INTERNET BASED REMOTE CONTROL OF AN AIR-FLOW AND TEMPERATURE PLANT

C. PETRESCU[1], C. LUPU[2], I. MATEI[3]

Această lucrare prezintă modalitatea de realizare a controlului de la distanţă a unei instalaţii didactice destinate reglării debitului de aer şi a temperaturii Sunt prezentate aspecte hardware şi software, arhitectura şi configurarea platformei experimentale care utilizează Internet-ul ca suport pentru comunicaţie. Este prezentată, pe scurt, modalitatea de realizare a reglării numerice, precum şi mecansimele care asigură siguranţa şi securitatea controlului de la distanţă. Scopul principal al aplicaţiei prezentate este acela de suport pentru realizarea lucrărilor practice în cadrul învatămîntului la distanţă.

This paper describes the setup behind the remote control of an air-stream and temperature didactical control platform. Issues related to the hardware and software architecture and configuration for remote control of the experimental platform using Internet as communication network are emphasized. A brief description of the control approach used and of the safety and security mechanism of the remote control setup are presented. The main purpose of the remote control application is to serve as an educational tool for distance learning.

Keywords: control systems, e-learning, internet based remote control.

1. Introduction

Today the Internet is widely used in education, commerce, industry and entertainment, as it represents a gate to a huge amount of information. Control systems education is currently exploiting the advantages of Internet and web technologies to develop distance learning paradigms. Remote user interaction with on-line experiments is one of the most in-vogue topics today in distributed computing, web applications, and distance learning. Applications range from purely control educational tools to real time process control. Real time process control via Internet is a challenging topic, because usually the Internet is not a safe environment, introducing stochastic delays and packet loses. Therefore great care must be taken in real time control application via Internet. Process

[1] Lecturer, Dept. of Automatics and Computer Science, University POLITEHNICA of Bucharest, Romania, e-mail: catalinp@indinf.pub.ro
[2] Lecturer, Dept. of Automatics and Computer Science, University POLITEHNICA of Bucharest, Romania, e-mail: cip@indinf.pub.ro
[3] Teaching Assistant, Dept. of Automatics and Computer Science, University POLITEHNICA of Bucharest, Romania,

monitoring and control through the Internet provides several advantages (decrease personnel costs, infrastructures performance optimization, preventive maintenance, etc.).

E-learning offers potentially universal access to "best-in-class" content, regardless of location, and it can transform education and training from a passive consumption experience to a more flexible and student-centered experience. Many computer-based or web-based training programs are available nowadays. Even in traditional face-to-face classes, several web-based homepages support students' study preparation, review and discussion off- or on-campus. Relevant educational remote control applications can be found in [1] [4] [8].

In the following, the hardware/software architecture and configuration of an air-stream and temperature control system with remote control possibility (via Internet) will be described. The current paper is a continuation of the work presented in [7].

2. Remote control setup

The experiment presented in this paper was developed as laboratory activity for “System Identification and Control” course, lectured in the “*Politehnica*” University of Bucharest, Faculty of Automatic Control and Computers. The laboratory aims to o familiarize students with issues regarding system identification and numerical process control.

The control system over the Internet is organized in a modified hierarchal structure [11] consisting of a main controller and a remote close loop system. Periodically, the main controller computes and sends the reference signal, in a frame or a packet, via the network to the remote system. The remote system then process the reference signal to perform local close-loop control and returns the sensor measurement to the main controller. This structure is widely used in several application including mobile robots [12] and teleoperation [10]. In our case we simplified the hierarchical structure by replacing the main controller with a reference generator.

The air-stream and temperature control plant serves as a didactical platform for testing several model identification techniques and for analyzing the process behavior with respect to different control algorithms. The experiment has two component: a hardware component and a software component, which will be emphasized in the next sections.

Control platform

The air-stream and temperature control plant, showed in Fig. 1, is intended for the laboratory activities of control engineering courses. It has two output measurement points (air pressure and air temperature) and two control inputs (fan speed and heater power). A fan provides an air-stream along a tube and a heating element feeds heat into the air-stream. The temperature can be measured at four different locations of the tube. The volume of the air flow is measured by a differential pressure sensor, which detects the pressure in front of a metering orifice in comparison with the constant supply pressure. The two sensors (industrial standard) are mounted at distinct positions of the tube and offer a standard 0…10V output signal. The heating capacity as well as the air flow may be manipulated by external 0...10V input signals.

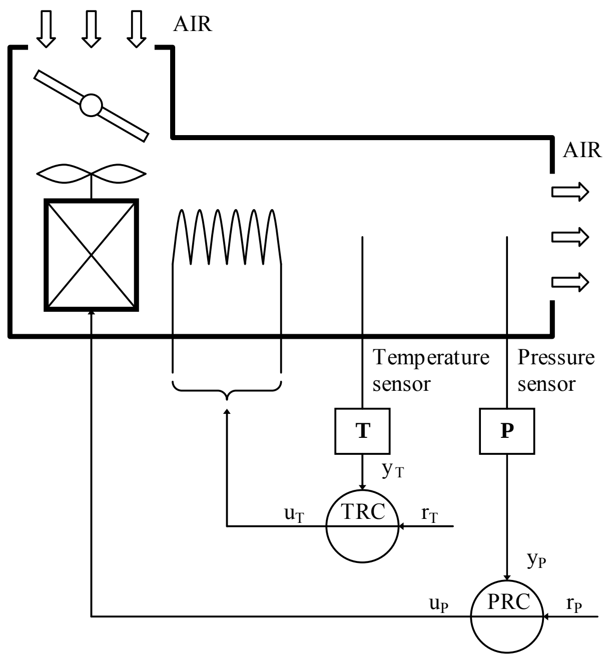

Fig. 1. The air-stream and temperature control plant

After connecting additional external controllers, the laboratory experimental setup "Air-stream and Temperature Control Plant" may implement the following closed control loops:

- simple control loop with a 1st-order control plant (Air-stream control with uncontrolled temperature);
- simple control loop with a 2nd-order control plant (temperature control with uncontrolled air-stream);
- cascaded control loop with temperature control and cascaded air-stream control;
- two separated control loops with independent temperature control and air-stream control.

For the remote control system, presented in this paper, a control approach formed by two separate independent close loops (one for temperature control and one for pressure control respectively) was chosen, as presented in Fig. 1 (y_T – temperature output, r_T – temperature set-point, u_T – heater control input; y_P – pressure output, r_P – pressure set-point, u_P fan control input; u_φ–manual air opening; PRC and TRC are the controllers for the pressure and temperature control loops, respectively).

Remote Control System–Hardware Configuration

The hardware devices used for the remote control system were:

- An air-stream and Temperature Control Plant;
- A PC compatible IBM with an acquisition board through which the computer is connected to the plant;
- Video camera for real-time image broadcasting;
- Server machine for Internet connection.

The hardware configuration of the remote control system is depicted in Fig. 2.

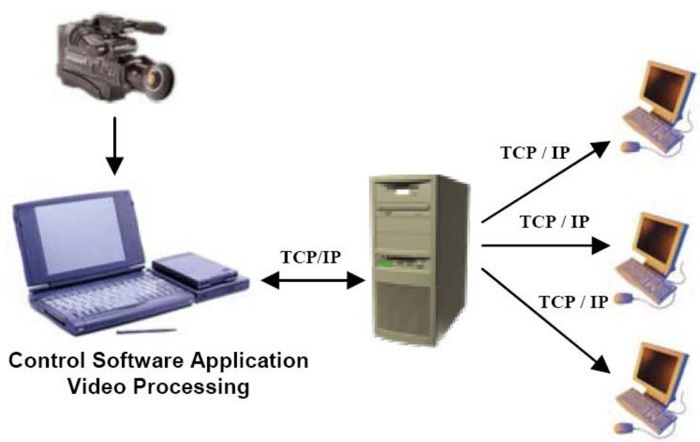

Fig. 2. Hardware configuration of the remote control system

Remote Control System–Software Architecture

The software architecture is based on two subsystems which act as user interfaces for on-place as well as remote control of the plant, as it can be noticed in Fig. 3.

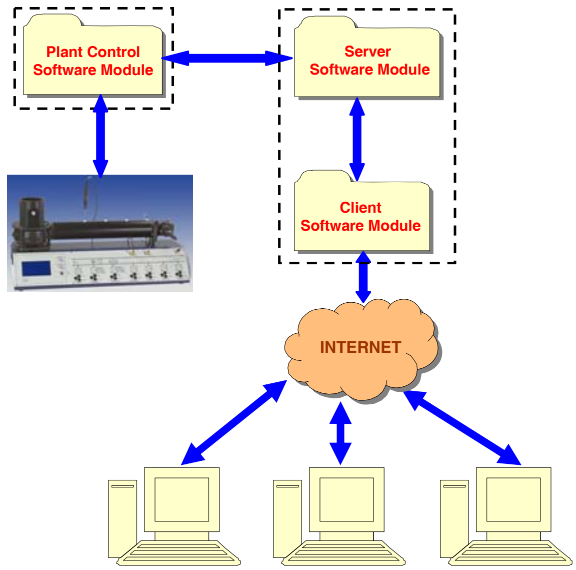

Fig. 3. Software architecture of the remote control system

The first subsystem is a Plant Control Software Module represented by a C++ program written in Lab Windows/CVI Environment. It runs on the computer connected to the process and allows direct control of the platform. This application represents a gate for the data leaving or coming to or from the process respectively. The second subsystem consists in two software module organized in a client-server arhitecture.

On place-process control

The on-place process control is assured by a *Plant Control Software Module* created in the Lab Windows/CVI environment. The program acts also as a bridge between the platform and the remote subsystem, exchanging data between the two entities. The core of the Plant Control Software Module are the control algorithms for the two close loops. There are some default control algorithms but they can be easily replaced by new ones intended for testing. The Plant Control Software Module provide a number of control services such as:

- Set point changing (the user can set different values for the step reference experimenting this way different operating points of the platform's processes);
- Automate/manual control (the user can choose between letting the control algorithm to generate de control signal or setting this value himself; this

service is useful for finding out the tip of input/output characteristic of the process);

- Control algorithms configuration (this service is useful for testing user designed control algorithms belonging to a class of digital control algorithms).

To make this module as friendly as possible for the user, the program has a graphical user interface (GUI) presented in Fig. 4, which makes the use of the program very easy and natural.

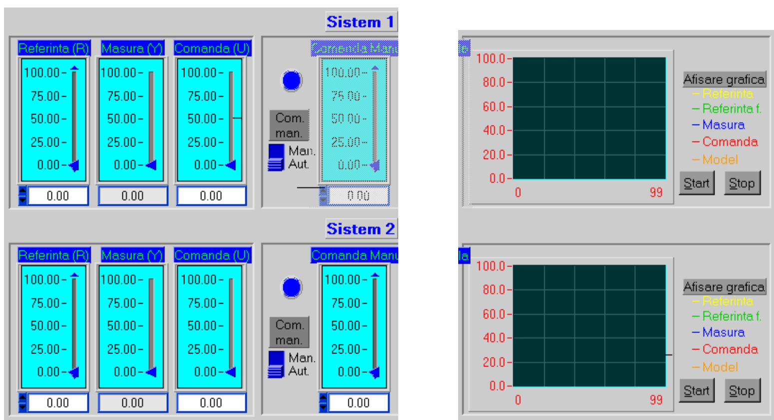

Fig. 4. a) GUI for control operations; b) GUI for process evolution

Remote Process Control

The center of the remote subsystem are two software modules: Server Software Module and Client Software Module, programs oriented in a client-server architecture [9] which will be described in the following:

Server Software Module: entity which accepts requests and offers services to the clients. For this experiment the server is a Java program designed by us, hosted by a HP LH3 server machine and not visible to clients. The server module connects itself to the Plant Control Module and becomes a proxy for the clients connected to it. The server is a multithreading program that runs in the background and periodically sends data about the state of the platform to the clients (the period of data transmission is determined by the dynamics of the two processes). In theory it can accept an infinite number of client connections, but the number is limited by the software in order to maintain a certain level of performances and functionality.

Client Software Module: software entity which sends requests to the server and process the received data. For this experiment the client is a Java applet incorporated in a web page. Next to the applet there are presented live images with the two process evolution. For security reasons, imposed by the Java language, the applet and the server must reside on the same machine. The client application has a graphical interface presenting in a natural manner information (received from server) about the state of the platform and sending, in return, data to the process. The access to the web page requires an ID and a password and it is supervised by an Apache web.

The Client Software Module offers the same facilities as the Plant Control Software Module. The user can modify the set point, change the automate/manual regime and generate manual commands as presented in Fig. 5. In the same time the user can see the time evolution graphic of the set-points, output and inputs of the two processes.

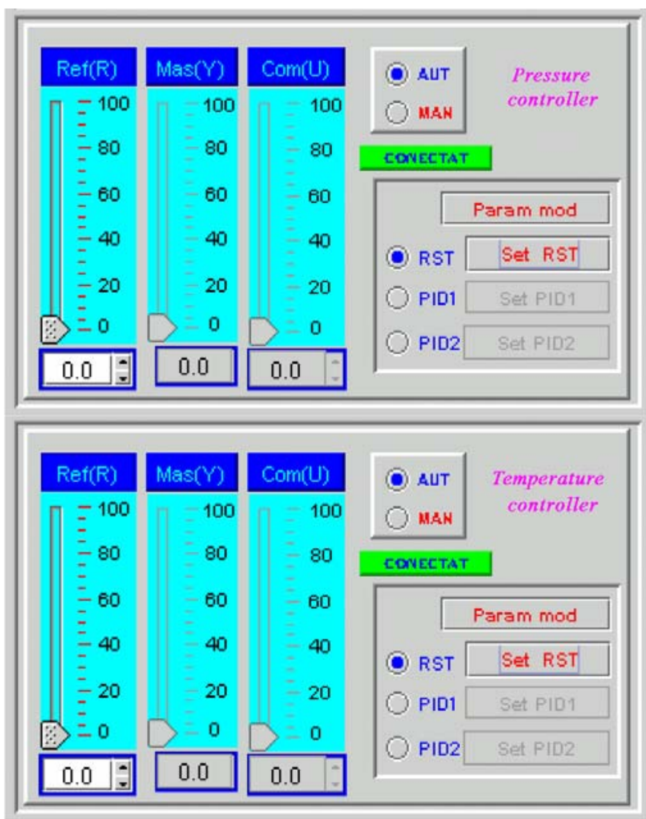

Fig. 5. Client Software Module Graphical User Interface: GUI for control operations

Also, in order to increase the interaction with the didactical platform, the user is able to design its own control algorithms and send them to the process for testing. In Fig. 6 a GUI for changing the parameters of the control algorithm is presented. The GUI provides text boxes for each coefficient of the RST and PID control algorithms. The user has the possibility to set new control algorithms and to observe how the processes behaves with the new configuration. The new algorithms are sent to the local application (by means of an implemented TCP/IP based protocol) where the updated control algorithm is applied to the process. The GUI provides also the option (under the form of a button) for restoring the default parameters. For security reasons, in case of bad algorithms (algorithms which lead to instability) the local application will take over and replace the bad algorithms with the default one. The plant has also a safety mechanism which consists in shutting down the heater and starting the fan when the temperature exceeds a certain maximum value.

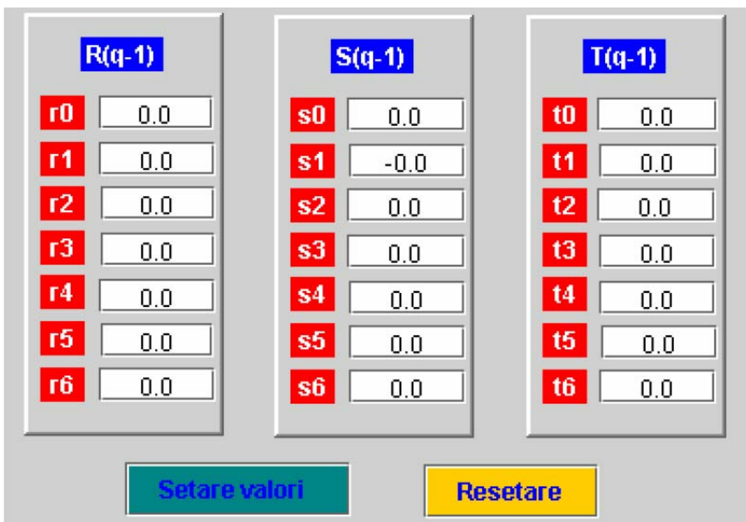

Fig. 6. GUI for control algorithms parameter settings

An important aspect is the study of the Internet connection performances with respect to the network speed, congestion and other parameters that might affect the quality of the remote control laboratory lessons. In direct structure, the network influence (by introducing time-varying delays and packet loses) may damaging to the stability of the close loop. Still in our case, the use of a modified hierarchical structure eliminates the risk of a network delay induced instability. The delay will may affect only the update of the parameters by the client program or the receiving of the new data by the platform, which are not crucial operations. However, the LAN-s in our university are connected through high-speed networks (fiber optics) with speed transfers up the 100 Mbps, which, at least from a local perspective, makes the Internet a relative safe environment.

The access manager

A significant issue related to the remote experiment is the procedure for remotely accessing the air-stream and temperature plant, how it is done and what restrictions are imposed. In order to get permission to connect to the platform a student must first send a request (generally by e-mail) to the administrator. The student is then notified of the acceptance of his request (receiving a user ID and a password), a time window being set (taking in account the other activities involving the platform and the student requests) in which the student may perform the task involving the control laboratory. Only one student is allowed to remotely access the platform per one time window (an exception is made for the administrator who can intervene at any moment). The access manager is a database application (Fig. 7) which contains all the information regarding the connections to the platform: user ID, password, time window and day. Using this software tool the administrator performs all the necessary tasks for the schedule procedure. A log of all accesses to the platform is kept in order to have a clear image the remote activities.

	Id	User	Start Time	End Time	Start date	Stop date
Edit | Delete	1	sieben	00:00:00	23:59:00	2000-01-01	2010-12-31
Edit | Delete	21	admin	00:00:00	23:59:59	2000-01-10	2010-12-15
Edit | Delete	2	teacher	00:00:00	23:59:59	2002-01-10	2010-12-15
Edit | Delete	55	hilde	00:00:00	00:00:00	2000-01-01	2000-01-01
Edit | Delete	35	master	00:30:00	23:45:00	2000-10-25	2005-04-10
Edit | Delete	52	dpopescu	00:00:00	00:00:00	2000-01-01	2000-01-01
Edit | Delete	50	john	00:00:00	23:59:59	2000-01-01	2010-01-01
Edit | Delete	53	catalin	00:00:00	23:59:59	2000-01-01	2010-01-01
Edit | Delete	54	ciprian	00:00:00	00:00:00	2000-01-01	2010-01-01
Edit | Delete	56	mamae	00:00:00	00:00:00	2000-01-01	2000-01-01
Edit | Delete	58	root	00:00:00	23:59:59	2000-01-01	2010-01-01
Edit | Delete	59	Loto	23:00:00	00:00:00	2000-01-01	2007-01-01

Fig. 7. Access manager configuration window

3. Air stream and temperature control

The air-stream and temperature plant has two output measurement points to be controlled: temperature output and pressure output. The input-output transfers in the plant and the influences between the two outputs are depicted in Fig. 8:

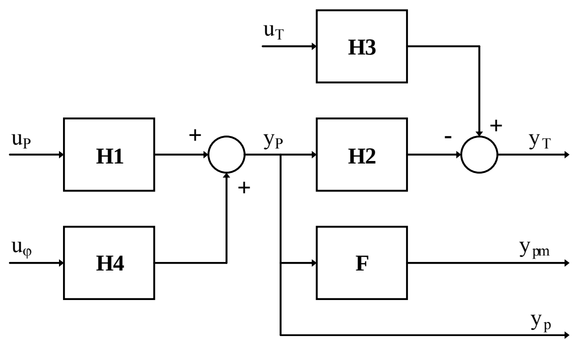

Fig. 8. Air-stream and temperature plant scheme

with:

$$\begin{aligned} y_T(s) &= H_3(s)u_T(s) - H_2(s)y_P(s) \\ y_P(s) &= H_1(s)u_P(s) - H_4(s)y_\varphi(s) \\ y_{pm}(s) &= F(s)u_P(s) \end{aligned} \tag{1}$$

where: $H_1(s)$ designates the pressure control input to pressure unfiltered output transfer, $H_2(s)$ is the pressure output to temperature output transfer, $H_3(s)$ represents the transfer between the temperature control input and temperature output, $H_4(s)$ is influence of the air opening level to the pressure output and $F(s)$ is a transfer function filtering the pressure ($y(s)$and $u(s)$ are the Laplace transforms of the mentioned input and output signals). It can be noticed that the temperature is deeply influenced by the pressure value, therefore we can say that the pressure acts as a disturbance for the temperature output (Fig. 8).

All the transfers presented in Figure behave like a first order transfer function with delay. The following values were experimentally determined:

$$H_1(s) = e^{-0.25s}\frac{0.425}{2.40s+1} \quad H_2(s) = e^{-0.70s}\frac{0.368}{8.35s+1} \tag{2}$$

$$H_3(s) = e^{-0.30s}\,\frac{0.543}{3.30s+1} \quad H_4(s) = \frac{0.030}{2.10s+1} \tag{3}$$

and filter was chosen as:

$$F(s) = \frac{1}{0.08s+1} \tag{4}$$

An RST control scheme [2][5] was used for the two close loops (the temperature close loop is presented in Fig. 9).

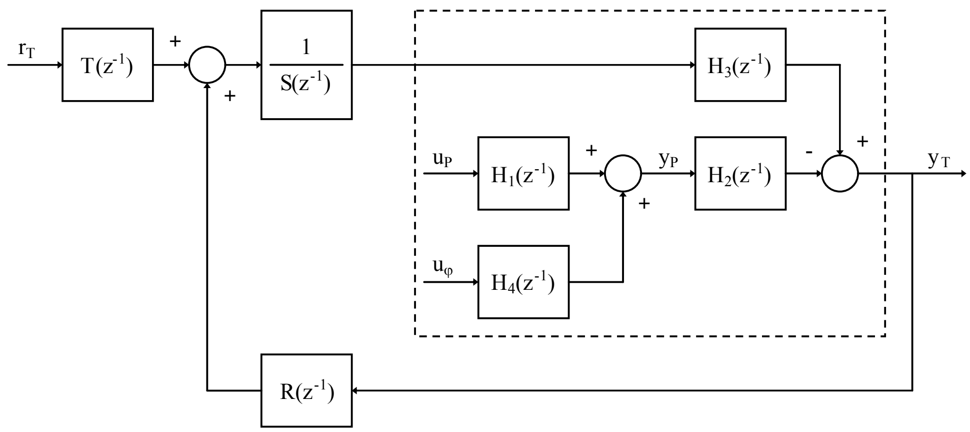

Fig. 9. Block diagram of temperature control close loop

Choosing h_T=0.3 as sampling period for the temperature close loop and h_P = 0.05 for the pressure close loop, the following RST numerical algorithms (after setting appropriate performance criterions) were obtained by using a software tool for control algorithm design–WinREG:

$$\begin{aligned} R_P(z^{-1}) &= 3.438488 - 3.277246z^{-1} + 0.063116z^{-2} \\ S_P(z^{-1}) &= 1 - 0.955681z^{-1} - 0.017777z^{-2} + 0.000628z^{-3} + 0.000952z^{-4} + \\ &\quad + 0.000922z^{-5} - 0.024465z^{-6} - 0.004578z^{-7} \\ T_P(z^{-1}) &= 114.3511 - 221.4958z^{-1} + 107.3691z^{-2} \end{aligned} \tag{5}$$

$$\begin{aligned} R_T(z^{-1}) &= 4.790591 - 3.983895z^{-1} \\ S_T(z^{-1}) &= 1 - 0.794108z^{-1} - 0.205892z^{-2} \\ T_T(z^{-1}) &= 21.19093 - 36.17732z^{-1} + 15.79309z^{-2} \end{aligned} \tag{6}$$

The behavior of the close loops and how the pressure loop influence the temperature is depicted in Fig. 10a. and Fig 10b.

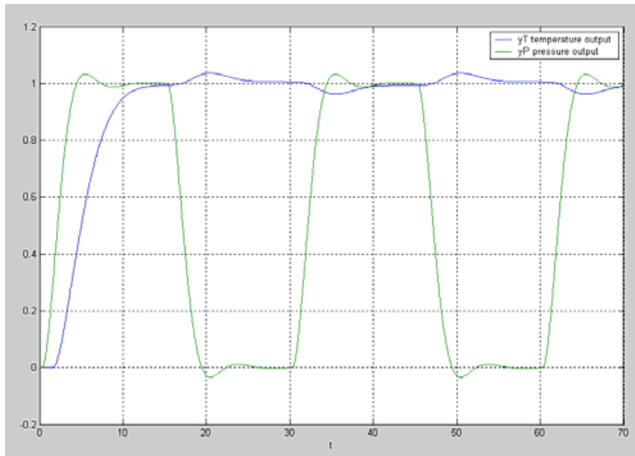

Fig. 10a. plant ututs time evolution

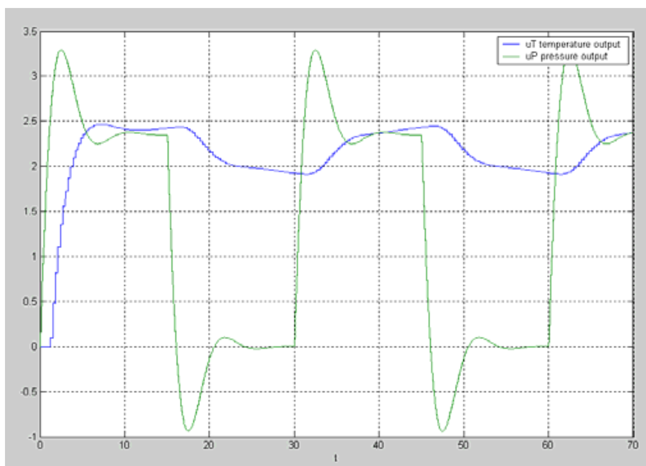

Fig. 10b. plant inputs time evolution

Besides the RST control scheme, other two particular cases of the mention control algorithm class were introduced: PID(1) and PID(2). In consequence the remote control application provides the means for comparison between a one freedom degree and a two freedom degree control algorithm.

It is well known the Internet has a stochastic behavior by introducing delays and packet losses. In our case the system stability is not affected by delays because only the reference is generate remotely. Still with some minor modifications our hierarchical structure could be transform into a direct structure [11] where the controller and the plant are located at different locations. This remote control setup is perfect for testing the robustness of the control algorithms with respect to variable delays introduced by the network. Simple and useful criterions for testing the robustness of a system to variable delays were proposed in [3] [6].

4. Conclusions

The structure presented here has been created as an educational tool for students, in order to experiment remote control of a didactical platform and perform comparative studies regarding the performances of different numerical control algorithms.

By remotely accessing the platform, using the Internet, a student is able to observe (real-time image broadcasting) the evolution of the process, to follow and change the state of the platform (i.e. to perform the lab activities) from any place, just accessing a web page (www.lab2.uv.pub.ro). This manner of distance access transforms the traditional way of education into a more flexible and more personal learning experience.

The platform described in this paper is used in a network of universities (Ecole Supérieure d'Ingénieurs d'Annecy, Polytech'Lille, etc.) created for promoting distance learning, as part of the European project: Socrates-PROG "Joint Development of Postgraduate Program in Electrical Engineering and Automatic Control".

R E F E R E N C E S

[1] *Casini M., Prattichizzo D., Vicino A.,* "Remote Control of a Lego Mobile Robot through the Web". Proceedings of the IBCE'04 IFAC Workshop (2004), Grenoble, France.

[2] *Dauphin-Tanguy G, Foulloy L., Popescu D.,* "Modelisation, Identification et Commande des Systemes". Romanian Academy Press, 2004.

[3] *Kao C.Y., Lincoln B.,* "Simple Stability Criteria for Systems with Time-Varying Delays". Automatica 40(8), p1429-1434, 2004.

[4] *Kondraske, G.V., Voltz R.A., Johnson D.H., Tesar D., Trinkle J.C., Price C.R.,* "Network based infrastructure for distributed remote operations and robotics research". IEEE Trans. Robot. Automat Name (1993), vol. 5, pp. 702-704.

[5] *Landau I. D.,* "Identification et Commande des Systemes", Ed. Hermes, Paris (1995).

[6] *Lincoln B.,* "Dynamic Programming and Time-Varying Delay Systems". PhD Thesis, Lund Institute of Technology, 2003.

[7] *Popescu D., Matei I., Luican I., Catalin C.,* "Remote Process Control–Virtual Laboratory for Distance Learning". Proceedings of the IBCE'04 IFAC Workshop (2004), Grenoble, France.

[8] *Tan K. K., Lee T. H., Sog C. Y,* "Internet-Based Monitoring of Distributed Control System-An Undergraduate Experiment". IEEE Trans. Education (2002), vol. 45, pp. 128-134.
[9] *Tannenbaum A,* "Computer Networks". Ed.Agora (1999).
[10] *Tarn T. J., Xi N.,* "Planning and control of the internet-based teleoperation". Proceedings of SPIE: Telemanipulator and telepresence technologies V, Vol. 3524, pp. 189-193, Boston, MA, 1999.
[11] *Tipsuwan, M. Y. Chow,* "Control methodologies in networked control systems". Control Engineering Practice 11 (2003), pp. 1099-1111, Ed. Elsevier.