E-LEARNING BY REMOTE LABORATORIES: A NEW TOOL FOR CONTROL EDUCATION

Marco Casini, Domenico Prattichizzo, Antonio Vicino

Dipartimento di Ingegneria dell'Informazione
Università di Siena, Siena - Italy
Email: {casini,prattichizzo,vicino}@ing.unisi.it

Abstract: This paper deals with the Automatic Control Telelab (ACT), a remote laboratory of automatic control developed at University of Siena. In particular, it focuses on a new chapter of the ACT referred to as "student competition". It is a mechanism through which students can compete to design the controller with best performance for a given remote experiment. Controllers which achieve the given performance requirements are stored according to a ranking criteria. This tool allows students to make control synthesis practice on real remote processes through the Internet, and to compare their favourite controller with competitors designed by other users.

Keywords: Control education, Remote laboratory, Control systems.

1. INTRODUCTION

The Automatic Control Telelab (ACT) is a laboratory developed at the University of Siena. Telelaboratories are instances of the more general distance education problems which is attracting a wide attention in the academic and government communities. Automatic Control is one of the technical areas which most exploited the new technologies to develop new tools for distance learning (Poindexter and Heck, 1999). A thorough treatment about control education by means of web technologies has been recently reported in (Dormido, 2002).

The web–based laboratories are divided in two classes: the virtual labs and the remote labs. The main difference between them is that virtual labs allow to remotely run simulations with possible animations of the controlled system (Merrick and Ponton, 1996; Lee *et al.*, 1998; Schmid, 1998), while remote labs are laboratories where students can remotely interact with real experiments. The ACT at the University of Siena is an example of remote laboratory and is attracting the interest of many students from our campus and from other national and international institutions (Casini *et al.*, 2001).

In remote labs, users can change control parameters, run the experiment, see the results and download data through a web interface. This is for instance the case of (Knight and DeWeerth, 1996), where a remote lab for testing analog circuits is described; in (Shaheen *et al.*, 1998), a remote chemical control process is implemented and in (Henry, 1998) several laboratory experiments are made available.

In (Exel *et al.*, 2000) 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 for instance unexpected non-linearities. On the other hand remote laboratories allow students to directly act with real processes, which is very important especially for engineering students.

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

One of the key features of the Automatic Control Telelab (ACT) is that students can choose a control law, change on-line the control parameters and design their own controllers simply through the Matlab/Simulink environment. This feature allows a remote user to synthesize his/her own controller without learning any special language other than the Matlab/Simulink software. It is the authors' opinion that usage of a standard language like Matlab/Simulink will dramatically encourage the exercise with remote labs in control classes.

This paper deals with a recent improvement of the ACT: the "student competition" chapter. In this part of the ACT structure, students, or groups of students, compete to gain the best controller performance for a given remote experiment. This is a very exciting experience and it is attracting a great attention from automatic control classes. A typical competition session starts with defining control system requirements on one of the real experiments of the ACT. Then students access the ACT and design their own controllers which will steer the process during the competition. The ACT server stores the controllers with students data in a database, computes the performance indexes and assigns a score to the controllers. Then a ranking is computed and it is possible to evaluate the ability of students to meet the assigned specifications.

The paper is organized as follows. Section 2 illustrates the main features of ACT. In Section 3 the student competition mechanism is described, whereas in Section 4 a competition example is provided. Section 5 deals with teaching experiences while some implementation aspects are described in Section 6. Conclusions are drawn in Section 7.

2. AUTOMATIC CONTROL TELELAB OVERVIEW

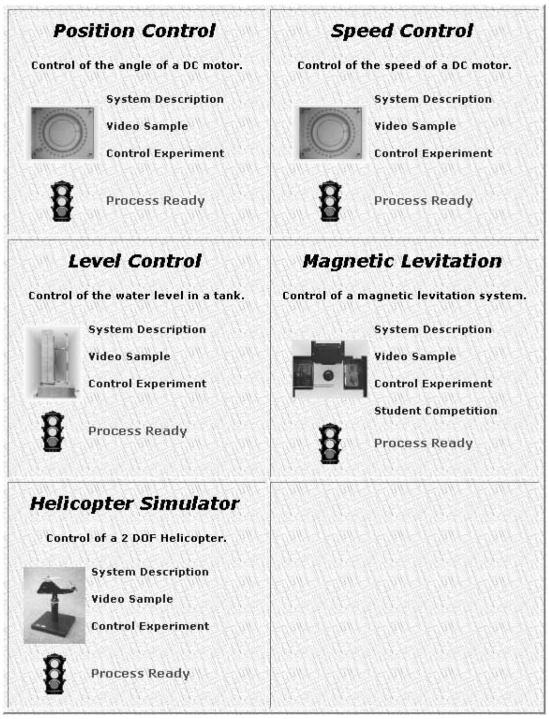
The Automatic Control Telelab is a remote laboratory mainly intended for educational purpose, and since 1999 it has been used in control systems courses (Casini, 1999; Casini *et al.*, 1999; Casini *et al.*, 2001). The aim of the project was to allow students to put in practice their theoretical knowledge of control theory in an easy way and without restrictions due to laboratory opening time and processes availability. At present, the ACT is accessible 24 hours a day from any computer connected to the Internet by means of a common browser like Netscape Navigator or Microsoft Internet Explorer; no special software or plug-in is required. If a user wants to design his/her own controller, the Matlab/Simulink software is required. A live video window is provided for each remote experiment session. One of the main features of the ACT is the possibility to integrate any user-defined controller in the control loop of the remote process. The controller synthesis is based on the Matlab/Simulink environment which is very popular in the control community. Since Matlab and Simulink packages are standard tools in control systems courses, there are no additional hurdles for a student who wants to design his/her own controller, which simply consists in a Simulink model similar to those commonly used to run a system simulation.

At present, four processes are available for remote control (Fig. 1): a DC motor, a tank for level control, a magnetic levitation system and a two degrees–of–freedom helicopter. The DC motor is used to control the axis angular position and velocity. The level control process has been included because, in spite of its simplicity, it shows nonlinear dynamics, whereas the magnetic levitation process, being nonlinear and unstable, shows very interesting properties to be analyzed in control theory education. Finally, the two degrees–of–freedom helicopter, being a nonlinear and unstable MIMO system, can be used in graduate control system courses.

Since like every remote lab the experiment hardware is controllable by one user at a time, from the web page showing the list of available experiments (Fig. 1) it is possible to know which processes are ready as well as the maximum waiting time needed to access the busy experiments.

3. STUDENT COMPETITION OVERVIEW

A typical remote laboratory allows users to run remote experiments using predefined or user-defined controllers. Students can run an experiment and see the dynamic response, but in general no information on controller performances is provided and it is not possible to know how controllers designed by other people behave on the same process. This is one of the reasons that motivated us to design a student competition mechanism for our remote laboratory. Through this tool a student knows about performance requirements his/her own con-



Position Control

Control of the angle of a DC motor.

System Description

Video Sample

Control Experiment

Process Ready

Speed Control

Control of the speed of a DC motor.

System Description

Video Sample

Control Experiment

Process Ready

Level Control

Control of the water level in a tank.

System Description

Video Sample

Control Experiment

Process Ready

Magnetic Levitation

Control of a magnetic levitation system.

System Description

Video Sample

Control Experiment

Student Competition

Process Ready

Helicopter Simulator

Control of a 2 DOF Helicopter.

System Description

Video Sample

Control Experiment

Process Ready

Fig. 1. Automatic Control Telelab's on-line experiments

troller must satisfy. Moreover a final ranking of the best controllers as well the time plots of the relative experiments are provided.

In the following some features of the ACT competition structure are described.

Remote exercises: in addition to standard control synthesis exercises, this tool allows a student to design a controller, which must satisfy some performance requirements, and to test it on remote real processes. At the end of the experiment, the performance indexes are automatically computed and shown to the user; if such indexes fulfil the requirements, the exercise is completed. An overall index is then computed (usually as a weighted sum of the previous indexes) and the controller is included in the ranking list.

Controller comparison: since this tool allows everybody to view the ranking concerning a competition, it is possible to know what kind of controller achieved better results. During the end-competition lesson, students who have designed the best controllers are invited to discuss their projects, while the lecturer shows why some control architectures work better than others.

Many competitions on the same process: it is possible to provide more than one competition benchmark on the same process, thus increasing the number of remote exercises available for students. Due to the software design of the competition structure of the ACT, new benchmarks can be added very efficiently.

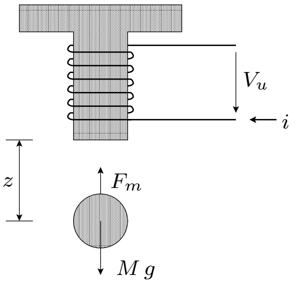


Fig. 2. Sketch of the process

It is the authors' opinion that competition can be considered as a new useful tool for distance learning and, at the same time, a tool which increases the potentiality of remote laboratories.

4. A COMPETITION SESSION DESCRIPTION

In this section, a competition session is described. In particular an example of competition regarding the process of magnetic levitation (see Fig. 1) is reported.

First of all, a student or a group of students who want to compete need to register by filling a form and obtaining a username and a password.

The user can analyze the mathematical model of the process (provided as a *pdf* file) as well as the required performance specifications. In this example it is required that, for a step reference, the settling time (5%) must be less than 1 second and the overshoot must be less than 40%. A more detailed description shows also the working point of the nonlinear benchmark.

The mathematical model of this process, sketched in Fig. 2, is summarized as follows

$$\begin{cases} M\,\ddot{z} &= M\,g - F_m \\ F_m &= k_m\,\dfrac{i^2}{z^2} \\ i &= k_a\,V_u \end{cases} \tag{1}$$

where z is the absolute distance of the center of the ball from the coil, M is the mass of the ball, F_m is the magnetic force, i is the current in the coil, and V_u is the input voltage of the coil ($0 \leq V_u \leq 5$); k_m and k_a are the magnetic constant and the input conductance respectively. The actual values of these coefficients are reported in Table 1.

Table 1. Magnetic levitation system parameters.

M	Mass of the ball	$20 \cdot 10^{-3}$ Kg
k_m	Magnetic constant	$2.058 \cdot 10^{-4}$ $N(m/A)^2$
k_a	Input conductance	0.5488 $1/\Omega$
g	Gravity acceleration	9.80665 m/s^2
k_y	Unit conversion	100 cm/m

Equation (1) can be rewritten with $x_1 = z$, $x_2 = \dot{z}$, $u = V_u$ (input command) and $y = k_y\, z$ (output in centimeters).

$$\begin{cases} \dot{x_1} &= x_2 \\ \dot{x_2} &= g - \dfrac{k_m\, k_a^2}{M}\, \dfrac{u^2}{x_1^2} = g - k_t\, \dfrac{u^2}{x_1^2} \\ y &= k_y\, x_1 \end{cases}$$

By substituting the actual values of parameters in the above equations, one obtains:

$$\begin{cases} \dot{x_1} &= x_2 \\ \dot{x_2} &= 9.80665 - 0.0031\, \dfrac{u^2}{x_1^2} \\ y &= 100\, x_1 \end{cases}$$

Since the competition is based on an experiment around the state ($x_{10} = 0.05m$, $x_{20} = 0$), students can choose to linearize dynamics

$$\begin{cases} \Delta \dot{x} = A\, \Delta x \,+\, B\, \Delta u \\ \Delta y = C\, \Delta x \,+\, D\, \Delta u \end{cases}$$

It follows immediately that $u_0 = \sqrt{\dfrac{g\, x_{10}^2}{k_t}} = 2.811$, thus linearized matrices are:

$$A = \begin{bmatrix} 0 & 1 \\ \dfrac{2\, k_t\, u_0^2}{x_{10}^3} & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 139.4389 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ -\dfrac{2\, k_t\, u_0}{x_{10}^2} \end{bmatrix} = \begin{bmatrix} 0 \\ -6.9719 \end{bmatrix}$$

$$C = [k_t \;\; 0] = [100 \;\; 0] \quad , \quad D = [0]$$

Now a linear controller, such as a PID or a lead-lag compensator, can be synthesized. Of course, advanced students can design controllers with nonlinear techniques.

In order to design the controller, students must run the Simulink environment on their own local computers, then download a template file (*template.mdl*) and connect the signals describing the output, the error and the command to design the desired controller. An example of a PID controller is shown in Fig. 3.

A special interface (Fig. 4) allows a student to describe the structure of his/her own controller (i.e. P.I.D. Controller) and to set the sample time of the experiment; if the controller is continuous time, the sample time is intended as the integration step of the Simulink solver. Moreover, the user has to specify the file containing the controller and, if needed, the Matlab workspace file (*.mat*) containing essential data for that controller. Those files will be uploaded to the server, compiled and, if no error occurs, executed on the real remote process.

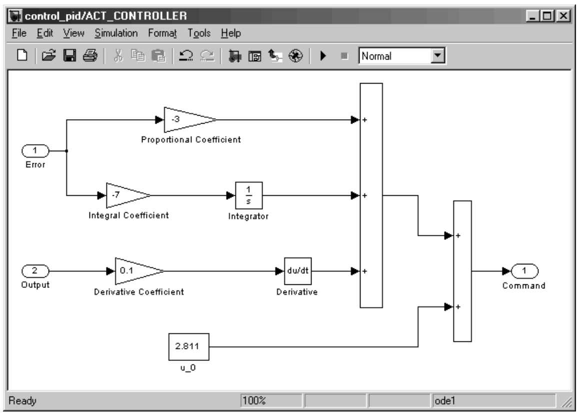
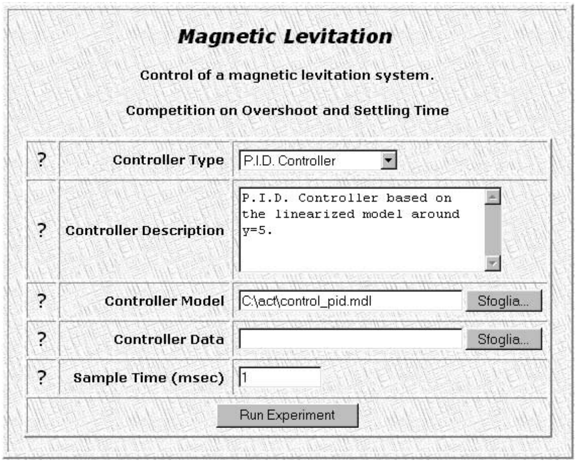


Fig. 3. A Simulink model for a PID controller



Magnetic Levitation

Control of a magnetic levitation system.

Competition on Overshoot and Settling Time

?	Controller Type	P.I.D. Controller
?	Controller Description	P.I.D. Controller based on the linearized model around y=5.
?	Controller Model	C:\act\control_pid.mdl
?	Controller Data	
?	Sample Time (msec)	1

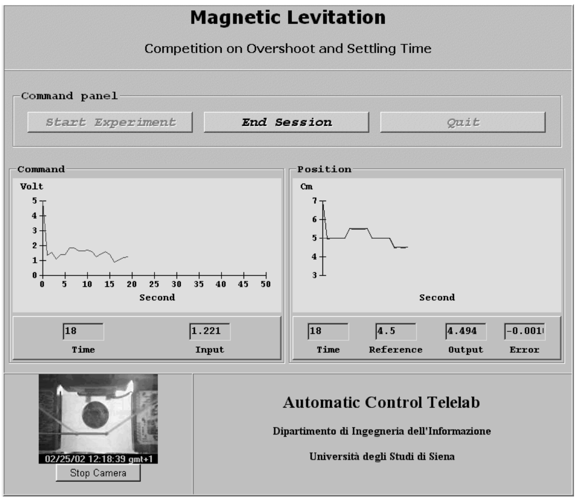
Fig. 4. The interface describing the controller features

A second graphical interface (Fig. 5) allows a user to start the experiment and to observe its behaviour through plots and the live video window.

At the end of the experiment, the performance indexes are computed and are displayed to the user. It is now possible to download a Matlab workspace file containing the full dynamics of the experiment and to view the time plots (Fig. 6). The ranking of the user controller is given as in Fig. 7.

Since several controllers can achieve the requested performance, an overall index is evaluated to build the ranking. This index is obtained by weighting each performance index. If a controller does not satisfy the requirements, the overall index is not computed.

It is possible for every user to view a controller report (Fig. 8) where information on ranking and other data, such as the controller description, the nickname of the user and his/her nationality and institution, are displayed.



Magnetic Levitation

Competition on Overshoot and Settling Time

Time	Input
18	1.221

Time	Reference	Output	Error
18	4.5	4.494	-0.001

Automatic Control Telelab

Dipartimento di Ingegneria dell'Informazione

Università degli Studi di Siena

Fig. 5. The interface showing the running experiment

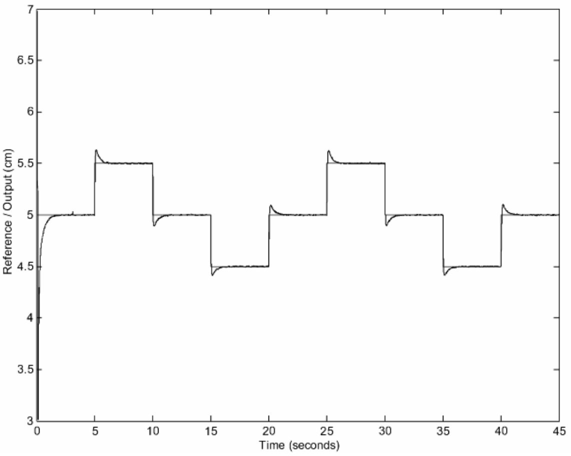
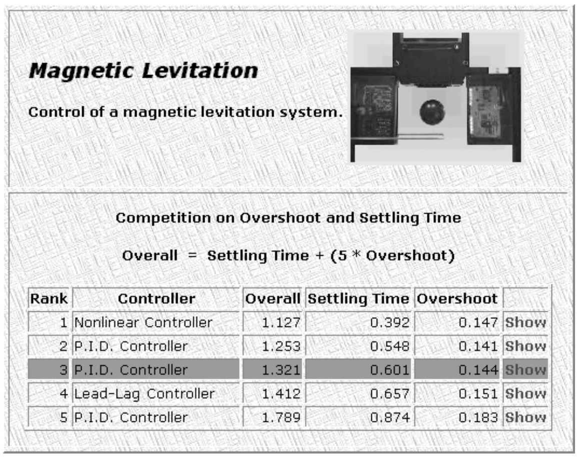


Fig. 6. Time plots of the experiment



Magnetic Levitation

Control of a magnetic levitation system.

Competition on Overshoot and Settling Time

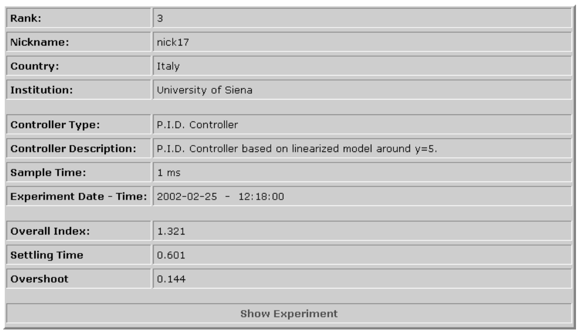
Overall = Settling Time + (5 * Overshoot)

Rank	Controller	Overall	Settling Time	Overshoot	
1	Nonlinear Controller	1.127	0.392	0.147	Show
2	P.I.D. Controller	1.253	0.548	0.141	Show
3	P.I.D. Controller	1.321	0.601	0.144	Show
4	Lead-Lag Controller	1.412	0.657	0.151	Show
5	P.I.D. Controller	1.789	0.874	0.183	Show

Fig. 7. Rank position of the controller

5. TEACHING EXPERIENCES

In spring 2002 undergraduate control system classes at the University of Siena used the student competition system.



Rank:	3
Nickname:	nick17
Country:	Italy
Institution:	University of Siena
Controller Type:	P.I.D. Controller
Controller Description:	P.I.D. Controller based on linearized model around y=5.
Sample Time:	1 ms
Experiment Date - Time:	2002-02-25 - 12:18:00
Overall Index:	1.321
Settling Time	0.601
Overshoot	0.144

Fig. 8. Controller report

First of all, the lecturer illustrated the physical model of the magnetic levitation system, emphasizing its unstable and nonlinear dynamics, and suggesting the students to linearize dynamics to design the controller.

Since ACT is accessible at any time, students had no problems to analyze the problem and test their own controllers during the days before the second competition class, where the lecturer answered students about their questions and difficulties, and helped them to solve some typical problems. For example, he suggested them to use a pre-filter on the reference to obtain smoother command signals, and, in general, better performances.

At the end of the competition almost all the students were able to design a satisfactory controller, and their feedback was really positive.

After an evaluation process, some conclusions about positive and negative aspects of this experience were drawn:

positive aspects: students seemed to be very interested and excited, and used this tool to put in practice many theoretical notions. Moreover, everyone tried to do his/her best to obtain a good position in the ranking. However, the real motivation for this kind of competition, is not to individuate a winner, but to give students a new tool which can help them to better understand some practical control design issues as well as to increase their interest about control systems and technology.

negative aspects: after a first phase when students were really involved in learning new tools for designing a good controller, many students spent plenty of time to tune controller parameters just to obtain the best controller in the ranking, without any additional educational improvement.

6. IMPLEMENTATION NOTES

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

The home page and other descriptive pages are stored in a unique server which is common to every process. Several pages regarding the student

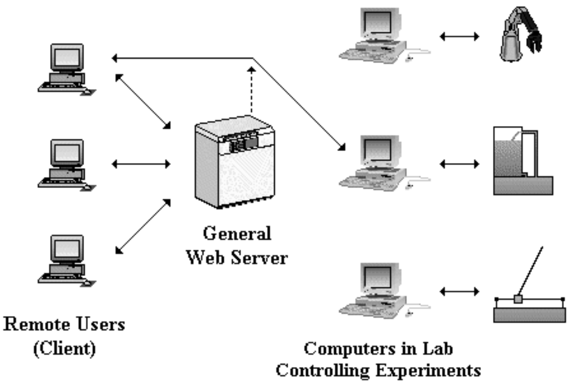


Fig. 9. General scheme of ACT connections

competition are dynamically generated by means of the PHP language, and all data about users and controllers are stored in a MySQL database.

When an experiment is chosen, the user host is re-addressed to the machine directly connected to the process (Fig. 9). Data are exchanged through the Internet by a TCP connection between the user (client) and the ACT server. Once the connection has been established, the server sends all the data the client needs, afterwards the process is ready to start.

Server programs run under Microsoft Windows NT/2000 operating system. They are executable files obtained by Real-Time Workshop (RTW), a Matlab toolbox which allows to transform a Simulink model into a C source. To realize all the special features of ACT, specific code is linked with the source code generated by RTW.

To run a controller designed by a user, the controller model is merged with a Simulink model interfacing with the process through a data acquisition board. Thus, the overall file obtained is compiled and executed.

On the client side, Java applets are used, so that platform compatibility is assured. In this way, the user can operate through a very easy to use graphical interface.

To allow video transmission, the software Webcam32 (Kolban, 2002) is used. It is based on a Java applet to display on-line video, so it is not necessary for the user to install special software to perform this task.

7. CONCLUSIONS

The student competition mechanism of the Automatic Control Telelab has been described. The competition session is a natural extension of the remote laboratory at the University of Siena. It is the authors' opinion that competition stimulates students' interest thus improving the learning capabilities. In spring 2002 control system classes at the University of Siena used the ACT competition system with great interest and excitement. The ACT's home page is: http://www.dii.unisi.it/˜control/act.

REFERENCES

Casini, M. (1999). Designing a Tele Laboratory for control of dynamic systems through Internet. Master thesis (in italian). Università degli Studi di Siena.

Casini, M., D. Prattichizzo and A. Vicino (1999). The automatic control telelab: a user-friendly interface for distance learning. Technical report. Università di Siena, Italy.

Casini, M., D. Prattichizzo and A. Vicino (2001). The automatic control telelab: a remote control engineering laboratory.. In: *Proc. of* 40*th IEEE Conference on Decision and Control*. Orlando. pp. 3242–3247.

Dormido, S. (2002). Control learning: present and future. In: 15*th IFAC World Congress b'02*. Barcelona.

Exel, M., S. Gentil, F. Michau and D. Rey (2000). Simulation workshop and remote laboratory: two web-based training approaches for control. In: *Proc. of American Control Conference*. Chicago. pp. 3468–3472.

Henry, J. (1998). *Enginering lab on line*. University of Tennessee at Chattanooga. http://chem.engr.utc.edu.

Knight, C.D. and S.P. DeWeerth (1996). World wide web-based automatic testing of analog circuits. In: *Proc. of 1996 Midwest Symposium Circuits and Systems*. pp. 295–298.

Kolban, N. (2002). Webcam32 - the ultimate webcam software. Technical report. http://surveyorcorp.com/webcam32.

Lee, K-M, W. Daley and T. McKlin (1998). An interactive learning tool for dynamic systems and control. In: *Proc. of International Mechanical Engineering Congress & Exposition*. CA.

Merrick, C.M and J.W. Ponton (1996). The ecosse control hypercourse. *Computers in Chemical Engineering* **20, Supplement**, S1353–S1358.

Poindexter, S. E. and B. S. Heck (1999). Using the web in your courses: What can you do? what should you do?. *IEEE Control System* **19**(1), 83–92.

Schmid, C. (1998). The virtual lab VCLAB for education on the web. In: *Proc. of American Control Conference*. Philadelphia. pp. 1314–1318.

Shaheen, M., K. Loparo and M. Buchner (1998). Remote laboratory experimentation. In: *Proc. of American Control Conference*. Philadelphia. pp. 1314–1318.