The Automatic Control Telelab: a User-Friendly Interface for Distance Learning

Marco Casini Domenico Prattichizzo Antonio Vicino

Dipartimento di Ingegneria dell'Informazione,

Università di Siena, via Roma 56, 53100 Siena, Italia

{casini,prattichizzo,vicino}@ing.unisi.it

Corresponding Author: Domenico Prattichizzo, Dipartimento di Ingegneria dell'Informazione, Università di Siena, via Roma, 56 – 53100 Siena, Italia Phone: +39-0577-234609, fax: +39-0577-233602, EMail: prattichizzo@ing.unisi.it

Abstract

In this paper a remote laboratory of automatic control is presented. The main target of this laboratory is to allow students to easily interact with a set of physical processes through the Internet. The student will be able to run experiments, change control parameters, and analyze the results remotely. The Automatic Control Telelab (ACT) allows the user to design his/her own controller by means of the Matlab/Simulink environment, and to test it at the actual plant through a user-friendly interface. An additional feature of ACT is its architecture, allowing for an easy integration of new processes for control experiments. The ACT is reachable at http://www.dii.unisi.it/˜control/act.

Keywords

Education, Web, Distance learning, Remote laboratory, Virtual laboratory, Control Systems.

I. Introduction

Recent years have witnessed dramatic changes in graduate education because of Internet and Web technologies. Web-based teaching, distance learning, electronic books, and interactive learning environments will play increasingly significant roles in teaching and learning processes in the near future.

This paper describes an experience of designing and building a tele-laboratory of automatic control at the University of Siena. Tele-laboratories are expressions of a more general distance education which is attracting wide attention in the academic and government communities. Reference [1] analyzes the state of the art and outlines the future perspectives of on-line distance education in the United States. In that paper, the point is made that the technology to meet the need of distance education is already available, and it will further improve in the near future. Moreover, software engineers consider on-line education as an important emerging market and, therefore, a business opportunity. An overview of Web-based educational systems is given in [2] where an accurate description of the actual problems concerning the design of Web-based educational systems and their possible solutions is also provided.

In [3] Poindexter and Heck present a survey on the usage of Web-based educational systems in the control area. In this paper, authors overview the level of integration between Internet and control courses. Special attention is devoted to the automatic control laboratories accessible through the Web. Tele-laboratories are divided into two classes: *virtual labs* and *remote labs*. The first ones are systems which can run simulations remotely with possible animations of the controlled system. One of the first instances of virtual labs has been developed with CGI scripts and Java applets at the University of Edinburgh, Scotland [4]. The student typically connects from a remote client to the virtual lab server, chooses the experiment, changes some parameters and runs the simulation. Then he or she looks at the simulation results through some graphical interface or downloaded data stored in a given format.

The MATLAB® environment has been used as the core of the virtual lab in [5] and in [6]. A plug-in is required to launch MATLAB on the remote machine from within a Web browser. The advantage of MATLAB over Java is that the first is a standard computational tool for control system applications.

Therefore, it is considered a standard tool for students of control courses so that they do not need to learn new languages to run a virtual lab session. In fact, several toolboxes are available in the Matlab environment for control applications, e.g. optimization, μ-control, and system identification. Standardized Computer Aided Control System Design (CACSD) tools, mainly based on the Matlab/Simulink environment, have been thoroughly discussed in [7]. In [6] two more plug-ins are needed to run a virtual lab session: one is necessary to display MATLAB figures in the browser windows, and one to use the Virtual Reality Modelling Language (VRML) to render the simulation results. In [5], students can run two simulations: a magnetically levitated vehicle and an automated highway. In [6], ball and beam, tanks' level control, and gyro pendulum simulations are available.

Remote labs are laboratories where students can interact with actual experiments via Internet. Usually, remote operators through a Web interface can change several control parameters, run the experiment, see the results and download data. This situation is for instance, the case of [8], where a remote lab for testing analog circuits is described; in [9], a remote chemical control process is implemented, and in [10] several laboratory experiments are made available.

The complexity of designing the hardware and software architecture of the remote laboratory dramatically increases when one of the features required is designing the controller within a remote Web session. One such instance is the case of the remote lab developed at the College of Engineering of Oregon State University [11], [12]. Here, students can remotely control a robot arm by changing some parameters but, more interestingly, transmitting the control program which changes the dynamics of the closed-loop system. The student can edit his/her own controller and paste it in an applet window. Then this code is uploaded to the server machine of the remote lab, compiled and executed on the robot arm, becoming an actual remote lab experiment. The user interface consists of a graphical network application referred to as SBBT (Second Best to Being There). The design has three main features: collaboration with peers, active presence, and complete control of the remote experiment. Another approach has been investigated in [13] where the user can run an experiment using a controller which resides not only on the server but also on the client, compiling and executing it on the user machine. In this case, some issues about network reliability and delays have been addressed. The audio and video feedback is very important to increase the effect of telepresence, as shown in [14] where authors implemented an interesting pan, tilt, and zoom control of cameras grabbing the experiments of the remote lab.

In general a remote laboratory can use a well-known software environment, such as LabVIEW® [15] or MATLAB/Simulink® [16], [17], but it can also use a special purpose one, as in [18], where a flow process is analyzed.

In [19] a comparison between virtual labs and remote labs is presented. The authors examine a common experiment (ball and beam) from these two points of view and conclude that virtual labs are good to assimilate theory, but they cannot replace real processes since a model is only an approximation which cannot reproduce all the aspects of the process, such as unexpected non-linearities. To avoid these issues, remote laboratories which allow a student to interact directly with real processes should be used; this opportunity takes a fundamental role especially for engineering students.

From a pedagogical point of view, remote labs, allowing for designing the whole control law, are more stimulating. Typically, the price to pay to obtain the controller design feature is that students must learn and use new control languages which are designed for the remote lab and cannot take advantage of control functions developed in other contexts.

In this paper, the realization of a remote lab, the Automatic Control Telelab (ACT) developed at the University of Siena, is presented. Preliminary results have been discussed in [20]. ACT allows students to choose a control law, change on-line the control parameters, and design their own controller simply through the MATLAB/Simulink environment. The main feature of the Automatic Control Telelab is that a remote user can synthesize his/her own controller without learning any special language. To know and run the MATLAB/Simulink environment is all that is needed. The authors' opinion is that the use of a standard language like MATLAB/Simulink will dramatically encourage the exercise with remote labs in control classes. Moreover, the ACT is accessible in a very easy way from every computer connected to the Internet through any browser, without the necessity of downloading any special software or plug-in. The authors think this fundamental point makes the lab accessible to everyone.

The paper is organized as follows. Section 2 describes the common features of remote laboratories and the specific characteristics and some implementation notes about the Automatic Control Telelab. Section 3 describes the ACT operations from a more practical point of view. Section 4 describes some teaching experiences, while conclusions are drawn in section 5.

II. Automatic Control Telelab overview

The Automatic Control Telelab (ACT) is a remote lab whose main goal is to create an educational environment that allows a remote student to interact with some real processes through the Internet [21].

A. Common features of remote labs

Like other remote labs [22], [23], [24], ACT implements the main features described below.

Activeness: A remote lab which is far from the user facilitates the active learning process. The student goes through a full set of operations as if he were in a traditional lab. He or she can make decisions on what to do, having the freedom to make mistakes and learn from them.

Full availability: Similar to most applications that run over the Internet, a remote lab is available 24 hours a day. This availability implies that the safety system must always be active. Every experiment of the remote lab must be equipped with a self-protection device to prevent damages to the components and local users present in the lab. Without such devices, breaks and electric shocks could cause damage and break the system availability.

Accessibility: The use of the Internet allows students to connect to the lab with extreme ease from every part of the world. All that is needed is an Internet connection and a browser, such as Netscape

Navigator® or Microsoft Internet Explorer® , to reach the home page of the remote lab. Additionally, students with motor problems can easily learn and operate staying at home.

Instrumental easiness: A student who wants to interact with a remote lab must know the essentials of control systems, such as modeling a process and designing a controller. He can avoid the details and problems encountered by lab's technicians in making the experiments fully operative.

B. Main features of ACT

In addition to the previously described features, common to every remote lab, ACT enjoys the following additional features which exploit most of the advantages of the Matlab/Simulink environment, in particular for the remote controller design.

Easy interface: Simplicity of use is essential to estabilish an interface that can be used by everyone [3], then the user can focus his/her efforts on interacting with the experiment without wasting time in trying to understand how to proceed through the site.

ACT is based on intuitive HTML pages and Java applets, both fully supported by the latest versions of the main browsers. Help pages are also provided for detailed information. The use of plug-ins or external programs are not required unless the user wants to design his/her own controller; in this case the use of MATLAB and Simulink is required.

Remote Simulink-based controller design: The most important feature of ACT is the design of a controller using Simulink. MATLAB and Simulink are standard tools in the control community, and they are adopted in many basic and advanced courses. Since the controller is a Simulink model, the user does not spend time to learn a special programming language to implement the controller; a basic knowledge of MATLAB and Simulink environment is only required. Besides, through the use of the Simulink graphical interface, a large set of blocks can be chosen and connected in a very easy way. Thus, a continuous, discrete or hybrid controller can be dedsigned without any additional hurdles. Since MATLAB provides many tools for data analysis, at the end of the experiment a file in MAT-

LAB workspace format (.mat), which contains all the data collected during the experiment, can be downloaded for off-line analysis.

Controller type: Every experiment in the remote lab can be controlled according to two modes: predefined or user-defined. In the first case, the student will choose a control law in a predefined set and then assign the value of typical parameters. For example, the student can select a PID controller to control the experiment and choose the value of proportional, integral and derivative coefficients. Rather than using a predefined controller, the user can design a new one (as described in the previous point) and send it to the ACT server. The new controller will be used to control the experiment.

On-line reference and controller parameter changes: If a student chooses a user-defined controller, he can also decide what references he wants to run during the experiment. He can select some references among a set or create new ones building a Simulink subsystem. Moreover, while an experiment is running, ACT allows the user to change some typical controller parameters on-line (e.g., for a PID controller, the user can change the proportional, integral, and derivative coefficients, while the experiment is running).

Reference change: As in the previous point, the system reference can be changed while an experiment is running. Thus the user does not have to start a new experiment to verify the response of the system for different input signals. The user can run the experimente for a given time length. At the end of this period, the experiment ends and the user is disconnected.

Lab presence: For effective distance learning, the users must have a sense of “presence” in the laboratory. Therefore, a live video is provided, so that students may view the real process while the experiment is in progress. This key point distinguishes between remote labs and virtual labs, the latter only providing software simulations of physical processes. Work is in progress to improve the sense of “presence”, for instance, through an audio feedback system.

Easy addition of new processes: The Automatic Control Telelab has been designed to allow the lab administrator to increase the number of processes available on-line in an efficient way. To connect a new process to the ACT, only two files are needed: one is a Simulink model file, while the other is a simple text file containing information on the process.

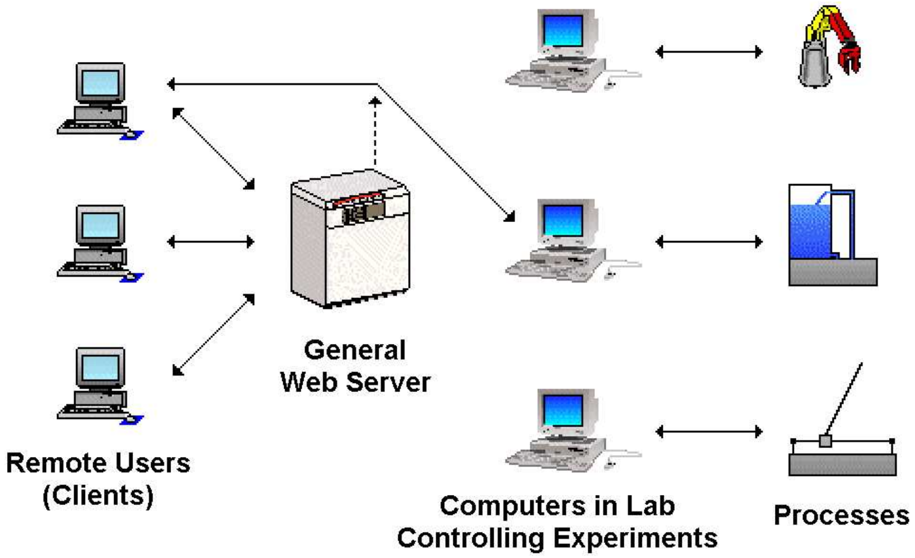


Fig. 1. General scheme of ACT connections.

C. Software and hardware notes

In this section a brief description of implementation aspects is provided.

The home page and other descriptive HTML pages are stored in a unique server which is common to every process. When an experiment is chosen, the user host is re-addressed to the machine connected to the process (Fig. 1). Data is exchanged through the Internet by a TCP connection between the user (client) and the ACT server which is directly connected to the process. Once the connection is established, the server sends all the data the client needs (controller parameters, references, etc.); afterwards, the process is ready to start.

Server programs run under Microsoft Windows 2000® operating system; they are executable files obtained by MATLAB Real-Time Workshop® (RTW), which allows the transformation of a Simulink model to a C source code.

To realize all the features of ACT, a specific code is linked with the source code generated by RTW, making it possible to establish a TCP connection, to change controller parameters and references on-line, and to obtain real-time requirements. The user-defined controller function is implemented by merging the controller model (control.mdl) with the selected process Simulink model (source.mdl). Thus, the file obtained is compiled and executed by means of RTW (Fig. 2). The resulting file (process.exe) will exchange data with the Java applet which implement the user interface.

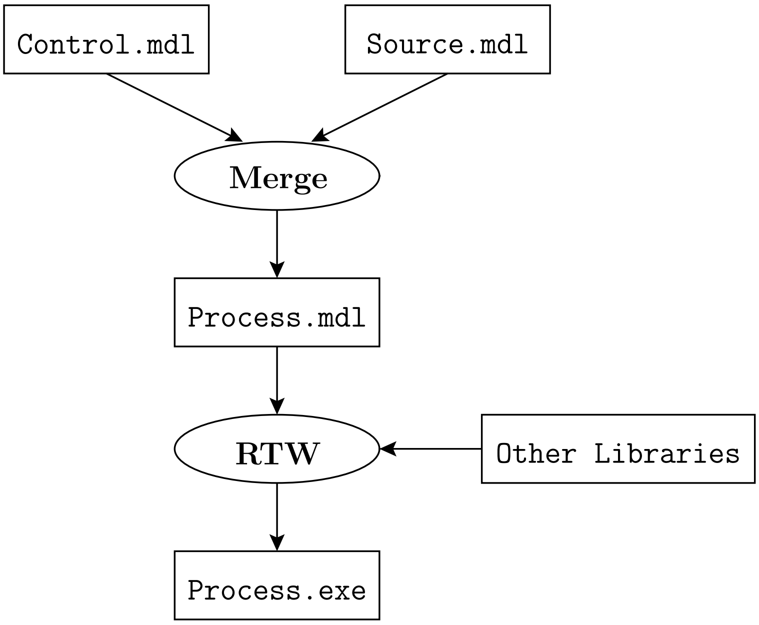


Fig. 2. Integration of the user-defined controller.

On the client side, Javascript and Java applets are use, so that platform compatibility is assured. In this way, the user can operate through a user–friendly graphical interface.

To allow video transmission, the software Webcam32® [25], developed by Neil Kolban, is used. It uses a Java applet to display the live video of the experiment without requiring specific plug-ins to perform this task.

A worthwhile point is that network delays do not deteriorate the performance of the control system which runs locally. Network delays can only delay changes of input references and controller parameters sent by remote users and data plots and videos sent back to the users.

At present three hardware processes are available for remote control: a DC motor, a tank, and a magnetic levitation system (Fig. 3). While the DC motor is a simple process to be controlled, the tank exhibits system nonlinearities, and the magnetic levitation system is a nonlinear and unstable process. The authors intend to increase the number of available processes, adding some MIMO (Multi-Input Multi-Output) systems as a two degrees of freedom helicopter (CE 150 Helicopter by Humusoft) and a six degrees-of-freedom robot arm (PUMA 560).

As with every remote lab, security aspects are crucial. Since everyone can connect to ACT at every time everyday, hard software constraints (saturations) and hardware constraints have been added to prevent breaks and electrical accidents. Moreover, since the user can make the control systems unstable, physical safety protections are present. For example, in the case of the magnetic levitation experiment, two transparent shields prevent the ball from falling during the normal operating conditions or at the start of the experiment.

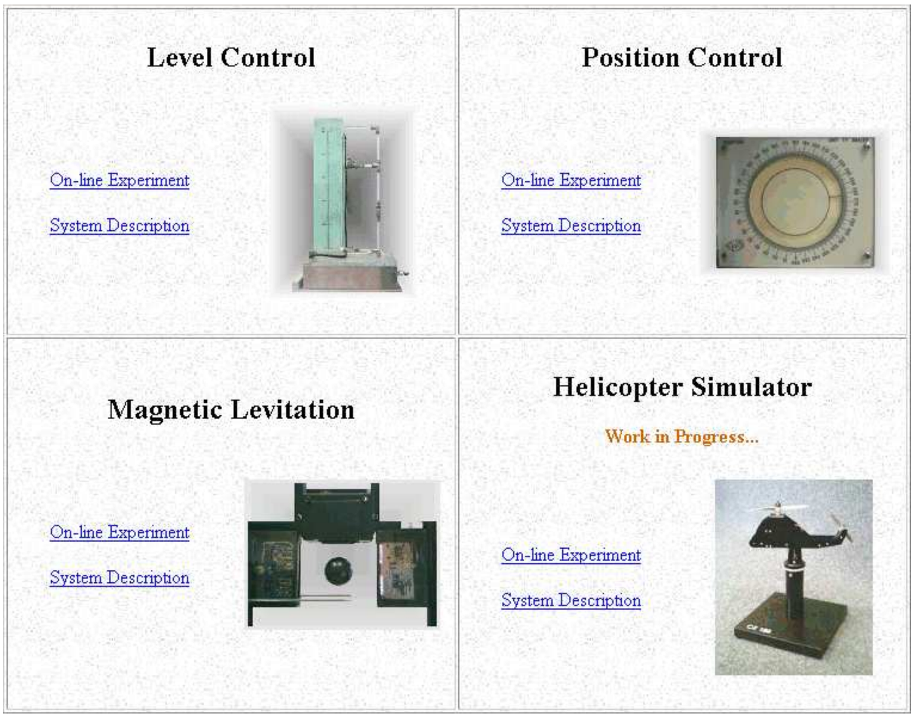


Fig. 3. Automatic Control Telelab's on-line experiments.

III. Automatic Control Telelab operations

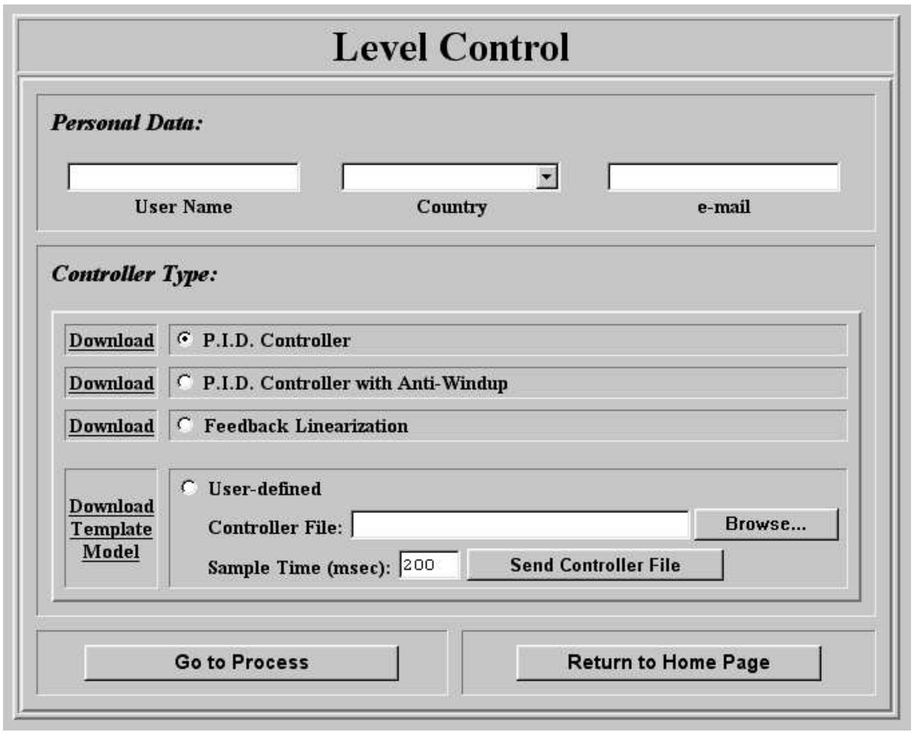
From the home page of ACT, the user may perform general operations, such as reading help files, contacting lab people, sending comments, and viewing the set of processes available for remote control.

All these pages are stored in the main web server which is common to every experiment.

After choosing a process, one can view a page describing the physical system and a mathematical model. Moreover, a Simulink model is provided to allow students to test a controller before running the actual experiment.

A. The Control Type Interface

The control type interface has been provided to allow the user to choose the controller for the experiment, (Fig. 4). This interface is divided into two parts: the personal data panel and the controller panel.



Level Control

Personal Data:

User Name
Country
e-mail

Controller Type:

● P.I.D. Controller
○ P.I.D. Controller with Anti-Windup
○ Feedback Linearization
○ User-defined
Controller File:
Sample Time (msec): 200

Fig. 4. The Control Type Interface for ACT. This is the display that the remote user will use to choose the controller to control the process.

- *Personal Data Panel:* The task of this part is to identify the user who will interact with the remote lab. The user must indicate his/her name, nationality, and e-mail address. This information will be stored and used for statistics about ACT users. Access restrictions can be set to avoid undesirable users.

- *Controller Type Panel:* In this panel, the user can choose his favorite controller for the process. He can select a predefined controller or a customized one. In the second case, the user must choose the sample time and send the file containing the controller to the ACT server; this file must be a Simulink model. To help the user in this phase, a Simulink model template which simplifies the design of the controller can be downloaded. In fact, one only needs to connect the provided signal blocks about references, outputs, and commands with the Simulink designed controller. An example of a user-defined PID controller is shown in Fig. 5. Note that an interesting feature of the ACT is that it is possible to make some parameters and gain blocks tunable on-line. They can be obtained simply by naming them with the prefix "ACT_TP_".

After providing the personal data and choosing the controller, the user can proceed to the next interface, the "experiment interface".

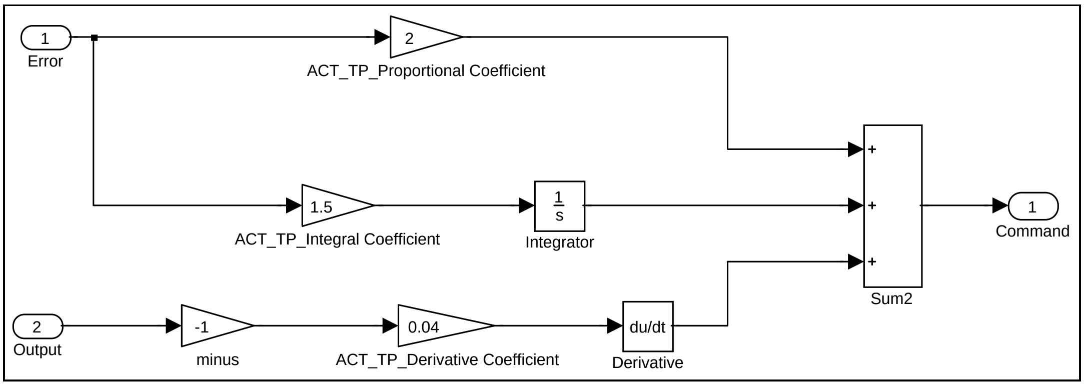

1
Error
2
ACT_TP_Proportional Coefficient
1.5
ACT_TP_Integral Coefficient
1/s
Integrator
+
+
+
Sum2
1
Command
2
Output
-1
minus
0.04
ACT_TP_Derivative Coefficient
du/dt
Derivative

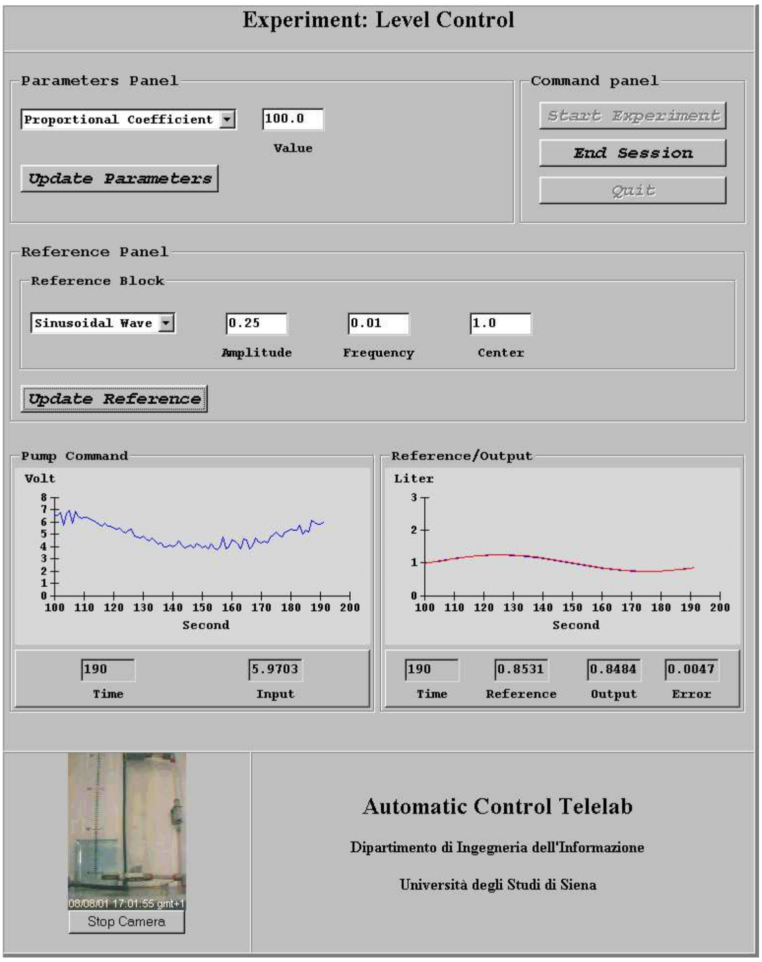
Fig. 5. Example of a PID controller with tuning parameters.

B. The Experiment Interface

The user interacts with the process through the experiment interface represented in Figure 6.

This interface is divided into five parts: (1) command panel, (2) controller parameters panel, (3) reference panel, (4) experiment dynamics window, and (5) video window of the experiment.

This interface provides the tools necessary to control the experiment and to change the reference and the controller parameters as well.



Experiment: Level Control

Parameters Panel

Proportional Coefficient: 100.0 (Value)

Reference Panel

Reference Block

Sinusoidal Wave

Amplitude: 0.25

Frequency: 0.01

Center: 1.0

Pump Command

Time: 190

Input: 5.9703

Reference/Output

Time: 190

Reference: 0.8531

Output: 0.8484

Error: 0.0047

08/08/01 17:01:55 gmt+1

Automatic Control Telelab

Dipartimento di Ingegneria dell'Informazione

Università degli Studi di Siena

Fig. 6. The Experiment Interface for ACT. Through this interface the remote user can interact with the experiment.

- *Command panel:* This panel shows only three buttons which start, stop, or quit (before starting) the experiment.
- *Controller parameter panel:* In this panel, the typical parameters of the controller are displayed. The user can modify them and send the new values to the server by pressing the "update parameters" button. When the TCP/IP packet containing the new values reaches the server, the parameters are immediately changed.
- *Reference panel:* This section gives the user the opportunity to choose the reference signal among a set of possible choices. Like the previous point, when the ACT server receives the TCP/IP packet, the reference is changed accordingly.
- *Experiment dynamics window:* To monitor the dynamics of the process, some graphical and numerical

panels are provided. The number of these panels will change with the number of inputs and outputs of the process. For a SISO (Single-Input Single-Output) process (Fig. 6), there is a panel for the input (command) and a panel for the reference and the output. The graphical and numerical panels are updated every second.

- *Video window:* A video of the experiment is shown, useful especially for processes with moving parts. In this case, the user can observe the process dynamics during the experiment having the feeling of being really present in the laboratory.

At the end of the experiment, the user can download the file containing the detailed dynamics of the experiment in MATLAB format (.mat). Thus, the user can use this information for off-line analysis.

IV. Teaching experiences

The Automatic Control Telelab has been used since 1999 in some practical undergraduate classes of "Automatic Control" courses. A typical lesson can be summarized as follows:

- The lecturer describes the process to be controlled and explains its main features, such as dynamic properties, special nonlinearities, etc.
- Students divide into groups and try to design a controller, testing it on the provided Simulink model.
- If a controller passes the test phase, it is sent to the ACT server, and it is used to perform the actual experiment.
- After the experiment has been executed, data is downloaded and analyzed to gather performance indexes, such as overshoot, rise time, and settling time.
- Different kinds of controllers can be compared to show which are their features in terms of dynamic or static performances.

During these lessons, students seem to be very excited, basically because they can put in practice their control theory knowledge through the Internet. Moreover, they are stimulated in a sort of competition to design the best controller. Since the mathematical model generally catches the main features of the system, students can evaluate the differences between the model and the actual plant and can tune on-line some parameters to obtain a controller which works better with the real process.

In addition to the University of Siena, other institutes are using the ACT in their Automatic Control courses, for instance, the University of Pisa, the Polytechnic of Milan, and the University of Bologna. Moreover, the ACT server has been installed in the Department of Aeronautics and Astronautics at MIT, to allow a decentralized and coordinated control of a team of vehicles (helicopter simulators).

V. Conclusion

The Automatic Control Telelab provides a sophisticated but easy-to-use mechanism to control a remote process through the Internet. This laboratory is intended to allow control system students to operate on real processes without being present in the laboratory. A student can choose a predefined controller or design a customized one (through a Simulink model) to control the process. While the experiment is running, it is possible to change controller parameters and reference signals.

Since the software is quickly adaptable to various kind of processes, adding a new process to the remote lab is quite easy. As the Automatic Control Telelab is still in a development phase, the number of available processes is limited (DC motor, tank, and magnetic levitation system). Future work will be devoted to add some MIMO processes and, in particular, a two degrees-of-freedom helicopter and a six degrees-of-freedom robot arm. Additional flexibility will be added to the user interface in order to manage data plots and video feedback for MIMO systems.

Further future improvements regard the development of competitions between student controllers and a special interface for the identification of the available processes, to allow the use of ACT also in “System Identification” courses. The ACT’s home page is: http://www.dii.unisi.it/˜control/act.

References

[1] D.A. Harris, "Online distance education in the united states," *IEEE Communications Magazine*, pp. 87–91, March 1999.

[2] A. Ausserhofer, "Web-based teaching and learning: a panacea ?," *IEEE Communications Magazine*, pp. 92–96, March 1999.

[3] S.E. Poindexter and B.S. Heck, "Using the web in your courses: What can you do? what should you do?," *IEEE Control System*, pp. 83–92, Febraury 1999.

[4] C.M Merrick and J.W. Ponton, "The ECOSSE control hypercourse," *Computers in Chemical Engineering*, vol. 20, Supplement, pp. S1353–S1358, 1996, [Online]. Available: www.chemeng.ed.ac.uk/ecosse/control/sample/index.html.

[5] K.M Lee, W. Daley, and T. McKlin, "An interactive learning tool for dynamic systems and control," in *Proc. of International Mechanical Engineering Congress & Exposition*, CA, November 1998.

[6] C. Schmid, "The virtual lab VCLAB for education on the web," in *Proc. of American Control Conference*, Philadelphia, PA, June 1998, pp. 1314–1318, [Online]. Available: www.esr.ruhr-uni-bochum.de/VCLab/.

[7] W.E. Dixon, D.M. Dawson, B.T. Costic, and M.S. de Queiroz, "Towards the standardization of a Matlab-based control systems laboratory experience for undergraduate students," in *Proc. of American Control Conference*, Arlington, VA, June 2001, pp. 1161–1166.

[8] C.D. Knight and S.P. DeWeerth, "World wide web-based automatic testing of analog circuits," in *Proc. of 1996 Midwest Symposium Circuits and Systems*, August 1996, pp. 295–298.

[9] M. Shaheen, K. Loparo, and M. Buchner, "Remote laboratory experimentation," in *Proc. of American Control Conference*, Philadelphia, PA, June 1998, pp. 1314–1318.

[10] J. Henry, *Enginering lab on line*, University of Tennessee at Chattanooga, 1998, [Online]. Available: chem.engr.utc.edu.

[11] B. Aktan, C.A. Bohus, A. Crowl, and M.H. Shor, "Distance learning applied to control engineering laboratories," *IEEE Transactions on education*, vol. 39, no. 3, pp. 320–326, August 1996.

[12] A. Bhandari and M. Shor, "Access to an instructional control laboratory experiment trough the world wide web," in *Proc. of American Control Conference*, Philadelphia, PA, June 1998, pp. 1319–1325.

[13] J. W. Overstreet and A. Tzes, "An internet-based real-time control engineering laboratory," *IEEE Control Systems Magazine*, vol. 19, no. 5, pp. 19–34, October 1999.

[14] J. Zhang, J. Chen, C.C. Ko, B.M. Chen, and S.S. Ge, "A web–based laboratory on control of a two–degree-of–freedom helicopter," in *Proc. of Conference on Decision and Control*, Orlando, Fl, December 2001, pp. 2821–2826.

[15] V. Ramakrishnan, Y. Zhuang, S.Y. Hu, J.P. Chen, C.C. Ko, Ben M. Chen, and K.C. Tan, "Development of a web-based control experiment for a coupled tank apparatus," in *Proc. of American Control Conference*, Chicago, IL, June 2000, pp. 4409–4413, [Online]. Available: vlab.ee.nus.edu.sg/vlab/control/.

[16] T.F. Junge and C. Schmid, "Web-based remote experimentation using a laboratory-scale optical tracker," in *Proc. of American Control Conference*, Chicago, IL, June 2000, pp. 2951–2954.

[17] J. Apkarian and A. Dawes, "Interactive control education with virtual presence on the web," in *Proc. of American Control Conference*, Chicago, IL, June 2000, pp. 3985–3990, [Online]. Available: www.controlab.com.

[18] G. Choy, D.R. Parker, J.N. d'Amour, and J.L. Spencer, "Remote experimentation: a web-operable two phase flow experimnet," in *Proc. of American Control Conference*, Chicago, IL, June 2000, pp. 2939–2943.

[19] M. Exel, S. Gentil, F. Michau, and D. Rey, "Simulation workshop and remote laboratory: two web-based training approaches for control," in *Proc. of American Control Conference*, Chicago, IL, June 2000, pp. 3468–3472.

[20] M. Casini, D. Prattichizzo, and A. Vicino, "The automatic control telelab: a remote laboratory of automatic control," in *Proc. of* 40*th IEEE Conf. on Decision Control*, Orlando, FL, December 2001, pp. 3242–3247.

[21] M. Casini, "Designing a tele laboratory for control of dynamic systems through internet," M.S. thesis, Università degli Studi di Siena, June 1999.

[22] B. Aktan, "Distance learning applied to control engeneering education," M.S. thesis, Oregon State University, June 1996.

[23] L. Zurawski, "Internet laboratory tests control system parameters," Tech. Rep., University of Tennessee at Chattanooga (UTC), August 1999, [Online]. Available: chem.engr.utc.edu.

[24] R.E. Gantenbein, T.L. James, J.R. Cowles, and W. Paloski, "TELELAB: A virtual laboratory for acquisition and distribution of scientific data on the internet," Tech. Rep., Department of Computer Scienze - University of Wyoming, [Online]. Available: www.cs.uwyo.edu/˜rex/telelab2.html.

[25] N. Kolban, "Webcam32 - the ultimate webcam software," [Online]. Available: surveyorcorp.com/webcam32.