An Internet-based Process Control Laboratory Project

Authors:

Valentín Pérez-Herranz, Depto. Ingeniería Química y Nuclear, E.T.S.I. Industriales.,Universidad Politécnica de Valencia, P.O. Box. 22012, E-46071, Valencia, Spain, vperez@iqn.upv.es
Anna Igual Muñoz, anigmu@iqn.upv.es
José L. Guiñón, jlginon@iqn.upv.es
José García-Antón, jgarciaa@iqn.upv.es
Salvador Cardona Navarrete, scardona@iqn.upv.es

Abstract — *This WORK describes a laboratory experiment for conductivity control in a chemical stirred tank reactor, using the Internet. The system is based on a client-server architecture. The server application is used as the data acquisition and control system, and it communicates with the clients situated in the computer laboratory. The client applications are used to manipulate the data that can be saved in a file compatible with conventional spreadsheet programs.*

Index Terms — *Process control, Internet applications, PID control, Laboratory experiments.*

INTRODUCTION

Process dynamics and control are important issues in chemical engineering undergraduate curricula. Students often consider these courses as an abstract study with extensive mathematical derivations, and they will find the courses much more interesting and useful when they are practice oriented [1]. However, many difficulties are encountered when teaching process dynamics and control as part of a chemical engineering degree [2, 3]. These difficulties range from the lack of time available to teach even the most fundamental principles, to providing sufficient engineering practice of the theory taught in the classroom, besides the increasing in the class sizes and the decrease in resources. These problems cause students to have to carry out fewer experiments or to have to work in large groups.

One solution to the problems mentioned before is to use the new software applications for the development of virtual instruments and tools for computer interconnection through the Internet to design applications for process control laboratory courses. The advances in software engineering techniques make it possible to build open, interoperable, modular, and extensible distributed measurement systems, which can be used to monitor and control the physical environments. On the other hand, the rapid growth of fast and reliable communication networks has allowed an easy exchange of information and commands between computers connected to sites of wide area networks such as the Internet [4, 5].

One advantage of this type of operation is the easy and efficient use of expensive or complex instrumentation systems. The remote access to measurement systems allows the interaction between students and instrumentation without the need to physically move people or bulk instrumentation. A second advantage coming from the availability of a remote instrumentation is the possibility of arranging measurement systems which are both controlled from a single position and able to perform complex measurements strictly related to the site where instrumentation is located. This working option is of interest for universities with a large number of students due to the high cost of the experimental laboratories.

Many papers have been published describing several types of process control laboratory experiments and course objectives and approaches [6-9]. In this paper, a laboratory experiment for conductivity control in a chemical stirred tank reactor, using the Internet, is described. The apparatus is fairly simple, it is safe to operate and the investment in process equipment and instrumentation is modest. The project progress from understanding the process, to analyze the dynamics responses, to add controllers, to finally implement the controllers to the system. The technique proposed is based on a client-server architecture, which is described below. With this technique, many users can simultaneously share remote measurement and control by using suitable client procedures that interact with a server program running on a computer physically connected to the system to be controlled.

DESCRIPTION OF THE CLIENT-SERVER ARCHITECTURE

A client-server environment as shown in Figure 1 has been designed and implemented. The general architecture consists of a set of 30 **Client** measurement stations located in a computer laboratory, all networked to a **Server** station, which is located in the laboratory where the plant to be controlled is situated. The server acts as the data acquisition and control system. The acquired data are distributed through the network to the rest of the clients where the data are analyzed. In order to remotely control the plant from the computer laboratory, one of the computers, named **Remote Controller** in Figure 1, is used to send information to the server station. This information is given in terms of the controller status, set-points and the parameters of the controller in the automatic operation mode, and the value of the manipulated variable in the manual operation mode. This feature allows the experimental system to be operated via any computer connected to the Internet. In particular, it allows the process control hardware to be operated from the lecture room, thus allowing the system to be used for lecture theater demonstrations.

To enable the system to be operated, three applications have been developed that allow the user to access similar displays to those available on the control system:

- The **Server** application is constantly running in the computer server station. It contains the network related procedures on the server side. The server also contains the procedures related to instrument management. It has been designed to operate with serial ports and is used to manage acquisition boards, which are employed for control purposes.
- The **Client** application contains the procedures that are related to the network of the client side as well as to the measurements processing. This application permits the data to be saved in order to be manipulated further with conventional spreadsheet programs.
- The third application, called **Remote Controller**, contains the procedures that are related to the user interface of the parameters of the controller. This application has been developed in order to avoid students from different computers from sending contradictory orders to the controller at the same time.

The applications have been developed using the LabVIEW program from NATIONAL INSTRUMENTS. LabVIEW is a graphical programming language for building data acquisition and instrumentation systems, and is ideally suited for the collection and distribution of data for basic process monitoring and control applications. It facilitates fast program development, provides many powerful built-in functions, and has universal acceptance in research and manufacturing settings [10, 11]. Applications are written by placing and connecting icons that represent various built-in and user-developed functions on a block diagram. The connection of icons provides data flow execution of program nodes.

The remote control of the experiments and the connection between the server and the clients was done using DataSocket. DataSocket has been designed for sharing and publishing live data in measurement and automation applications over the Internet and responds to multiple users without the complexity of low-level TCP programming. To perform these tasks using DataSocket technology two basic steps must be performed:

- Open a DataSocket connection using the name chosen to identify the data.
- Write the data to that connection.

The graphical user interfaces (GUI) of the server station, clients and remote controller are shown in Figures 2 to 4 respectively. In the GUI of the server station, Figure 2, the actual controller status, the parameters of the controller, set point and process variables are indicated. The box situated at the left top corner of the GUI indicates the IP address of the computer where the remote controller application is running. The controlled variable and the signal sent to the final control element are represented in two graphics. The example shown in Figure 2 corresponds to the closed loop response of the system to changes in the set point when a PI controller is used.

In the client GUI, Figure 3, the data received from the server station are represented. The example shown in Figure 3 corresponds to the open loop response of the system to step changes in the voltage applied to the final control element. Similarly to Figure 2, the IP address of the server station is indicated in the box situated at the left top corner of the GUI.

The GUI of the remote controller is shown in Figure 4. In this application, the control status, voltage applied to the final control element, set point and controller parameters can be modified and sent to the server station, whose IP address must be given to the program. This application is protected with a password in order to avoid running this application simultaneously from different computers.

DESCRIPTION OF THE EXPERIMENTAL EQUIPMENT

Figure 5 shows a simplified diagram of the experimental arrangement. It consists of a 2 l cylindrical chemical stirred tank reactor (CSTR) in which the control of the solution conductivity takes place. For this purpose, two streams enter the reactor. One of the streams is tap water with a flow rate F_W and conductivity λ_W. This stream acts as disturbance and is fed to the reactor using a centrifugal pump. The other stream is a 0.02 M solution of hydrochloric acid with a flow rate F_A and conductivity λ_A. The hydrochloric acid flow rate, F_A, which is applied using a MAXTERFLEX, L/S, 10-47 peristaltic pump, is used to control the conductivity, λ, of the solution inside the reactor. The liquid in the tank overflows, then the volume of the liquid in the tank is considered constant.

Conductivity values are measured using a CRISON conductivity-meter. The data are fed into the computer via a Lab-PC+ data acquisition (DAQ) board from National Instruments. The flow, F_A, is controlled through a 0-5 V analog voltage signal, which is applied to the peristaltic pump by the analog output of the DAQ board. The voltage applied to the peristaltic pump and the flow, F_A, are related by a proportional gain, K_F = 17.2 l/hV. The main parameters are measured with a sample time of 1 s. In order to smooth drastic variations during conductivity measurement, the arithmetic mean value of 20 conductivity values are calculated. All the runs are carried out at room temperature.

It is possible to run the experiment with or without control. Control strategies available are P only, PI or PID. When running without control, the acid flow in, F_A, can be modified through the voltage applied to the peristaltic pump. In this case, the open loop response of the system is given. When using the controller, the set point and the parameters of the controller can be modified. Then, the effect of the parameters of the controller on the closed loop response can be studied by changing the values directly on the remote controller program (Figure 4). For a P only controller, the parameters τ_I and τ_D are zero, while for a PI controller, τ_D is zero.

CONTROL EXPERIMENTS

The experimental project is carried out in two sessions. In the first session the dynamic behavior of the system is studied by modifying the acid flow, F_A, through the voltage, V, applied to the peristaltic pump, and measuring the conductivity, λ, in the tank until the new steady state is reached. The process reaction curve is obtained from the conductivity data and the process response is adjusted to a first order system with time delay. Several experiments are conducted for different nominal conditions of the manipulated variable as shown in the graphics of Figure 3.

In the second session, the parameters of the controller are calculated applying the Ziegler Nichols open-loop method of tuning controllers and the techniques of frequency response. The results obtained by the two methods are compared.

In the first method, the process parameters, time delay, τ_d, process gain, K, and time constant, τ, are related to the parameters of the controller, controller gain, K_C, reset time, τ_I, and derivative time, τ_D. The method is outlined as follows:

- Look at the open loop response of the process to a step change in the manipulated variable.
- Evaluate the process parameters.
  - Steady-state gain, $K = (\lambda_2-\lambda_1)/(V_2-V_1)$
  - Time delay, τ_d
  - Time constant, τ
- Calculate the parameters of the controller according to Table 1.

In the second method Bode stability criterion is applied. The parameters of the controller are calculated through the following steps:

- Using the open loop transfer function and a proportional only control, the proportional gain of the controller that gives a continuous oscillation is determined. This will be the ultimate gain, K_u.
- Determine the frequency of the continuous oscillation, or crossover frequency, ω_{CO} and the ultimate period of sustained cycling, $P_U = 2\pi/\omega_{CO}$.
- Calculate the controller parameters according to Table 2.

Once the controller parameters are calculated using the two methods, students must compare the parameters of the controller obtained by both methods. Finally, different controllers can be applied to control the conductivity inside the tank. The performance of the controllers implemented to the process is studied for set point tracking and disturbances rejection. The effect of the parameters of the controller on the closed loop response can be studied, i.e. the effect of the proportional gain on the off set when a proportional only controller is used.

CONCLUSIONS

An experimental laboratory project to control conductivity in a CSTR through the Internet has been developed. For this purpose, client-server applications have been developed to allow PCs to communicate with the control system. These applications enhance the flexibility of the system for teaching purposes, and are especially suitable for universities with a large number of students. The system can also be used for distance education and remote control of process equipment.

This facility can be applied to control other systems such as pH control, level control, or temperature control with little changes in the process equipment. Moreover other control algorithms can be applied to the control system with little programming effort.

ACKNOWLEDGMENTS

We thank María Asunción Jaime for her help in translating this paper into English.

REFERENCES

[1] Jovan, V. Petrovcic, J. "Process Laboratory: A New Necessary Resource in Control Engineering Education" *Comp. Chem. Eng.*, **20**, 1996, 1335.

[2] Lewin, D. R., Rockman, J., Lavie, R. "Teaching Advanced Process Control to Undergraduates" *Comp. Chem. Eng.*, **20**, 1996, 1347.

[3] Lennox, B., Brisk, M. " Network Process Control Laboratory" *Chem. Eng. Ed.*, **32**,1998, 314.

[4] Grimaldi, D., Nigro, L., Pupo, F. "Java-Based Distributed Measurement Systems". *IEEE Trans. Instrum. Meas.*, **47**, 1998, 100.

[5] Bertocco, M., Ferraris, F., Offelli, C., Parvis, M. "A Client-Server Architecture for Distributed Measurement Systems". *IEEE Trans. Instrum. Meas.*, **47**, 1998, 1143.

[6] Lira, C. T. "Computer Control of a Distillation Experiment" *Chem. Eng. Ed.*, **26**, 1992, 38.

[7] Bequete, B. W., Schott, K. D., Prasad, V., Natarajan, V., Rao, R. R. "Case Study Projects in an Undergraduate Process Control Course" *Chem. Eng. Ed.*, **32**, 1998, 214.

[8] Gatzke, E. P., Vadigepalli, R., Meadows, E. S., Doyle, F. J. "Experiences with an Experimental Project in a Graduate Control Course" *Chem. Eng. Ed.*, **33**, 1999, 270.

[9] Luyben, W. L. "A Feed-Efluent Heat Exchanger / Reactor Dynamic Control Laboratory Experiment" *Chem. Eng. Ed.*, **34**, 2000, 56.

[10] Davis, R. A. "Create Virtual Unit Operations with your Data Acquisition Software" *Chem. Eng. Ed.*, **29**, 1995, 270.

[11] Gostowski, R. "Teaching Analytical Instrument Design with LabVIEW" *J. Chem. Ed.*, **73**, 1996, 1103.

FIGURES AND TABLES

TABLE 1
ZIEGLER NICHOLS OPEN LOOP RULES FOR CONTROLLERS DESIGN.

Controller Type	Gain, K_C	Reset time, τ_I	Derivative time, τ_D
P	$\frac{1}{K}\frac{\tau}{\tau_d}$		
PI	$\frac{0.9}{K}\frac{\tau}{\tau_d}$	$3.3\ \tau_d$	
PID	$\frac{1.2}{K}\frac{\tau}{\tau_d}$	$2.0\ \tau_d$	$0.5\ \tau_d$

TABLE 2
ZIEGLER NICHOLS CLOSED LOOP RULES FOR CONTROLLERS DESIGN.

Controller Type	Gain, K_C	Reset time, τ_I	Derivative time, τ_D
P	$\frac{K_u}{2}$		
PI	$\frac{K_u}{2.2}$	$\frac{P_u}{1.2}$	
PID	$\frac{K_u}{1.7}$	$\frac{P_u}{2}$	$\frac{P_u}{8}$

FIGURE 1
CLIENT-SERVER ARCHITECTURE FOR THE REMOTE CONTROL.

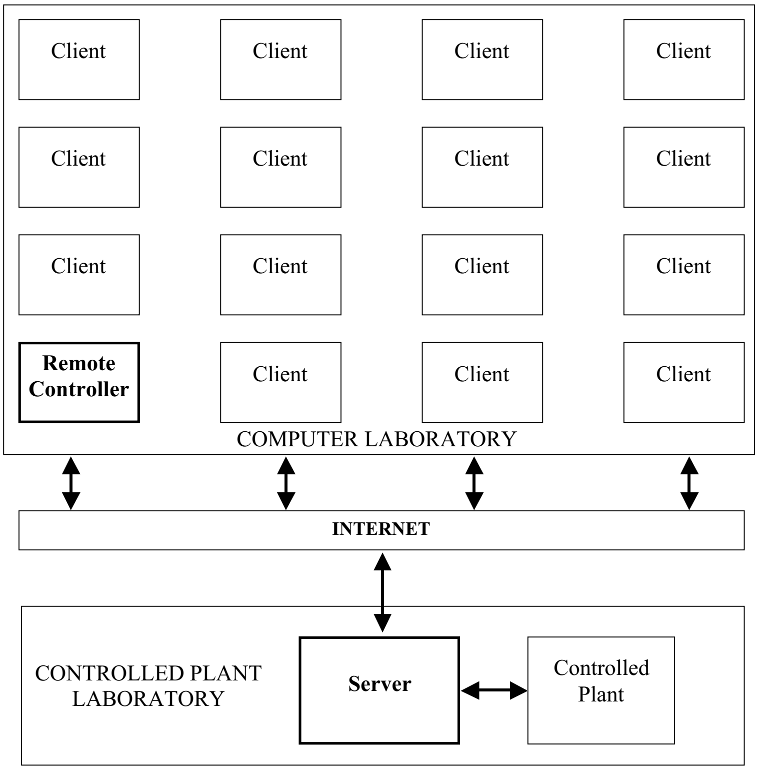

FIGURE 2
GRAPHICAL USER INTERFACE (GUI) OF THE SERVER.

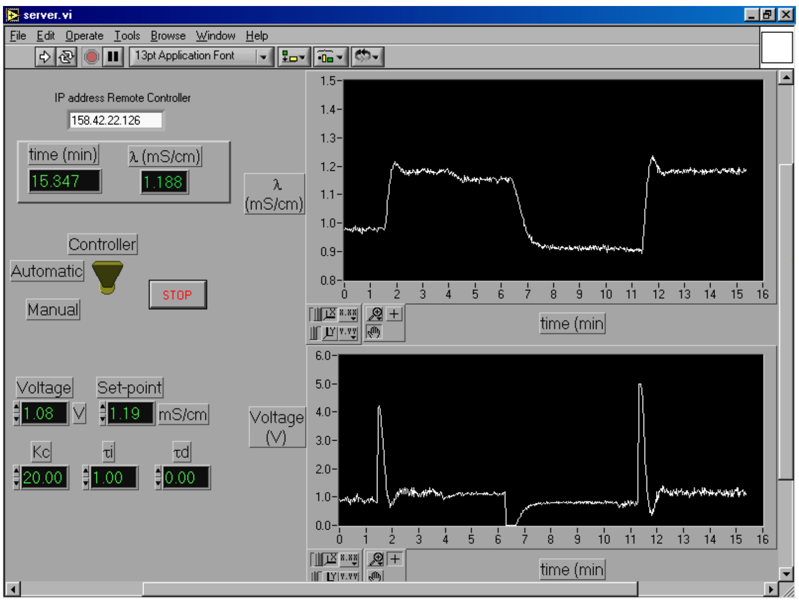

FIGURE 3
GRAPHICAL USER INTERFACE (GUI) OF THE CLIENTS.

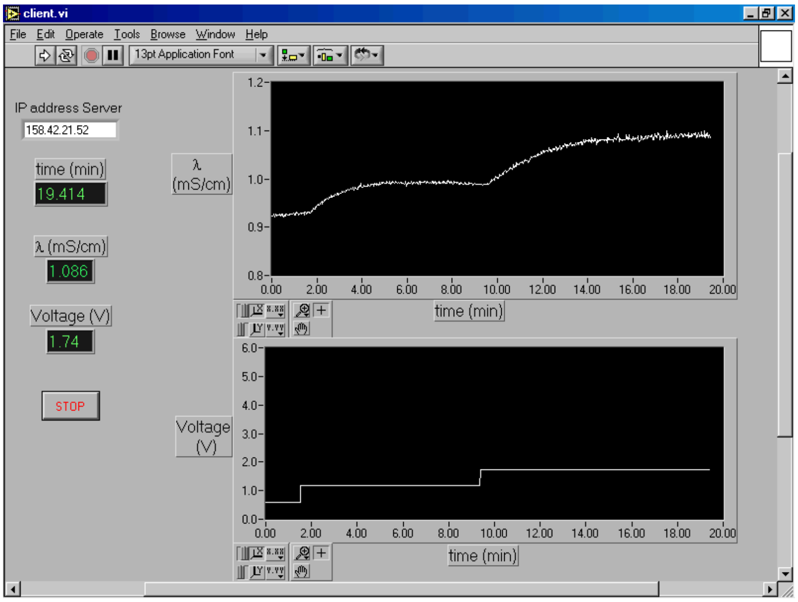

FIGURE 4
GRAPHICAL USER INTERFACE (GUI) OF THE REMOTE CONTROLLER NTS.

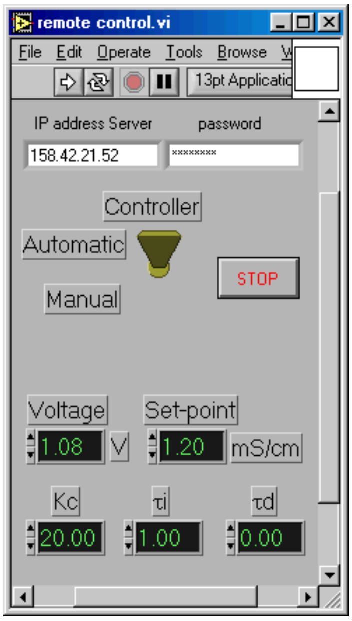

FIGURE 5
DIAGRAM OF THE EXPERIMENTAL SETUP.