A STRAIGHTFORWARD PROPOSAL FOR LOW-COST DEVELOPMENT OF VIRTUAL AND REMOTE CONTROL LABORATORIES

M. Vallés, A.Valera, J.L. Díez, P. Albertos

Dpto. Ingeniería de Sistemas y Automática
Universidad Politécnica de Valencia
Camino de Vera 14, 46022 Valencia (Spain)

Abstract: Applications of Information Technology in Process Control range from control educational tools to real process control. Advantages are related to an increase in flexibility for engineers (specially interesting for industrial operators training in new techniques), a decrease in maintenance and supervision costs and a best managing and performance of the plant.

This paper presents a straightforward and simple approach for the remote control and simulation of real processes using Matlab and Quanser Consulting tools. Different examples and applications are shown.

Keywords: Remote control, Control applications, Control education, Computer control, Data acquisition, Robot control.

1. INTRODUCTION

Although there is an increasing trend (Yang, *et al.*, 2003) in applications of Information Technology (IT) in real Process Control (for example: monitoring and control through the Internet, tele-working, tele-medicine, or tele-robotics) origins could be found in educational approaches (Paproch, 1998), (Johansson, *et al.*, 1998), because the use of Internet led to an increase in time and space flexibility of the educational process (Poindexter and Heck, 1999).

The extended use has been possible provided the increasing number of computers at home connected to Internet, and it could decrease the actual teaching hours at University in undergraduate programmes (Overstreet, 1999). Nowadays, relation between Internet an Process Control is far from an experimental phase but in a mature state, providing practical applications at those companies that offers online courses or where employees training plays a role. Distance education takes more advantage of the new technologies than standard education. But when the latter makes appropriate use of IT as an effective learning support resource a "blended learning" appears that seems to be the future educational trend (Bersin, 2003).

Nevertheless, in addition to current Internet delays (Yang, *et al.*, 2003), complexity and expensiveness usually arise when trying to implement plant remote control. Some insight in the solution of these problems is given in this work, especially for those institutions where Matlab® is the standard control design tool. The proposed solutions complement Matlab for remote control and aims to two premises: simple (for server provider) and economical (for remote user).

After outlining configuration needs for virtual (simulated) and remote (real) control approaches, this paper presents in subsequent sections Matlab-based methods for developing virtual and remote processes control. These sections are illustrated with different examples.

2. VIRTUAL AND REMOTE CONTROL LABORATORIES CONFIGURATION

Based on Internet, two different options can be found for setting up laboratories in learning environments: virtual laboratories and remote laboratories. A virtual laboratory allows, for example, continuous access to a simulated process in a computer.

Halfway between traditional and virtual laboratories are the remote laboratories that offer a real experiment to remote users. The incorporation of webcams allows observing the evolution of real systems in addition to variables of interest. Most of the equipment needed for setting up a virtual/remote laboratory is available in traditional laboratories. The only additional element required is an interface between the local application and the Web server.

If Matlab is used as control design tool, then Matlab Web Server (MWS) is a very powerful tool that helps to develop virtual laboratories (Díez, et al., 2002). However, direct remote process control is not possible if this software is used alone, and MWS must be complemented with some software functions (as those presented in this work) or additional commercial packages as WinCon. This section shows simple and economical approaches for hardware and software configuration needs in remote simulation and real processes control using Internet platforms and MWS.

2.1 Matlab-WinCon Configuration

WinCon is a real-time application (under Windows 2000/XP) that allows running code generated from a Simulink diagram in real-time on the same PC (local PC) or on a remote PC (Quanser Consulting®, 2002). With WinCon, data from the real-time running code may be plotted on-line, and model parameters may be changed on the fly through *WinCon Control Panels* as well as Simulink. WinCon software actually consists of two distinct parts that communicate using the TCP/IP protocol: *WinCon Client* (WCC) and *WinCon Serve*r (WCS).

WCC is the real-time software component that runs the code generated from the Simulink diagram at the sampling rate specified. It receives the controller from WCS and it runs it in real-time. In addition, it maintains the communication with WCS and it streams data with servers in real-time.

WCS is the software component that performs the compilation of Simulink diagrams (with Matlab's Real-Time Workshop) to create a real-time WinCon controller executable. It also downloads the controller to WCC and it starts and stops the execution on the client. In addition, WCS maintains communication with Simulink to perform changes of controller parameters, and it plots and saves system data response.

WinCon supports two possible configurations: the local configuration and the remote configuration. In the local configuration WCC and WCS run on the same machine and simultaneously. In the remote configuration WCC runs on a different machine from WCS, and TCP/IP protocol is used. Each WinCon Server can communicate with several WinCon Clients, and reciprocally, each WinCon Client can communicate with several WinCon Servers. Fig. 1 shows a WinCon remote configuration with a single WinCon Server and multiple WinCon Clients running as nodes on a TCP/IP network.

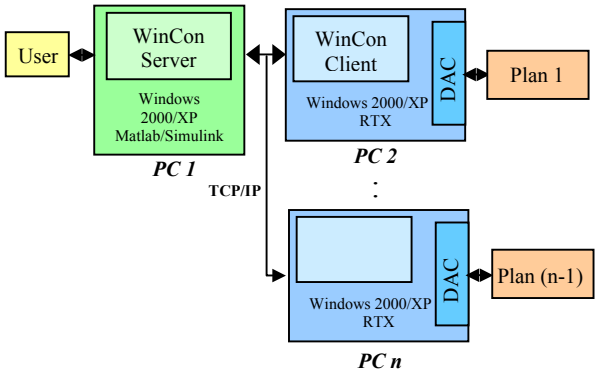

Fig. 1. WinCon remote configuration.

2.2 Matlab Web Server Configuration

Matlab Web Server is an integrated software that enables 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 in a Web browser. A set of programs (a multithreaded TCP/IP server, a TCP/IP client and some configuration programs) enables remote Web access to Matlab (MathWorks, 2003).

For an extensive explanation of the main features of *MWS* and commands used in the development of basic applications see (Díez, et al., 2002), where the configuration for authors' equipment is explained and examples of virtual/remote laboratories are presented.

The process of developing *MWS* applications will require generating an HTML input document for data submission to Matlab, a Simulink model or a Matlab M-file to process input data and to compute results and an HTML output document for the display of Matlab's computations.

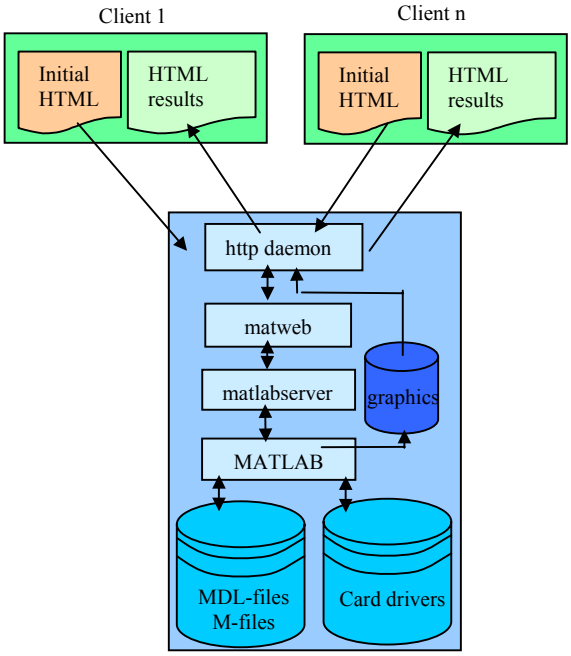

Fig. 2. *MWS* file structure.

However, *MWS* does not allow on-line access to the low level hardware. Therefore, remote laboratories and remote industrial process control is not possible under the *MWS* software provided by Mathworks. To allow remote laboratories and remote control of real systems using *MWS*, an additional configuration with several data acquisition cards has to be used and different routines have been programmed to access these cards from the Matlab environment and the web server.

The routines that have been programmed to access these cards from the Matlab environment allow A/D conversion, D/A conversion, Encoder signal management, Digital input/output access and timer programming. All these functions have been programmed using C++ language. The Matlab executable functions are obtained as DLL files with the *mex* command, using Visual C++ v6.0 compiler.

3. DEVELOPMENT OF VIRTUAL LABORATORIES

For building virtual laboratories, the software that has been used is MWS instead of WinCon because the latest is "too powerful" for being used at these applications. It is more appropriated for controlling real processes remotely. So, only MWS approximation is described at this section.

3.1 Virtual Laboratories using M-files

The M-file for virtual applications is a normal Matlab function that accepts input from HTML input documents and returns the results to an HTML output document. Any available M-file developed for a different application, can be easily adapted in a MWS format then making it accessible through the Internet.

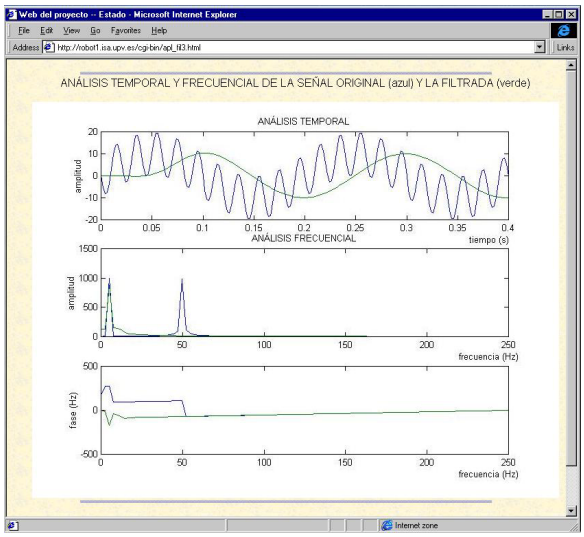

Fig. 3. M-file based virtual laboratory example**.**

Different options exist for showing results and the choice must depend on the kind of result. The simplest way to show output results is to display a scalar or character string. If the Matlab function output is more complex, the results can be shown in a matrix. A table, based on the information included in a Matlab variable, can be dynamically generated for this purpose by means of an HTML command *table*. Finally, graphics can be generated when much information must be displayed. *MWS* generates *jpeg* graphics from Matlab. A graphic identifier (*mlid*) must be defined and used subsequently for file name creation. An example of this kind of laboratory is shown in Fig. 3.

3.2 Virtual Laboratories using Simulink Models

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 normal Simulink model with all the simulation blocks (reference blocks, continuous and discrete transfer functions, scopes, etc.). The model can be a general simulation scheme where the user specifies the particular parameters, such as transfer functions, numerators and denominators, sample and simulation periods, or total simulation time. Because users provide this information in the input HTML document, it is very easy to change the simulation model parameters.

A Matlab M-file is needed to implement the virtual application. This M-file must update the model parameters and start the simulation. To do so, this file must open the model and update the system parameters using the *set_param* command. After that, the *sim* command will simulate the Simulink model

3.3 Virtual Laboratories using Video Files

Even though remote laboratories (shown in next section) allow users to establish the control of real processes without the software tools and the real system, the limitation of a single user access to the remote real process diminishes its effectiveness.

This limitation can be overcome if virtual processes are used (Díez, et al., 2002). A virtual process could be defined as a set of software functions and procedures that simulates a real process response, using (standard video) *avi* files. If the virtual process equations are very close to the physical equations of the system (that is, a white box model is available), the virtual and the real system responses will be similar. In this way, the students/operators would be able to analyse the system's characteristics (for example, stability or steady state error) in an intuitive way.

From version 5 Matlab can convert movie files to *avi* (Audio Video Interleave) files. *avi* files are the most common files for PC video. Commands such as *mov2avi*, *avi2mov*, *avifile*, *addframe* or *getframe* allow Matlab to work with videos externally generated or to generate animations to be visualized in any other PC.

Figure 4 shows an output HTML page for the virtual process control. In the center of the figure an example of a virtual process is showed: a virtual DC motor (based on the Servomotor SFT 154 of Feedback). The system's dynamics is described by the difference equations of the motor's discrete transfer function. The signals that represent the motor position and the control action evolve simultaneously with the motor cursor. This way, users can analyse, in a very direct and easy way, whether the system response is underdamped or overdamped, the maximum overshoot and settling time. Virtual processes can be used in a wide number of activities, and they are described in detail in (Díez, et al., 2002).

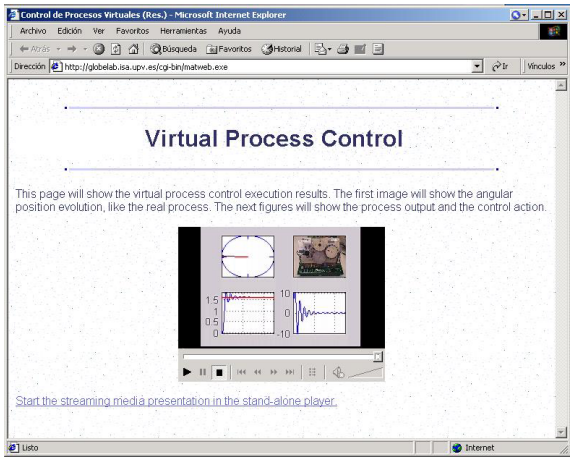

Fig. 4. Virtual Process HTML output page

3.4 A Virtual Laboratories Repository

Some examples of virtual laboratories have been developed with *MWS* by appropriately designing a set of web pages, to test tool capabilities and the easy access of the whole simulation framework by means of Internet. Nowadays, available virtual laboratories are related to "IIR Digital Filter Design" and "Controllers Design", both of them quite common in basic control courses. The *IIR Filter Design Laboratory* includes different activities (signal generation, filter design, and filter behavior test). The activities related with the *Controller Design Laboratory* are open loop system analysis, closed loop system analysis, and controller design.

Portal structure consists of hyperlinks from a web main page to the controller and filter design main pages. From URL http://globelab.isa.upv.es all laboratories presented in this paper can be accessed. In this way, any worker or student following a course that describes some concepts related to developing web pages, can follow the virtual laboratory work proposed there. The only requirement is to have a PC connected to Internet and a Web browser and no Matlab version is required on the remote computer.

4. REMOTE LABORATORIES DEVELOPMENT

Two different operating options are available in remote labs: batch and on-line. On-line operation considers control algorithms at the remote computer, the control actions and sensors being the information transmitted through Internet. The reference and parameters can be changed while the experiment is being carried out, but variable delays appear as a consequence of Internet traffic. Batch operation 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 have been implemented: a batch laboratory based *MWS* (and complementary functions) and an on-line laboratory designed using *WinCon*. In this way, any user having a PC connected to Internet and a Web browser can control an industrial process in batch, and no Matlab version is required on the remote user computer (see (Díez, et al., 2001) or (Valera et al., 2005) for details). For on-line remote control WCS would also be required. Additionally, a webcam has been included using Windows Media Technologies free software (Microsoft, 2000). This camera allows observing the evolution of real systems.

Figure 5 shows the general laboratory setup used for those experiments, where the *Advantech*™ industrial data acquisition cards PCL-833, PCI-1711 and PCI-1720 and the *Quanser Consulting* card MultiQ-2E have been used.

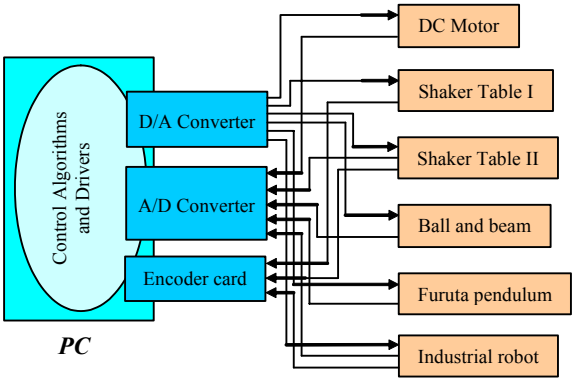

Fig. 5. Remote Laboratory Setup

From a technical point of view, this control architecture works as follows: the tachometers give the velocity and the position of the dc motors. These signals are introduced in the computer server using the A/D converters. The position information also can be obtained using the encoders signals. These signals are introduced in the computer server using the encoder card. With this information the computer server calculates and applies the control actions using the D/A converters. Additionally, other card drivers that could help in controlling more complex systems have also been programmed, like the digital inputs/outputs.

In spite of some other physical process that have been used and tested, this work will present two remote laboratories. The first one is a batch remote laboratory of is an *Industrial Gantry Robot* (Figure 6). This robot has three degrees of freedom: the X-Y-Z axes, with 2.5, 1.5 and 1 meters respectively. Some original control unit stages of these industrial robots have been modified following the laboratory setup of Figure 5 to obtain a flexible control architecture based on PC. For the implementation of this remote robot control, an HTML file for data submission (kind of controller, movement references, sample time, etc.) to Matlab is needed.

Fig. 6. Industrial gantry robot controlled remotely

For the remote robot control, two options have been used for the implementation of these controllers. The first one is based on Matlab scripts. These scripts are standard Matlab functions that access to the data acquisition cards and calculate the robot control inputs.

The second option is based on Simulink schemes. Figure 7 shows a controller for this robot. The Simulink scheme has some especial blocks (Analog Input, Analog Output, etc.) to obtain and to provide the signals of the system. For this kind of control implementations, some additional *Active Server Pages* (ASP) functions have been programmed.

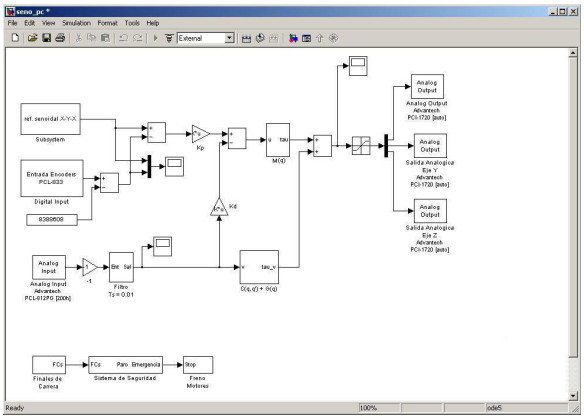

Fig. 7. Simulink scheme used for the robot control

Figure 8 shows the robot response. In this case, a circular reference for the movement has been specified. After an short underdamped system response, the figure shows how the robot movement follow without problems the reference. This kind of figures can be attached in the HTML results pages of the remote laboratories.

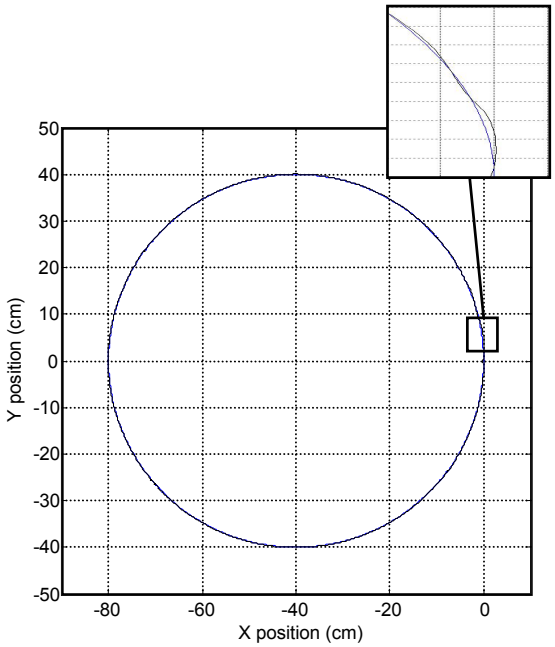

Fig. 8 X-Y reference and robot response

Finally, an on-line remote control implementation (using WCS) based on *Shaker Table II*, of Quanser Consulting, is presented. This process, showed in figure 9, is composed by a table and a two floors structure. The table is controlled by a DC brushless motor. An encoder collects the table position, and the system acceleration is measured by the accelerometers (one is mounted on the table, and one on each floor). With this information the controller calculates the required control action. Several kinds of controls have been implemented: PID, state-space or nonlinear controllers

Fig. 9. Shaker Table II controlled with a remote laboratory

Figure 10 shows a *WinCon Control Panel* of a controller. The control panels allow the user to modify model parameters in real-time. In this case users can modify the reference (the wave form, the amplitude and the frequency), and the kind of controller. In addition, the control panel can show the system variables, like the table position and reference, the control input, the frame accelerations, etc.

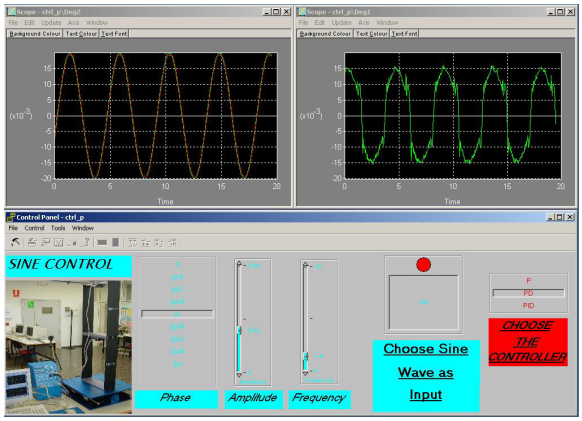

Fig. 10. WinCon control panel for the Shaker II experiment

5. CONCLUSIONS

Virtual and remote laboratories provide an innovative experimental tool allowing control lecturers: to optimize laboratory resources (for instance, sharing plants or software), to develop innovative classroom material (making use of advanced pedagogical methodologies), or to reinforce students' knowledge in a time-space flexible framework.

This paper presented two different approaches for developing virtual and remote laboratories based on commercial technologies (WinCom by Quanser Consulting, and Matlab by Mathworks). They provided straightforward methodologies for the developer of the teaching material (control lecturer) and low-cost for the user (student). Finally, several examples on virtual and remote labs (based on industrial and educational process) have been presented.

At http://globelab.isa.upv.es are available six remote laboratories (a DC Motor, two Shaker Tables, the Ball and Beam experiment, a Furuta Pendulum and an Industrial Robot) that allow experiments from basic to advanced control. A number of virtual laboratories are also available from the website.

Virtual and remote laboratories available have been tested for two years as a support for some regular lectures at Universidad Politécnica de Valencia. For example, at "An Introduction to Computer Control" (High Technical School of Industrial Engineering) and at "Industrial Computer Science Laboratory" (High Technical School of Computer Science School) controllers calculated by students are run remotely. System response is analyzed subsequently.

These experiments satisfy students much more than simulations, as far as they can observe controller working on a real system trough the webcam. Motivation is increased and some kind of competition is created among students in order to design better controllers.

After this test phase, it is expected to open access to all remote and virtual laboratories for the use of all students of Systems Engineering and Control Department at Universidad Politécnica de Valencia.

REFERENCES

Bersin, J. (2003). *What Works in Blended Learning*, ASTD: American Society for Training & Development.

Díez, J.L., M. Vallés, A. Valera and J.L. Navarro (2001). Remote Industrial Process Control with Matlab Web Server, in *Proc. IFAC-Internet Based Control Education* (ISBN:0080439845), Madrid.

Díez, J.L., M. Vallés and A. Valera (2002). A Global Approach for the Remote Process Simulation and Control, in *Proc. 15th IFAC World Congress of Automatic Control* (ISBN:008044184X), Barcelona.

Johansson, M., M. Gäfvert and K. J. Aström (1998). Interactive Tools for Education in Automatic Control, *IEEE Control Sytems Magazine*, **vol. 18**, pp. 33-40.

Microsoft Corp. (2000). *Microsoft Windows Media™ JumpStart CD*.

Overstreet, J. W. and A. Tzes (1999). An Internet-Based Real-Time Control Engineering Laboratory, *IEEE Control Sytems Magazine*, **vol. 19**, pp. 19-34.

K. Paproch (1998). *Distance Learning: The Ultimate Guide,* London: Sage Publications.

S.E. Poindexter and B.S. Heck (1999). Using the Web in your Courses: What can you do? What should you do?", *IEEE Control Sytems Magazine*, **vol. 19**, pp. 83-92.

Quanser Consulting, *WinCon 4.1: Hard Real-time Performance at your Fingertips (User's Guide)*, 2002.

The MathWorks Inc., *Matlab Web Server user's guide*, [Online], 2003. Available at http://www.mathworks.com/access/helpdesk/help/toolbox/webserver/webserver.html

Valera, A, J. L. Díez, M. Valles and P. Albertos. (2005). Virtual and Remote Control Laboratory Development., *IEEE Control Sytems Magazine*, **vol. 25**, pp. 35-39.

Yang, S.H., X. Chen, and J.L. Alty (2003). Design issues and implementation of Internet-based process control systems", *Control Engineering Practice*, **vol. 11**, pp. 709-720.