laboratory setup of the equipment.

the sharing of expensive instruments and equipment, and represent the next important step in remote distance learning.

In this paper, the implementation of a remote laboratory control experiment on a coupled tank apparatus is presented. Four commonly used control structures (manual control, proportional integral derivative (PID) control, general state-space control, and fuzzy logic control) are implemented. The experiment corresponds to an industrial control-related experiment for undergraduate and graduate students in NUS, with the virtual laboratory also providing a platform for researchers to test control algorithms. LabView [6], Java, HTML (Hyper Text Markup Language), as well as scripting languages like JavaScript and VBScript have been used in the implementation. A distinctive feature is the provision of audio and video feedback through the use of video conferencing based on Microsoft NetMeeting [8], with a camera mounted on a movable platform so that the user can control both the zooming and viewing angle. The equipment setup of the laboratory is depicted in Fig. 1.

This paper is organized as follows. Section II describes the structure of the virtual laboratory, including its hardware and software structures. Discussion is given on the use of transmission control protocol (TCP) as opposed to common gateway interface (CGI) for client–server communication as well as other distinctive features of the implementation from the point of providing audio–visual feedback. Section III presents a model of the coupled tank apparatus. Section IV presents the structures of the different controllers that can be used and the corresponding user interfaces. Conclusions are presented in Section V.

Manuscript received January 5, 2000; revised May 30, 2000.

The authors are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117576.

Publisher Item Identifier S 0018-9359(01)01286-9.

## II. STRUCTURE OF VIRTUAL LABORATORY

### A. Hardware Structure

Fig. 2 depicts the hardware structure of the virtual laboratory. As shown, a client PC is accessing the laboratory remotely through a connection to the HTTP server (provided with LabView [6] as part of its Internet Toolkit) which hosts the web site for conducting the experiment. The PC running LabView functions as the instrument controller. Connected to the coupled tank through a 50-pin Data Acquisition (DAQ) card, the PC implements local control of the coupled tank apparatus by supplying two input voltages to the pumps of the apparatus. A camera connected to a PC (serving as the video server) running Microsoft NetMeeting provides visual feedback to the user on the changes in water levels in the coupled tank. A microphone connected to the same PC captures the sound of the coupled tank's motor and provides audio feedback. The pan and tilt of the camera as well as its zoom setting are controlled by a circuit board connected to the instrument controller. Live video captured by the camera and the sound captured by the microphone are sent to the client through the use of a video conference session. This is a point-to-point transfer implying that only one user can view the video at a time.

### B. TCP for Client Server Communication

CGI and TCP are two popular methods for implementing client-server communication. As illustrated in Fig. 3, CGI programs running on a web server sending and receiving parameters from the users' web browsers have become an Internet standard. The client uses an Internet browser to connect to the web server, view the web page, send the CGI request to the server and finally receive the response from the server. Since CGI connection is based on HTTP, which is a per-session-basis protocol, the connection between the client and the server is closed once the client receives the response from the server.

In contrast to a CGI connection, a TCP connection established by a client with a server remains in place until the client closes the connection. Since the establishment of a TCP connection implies that the connection will be in place until the client closes it, TCP is the ideal communication protocol for implementing web-based instrument control that requires frequent parameter adjustments. CGI, on the other hand, is not so suitable for this purpose due to its operation on a per-session basis. Fig. 4 illustrates the communication between a client and server using TCP.

Communication between the client and server can be set up conveniently using Java. Java programs are compiled to platform independent codes, called bytecodes, of a hypothetical machine called the Java virtual machine (JVM). These bytecode programs, often called applet, can be included in a web page in the same way as an image. In addition, they can be viewed and downloaded by a Java-enabled browser to be executed by the browser's JVM. Using Java, the connection between the client and the server side can be set up through sockets. When the user downloads a Java applet from the web server, the applet (running on the client machine) represents the client side of the connection. Java was preferred to other programming languages

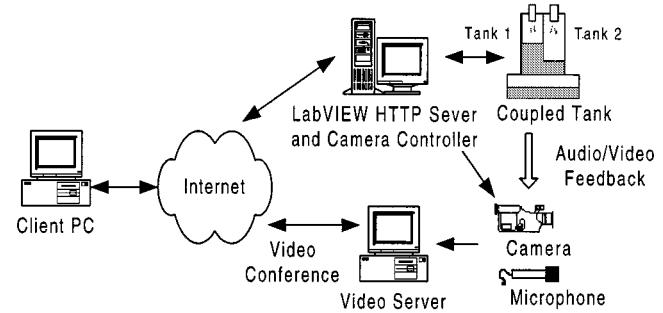


Fig. 2. Hardware structure of the virtual laboratory.

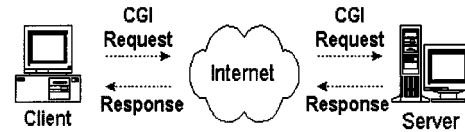


Fig. 3. A typical CGI connection process.

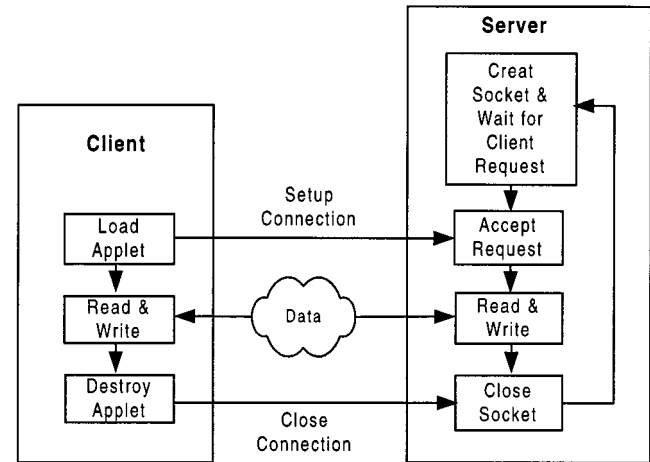


Fig. 4. Client-server interaction using TCP.

for implementing client-server communication due to its platform independent nature and extensive network programming support.

### C. Audio and Video Feedback

A striking feature of the implementation is the use of video conferencing as opposed to multicast schemes to provide the user with audio-visual feedback. During the initial stages of development, real-time streaming of image data was attempted using the two most popular commercial solutions available at the time. These solutions involved the use of RealServer G2 [9] and NetShow [10] media servers to stream live video as well as the use of commercial decoders RealPlayer G2 and Windows Media Player at the client side. However, these solutions were observed to be impractical, as a time lag of the order of ten seconds was noticed between the actual occurrence of an event in the laboratory and its presentation at the client. This delay was caused predominantly due to software overheads, which arose because multicast systems were optimized to support a large number of clients. The multirate encoding (based on the

bandwidth available to the target audience), Buffering (during network congestion) and intelligent transmission (maintaining continuity of video at the expense of quality during network congestion) features of the multicast schemes were also identified as potential bottlenecks.

Since the NUS campus network, NUSNET-III [11], provides the user with bandwidth comparable to a T1 connection, bandwidth is not a constraint for the implementation of remote laboratory experiment within the campus and video conferencing was adopted to provide live video to the client. For this purpose, Windows NetMeeting 3.0 was chosen. NetMeeting uses the H.323 standard, which is comprised of International Telecommunications Union (ITU) approved protocols for audio, video, and data conferencing over TCP/IP networks. The standard is compatible with H.261 and H.263 video codecs.

As NetMeeting 3.0 also supports the transfer of audio, a microphone connected to the instrument controller is used to capture the sound made by the coupled tank's motor. The latter can then be heard at the client side by attaching speakers to the client computer. This feature is intended to provide the user as realistic a feel as possible while performing the experiment.

The implementation uses an ActiveX control to embed the live audio and video on web pages. This control is part of the Windows NetMeeting Resource Kit [12], and is invoked and controlled using VBScript, a scripting language developed by Microsoft Corporation. When the user activates ActiveX and calls the video server (explicitly by specifying the IP (Internet Protocol) of the video server), a call is placed. The NetMeeting program on the video server is configured to receive calls immediately without any user intervention on the server side to accept the call. The program is also configured to send out live audio and video at the start of each call. The audio and video at the client side can be stopped in a similar manner through ending the call or clicking the "Stop A/V" button.

From the above, it is apparent that only one user can receive video at a time. This is logical since only one user is allowed to access the web-based laboratory at a time. In fact, this single user feature is implemented by restricting access to the experiment's web pages. The user is authenticated by a CGI program residing on the HTTP server, and can access the web pages for performing the experiment if no one else is carrying out the experiment.

#### D. Camera Control

Yet another special feature of the implementation is the camera's pan, tilt, and zoom control. The camera is mounted on a pan and tilt unit, which can be controlled through a circuit board. The board is in turn connected through RS-232 to the controller PC's serial port. The commands that the controller PC can issue are described in Table I.

Web-based control is implemented by continuously running a camera control program on the machine housing the HTTP server to receive command strings from the client (through a TCP channel) for the local control of the circuit board. At the client end, a Java applet sets up the sockets for TCP communication with the server. When the user clicks a button intended for camera control, a procedure in the applet that generates the

TABLE I  
COMMANDS FOR CAMERA CONTROL CIRCUIT

FUNCTION	ASCII	HEX	DEFINITION
<i>Camera</i>	#	23	Addresses the intended receiver/driver (camera). This command must be preceded by the camera number.
<i>Aux Off</i>	B	42	Turns an auxiliary off. This command must be preceded by the auxiliary number.
<i>Aux On</i>	A	41	Turns an auxiliary on. This command must be preceded by the auxiliary number.
<i>Focus Far</i>	F	46	Focus lens far.
<i>Focus Near</i>	N	4E	Focus lens near.
<i>Iris Close</i>	C	43	Close lens iris.
<i>Iris Open</i>	O	4F	Open lens iris.
<i>Lens Wide</i>	W	57	Zoom lens wide.
<i>Lens Tele</i>	T	54	Zoom lens telephoto.
<i>Pan Left</i>	L	4C	Pan camera left.
<i>Pan Right</i>	R	52	Pan camera right.
<i>Tilt Up</i>	U	55	Tilt camera up.
<i>Tilt Down</i>	D	44	Tilt camera down.
<i>Call Shot</i>	\	5C	Calls a preposition scene. This command must be preceded by a scene number.
<i>Set Shot</i>	^	5E	Sets (stores) a preposition scene. This command must be preceded by a scene number.

corresponding command string and writes it to the TCP connection is invoked.

#### E. Software Structure

This subsection explains how web-based control of the coupled tank is implemented. The PC running the LabView HTTP server also runs the LabView "G" language programs that set up the server side of the TCP communication with the client. Fig. 5 gives the software structure of the system at runtime. As shown, there are four Java applets running on the client for handling the necessary communication with the server and plotting the response curve using the data obtained from the server. A fifth applet communicates with a server program by sending it command strings for camera zoom and pan/tilt control. An ActiveX control executes on the client and receives audio and video data from the video server.

On the server, there are four programs handling communication with the client and receiving the controller parameters and the reference input (set point) values input by the user. These parameters are passed on to the controller modules (coded in LabView G) which implement the local control of the coupled tank. In the case of manual control, however, the received voltage values are directly fed to the coupled tank. The program for controlling the mounting unit of the camera is coded in Visual Basic and also resides on the same machine. The functionality of the LabView G programs for receiving the parameters from the user is now described.

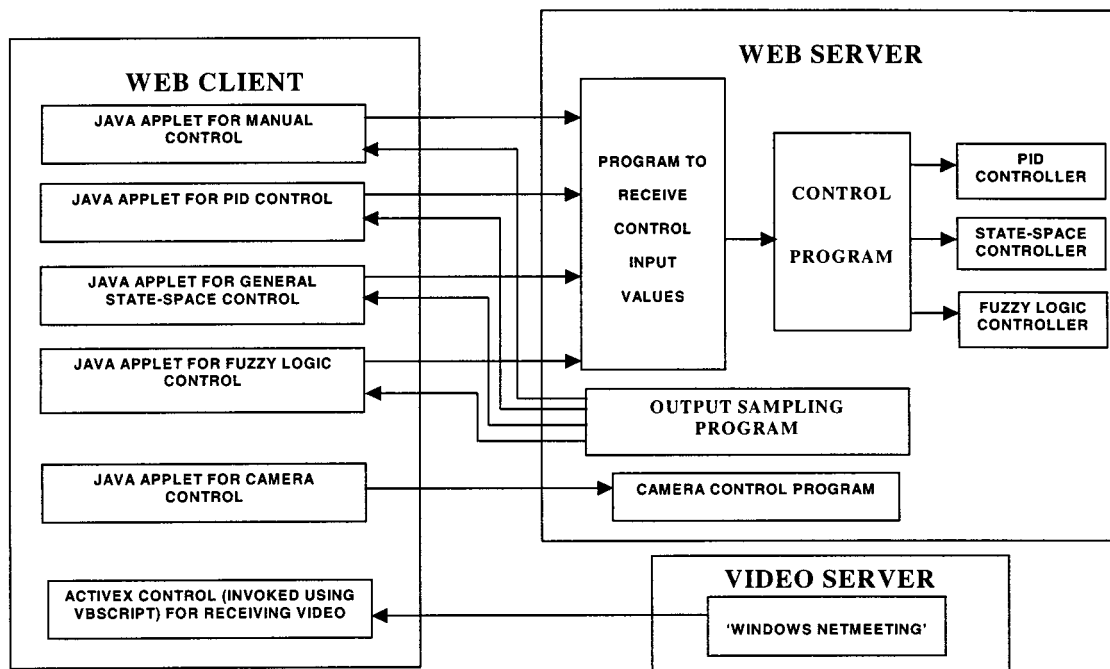


Fig. 5. Software structure of the virtual laboratory.

LabView supports network communication protocols, such as TCP and User Datagram Protocol (UDP), implemented in the form of Virtual Instruments' sub VI's. LabView uses the following VI's to communicate through the Internet.

- 1) The TCP Listen.vi: This VI waits for an incoming TCP connection request. It also returns a connection ID when the TCP connection is created.
- 2) The TCP Read.vi: This VI receives a specified maximum number of bytes to read from the specified TCP connection.
- 3) The TCP Write.vi: This VI writes the string data to the specified TCP connection.
- 4) The TCP Close Connection.vi: This VI is used to release the TCP connection.

The server's G programs receive information on the two set points and the parameters for the various controllers and pass these on to the two controller blocks (one for each input of the coupled tank). The output of each controller block is converted to an analog voltage by the DAQ card and fed to the coupled tank. The two output analog signals from the coupled tank are sampled and the voltage information is sent to the client. At the client, the voltage information is then used to draw the curve corresponding to the system response by means of a Java applet.

### III. MODELING THE COUPLED TANK APPARATUS

As shown in Fig. 6, the coupled tank apparatus is designed for the teaching and research of process control principles. It consists of two perspex tower-type tanks mounted above a reservoir which stores water. The head of water in the tanks can be read from the scale in front of the tank. Each tank is fitted with an outlet, which is in turn connected to a plastic hose for returning water to the reservoir. The outflow rate of water returning to the

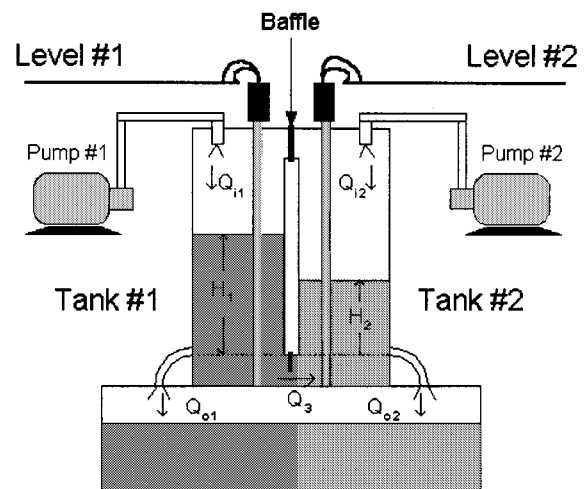


Fig. 6. Coupled tank apparatus.

reservoir is approximately proportional to the head of water in the tank. The water level in the tanks is monitored by two capacitance probes, each of which, along with some electronic circuits, provides an output signal proportional to the level of water in the corresponding tank. This output voltage is in the range of 0 to +10 V. Complex tank arrangements are possible by varying the level of the internal baffle, which controls the inter-tank resistance. The controlling PC feeds two analog input signals to the coupled tank through the DAQ card. The two analog outputs corresponding to the water level in the tanks are sampled at a rate set by the user.

In general, with the baffle raised, the coupled tank can be modeled as a two-input-two-output system. The nonlinear and linearized state-space models of the coupled tank are presented below.

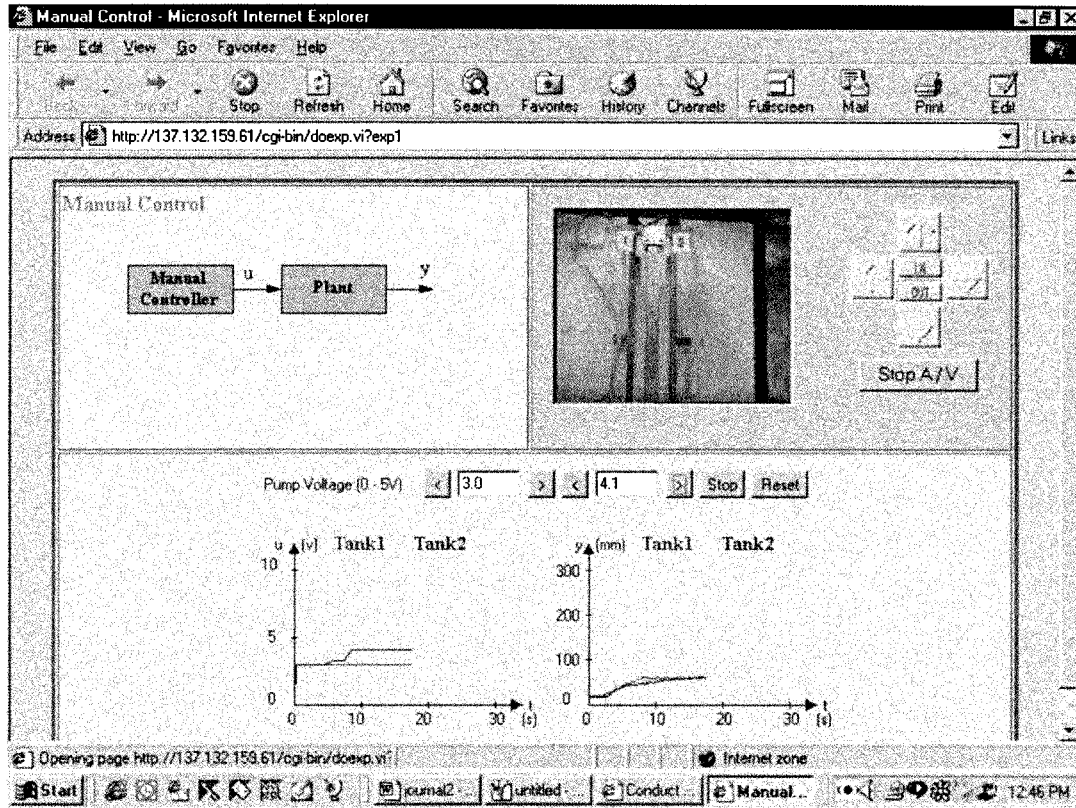


Fig. 7. User interface for manual control.

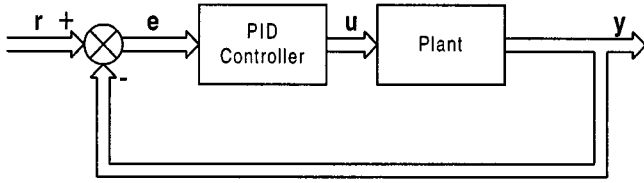


Fig. 8. Block diagram of experimental system using a PID controller.

Specifically, the various parameters of the system are

$H_1, H_2$  = height of fluid in the tanks

$A_1, A_2$  = cross-sectional areas of the tanks

$Q_{i1}, Q_{i2}$  = pump flow rate into tanks

$Q_{o1}, Q_{o2}$  = flow rate of fluid out of the tanks

$Q_3$  = flow rate of fluid between the tanks.

Using Bernoulli's equation for steady nonviscous incompressible flow (which states that the outlet flow in each tank is proportional to the square root of the head of water in the tank), we have

$$Q_{o1} = \alpha_1 \sqrt{H_1} \quad (1)$$

$$Q_{o2} = \alpha_2 \sqrt{H_2}. \quad (2)$$

Similarly, as the flow between the two tanks is proportional to the square root of the head differential

$$Q_3 = \alpha_3 \sqrt{H_1 - H_2}. \quad (3)$$

The proportionality constants,  $\alpha_1$ ,  $\alpha_2$ , and  $\alpha_3$ , in the above equations are dependent on the coefficient of discharge, the cross sectional area of each orifice, and the gravitational constant. The coupled tank apparatus with two inputs and two outputs and the baffle completely raised can be modeled in state space as follows.

Combining (1)–(3), the following set of nonlinear, state-space equations representing the system dynamics of the coupled tank can be obtained

$$\begin{aligned} A_1 \frac{dH_1}{dt} &= Q_{i1} - \alpha_1 \sqrt{H_1} - \alpha_3 \operatorname{sgn}(H_1 - H_2) \sqrt{|H_1 - H_2|} \\ A_2 \frac{dH_2}{dt} &= Q_{i2} - \alpha_2 \sqrt{H_2} - \alpha_3 \operatorname{sgn}(H_2 - H_1) \sqrt{|H_1 - H_2|}. \end{aligned} \quad (4)$$

Furthermore, the dynamics of the pumps connected to the tanks can be approximated by

$$\begin{aligned} Q_1 &= k_1 u_1 + c_1 \\ Q_2 &= k_2 u_2 + c_2 \end{aligned} \quad (5)$$

where  $Q_1$  and  $Q_2$  are the inflow rates of tanks 1 and 2,  $u_1$  and  $u_2$  are the control signals for pumps 1 and 2, and  $k_1, c_1$  and  $k_2, c_2$  are the associated constants for pumps 1 and 2, respectively.

Thus, if the coupled tank is controlled with the prefeedback law

$$\begin{aligned} Q_{i1} &= \left[ \alpha_1 \sqrt{H_1} + \alpha_3 \operatorname{sgn}(H_1 - H_2) \sqrt{|H_1 - H_2|} - c_1 \right] \\ &\quad + (k_1 u_1 + c_1) \\ Q_{i2} &= \left[ \alpha_2 \sqrt{H_2} + \alpha_3 \operatorname{sgn}(H_2 - H_1) \sqrt{|H_1 - H_2|} - c_2 \right] \\ &\quad + (k_2 u_2 + c_2) \end{aligned} \quad (6)$$

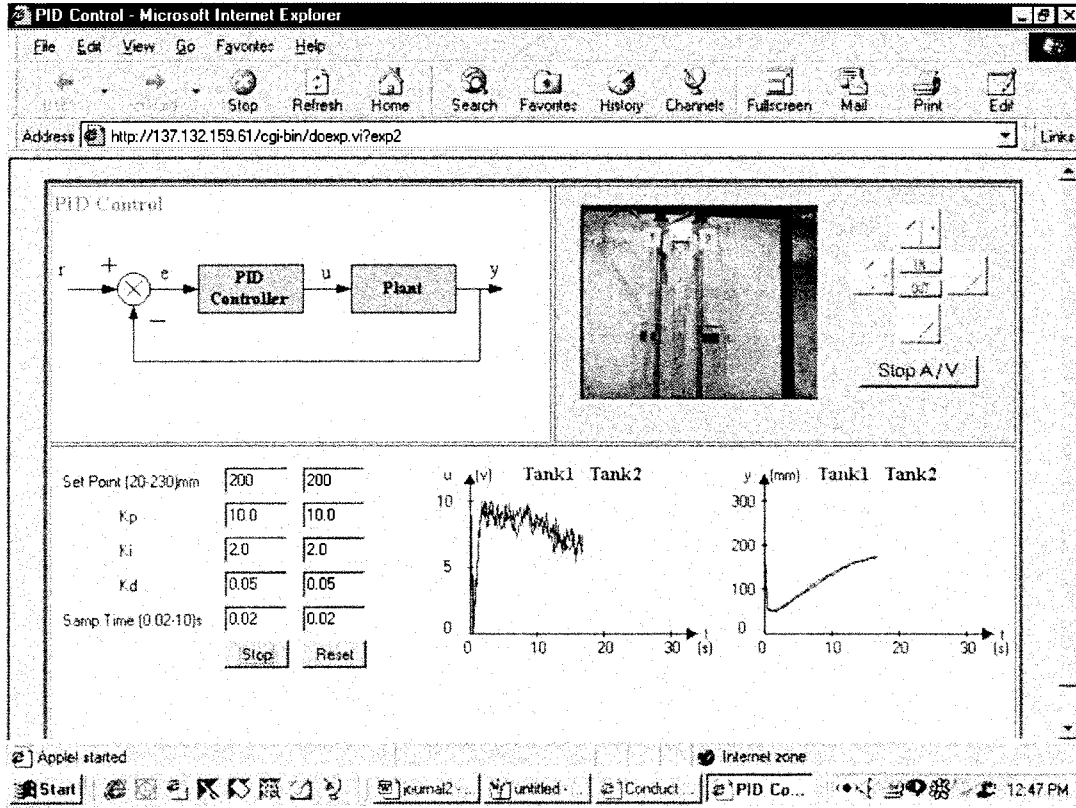


Fig. 9. User interface for PID control.

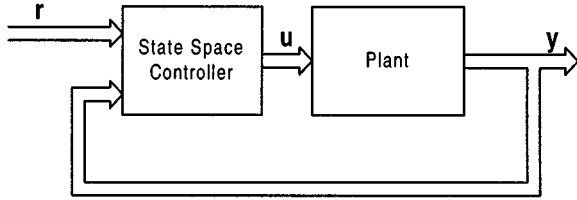


Fig. 10. Block diagram of experimental system using a state-space controller.

the system can be simplified as

$$\begin{aligned} \frac{dH_1}{dt} &= \frac{k_1}{A_1} u_1 \\ \frac{dH_2}{dt} &= \frac{k_2}{A_2} u_2. \end{aligned} \quad (7)$$

Note that the above procedure is really a good example of feed-back linearization, as the resulting system dynamics in (7) is linear.

#### IV. CONTROLLING THE COUPLED TANK APPARATUS

Four different types of controllers which have been implemented will now be described.

##### A. Manual Control

Manual control consists of directly feeding the control inputs given by the user to the pumps without the use of any automatic controller. The user is expected to adjust the values of the control inputs to achieve the desired water levels in the tanks. The user interface for manual control is shown in Fig. 7. Note that this can be used to determine critical frequencies of the process,

which will in turn provide useful information for selecting various gains in PID control.

##### B. PID Control

Fig. 8 shows the block diagram of the overall system, which comprises of the PID controller and the plant or coupled tank. Here  $r$  is the reference input or the set points for the water levels in the tanks. The continuous time response of a PID controller is given by

$$u = K_p e + K_i \int e dt + K_d \dot{e} \quad (8)$$

where  $e = r - y$  is the tracking error,  $K_p$ ,  $K_i$ , and  $K_d$  are the proportional, integral, and derivative gains, respectively. In the coupled tank experiment, the PID controller is realized through a software program and the appropriate discrete-time input is

$$u(k) = K_p e(k) + K_i T_s \sum_{j=0}^k e(j) + K_d \frac{e(k) - e(k-1)}{T_s}. \quad (9)$$

While PID control is selected, the user is expected to input the values of the two reference inputs (set points) and the values of the proportional, integral and differential gains. These values can be obtained by tuning the controller using various established procedures. The user interface for PID control is shown in Fig. 9.

##### C. Generalized State-Space Control

Fig. 10 shows the block diagram for the control of the coupled tank using a generalized controller modeled in state space. Here,  $r$  is the reference input,  $y$  is the measured output of the coupled-

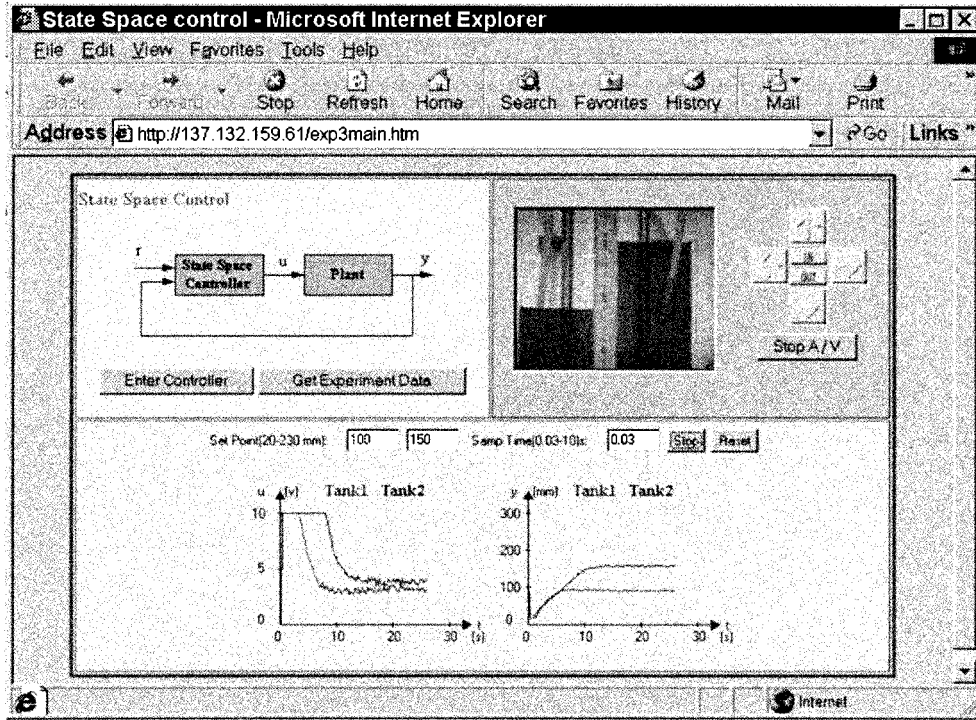


Fig. 11. User interface for general state-space control.

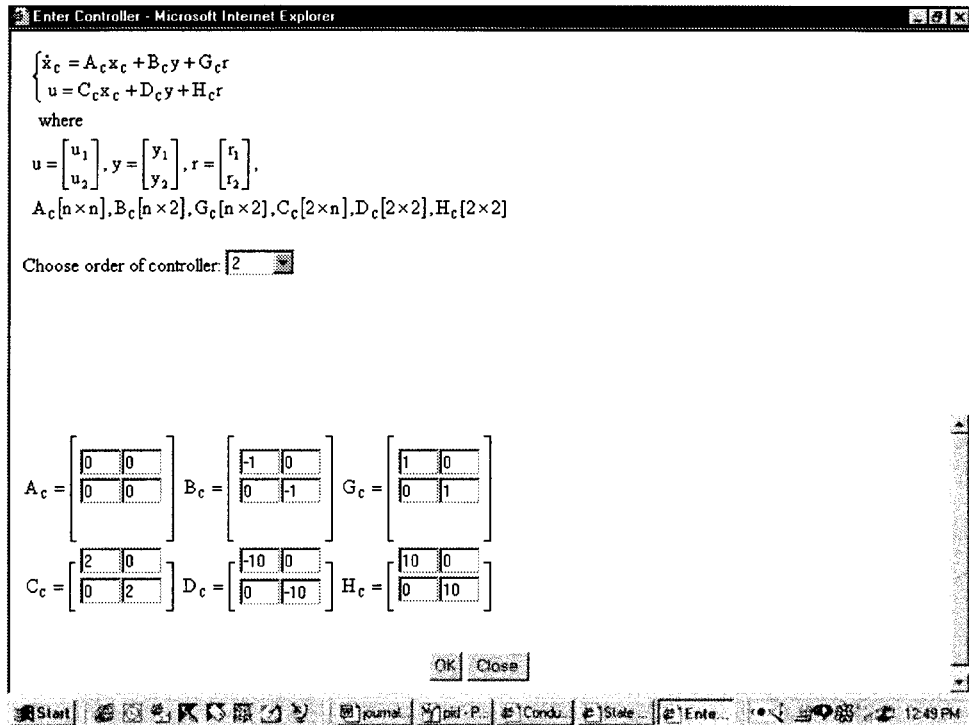


Fig. 12. User interface for input of matrices.

tank, and  $u$  is the control input. The state model of the controller is given by

$$\dot{x}_c = A_c \dot{x}_c + B_c y + G_c r \quad (10)$$

$$u = C_c x_c + D_c y + H_c r \quad (11)$$

where  $x_c$  is the state of the controller, and  $A_c$ ,  $C_c$ ,  $D_c$ ,  $H_c$ ,  $B_c$ , and  $G_c$  are constant matrices of appropriate dimensions.

The user interface for performing general state-space control is shown in Fig. 11. In conducting experiment using this general state-space controller structure, the user is first prompted to enter the order of the controller that he or she has designed. Based on this value, a user interface will automatically appear for the user to enter the various matrices. Fig. 12 shows the interface when the order is chosen to be two. The data entered will be

sent to the software controller on the server once the user clicks the “OK” button.

#### D. Fuzzy Knowledge-Based Control

A fuzzy knowledge-based controller (FKBC) has also been developed to control the coupled tank. Fig. 13 shows the system block diagram, whereas Fig. 14 shows the principal components of the FKBC.

As shown, the three principal blocks of the FKBC are the fuzzification module, the inference engine and the defuzzification module. The fuzzification module includes a normalization block for mapping the physical values of the control state variables onto the normalized domain. This block also maps the normalized control output variable onto its physical domain. The knowledge base of the fuzzy controller consists of a database and a rule base. The former includes information on the membership functions and the normalization/denormalization scaling factors. The rule base represents the control policy adopted. For implementing fuzzy control on the coupled tank, the process state variables are the error ( $e$ ) and the change-of-error ( $\dot{e}$ ). The control output variable is the control output ( $u$ ). The term sets of  $e$ ,  $\dot{e}$  and  $u$  are denoted by  $L(e)$ ,  $L(\dot{e})$  and  $L(u)$ , respectively. They contain the same set of linguistic values, i.e., NL (negative low), ZR (zero) and PL (positive low). Thus

$$L(u) = L(e) = L(\dot{e}) = \{NL, ZR, PL\}$$

The user interface developed for fuzzy logic control is shown in Fig. 15. The default rule base shown in two tables at the bottom right corner can be modified by the user by clicking on the table entries. The membership functions for err (error), derr (derivative of error), and out, (output) can be modified by clicking on the *err*, *derr*, and *out* buttons respectively. On clicking these buttons a new window pops up as shown in Fig. 16 to enable the user to edit the membership function.

By clicking on the *Left*, *Middle*, and *Right* buttons, the user can choose the function to be edited. A vertical line then appears which can be dragged using the mouse to the desired position. The default membership functions can be restored by pressing the *Default* button.

The inference engine has been realized to use individual rule-based inference. The normalization/denormalization scaling factors used are 10.0, 50.0, and 1.0 for  $e$ ,  $\dot{e}$  and  $u$ , respectively. Current efforts are focused on developing and implementing a learning algorithm for the rule base, membership functions, and scaling factors.

#### V. IMPACTS OF VIRTUAL LABORATORIES IN TEACHING AND RESEARCH

The virtual laboratory described in this paper is currently being utilized in the teaching of both undergraduate and postgraduate courses in the Department of Electrical Engineering, National University of Singapore. The laboratory session for the undergraduate course consists of identifying a physical model for the coupled-tank system based on input–output data, as well as designing a PID controller and different fuzzy logic controllers (such as Takagi-Sugeno, Mamdani, etc) for the system. Due to the complexity in tuning the PID controller parameters,

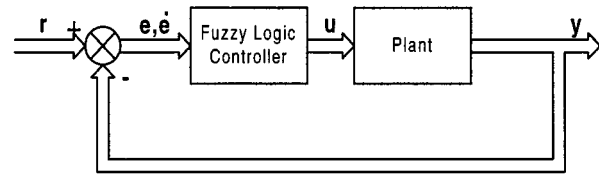


Fig. 13. Block diagram of experimental system using a fuzzy logic controller.

the membership functions, and the rule-bases of the fuzzy controller, students are often short of time in implementing their controllers within the scheduled laboratory session of 3 h.

Through the virtual laboratory, however, each student is now able to carry out the laboratory according to his/her own pace and time schedule. The manual control in the virtual laboratory also allows students to collect real-time coupled-tank input–output data for the identification of a physical model. This is particular useful since an accurate model would help in the design of a good closed-loop controller. Besides, students could also interplay or fine-tune the controller parameters and membership functions, and the resulted control performance could be visualized and compared since they are based upon the same virtual laboratory platform.

A postgraduate course on optimal control systems also incorporates the virtual laboratory in its formal teaching. In this course, students are introduced to solve both classical optimal control system design problems, such as linear quadratic regulator (LQR) control and linear quadratic Gaussian (LQG) control, and advanced robust control problems, such as  $H_2$  optimal control and  $H_\infty$  control. Since half the class consists of part-time students holding full-time employment in industry, classes are conducted in 3-h sessions in evenings once a week. Due to security, manpower, and other constraints, teaching laboratories are generally closed after working hours and it was almost impossible to schedule laboratory experiments or to assign homework assignments on hardware implementations in the past.

With the help of the virtual laboratory, which is available 24 h a day, the course lecturer is now able to introduce a new teaching element on actual experimentation. The virtual laboratory is used throughout the course homework assignments, in which students are required to solve an actual problem on the control of the flow levels in the coupled-tank system using techniques learned in the class, and implement their designs on the actual system through the Internet. To the part-time students who can logon and use the system whenever they are free, the virtual laboratory provides a highly flexible access to a real experiment. Feedback and comments from the students are generally very positive. Many students feel that it is superb to see that the controllers they have designed actually work in the real system.

The virtual laboratory offers an excellent and convenient platform for researchers to test and implement their new algorithms as well. As an example, based on a robust and perfect tracking technique developed recently in [13], a controller for the coupled-tank system in the virtual laboratory has been implemented by the authors. While the controller is theoretically capable of tracking any reference signal with arbitrary fast settling time



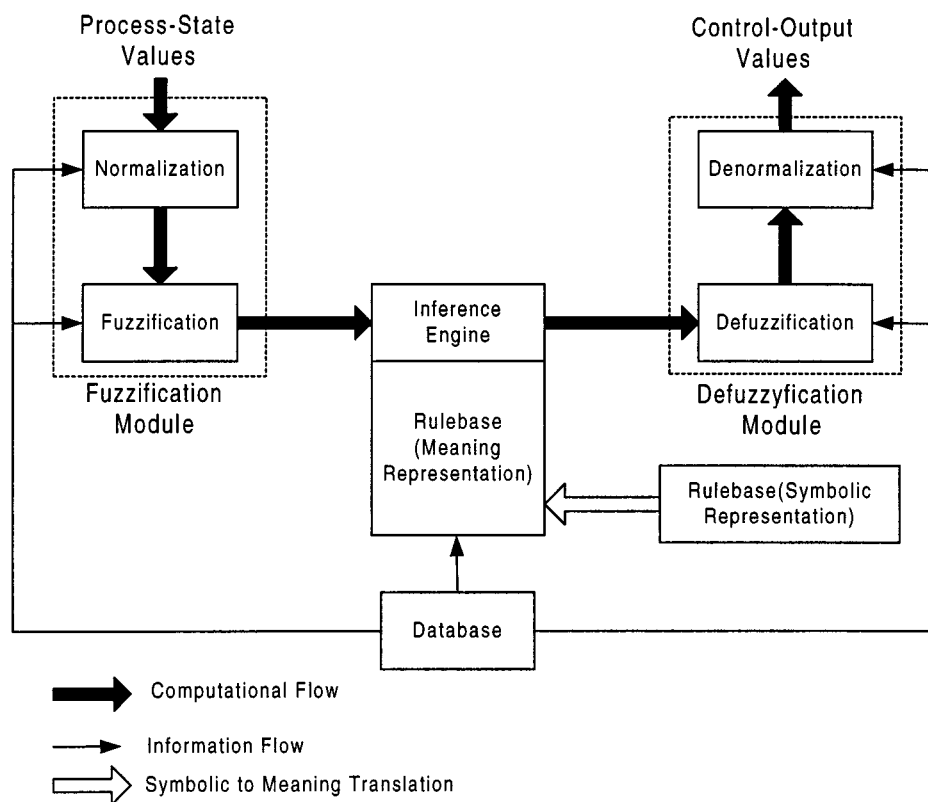


Fig. 14. Principal components of fuzzy knowledge-based controller.

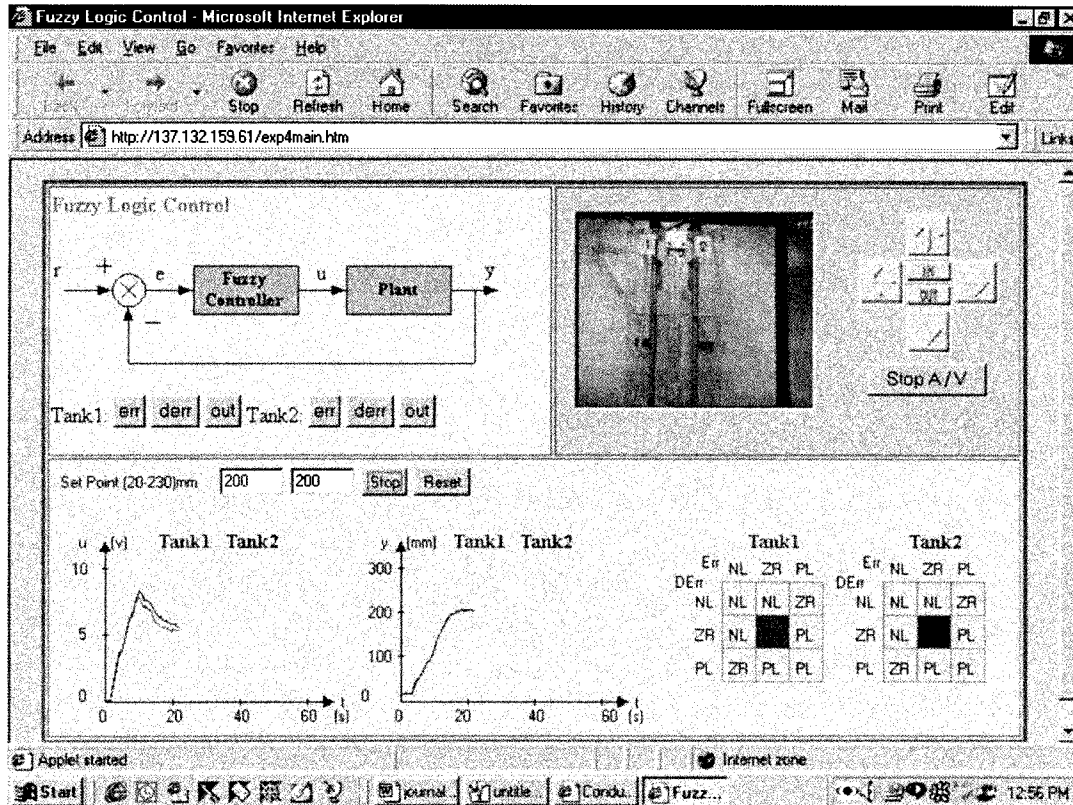


Fig. 15. User interface of fuzzy knowledge-based control.

from any initial condition, it requires in general an infinite gain to guarantee such a performance. With the flexibility of the vir-

tual laboratory, it is simple and easy to tune certain design parameters to meet the control constraints of the physical system

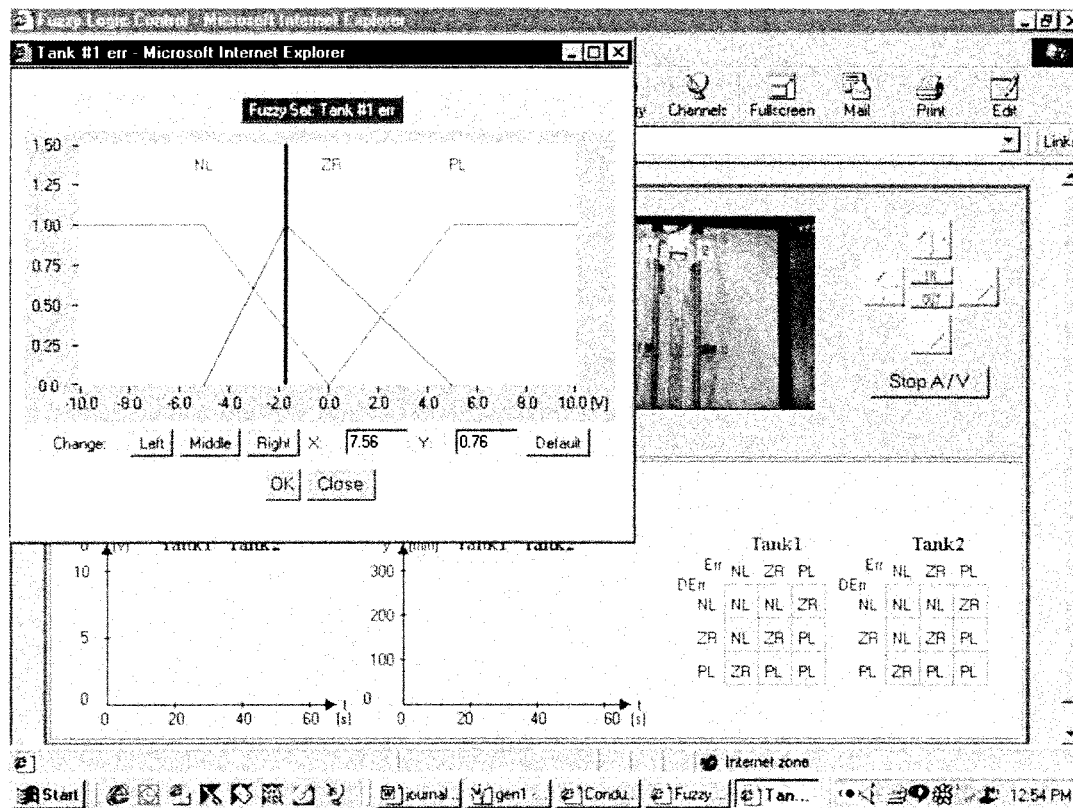


Fig. 16. User interface for editing membership functions.

with some tradeoff in the overall performance. The design was very successful and the results have been submitted for possible publication. Last, it should be emphasized that the virtual laboratory enables making performance comparison of different techniques very easily and meaningfully, as the various techniques are all implemented under the same setting.

## VI. CONCLUSION

This paper has described the design and development of a web-based laboratory experiment on a coupled tank apparatus and discussed its impact from both a teaching and research points of view. New and attractive features, such as the use of video conferencing for providing audio-visual feedback to the user and the provision for adjustment of the pan/tilt and zoom of the camera capturing the real-time video, have been incorporated. Strategies that have been implemented include manual control, PID control, general state-space control, and fuzzy control. Accessible at <http://vlab.ee.nus.edu.sg/vlab/control/>, the virtual laboratory has been used for the teaching of undergraduate and postgraduate courses as well as for research purposes. In teaching, the possibility of anytime anywhere access allows undergraduate students who are not able to finish the actual experiment to be able to continue to work on the subject after the formal laboratory session on their own time through the Internet. For part-time postgraduate students, the system solves security and manpower problems associated with the operation of teaching laboratories in the evening. The virtual laboratory provides the course lecturer

with the possibility to use a real couple-tank apparatus as a principal teaching tool for homework assignments for the first time. In research, the system has the advantage that important parameters can be remotely tuned easily. Also, performance comparison with different techniques can be obtained very easily and meaningfully, as the various techniques are all implemented under the same setting.

## REFERENCES

- [1] 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," *Inst. Elect. Eng. Sci. Educ. J.*, vol. 9, no. 2, pp. 69–76, 2000.
- [2] M. Shor and A. Bhandari, "Access to an instructional control laboratory experiment through the World Wide Web," in *Proc. 1998 Amer. Contr. Conf.*, Philadelphia, PA, 1998, pp. 1319–1325.
- [3] M. Shaheen, K. A. Loparo, and M. R. Buchner, "Remote laboratory experimentation," in *Proc. 1998 Amer. Contr. Conf.*, Philadelphia, PA, 1998, pp. 1326–1329.
- [4] "Web-based virtual laboratory, Nat. Univ. Singapore," <http://vlab.ee.nus.edu.sg/vlab>.
- [5] 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," in *Proc. 1999 IEEE Hong Kong Symp. Robot. Contr.*, Hong Kong, July 1999, pp. 756–760.
- [6] LabView, *User Manual*: National Instruments, 1998.
- [7] Coupled Tank Control Apparatus, *Model: PP-100, Operator and Service Manual*. Singapore: Kent Ridge Instruments, 1999.
- [8] *Windows NetMeeting 3.0 Help*: Microsoft Corporation, 1999.
- [9] *RealServer G2 Help*: RealNetworks, 1999.
- [10] *NetShow 3.0 Documentation*: Microsoft Corporation, 1999.
- [11] "Home page of NUSNET-III," <http://www.cc.nus.edu.sg/>.
- [12] *Windows NetMeeting Resource Kit*: Microsoft Corporation, 1999.
- [13] B. M. Chen, *Robust and  $H_\infty$  Control*. New York: Springer-Verlag, 2000.

**C. C. Ko** (M'82–SM'93) received the B.Sc. (1st Class Honors) and Ph.D. degrees in electrical engineering from Loughborough University of Technology, U.K.

In 1982, he joined the Department of Electrical Engineering, National University of Singapore, where he is presently an Associate Professor and Deputy Head (Research) of the Department. His current research interests include digital signal processing, adaptive arrays and mobile communications and he has written more than 150 technical publications in these areas.

Dr. Ko has served as an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING.

**Ben M. Chen** (S'89–M'92) was born in Fuqing, Fujian, China, on November 25, 1963. He received the B.S. degree in mathematics and computer science from Amoy University, Xiamen, China, in 1983, M.S. degree in electrical engineering from Gonzaga University, Spokane, Washington, in 1988, and Ph.D. degree in electrical and computer engineering from Washington State University, Pullman, WA, in 1991.

He was a Software Engineer from 1983 to 1986 in the South-China Computer Corporation, China, and was a Postdoctoral Associate from 1991 to 1992 at Washington State University. He was with the Department of Electrical Engineering, State University of New York at Stony Brook, from 1992 to 1993, as an Assistant Professor. Since August 1993, he has been with Electrical Engineering Department, the National University of Singapore, where he is currently an Associate Professor. His current research interests are in linear control and system theory, control applications, development of internet-based virtual laboratories and internet security systems. He is the author of the books, *Robust and H-infinity Control* (London: Springer, 2000), and *H-infinity Control and Its Applications* (London: Springer, 1998), and coauthor of the books, *Loop Transfer Recovery: Analysis and Design* (London: Springer, 1993), *H2 Optimal Control* (London: Prentice Hall, 1995), and *Basic Circuit Analysis* (Singapore: Prentice Hall, 1st Ed., 1996; 2nd Ed., 1998).

Dr. Chen was an Associate Editor in 1997–1998 on the Conference Editorial Board of IEEE Control Systems Society. He currently serves as an Associate Editor of IEEE TRANSACTIONS ON AUTOMATIC CONTROL.

**Jianping Chen** was born in Fujian, China in 1971. He received the B.Eng. degree from Shanghai Jiaotong University in 1993 and M.Eng degree from Xiamen University in 1999 respectively, both in control theory. From 1993 to 1996, he researched on communication engineering in South-Cube High-Tech Co. Ltd, Xiamen.

From January 1998 to June 1999, he researched on intelligent control and robotics in the National Key Laboratory of Intelligent Science and System, which is located in the Computer Science Department of Tsinghua University. His current research interest is network security and intelligent control.

**Yuan Zhuang** was born in Anhui, China, in 1975. He received the B.Eng. degree in electrical engineering from Zhejiang University, China, in 1997. Since 1998, he had been pursuing the M.Eng. degree at in the National University of Singapore.

His current research interest is remote instrument control, distributed system application and web technology development.

**Kay Chen Tan** (S'95–M'99) received the B.Eng. (First-class honors) and Ph.D. degrees in electrical and electronic engineering from the University of Glasgow, U.K., in 1994 and 1997, respectively.

He was with the Centre for Systems and Control (CSC) and the Evolutionary Computing Group at Glasgow before joining the Department of Electrical and Computer Engineering in National University of Singapore (NUS) as an Assistant Professor in 1997. He has published more than 30 technical papers and served as an IPC Member for many international conferences. His research interests include evolutionary computation, multiobjective optimization, intelligent control, and design automation.