A Remote-Control Engineering Laboratory

Corneliu Lazar, *Member, IEEE*, and Sorin Carari

***Abstract*—In this paper, a networked control-system laboratory for the remote control of processes is presented. The laboratory allows the students to develop network-based control systems with the use of an architecture consisting of I/O devices, communication modules, and server–client applications implemented with supervisory control and data acquisition environment facilities, and to operate on real pilot plants through an intranet and the Internet. Three examples were presented, demonstrating the potentiality of the laboratory to remotely control the processes and to develop new network-based control-system structures.**

***Index Terms*—Control engineering, networked control system (NCS), remote experiment, spatially distributed systems, supervisory control and data-acquisition (SCADA) systems.**

I. Introduction

NETWORKING is the foremost requirement for current structures in process control instrumentations. By means of control equipment that is able to transmit the information through a shared-communication medium, it is possible to build network-oriented structures in process automation systems. These structures have numerous advantages in comparison with the traditional process automation techniques component oriented on the basis of hardware components, such as sensors, actuators, transmitters, and controllers arranged in the conventional hierarchical fashion. Networks enable remote data transfers and data exchanges among users, reduce the complexity in wiring connections, and provide easy maintenance. Field buses are used to network sensors, actuators, and control devices inside a network-based control system. Using field-area networks, a new model for a networked control system (NCS) is presented in [1], and the Internet integration of field bus systems is shown in [2]. At the field level, wireless communication systems are a solution for the implementation of an NCS [3].

Due to these benefits, many industrial companies and institutes apply reconfigurable distributed NCSs for remote-control purposes and factory automation [4]–[7]. Control applications can be connected to the Internet in order to perform remote control at much farther distances than in the past, without investing on the whole infrastructure. Replacing specialized networks with the Internet is a new trend in industrial informatics. The quality of control in NCSs depends on the network traffic which introduces delays. In [8], the optimization of the quality of control using controller and message-scheduling codesign is presented. Another methodology given in [9] and [10] enables existing controllers for networked control and teleoperation by middleware that allows one to modify the command based on a gain-scheduling algorithm corresponding with the current network traffic. The influence of the denial of service attacks, which produces disturbance inside the Internet, to the NCS performance is analyzed in [11]. The state of the art of the research in the field of control and communication in networked real-time systems was the subject of special issues of several journals during the last years, such as IEEE Control Systems Magazine (February, 2001) [12], IEEE Transactions on Automatic Control (September, 2004) [13], or Proceedings of IEEE (January, 2007) [14].

There are two general NCS configurations, namely direct and hierarchical structures [5]. The NCS in the direct structure is composed of a controller and a remote system containing a physical plant, sensors, and actuators. The controller and the plant are physically located at different areas and are directly linked by a data network in order to perform remote closed-loop control. The control-action and sensor measurements are exchanged by means of a shared-communication digital network.

The basic hierarchical structure consists of a main controller and a remote closed-loop system. Periodically, the main controller computes and sends the reference signal in a frame or a packet via a network to the remote system. The remote system then processes the reference signal to perform local closed-loop control and returns the sensor measurement to the main controller for networked closed-loop control. The networked control loop usually has a longer sampling period than the local control loop because the remote controller is supposed to satisfy the reference signal before it processes the newly arrived reference signal.

The connection between new network-based control systems and practice is one of the most difficult lessons to teach in control engineering. Many universities developed remote-control engineering laboratories based on NCS, such as those given in [15]–[19]. For several years at the Automatic Control and Applied Informatics Department in "Gh. Asachi" Technical University of Iaşi, we have experimented with different teaching techniques, such as demonstrations, projects, and laboratory work in the field of NCSs [20]–[23].

This paper presents the experience of the control group at the Department of Automatic Control and Applied Informatics from Iaşi in developing an NCS laboratory. Two systems were developed to illustrate the direct and the hierarchical NCSs. For the direct structure, a supervisory control and data acquisition (SCADA) environment has been chosen to implement the network-based control architecture, whereas for the hierarchical structure, the entire system was developed. These control systems allow the user to remotely choose a predefined controller to steer the process or to design a new one.

Manuscript received March 29, 2007; revised February 13, 2008.
The authors are with the Department of Automatic Control and Applied Informatics, "Gh. Asachi" Technical University of Iaşi, 700050 Iaşi, Romania (e-mail: clazar@ac.tuiasi.ro; scarari@ac.tuiasi.ro).
Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.
Digital Object Identifier 10.1109/TIE.2008.920650

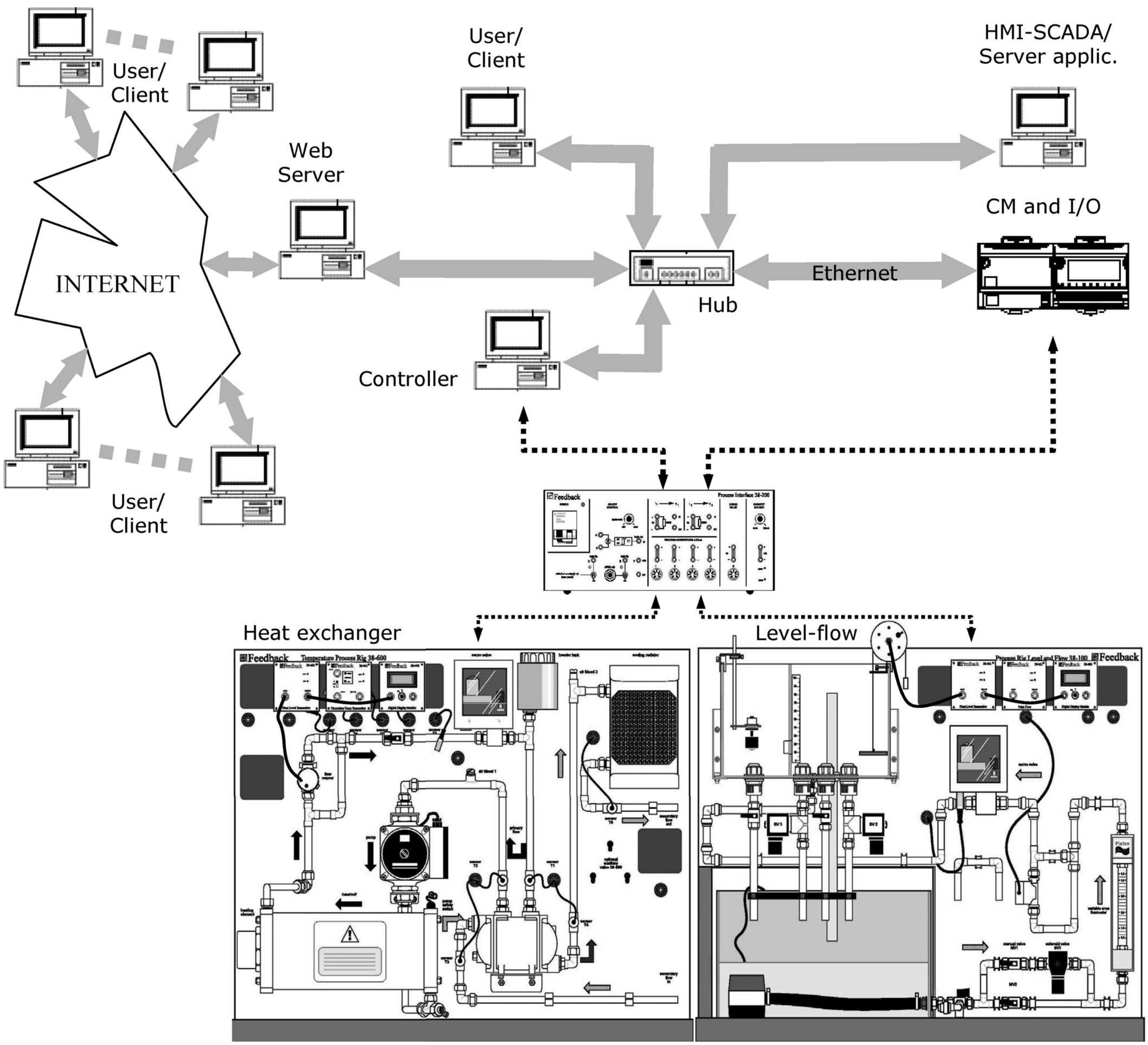

Fig. 1. Remote-control architecture.

Using SCADA software facilities, the students themselves can develop new NCSs for the pilot plants from the laboratory. The main advantage of the network-based control structure is the user interface, which allows the user to tune the controller and to test it through the remote laboratory. During the experiments, it is possible to change the set point, the operating mode, and some typical controller parameters. Experimental results can be displayed, showing the real running experiment, and can be checked through online plots.

The NCS laboratory is mainly intended for educational use, and it has been employed in remote-control engineering and industrial informatics courses. The aim of the laboratory is to allow students to put their theoretical knowledge of control and networking in practice in an easy way and without restrictions regarding the process availability.

This paper is organized as follows. Section II illustrates the main features and the architecture of the NCS laboratory. In Section III, typical working sessions are described. Section IV provides conclusions and future developments.

II. NCS Laboratory Configuration

In order to achieve the remote control of pilot plants from the NCS laboratory, a Web server is used, which assures the processes distribution for different users (clients) via an intranet and the Internet, as shown in Fig. 1. The processes distribution is realized by using an NCS architecture, which implements both configurations, namely the direct and hierarchical structures. SCADA software enables programmers to create distributed network control applications having supervisory facilities and a human–machine interface (HMI). As a SCADA environment, LOOKOUT is used for the direct structure, and a self-made software is used for the hierarchical structure. All external signals start and arrive at the HMI/SCADA computer.

The laboratory architecture allows one to run experiments while interacting with instruments and remote devices. The I/O remote device permits data acquisition from sensors and supplies the control signals for actuators, using A/D and D/A converters.

The NCS architecture offers the possibility to remotely choose a predefined control structure to handle the process variables or to design a new control application using SCADA software facilities. In the first case, which is using the remote-control architecture, students have the possibility of practicing their theoretical knowledge of process control and networking in an easy way due to the process access by a user-friendly interface. The second opportunity offered to the students is to design a new networked control architecture which allows the creation of a new HMI/SCADA application to remotely control a process.

The software architecture can be split into two parts. One concerns the control of the physical process — the server side, and the other relates to the user interface — the client side. The server runs on the Microsoft Windows NT platform.

The server application contains the HMI interface and fulfils the following functions:

1) implements the control strategies;
2) communicates with I/O devices through object drives;
3) records the signals in a database;
4) defines the alarms.

The client process contains an HMI interface, which is similar or not with those from the server application, and has the following characteristics:

1) allows one to remotely modify the parameters defined by the server application through a Web site;
2) communicates with server application;
3) displays the alarms defined by the server application.

The client application runs on a Web server and can be accessed by using a Web browser. In order to run the client process, the LOOKOUT Web client software has to be installed on the user client computers. At the same time, it is necessary that the HMI/SCADA computer and the process should operate and that the Web server should be running.

The NCS in the direct structure is composed of a computer of the intranet, which is called HMI/SCADA, which achieves local communications with the process using Ethernet protocols. The remote processes, which are two pilot plants — one for level-flow control and one for the heat exchanger — are connected with the communication module (CM) and are able to transfer data from/to the I/O device to/from the HMI/SCADA computer via a communications system. A remote-control application was also developed for a dc motor, which involved the control of the shaft angle or rotation speed. The CM and I/O devices are implemented with National Instruments modules FP1600 (Ethernet), for communication, and dual-channel modules (1.5-ms scan time for analog I/O), for I/O devices. This hardware, together with the LOOKOUT SCADA software, clearly emphasizes the concepts of the distributed control systems and the NCSs in comparison with other SCADA software, such as Labview.

Considering that NCS in the direct structure operates over a network, the total delay involved in closed-loop communications can be divided in three parts, namely: the communication delay from the sensor to the controller, the computation delay at the controller, and the communication delay from the controller to the actuator. The computation delay is considered small in comparison with the communication delay, which is also small if the user is located at a computer from the intranet; however, it can be significant if the user employs the Internet to remotely control a laboratory setup.

The NCS in the direct structure allows the development of different control architectures using the PID controllers from the LOOKOUT environment, starting with the basic feedback loop with one control variable and one measured signal and extending to complex control problems with many control variables and many measured signals (cascade control, feed-forward, and ratio control).

The NCS in the hierarchical structure comprises three components, namely: the remote closed-loop system, the Web server, and the clients. The controller communicates with the controlled process through a data acquisition (DAQ) board from National Instruments (AT-MIO-16E-10, with a sample rate of 100 kS/s) and implements the control strategy. The control law is expressed by using the difference equation

$$\begin{aligned} u(k) = {} & a_1 u(k-1) + \ldots a_n u(k-n) + q_0 e(k) \\ & + q_1 e(k-1) + \cdots + q_m e(k-m) \end{aligned} \tag{1}$$

where a_i, $i = \overline{1,n}$ and q_j, $j = \overline{0,m}$ are the controller parameters, e is the control error, and u is the control action. Based on (1), various control strategies can be implemented, such as discrete PID, pole placement, deadbeat, etc.

Due to the sample rate that is greater than that of the I/O device of the direct structure architecture, the NCS in the hierarchical structure can be used both for controlling the two pilot plants (level flow and heat exchanger) and for supervising the stepper motors or brushless dc servo motors [24]. The controller receives the commands for controller parameter settings or transmits data to a remote client by means of a Web server.

The Web server ensures the access of a remote client to the remote closed-loop system via the Internet by means of a set of Web pages. The controller does not have his own interface, and this interface can be remotely accessed only by using the Web server.

The NCS laboratory development was financed within a grant of 50 000 USD by the Romanian Government and the World Bank, and the laboratory has been used in control-engineering courses at the Technical University of Iaşi since 2003. A typical student laboratory session consists of studying the physical and mathematical models of a pilot plant from the laboratory, designing and validating a controller offline, setting and running the NCS real-time experiment, downloading data, and analyzing the control-system performance offline. The students appreciate the fact that they can remotely control real processes even outside of the laboratory work. Thus, they can complete or repeat the experiments in which they are interested.

Regarding the NCS safety, hardware and software actuator saturation is enforced to prevent the users from performing

Fig. 2. Laboratory pilot plants. Level flow and heat exchanger.

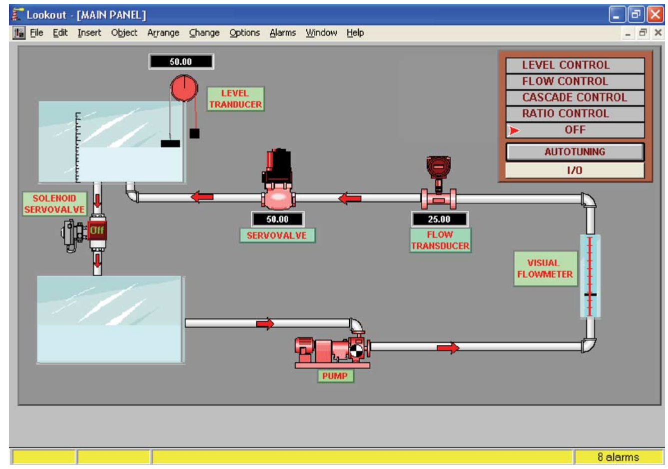

Fig. 3. Main panel for level-flow control system.

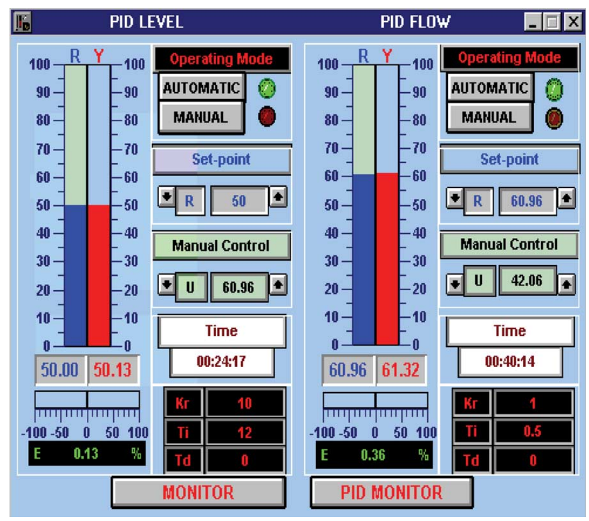

Fig. 4. Control panel for cascade-control structure.

dangerous operations. Similarly, saturation is enforced on the input reference.

In the case of more users who want to connect to the controlled process, the first-come–first-serve rule was introduced in order to solve the simultaneous access to the NCS laboratory. The first user will have the right to change the control-system parameters and the next ones only to read them. If someone controls the process from the HMI/SCADA computer, all remote users are passed to the read priority.

III. Experimental Session

This section presents three examples to illustrate how an NCS works or can be developed. The two pilot plants from the laboratory are shown in Fig. 2 and can be linked to remote-control architecture via a process interface and suitable I/O devices.

The first examples exemplify the laboratory works dedicated to control the pilot plants with the NCS in the direct and hierarchical structures. The last example is devoted to the project work, where the students learn to build their own HMI/SCADA applications for controlling various setups in the laboratory.

A. NCS in the Direct Structure

The NCS in the direct structure was designed and implemented by using LOOKOUT facilities to remotely control the two pilot plants from the laboratory, namely the level flow and the heat exchanger.

The level-flow system, which is shown in Fig. 2, uses water as the process fluid and allows one to study the principles of process control using liquid level and flow rates as the process variables to be controlled. This setup can be used to illustrate different control strategies. Thus, when employing networked control architecture and LOOKOUT facilities, a SCADA/HMI application was developed. This allows for the study level, flow, and cascade controls of the level/flow and, if employing the heat-exchanger process, the set-point ratio control.

The student graphic interface is shown in Fig. 3. It is loaded from the server when the editing session starts, and it runs inside the browser. For the user, the only visible part of the level-flow control application is the HMI that is implemented as a Web page. With the HMI, it is possible to monitor and control, in real time, the process that is connected to the system. The main panel contains the layout of the control-system and real-time data from the sensors and actuator and allows the choice of the control structure.

After the choice of the control structure, a new window containing the control panel and real-time data from the process is displayed. For the cascade-control structure, there are two control closed loops, which are the inner one, for flow control, and the outer one, for level control.

The control panel contains two PID controllers for flow and level, as is shown in Fig. 4. This panel contains the controller interface and allows one to change the controller tuning parameters and the set point and to modify the operating mode.

The PID controllers are similar to the industrial ones, having a user-friendly interface. Students can achieve numerous experiments in order to tune the controllers or to understand the influence of tuning parameters upon the control performances, without navigating through a complex menu of an industrial controller.

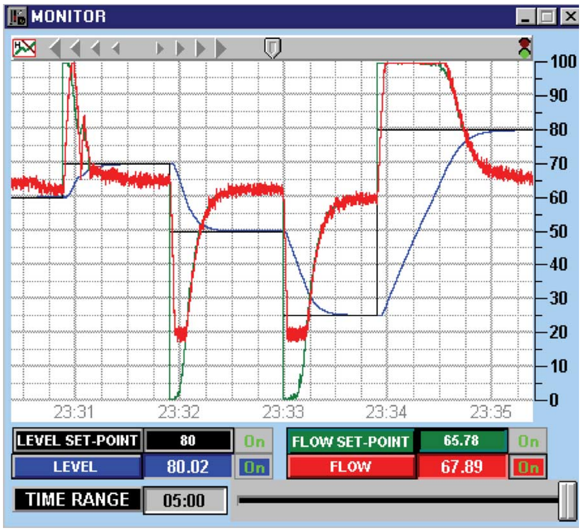

Fig. 5. Monitor window for cascade-control structure.

Using the monitor button from the controller panel, the set point and the controlled and command signals are displayed in a new window. For the cascade-control structure, the monitor window is shown in Fig. 5, with the level as the primary control variable and the flow as the secondary control variable. The monitor windows have the possibility of representing only the selected signals by using on/off buttons, by stopping the variables from displaying in order to analyze the signals, or changing the timescale through a slide button. All control-system real-time data are recorded in a database and can be employed for a future analysis.

For designing the PID controllers, students can apply empirical tuning based on the Ziegler–Nichols oscillation, on reaction-curve methods, or on analytical ones [25]–[27]. For the analytical procedures, first, a process model is derived by using the reaction-curve method and the manual operating mode of the controller. Having a process model, the PID controller is designed by using a pole-placement approach or optimization techniques.

An autotuning approach based on the reaction curve and model-parameter computing with the area method [27] is implemented and can be used for activating the autotuning button in the control panel. When the autotuning procedure is started, the autotuning panel from Fig. 6 is displayed, and a step signal with the desired amplitude introduced via the panel is applied at the process input.

After the detection of the steady-state regime, the parameters K_f, T_f, and the time delay L of the first order model with deadtime

$$G(s) = \frac{K_\mathrm{f}}{1+sT_\mathrm{f}} e^{-sL} \tag{2}$$

are computed. Having the static gain K_f, the average residence time T_ar is first determined by using the ratio

$$T_\mathrm{ar} = \frac{A_0}{K_\mathrm{f}} \tag{3}$$

where A_0 is the area between the step response, the horizontal K_f, and the y-axis.

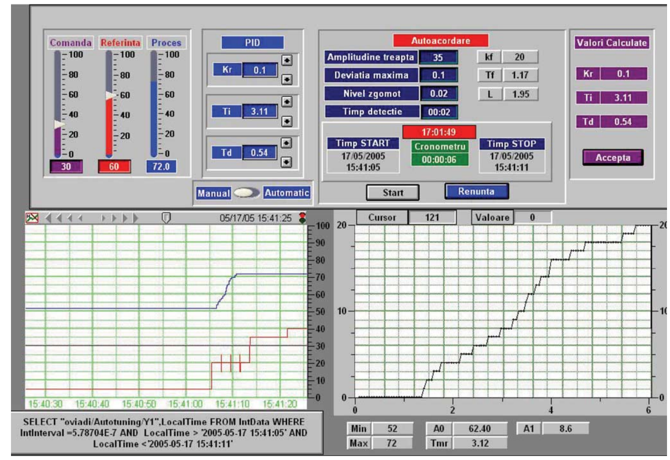

Fig. 6. Autotuning panel.

Computing the area A_1 under the step response up to the time T_ar, the time constant T_f and the deadtime L are thus given by

$$\begin{aligned} T_\mathrm{f} &= \frac{eA_1}{K_\mathrm{f}} \\ L &= T_\mathrm{ar} - T_\mathrm{f} = \frac{A_0}{K_\mathrm{f}} - \frac{eA_1}{K_\mathrm{f}}. \end{aligned} \tag{4}$$

Based on model parameters, the tuning parameters K_r, T_i, and T_d are determined and displayed. If the tuning parameters are accepted by the user, the PID controller will run with these new values. In the lower part of the autotuning panel, the process variables (set point, control action, and controlled output) are displayed together with a scaled representation of the controlled variable and the results obtained by applying the area method.

The heat-exchanger process is a two-loop system, using water as the process fluid, which allows the study of the principles of the process control using primary and secondary circuit temperatures as the process variables to be controlled.

A forced air cooler is also available. It accelerates the process dynamics using a constant input temperature, allowing a high temperature differential to be monitored for longer periods, and introduces load disturbance. This setup shown in Fig. 2 is used to illustrate the cascade control with feedforward load-disturbance rejection.

The primary controlled variable is the output temperature of the secondary circuit temperature, the secondary controlled variable is the flow from the primary circuit temperature and the load disturbance is the flow from the secondary circuit temperature. The students have to design the two controllers of the cascade control system using empirical or analytical methods and to determine a plant model having as input the measurable disturbance in order to design the feedforward controller. Starting the SCADA/HMI application, the main panel from Fig. 7 is displayed. The main panel contains a schematic description of the controlled process with information about sensors status and a button for switching to the control panel.

The control panel, which is shown in Fig. 8, ensures access to the controllers' tuning parameters of the two control loops of the cascade structure and to the feedforward controller, offers

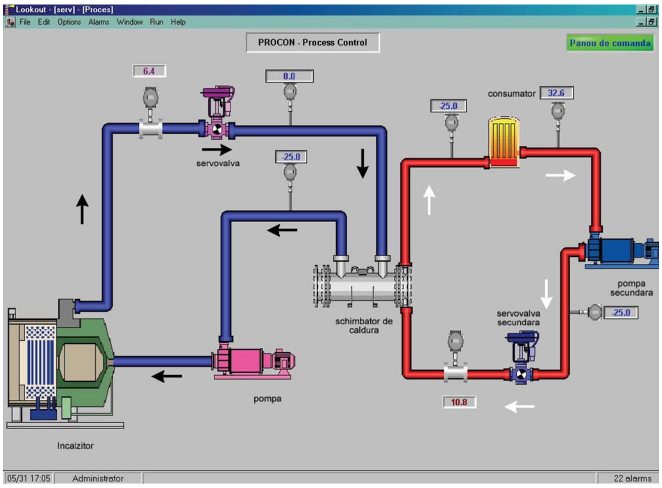

Fig. 7. Layout of heat-exchanger control system.

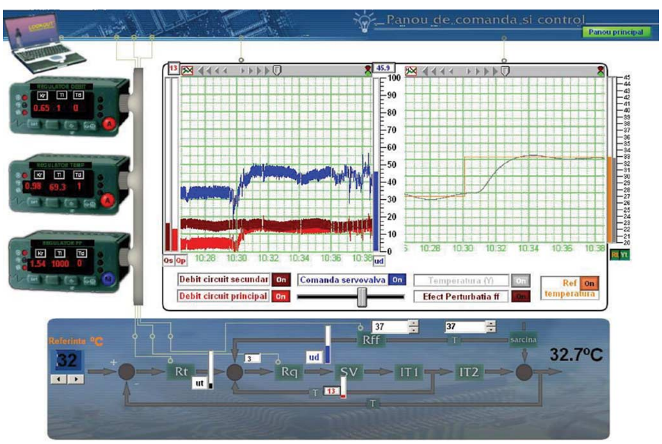

Fig. 8. Control panel for the heat-exchanger process.

the possibility to change the set point and the operating mode, and displays the evolution of the monitored signals.

In Fig. 8, the left trend shows the evolution of the control signals, while the right one displays the controlled variable (temperature at heat exchanger output) and the set point.

B. NCS in Hierarchical Structure

The NCS in the hierarchical structure was designed for Web-based remote-control experiments. The system uses recent Internet technologies, without need of an additional environment for data communication.

From the software point of view, the development of NCS was done by having in mind the idea that the system requires no additional software for a remote user. To accomplish that, the user interface was realized as sets of Web pages from where all the working parameters could be settled or the dynamic of the signals displayed.

When accessing the application, the start page shown in Fig. 9 is displayed. This page contains a schematic description of the NCS and the links to the pages for setting the working parameters for the DAQ board and for automatic and manual operating modes.

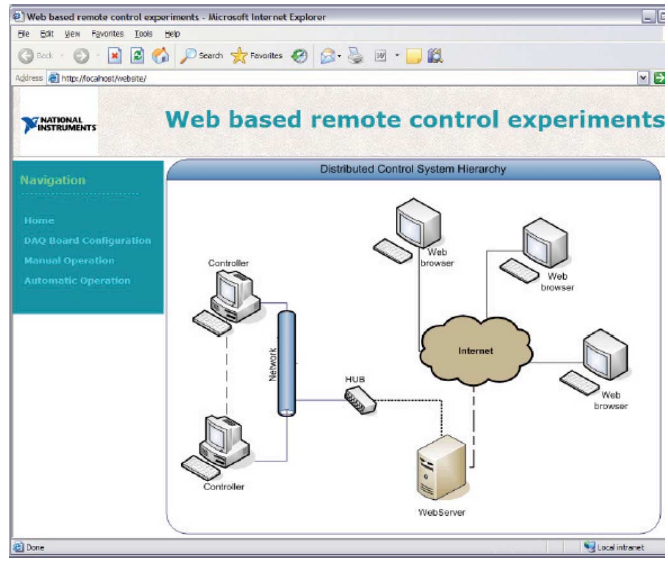

Fig. 9. Introductory page.

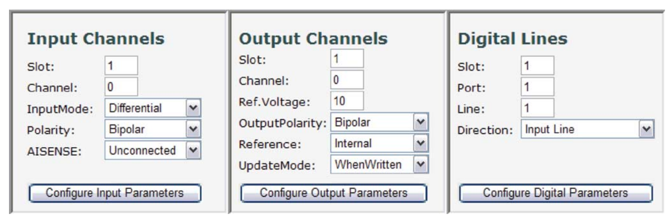

Fig. 10. DAQ board configuration.

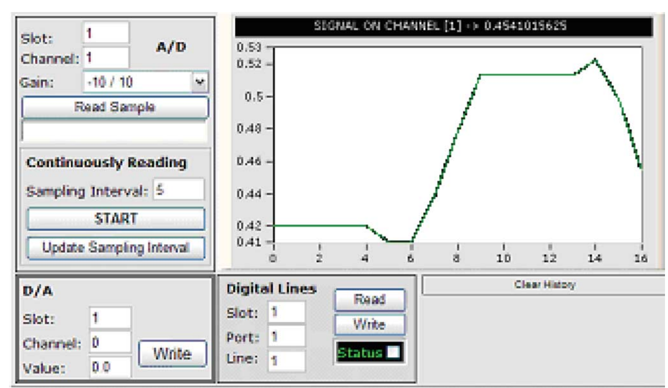

Fig. 11. Manual mode.

The first step is to configure the DAQ board according to the connections to the process using the page from Fig. 10. On the DAQ board configuration page, the DAQ channels have to be configured according to the process connections, namely: channel number, working mode (differential and single ended), polarity, etc.

After the I/O device is configured, two operating modes can be chosen, namely manual or automatic. In the manual-operation page, which is shown in Fig. 11, the user has the possibility to transmit control actions to the process directly by means of an output channel and to read the signal from a specified input channel.

The acquisition can be performed continuously at a specified sampling rate or by taking an instant sample at user demand.

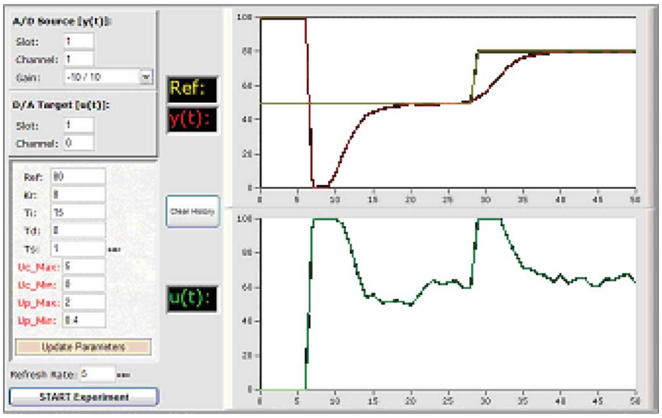

Fig. 12. Automatic mode.

The dynamic of the acquired signal is displayed graphically. The updates of the signal values are made at every specified sampling interval. Using this mode, an open-loop control can be performed.

In the automatic-operation mode, with the page given in Fig. 12, the process is controlled by means of a discrete PID controller derived from (1), with the following parameters:

$$\begin{aligned} a_1 &= 1 \\ a_i &= 0, \qquad \text{for } i \geq 2 \\ q_0 &= K_\mathrm{R}\left(1 + \frac{T}{T_\mathrm{i}} + \frac{T_\mathrm{d}}{T}\right) \\ q_1 &= -2K_\mathrm{R}\left(1 + \frac{T}{T_\mathrm{i}}\right) \\ q_2 &= K_\mathrm{R}\frac{T_\mathrm{d}}{T} \\ q_j &= 0, \qquad \text{for } j \geq 3 \end{aligned} \tag{5}$$

where T is sampling period, K_R is proportional gain, T_i is the integral time, and T_d is the derivative time.

The user must specify the A/D channel for reading the controlled output, the D/A channel for transmitting the control actions, the minimum and maximum voltage values corresponding to 0% and 100%, and the PID tuning parameters.

The dynamic of reference "Ref," output $y(t)$, and the control action $u(t)$ are displayed graphically. After setting the working parameters, the experiment is started by pressing the "START Experiment" button.

During the experiment, the user has the possibility to clear the graphics by pressing the "Clear History" button. The DAQ board configuration, together with the manual and automatic operating modes, allows the usage of the control system for all pilot plants from the laboratory. The result given in Fig. 12 was obtained by using the level-flow pilot plant; the measured signal being the level, and the command being the voltage transmitted to a servo valve.

C. Project Work

In the framework of the project, the students learn to build an NCS using the HMI/SCADA facilities for controlling various setups in the laboratory. First, the students are introduced to the concepts of the networked control architecture, the philosophy of HMI/SCADA applications, and the hardware equipment utilized in distributed control (communication and I/O distributed modules). Then, they start to create an HMI/SCADA application using the LOOKOUT software package.

The key concept used in LOOKOUT, which must be understood by the students, is to create the entire application just by using objects and by interconnecting properties of the object, without the need to write a program code. Among the advantages are that the students can concentrate on solving control problems and that the development and debugging times are reduced.

An application used to illustrate how to develop an NCS employing the principles of the SCADA environment incorporates the following requirements:

1) the automatic control of the plant;
2) the interconnection of the NCS components;
3) the graphic display of the process variables.

In order to accomplish all the requirements, the LOOKOUT includes several options for graphical displays, such as the buttons used for object connection and the panels for application display in secondary windows, where the links between created objects, such as text comments, pictures, and so on, are made. The graphic display, with the assistance of HyperTrend, expressions, and signal-value objects, improves the overall understanding and follows up on the dynamic of the plant.

The first requirement from the proposed task is, at the same time, the most simple control application that can be created by using a PID object and a set of buttons for setting the PID parameters or for controlling the PID behavior — whether manually or automatically. To connect the objects, one simply specifies that the value of a property is equal to the property of the linked object. It is also possible to create a complex expression which can change its values at runtime. Screenshots of possible panels for PID controllers are shown in Figs. 4 and 8, and how to display control-systems signals are shown in Figs. 5 and 8.

The students are also encouraged to find solutions to more complex problems, such as implementing structures for autotuning procedures or implementing complex control strategies (e.g., cascade, ratio, or feedforward control structures).

IV. Conclusion

In this paper, an NCS laboratory for remote-control-engineering teaching is presented. The laboratory allows the students to develop network-based control systems using an architecture based on I/O devices, CMs, and server–client applications implemented with SCADA environment facilities, and to operate on real processes through the intranet and the Internet.

Three examples were presented, demonstrating the potentiality of the NCS laboratory to remotely control two pilot plants and to develop new network-based control-system structures.

Work is in progress to upgrade the laboratory with new CMs, having included remote controllers which allow the implementation of the hierarchical configuration of an NCS using the LOOKOUT environment.

REFERENCES

[1] G. Pratl, D. Dietrich, G. P. Hancke, and W. T. Penzhorn, "A new model for autonomous, networked control systems," *IEEE Trans. Ind. Informat.*, vol. 3, no. 1, pp. 21–32, Feb. 2007.

[2] S. Eberle, "Adaptive Internet integration of field bus systems," *IEEE Trans. Ind. Informat.*, vol. 3, no. 1, pp. 12–20, Feb. 2007.

[3] F. De Pellegrini, D. Miorandi, S. Vitturi, and A. Zanella, "On the use of wireless networks at low level of factory automation systems," *IEEE Trans. Ind. Informat.*, vol. 2, no. 3, pp. 129–143, May 2006.

[4] T. C. Yang, "Networked control system: A brief survey," *Proc. Inst. Electr. Eng.—Control Theory App.*, vol. 153, no. 4, pp. 403–412, Jul. 2006.

[5] Y. Tipsuwan and M.-Y. Chow, "Control methodologies in networked control systems," *Control Eng. Pract.*, vol. 11, no. 10, pp. 1099–1111, Oct. 2003.

[6] F. L. Lian, J. R. Moyne, and D. M. Tilbury, "Network design consideration for distributed control systems," *IEEE Trans. Control Syst. Technol.*, vol. 10, no. 2, pp. 297–307, Mar. 2002.

[7] J. Garcia, F. R. Palomo, A. Luque, C. Aracil, J. M. Quero, D. Carrion, F. Gamiz, P. Revilla, J. Perez-Tinao, M. Moreno, P. Robles, and L. G. Franquelo, "Reconfigurable distributed network control systems for industrial plant automation," *IEEE Trans. Ind. Electron.*, vol. 51, no. 6, pp. 1168–1180, Dec. 2004.

[8] P. Marti, J. Yepez, M. Velasco, R. Villa, and J. M. Fuertes, "Managing quality-of-control in network-based systems by controller and message scheduling co-design," *IEEE Trans. Ind. Electron.*, vol. 51, no. 6, pp. 1159–1167, Dec. 2004.

[9] Y. Tipsuwan and M.-Y. Chow, "Gain scheduler middleware: A methodology to enable existing controllers for networked control and teleoperation—Part I: Networked control," *IEEE Trans. Ind. Electron.*, vol. 51, no. 6, pp. 1218–1227, Dec. 2004.

[10] Y. Tipsuwan and M.-Y. Chow, "Gain scheduler middleware: A methodology to enable existing controllers for networked control and teleoperation—Part II: Teleoperation," *IEEE Trans. Ind. Electron.*, vol. 51, no. 6, pp. 1228–1237, Dec. 2004.

[11] M. Long, C.-H. Wu, and J. Y. Hung, "Denial of service attacks on networked-based control systems: Impact and mitigation," *IEEE Trans. Ind. Informat.*, vol. 1, no. 2, pp. 85–96, May 2005.

[12] L. G. Bushnell, "Special section on networks and control," *IEEE Control Syst. Mag.*, vol. 21, no. 1, pp. 22–23, Feb. 2001.

[13] J. Baillieul and P. J. Antsaklis, "Special issue on networked control systems," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1421–1423, Sep. 2004.

[14] J. Baillieul and P. J. Antsaklis, "Control and communication challenges in networked real-time systems," *Proc. IEEE*, vol. 95, no. 1, pp. 9–28, Jan. 2007.

[15] A. Valera, J. L. Diez, M. Valles, and P. Albertos, "Virtual and remote control laboratory development," *IEEE Control Syst. Mag.*, vol. 25, no. 2, pp. 35–39, Feb. 2005.

[16] M. Casini, D. Prattichizzo, and A. Vicino, "The automatic control Telelab—A web-based technology for distance learning," *IEEE Control Syst. Mag.*, vol. 24, no. 3, pp. 36–44, Jun. 2004.

[17] M. Casini, D. Prattichizzo, and A. Vicino, "Operating remote laboratories through a bootable device," *IEEE Trans. Ind. Electron.*, vol. 54, no. 6, pp. 3134–3140, Dec. 2007.

[18] H. Hassan, C. Domínguez, J. M. Martínez, A. Perles, and J. Albaladejo, "Remote laboratory architecture for the validation of industrial control applications," *IEEE Trans. Ind. Electron.*, vol. 54, no. 6, pp. 3094–3102, Dec. 2007.

[19] D. Hercog, B. Gergiè, S. Uran, and K. Jezernik, "A DSP-based remote control laboratory," *IEEE Trans. Ind. Electron.*, vol. 54, no. 6, pp. 3057–3068, Dec. 2007.

[20] S. Carari, O. Chirea, and C. Lazar, "Distributed control system for educational training based on lookout and FieldPoint," in *Proc. NIWeek Worldwide Virtual Instrum. Conf.*, Austin, TX, 2003. CD-ROM.

[21] S. Carari, C. Lazar, and O. Chirea, "Remote control and monitoring of heating systems," in *Proc. 2nd Int. Symp. Remote Eng. Virtual Instrum.*, Brasov, Romania, 2004. CD-ROM.

[22] C. Andrici, S. Carari, C. Halauca, C. Lazar, and G. Baluta, "A remote-based real-time control laboratory application," in *Proc. 3rd Int. Symp. Remote Eng. Virtual Instrum.*, Maribor, Slovenia, 2006. CD-ROM.

[23] C. Lazar, S. Carari, and C. Halauca, "Networked control systems laboratory development," in *Proc. 16th Int. Conf. Control Syst. Comput. Sci.*, Bucharest, Romania, May 22–25, 2007, pp. 653–659.

[24] G. Baluta and C. Lazar, "Remote control laboratory development for electrical drive systems," in *Proc. 13th IEEE IFAC Int. Conf. Methods Models Autom. Robot.*, Szczecin, Poland, Aug. 27–30, 2007. CD-ROM.

[25] *Handbook of PI and PID Controller Tuning Rules*, Imperial College Press, London, U.K., 2003.

[26] K. H. Ang, G. Chong, and Y. Li, "PID control system analysis, design, and technology," *IEEE Trans. Control Syst. Technol.*, vol. 13, no. 4, pp. 559–576, Jul. 2005.

[27] K. J. Astrom and T. Hagglund, *PID Controllers: Theory, Design and Tuning*. Pittsburgh, PA: Instrum. Soc. Amer., 1995.

Corneliu Lazar (M'06) was born in Iaşi, Romania, on October 2, 1952. He received the M.Sc. degree in electrical engineering and the Ph.D. degree in automatic control from "Gh. Asachi" Technical University of Iaşi, Iaşi, Romania, in 1976 and 1991, respectively.

He was with the Havy Plant Company, Iaşi, from 1976 to 1981. He was an Assistant Professor in 1981, with the Department of Automatic Control and Applied Informatics, "Gh. Asachi" Technical University of Iaşi, where he was an Associate Professor in 1992 and where he has been a Full Professor in control engineering since 1997. He has authored or coauthored more than 150 conference and journal papers. His main research interests are in the fields of process control, visual servoing, and predictive control.

Dr. Lazar received the 2003 Academic Category Finalist Paper Award from National Instruments.

Sorin Carari was born in Suceava, Romania, on July 1, 1977. He received the M.Sc. degree in automatic control from "Gh. Asachi" Technical University of Iaşi, Iaşi, Romania, in 2002.

Since 2002, he has been an Assistant Professor with the Department of Automatic Control and Applied Informatics, "Gh. Asachi" Technical University of Iaşi. His research interests include networked control systems and PID controllers.

Mr. Carari received the 2003 Academic Category Finalist Paper Award from National Instruments.