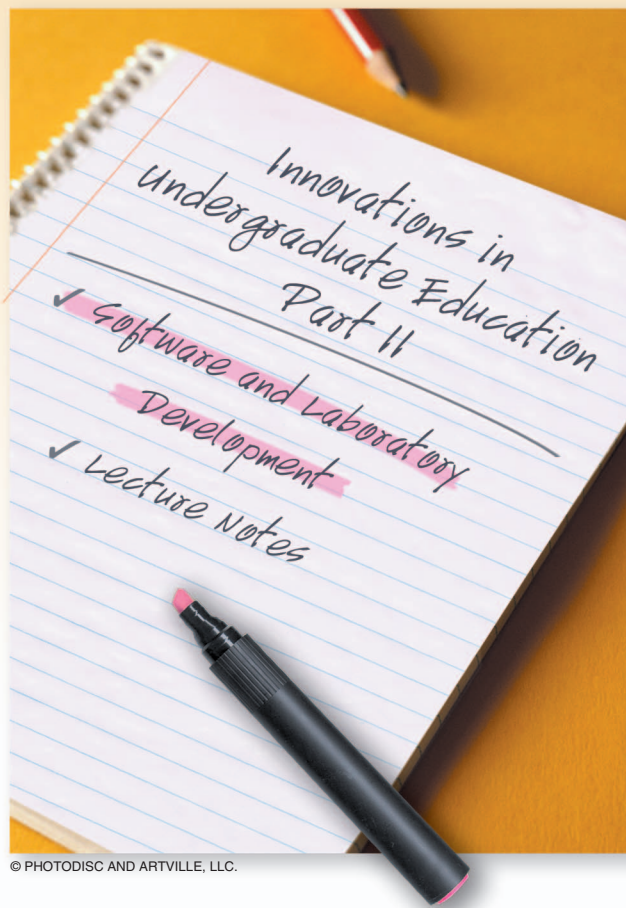# Virtual and Remote Control Laboratory Development

The approach has two main goals: simplicity for the lecturer and low cost for the student

© PHOTODISC AND ARTVILLE, LLC.

T he relationship between information technologies and process control has reached a new stage [1], encouraging the creation of applications such as monitoring and control through the Internet, as well as teleworking, telemedicine, and telerobotics. With regard to process control, information technologies were originally used in the field of education [2], where the Internet increases flexibility in the educational process regarding schedules and physical spaces [3] and facilitates remote control applications.

Issues of complexity, cost, and Internet delays [1] limit the ability to implement a remote control plant. This article proposes a solution to these problems, especially for institutions where MATLAB is the standard control design tool. The proposed solution supplements MATLAB by using a virtual process that allows video animations of simulated
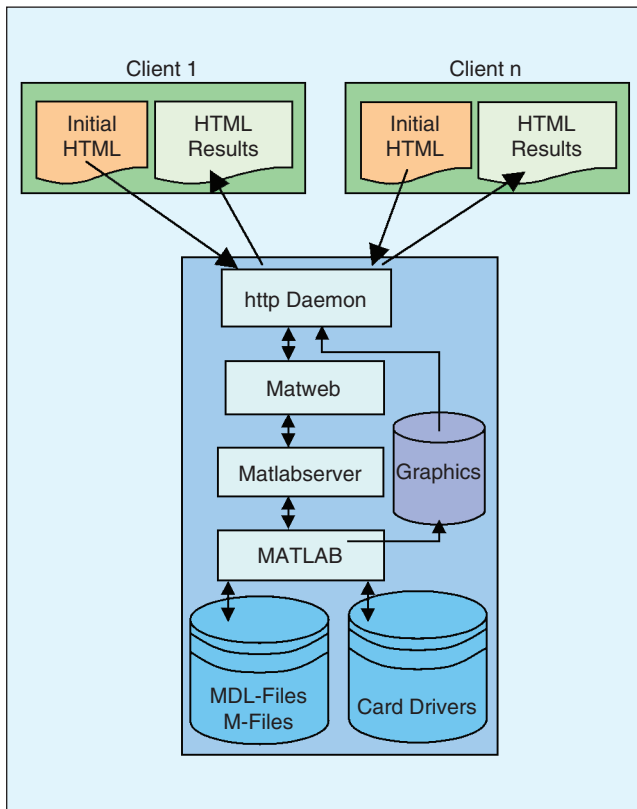
By Angel Valera, José Luis Díez, Marina Vallés, and Pedro Albertos
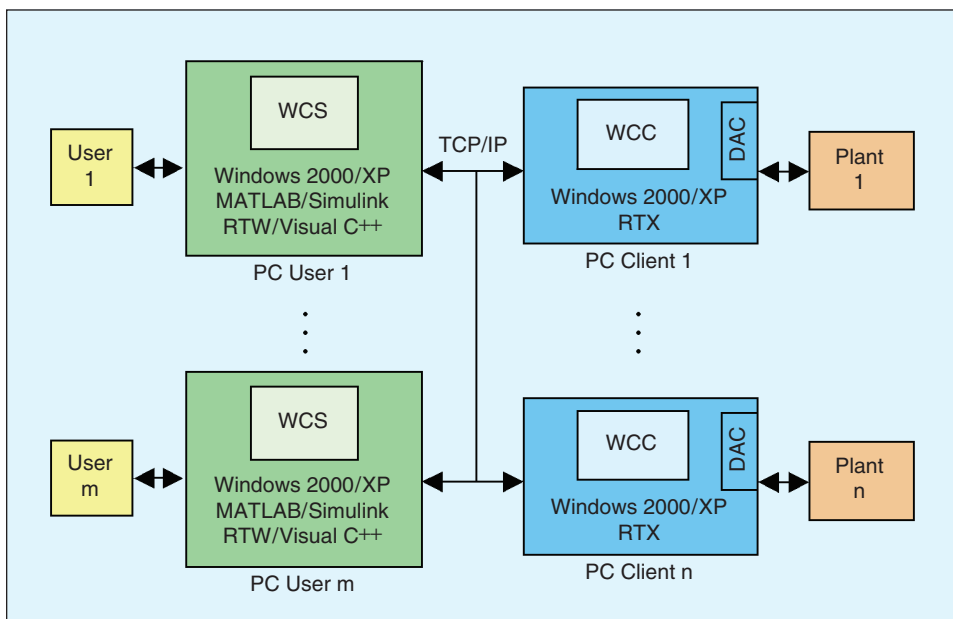
processes and remote control. The solution focuses on two main goals: simplicity for the service provider, that is, the lecturer, and low cost for the remote user, that is, the student.

MATLAB-Based Configurations for Virtual and Remote Process Control

Based on the Internet, either a virtual learning laboratory or a remote distance-learning laboratory is available for setting up a laboratory in a learning environment. A virtual laboratory allows continuous access to a simulated process on a computer. Remote distance-learning laboratories are an option halfway between traditional and virtual laboratories, allowing remote users to perform real experiments. To implement virtual and remote laboratories, we have used two integrated MATLAB-based software packages, namely, MATLAB Web Server (by Mathworks) and WinCon (by Quanser Consulting).

Figure 1. *MATLAB Web Server Configuration. To execute MATLAB applications over the Internet, the MATLAB Web Server configuration requires an HTML file for data submission to MATLAB, a Simulink model or an M-file to process input data and calculate results, and an HTML output document to return results to the user.*

MATLAB Web Server (MWS) is an integrated software package enabling MATLAB programmers to create applications that use the capabilities of the World Wide Web to send data to MATLAB for computation and to display the results on a Web browser [4]. For a thorough explanation of the main features of MWS and the commands used in the development of applications, see [5], where the configuration for the authors' equipment is described in detail.

However, the MWS software provided by Mathworks does not allow online access to low-level hardware and thus cannot be used to control remote laboratories or industrial processes. To allow remote laboratories and remote control in real systems, an additional configuration with several data acquisition cards is necessary, and subsequently, new routines must be programmed to access these cards from the MATLAB environment (see Figure 1). These routines are created by programming a set of functions in C++ to provide A/D conversion, D/A conversion, encoder signal management, digital input/output access, and timer programming. The MATLAB executable functions are obtained in the form of DLL files with the *mex* command, using the Visual C++ v6.0 compiler.

An alternative to MATLAB and MWS is the use of WinCon, which can run code generated by a Simulink diagram in real time on a remote PC [6]. WinCon enables data to be plotted online using real-time code and allows on-the-fly changes of model parameters by means of WinCon Control Panels or Simulink. In fact, WinCon software consists of WinCon Client (WCC) and WinCon Server (WCS), which communicate by using the TCP/IP protocol. Figure 2 shows the WinCon configuration.

Virtual Laboratories

MWS provides a straightforward approach to building virtual laboratories using MATLAB's M-files. The M-file for virtual applications is a standard MATLAB function that accepts HTML input documents (see Figure 3) and returns HTML output documents. Any existing M-file can be adapted to an MWS format, thus making it available over the Internet. Furthermore, virtual applications can be developed using Simulink models. In this case, an MDL-file and an M-file are necessary; the MDL-file is a standard Simulink model with all of the simulation blocks, while

Figure 2. *WinCon configuration. In the remote configuration, each WinCon Server (WCS) can communicate with several WinCon Clients (WCCs), and vice versa.*

the M-file updates the model parameters and starts the simulations. To update the model, the M-file must open the model and update the system parameters using the set_param command. Afterwards, the sim command runs the Simulink model.

Although some remote laboratories allow users to control a real process without using special software tools or a real system, access to the laboratory is permitted to only one user at a time. This limitation can be overcome by using virtual processes [5]. A virtual process is defined as a set of software functions and procedures simulating a real process response using, for instance, standard video .avi files. MATLAB commands such as mov2avi, avi2mov, avifile, addframe, or getframe allow animations to be generated and visualized on any other PC. Therefore, students can analyze system features such as stability and steady-state error, in an intuitive way, by playing a downloadable .avi file.
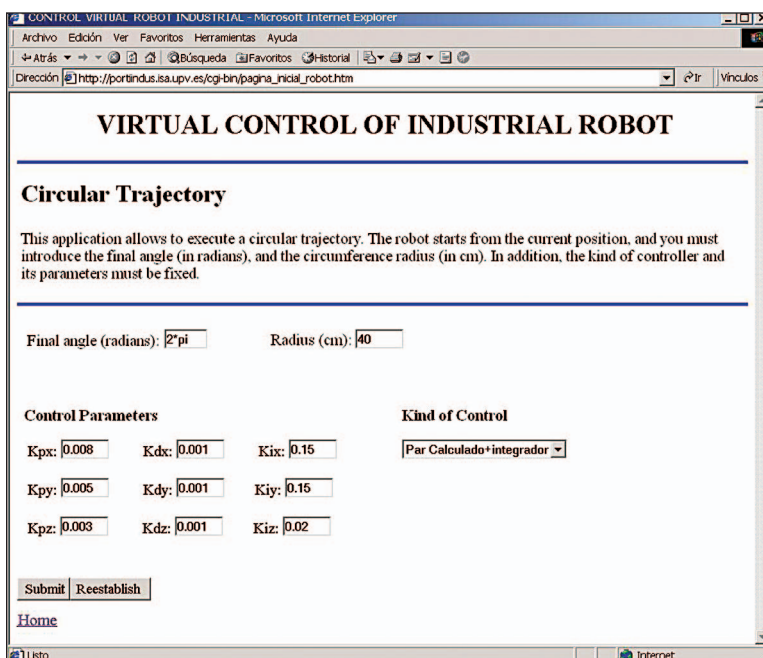
A sample virtual process control HTML output page (Figure 4) shows a virtual dc motor process based on the Servomotor SFT 154 manufactured by Feedback Instruments Ltd. The signals representing the motor position and control action evolve simultaneously with the motor cursor. Users can analyze whether the system response is underdamped or overdamped, as well as the maximum overshoot and settling time.

Several virtual laboratories, such as IIR Digital Filter Design and Controller Design, can be accessed through [7], which is currently available only in Spanish. Any student enrolled in the course can access the virtual laboratory and perform their work online. The only requirement needed is an Internet-connected PC and a Web browser. The cost is kept low for students since there is no need for a MATLAB version to be installed on the remote computer.

Remote Laboratories

The remote laboratories can be operated in batch or online. Online operation involves control algorithms on the remote computer, with control and sensor data transmitted over the Internet. The reference signal and parameters can be changed while the experiment is being conducted, but random delays appear as a consequence of Internet traffic. The batch operating option avoids Internet delays because the reference and controller parameters are sent to the server before the experiment starts, and, once finished, the process output is sent to the remote computer. Both

approaches are described in the following. Batch laboratories are based on MWS, including a number of supplementary functions and online laboratories, which are not supported by MWS, are based on the use of WinCon.

Figure 3. *Input page example for a virtual laboratory. Once the parameters for the virtual laboratory have been introduced into the input page by the user, the server processes the data.*
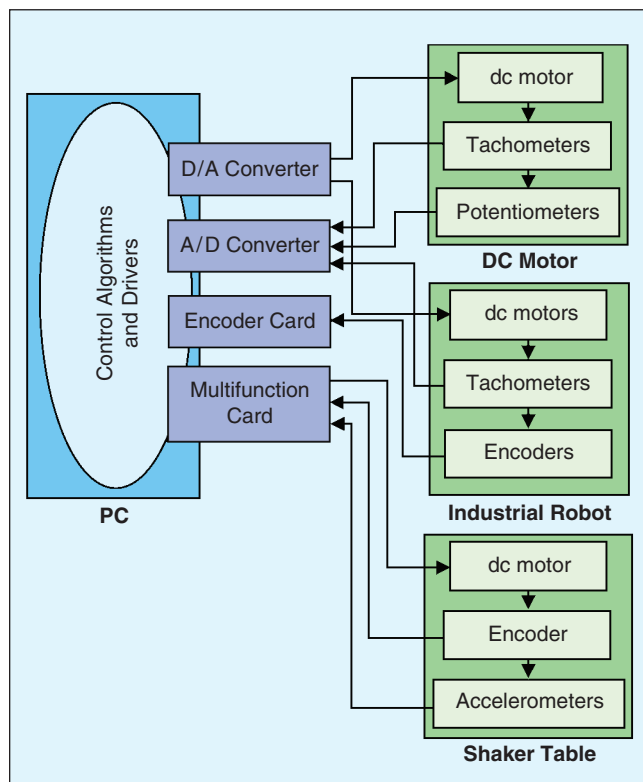

Figure 4. *Output page example for a virtual laboratory. The virtual laboratory results processed by the server are sent to an HTML output page. From this output page, learners can either repeatedly execute or download the .avi file, analyzing the system response in a simple, fast, and intuitive manner with a mouse click.*

Although other physical processes have been used and tested, this article focuses on a dc motor, an industrial robot, and a shaker table. Figure 5 illustrates the general laboratory setup for the experiments, using Advantech and

Based on the Internet, either a virtual learning laboratory or a remote distance-learning laboratory is available.

Quanser Consulting industrial data acquisition cards. The processes are continuously running, are periodically supervised by specialized technicians, and are hardware protected to avoid inadequate control inputs.

The dc motor and industrial gantry robot are remotely controlled in batch by MWS. For controller implementation, HTML files for data submission have been programmed
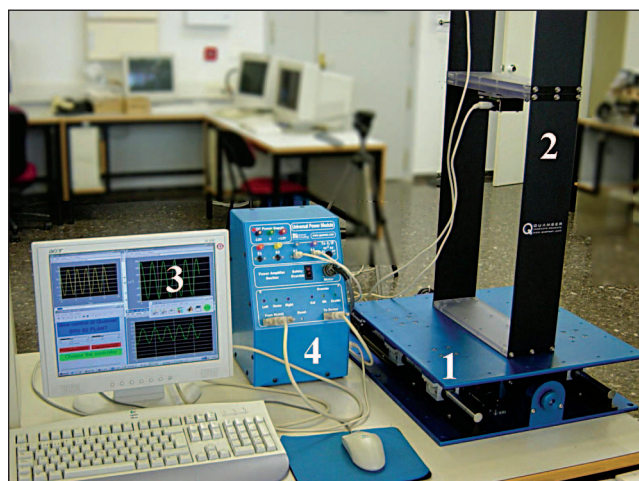
into the system to specify all M-file variables, and a submit input has been implemented to submit data to the program. The initial Web page submits reference values, controller parameters, and experiment time for MATLAB. A video hyperlink is included in the experiment's main page.

Finally, an online remote control application using WCS, applied to Shaker Table II by Quanser Consulting, has been implemented. This process is composed of a table and a two-floor structure. The table is controlled by a dc brushless motor. An encoder collects the table position, and the system acceleration is measured by three accelerometers, one mounted on the table, and two mounted on different floors. Several kinds of controllers have been implemented, including proportional-integral differential, state-space, and nonlinear controllers. Figure 6 shows the Shaker Table II.

Conclusions

This article describes Internet functions that include video file generation and real-time control as MWS features, which are useful for undergraduate courses. With these functions, virtual laboratories can be enhanced by using virtual processes, which in turn allow video animations of simulated processes. Furthermore, MWS allows the implementation of remote laboratories operating in batch mode. WinCom or any other suitable software can be used to implement online laboratories. As illustrated previously, through the application to several examples, these methodologies provide a straightforward approach for the developer of the teaching material, the control engineering lecturer, and a low-cost option for the student user.

Figure 5. *Remote laboratory setup. The potentiometers, encoders, tachometers, and accelerometers give the position, velocity, and acceleration of the system. These signals are fed into the computer server using the data acquisition cards. The computer server calculates and applies the control action. Additionally, a Web cam can be used to observe the operation of real systems.*

Figure 6. *Shaker Table II. This system consists of (1) a shaker table with a (2) mountable frame, (3) the control unit, and (4) the power supply unit.*

References

[1] S.H. Yang, X. Chen, and J.L. Alty, "Design issues and implementation of Internet-based process control systems," *Contr. Eng. Practice*, vol. 11, pp. 709–720, 2003.

[2] M. Johansson, M. Gäfvert, and K.J. Aström, "Interactive tools for education in automatic control," *IEEE Contr. Syst. Mag.*, vol. 18, no. 3, pp. 33–40, 1998.

[3] S.E. Poindexter and B.S. Heck, "Using the Web in your courses: What can you do? What should you do?," *IEEE Control Syst. Mag.*, vol. 19, no. 1, pp. 83–92, 1999.

[4] The MathWorks Inc., *MATLAB Web Server User's Guide*. [Online] 2003. Available: http://www.mathworks.com/access/helpdesk/help/toolbox/webserver/webserver.html

[5] J.L. Díez, M. Vallés, and A. Valera, "A global approach for the remote process simulation and control," in *Proc. 15th IFAC World Congress Automatic Control*, Barcelona, 2002, pp. 76–81.

[6] Quanser Consulting, *WinCon 4.1 User's Guide: Hard Real-time Performance at your Fingertips*, 2003 [Online]. Available: http://www.quanser.com

[7] *GlobeLab: Plataforma para la Docencia de Ingenier de Sistemas y Autom tica no-presencial y semi-presencial,* Research Project PID1208C [Online]. Available: http://globelab.isa.upv.es

Angel Valera (giuprog@isa.upv.es) received his degree in computer science in 1990 and his Ph.D. in 1998 from the Universidad Politécnica de Valencia, Spain. He joined the Department of Systems Engineering and Control in 1989. Since then, he has worked as a lecturer and assistant professor. He currently teaches automatic control, robotics, and CAD/CAM systems. His research interests include robot control, real-time systems, and control education. He can be contacted at the Dpto. Ingeniería de Sistemas y Automática, Universidad Politécnica de Valencia Camino de Vera 14, 46022 Valencia Spain.

José Luis Díez received an M.Sc. in industrial engineering in 1995 and his Ph.D. in control engineering in 2003, both from the Universidad Politécnica de Valencia, Spain. At the Department of Systems Engineering and Control he has taught courses on automation, linear systems control theory, and digital signal processing. His interests include complex systems modeling, clustering techniques, intelligent control, and control education.

Marina Vallés received her degree in computer science from the Universidad Politécnica de Valencia, Spain, in 1996 and completed a master's degree on CAD/CAM/CIM in 1997. Since 1999, she has been a lecturer at the Department of Systems Engineering and Control. She currently teaches automatic control and computer simulation. Her research interests include real-time systems, automatic code generation, and control education.

> With these functions, virtual laboratories can be enhanced by using virtual processes, which in turn allow video animations of simulated processes.

Pedro Albertos has been a professor of automatic control at the Universidad Politécnica de Valencia, Spain, since 1975, where he teaches courses on advanced and intelligent control systems and systems theory. He was a visiting professor at the University of Illinois Urbana-Champaign (1987), the Centre for Industrial Control Science of the University of Newcastle (1992), University of California at Santa Barbara (1996), and the Pulp and Paper Research Centre, University of British Columbia (2001). He was president of IFAC from 1999 to 2002. He has published over 200 technical papers in journals, technical conferences, and seminars. He is coeditor of *Control Engineering Solutions* (IEEE Press, 1997) and *Multivariable Control Systems*, (Springer, 2004) and an associate editor of *Control Engineering Practice* and *Automatica*.

Retroactive Combat

If we extend this argument still further we shall find that it is the existence of feedback in living organisms that gives error its constructive power. This is surely what we understand by the mechanism of experience. The way in which life extracts from ill its appropriate antidote can only be fully realized when we invoke the principle of feedback. Whether it be a question of temperature regulation, the pH of the blood, or the urea content of the blood, or whether it is necessary to expel a foreign body from the eye, nose or throat, it is the deviation from normality that causes the secretion of some hormone or other substance or causes some reflex mechanism to act in order to combat the imbalance. All microbic disease gives rise to a retroaction designed to restore the equilibrium that had been disturbed. Always the remedy is roughly proportionate to the disturbance which it combats.

— Quoted from P. De Latil,
Thinking by Machine,
The Riverside Press, Cambridge, MA, 1957.

© COREL