

# Teaching Industrial Control with the Assistance of a Web-based Interactive Simulator of Distributed Control Systems\*

GEORGE HASSAPIS

Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki,  
540 06 Thessaloniki, Greece. E-mail: chasapis@eng.auth.gr

*This paper addresses the issue of teaching the implementation of industrial control by using an interactive simulator of a typical distributed computer control system (DCS). This simulator is able to run in a web environment, which imposes an instructional model and provides access to hypertext. According to this instructional model, teaching of the industrial control implementation is done through the simulation of an application, which is wrapped-up with the relevant theory. The simulator is able to simulate the execution of the software that realizes the regulatory control algorithms, the startup and emergency control functions of an industrial process as well as the functions of the operator's station.*

## INTRODUCTION

DISTRIBUTED CONTROL SYSTEMS (DCS) [2] are computing facilities used widely in industry. They are local networks of microcomputer-based controllers, usually called local control units, and operator's stations with configurable hardware architectures and software modules. In addition to the regulatory control functions, which are based on feedback, feedforward, multivariable and other strategies, they can realize functions, which are related with practical plant operating issues. Such issues are the plant startup, the safety interlocks, the alarm annunciation strategies, the human operator intervention, the suppression of noise, the operation within the range limits of sensors and actuators and the capacity limits of interconnected equipment. Implementing such functions on a DCS requires the configuration of the hardware architecture of the DCS, the programming of the control functions and the testing of the DCS operation.

Various educators [1] believe and studies [3] confirm that student exposure to the practical issues of a complete control system, such as the implementation of the above functions by a DCS system, is a great stimulus for developing engineering intuition and bridge the gap between theory and practice. Therefore, in order to bring control systems education a step closer to real life and make the learning more effective, traditional courses on control systems could be enhanced with laboratory work that deals with the implementation of all the functions of an industrial control system. This new laboratory work must

include configuration, programming, testing and operation of complete industrial type facilities. However, setting up such experimental facilities is a very expensive task for an educational institution. The next best thing to real life facilities is the use of software simulators which are able to simulate all or some of the above mentioned functions along with their dynamic interaction with the physical plant. The use of simulators of other control system technology for training students has been reported at least in other two cases [11, 12].

In this work the development of a simulator is reported which can be used to prepare experiments on DCS-based industrial control implementations. In order to make learning more effective the simulator was made to run in a web environment that imposes an instructional model and allows access to hypertext. The instructional model supports an application-oriented teaching approach, according to which the student is required to study and implement the DCS-based control system of an industrial case. Structuring hypertext appropriately, knowledge and information at a depth and breadth that the needs of each individual student require can be retrieved. The view that learning becomes more effective if one uses instructional models that are based on the use of the web is supported by the results of studies on teaching VLSI design [7], DSP [8–10], adaptive neural networks [3] and other more general studies [4–6] on the use of the web for educational purposes.

The simulator was built with the purpose of setting up experiments on the control of continuous time industrial processes expressed in the form of a matrix of transfer functions or convolution coefficients. The experiments can include the implementation of plant startup functions, normal

\* Accepted 21 March 2002.

automatic plant operation, emergency control, plant monitoring and manual control. The considered application for teaching DCS-based industrial control is the interactive advanced control of a distillation column. This process is used extensively in the petroleum and petrochemicals industry. However, the simulator can be used to develop similar experiments for the continuous time processes that one can find in other industrial sectors. The pulp and paper industry, the power generation, the fertilizer production and the chemical industry are some representative sectors of continuous time processing.

### THE INSTRUCTIONAL MODEL

The instructional model is an extended version of Hanger's approach [10], according to which a simulation problem is defined and wrapped with the appropriate theory. The considered extension to this model is to increase the level of interaction of the student with the simulator functions and to provide access to knowledge at a depth that the individual needs of the student require.

The working steps that the student can take are shown in Fig. 1.

According to this figure, initially the student selects the simulation exercise and is exposed to a web page where the problem description is given and the strategy for handling it is explained. This strategy describes the issues explained in the introduction, such as startup and emergency procedures, automatic regulation of the controlled variables, noise filtering, alarm annunciation and operating constraints. Hyperlinks on keywords in the text allow the student to transfer control to a tutorial-like material where explanations and knowledge on a selected topic are provided. This explanation or knowledge may not be available in the local database but it is accessed through the appropriate links to other servers on the Internet.

When the student feels that he has understood the control problem and the recommended strategy for handling the problem, he may proceed with the implementation of the strategy by entering the simulation environment. This environment allows the student to define the architecture of the DCS and its connection with the process sensors and actuators. This work is usually called the DCS configuration. Also, the environment provides the necessary tools for programming all the functions of the strategy and the dynamic simulation of the process operation under the control of the DCS. For the dynamic simulation of the process, the student is required to describe the process dynamics, a step that is not part of any implementation procedure of a real-life system and is the only deviation from the work that is required to implement a real-life system.

During the simulated operation of the process under the control of the DCS all the typical displays and controls for carrying out the follow-up,

automatic and manual control actions that a professional operator does are provided to the student. Again, through references to hyperlinks the student is advised on how to run the plant during startup, emergency situations and normal operating conditions. His responses are recorded in the form of historic data of the controlled variables and analyzed off-line by the instructor.

### DEMONSTRATION EXPERIMENT

The functions of the simulation environment are demonstrated through the presentation of a developed experiment. This experiment concerns the implementation of an Interactive Advanced Control (IAC) strategy of a distillation column by a DCS. To carry out this experiment, a student is expected to spend two sessions of two hours duration each. Of course, the actual time might be longer if the student's background on control systems theory and computer basics is not the assumed one for this kind of exercise.

Initially, the student is exposed to a number of pages of text and figures in a window of the environment. An extract of this material, as seen on the screen, is given in Fig. 2.

According to the description given in these pages, the column can be operated in three different modes. Each mode is characterized by its control strategy. The first mode of operation concerns the column startup. In this mode, the operator through the controls of the operator's station drives the controlled variables of the column to their nominal operating values. This is achieved by manually adjusting the values of the manipulated variables. The differential equations that describe the way each controlled variable is influenced by changes in the manipulated variables are predominantly nonlinear. If the operator tries to drive the controlled variables to the operating point values by making large changes on the manipulated variables, he might drive them to an unstable state. Instead, the usual procedure is to change the manipulated variables in small increments, waiting for the controlled variables to reach steady-state values and repeat this action until the values of the operating point are reached. This operation is derived from the fact that the entire operating range of a controlled variable can be split in a number of subranges within which the changes of the variable in terms of changes in the manipulated variables can be approximated by a linear model. This model can be learned and followed by an operator much easier.

In the second mode of operation, deviations of the controlled variables from their nominal values, caused by disturbances on the crude feed rate, are compensated automatically by the IAC algorithm. If the algorithm fails to bring the variables back to their nominal values then the human operator makes adjustments on the manipulated variables. In the third mode of operation, the strategy

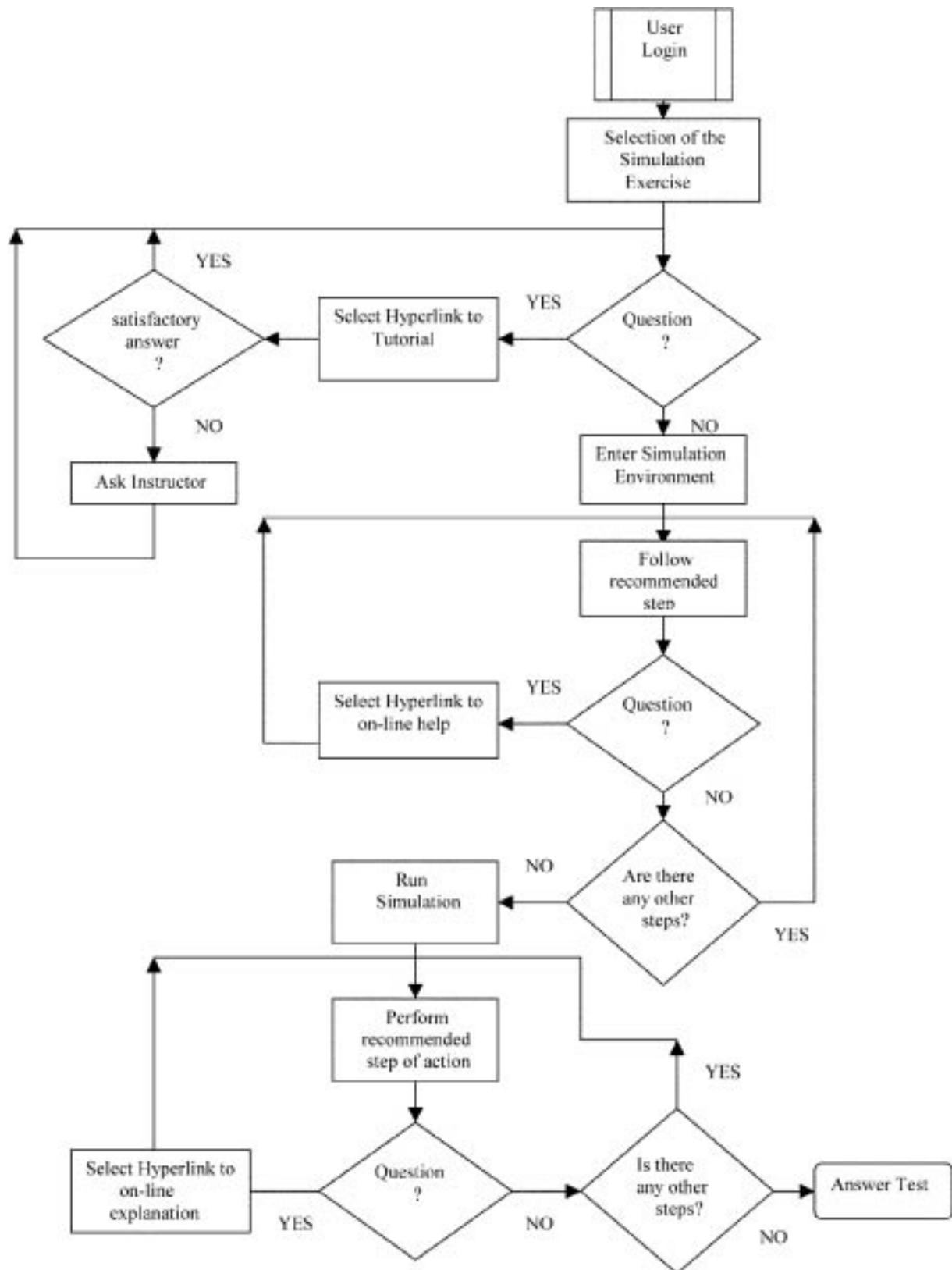


Fig. 1. The model procedural steps.

involves the continuous monitoring of key variables and the driving of them to safe states when limits are exceeded.

Next, the environment prompts the student to proceed with the implementation of these strate-

gies on a DCS. This task requires the following steps to be followed:

- Configuration of the hardware architecture of the distributed control system. This work

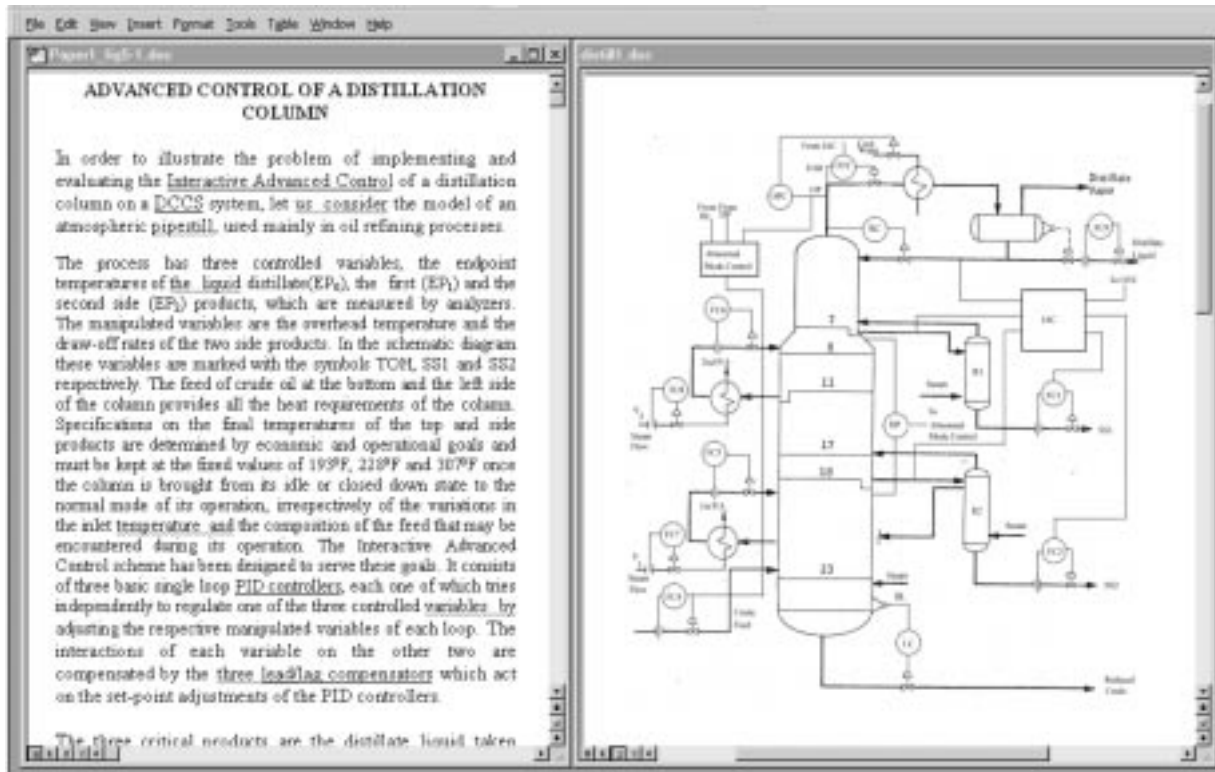


Fig. 2. Environment window in which the distillation control strategy is explained.

involves the finding of the number of the local control units that will carry out most of the control functions and the number of the operator's stations that will monitor the plant operation and execute the human operator commands.

- Assignment and connection of the appropriate sensors and actuators of the measured and manipulated variables to inputs and outputs of the local control units.
- Assignment of the control loops and monitored variables to the appropriate display group in the operator's stations.
- Programming the control and monitoring functions to the respective local control units and operator's stations.
- Testing the plant operation under the control of the programmed DCS.

For each one of these implementation steps the student is guided by the environment and enters the simulator, which provides to him similar software tools with those of a typical DCS. For space reasons, the following text shows how the above steps have been applied for implementing the part of the strategy that deals with the control of the endpoint temperatures of the distillate and the first side streams, during the normal mode of operation.

According to this strategy:

- the endpoint temperature of the distillate stream ( $EP_0$ ) is adjusted by PID closed loop control of the overhead temperature (TOH);
- the endpoint temperature of the first side stream

( $EP_1$ ) is adjusted by PID closed loop control of the side stream flow rate (SS1);

- the interactions of the distillate stream on the first side stream are compensated by the use of a lead/lag compensator;
- the measurement of both endpoint temperatures is extracted from the value of the received electrical signal which is filtered first by an exponential filter;
- the endpoint temperature of the first side stream should not exceed the upper limit of 200°C and the lower limit of 80°C during the normal mode of operation; otherwise alarm annunciation must be triggered;
- similarly, the endpoint temperature of the distillate stream should not exceed the upper limit of 193°C and the lower limit of 40°C; otherwise an alarm must be raised.

A display of the first window that appears on the screen for defining the DCS architecture is given in Fig. 3.

In this window the student is required to define the number of the local control units that the distributed system should consist of, and the number and the type of the inputs and the outputs of each local control unit. According to the schematic diagram shown in the window of Fig. 2, the DCS must have an overall number of 6 analogue inputs, 8 discrete inputs, 4 analogue outputs and 4 discrete outputs. Depending on the size of the selected local control unit of a certain manufacturer, these inputs and outputs may be more than a single local control unit can accommodate.

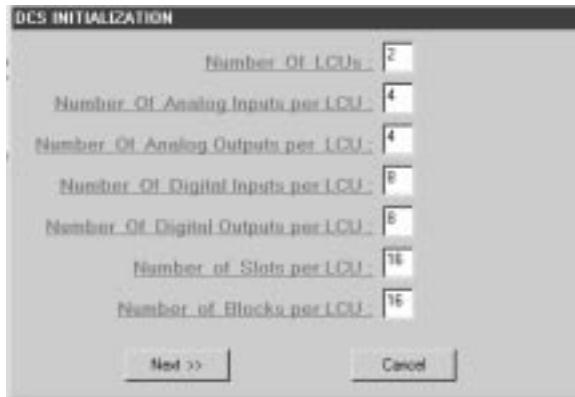


Fig. 3. An active window for the description of the distributed control system architecture.

Therefore, the student is required to find the number of the local control units that will match the overall number of inputs and outputs. Considering that each local control unit of a selected manufacturer has 4 analogue inputs, 4 analogue outputs, 8 discrete inputs and 8 discrete outputs, the respective form in Fig. 3 has been filled in with the appropriate numbers. The configuration of the architecture ends with the identification of the variables that will be monitored and the groups they will be assigned to in the operator's stations. A similar procedure is followed for the configuration of the operator's stations. The next step that the student is instructed to take is the programming of the control strategy. To proceed with this action he is exposed to the graphical environment shown in Fig. 4.

In this environment the student may draw a graph where:

- each input filtering and conversion function is represented by a circle;
- each PID, compensator and algebraic operation by a rectangle;
- the conversion of each output value to the value of the electrical signal by an inverse triangle.

In the graph shown in Fig. 4 the graphical symbols A/I1 and A/I2 program the sampling of the  $EP_0$  and  $EP_1$  temperatures. The names of the input conversion and filtering functions are embedded in these graphical symbols and their parameters are defined by calling successive fill-in forms. Slot1 and slot3 symbols program the PID control loop of the overhead temperature TOH, and the lead/lag compensator of the interactions of TOH with the  $EP_1$  variable. Slot2 and slots 4 to 7 program the PID control loop of the flow rate of the first side stream, SS1. These slots select the smallest from the values that the direct or the compensated output of the PID algorithm has. In this way the possibility of driving SS1 to an unsafe operating value is reduced. Symbols A/O1 and A/O2 program the functions of converting TOH and SS1 values to the appropriate output signal values. When the student completes the programming task of the local control unit he proceeds with the operator's station programming. This part involves the assignment of the variables that need to be monitored and/or manually adjusted to different display groups. In the case of the developed simulator, a number of eight bar graph displays of variables of control loop, discrete inputs and outputs can be accommodated on one screen. Placing certain variables in the same group depends mainly on criteria related with the interactions between variables. An operator must be

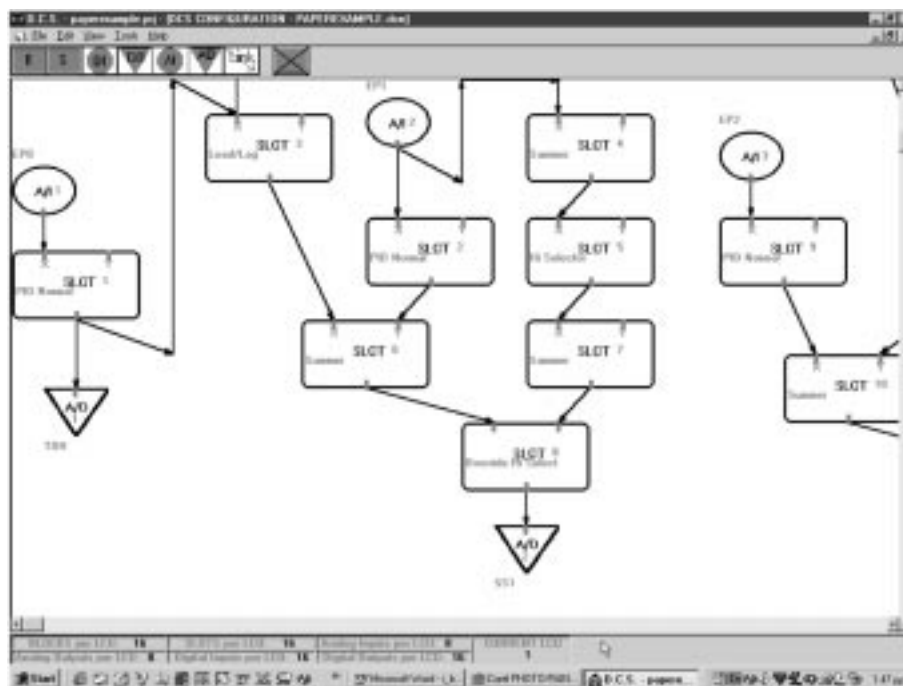


Fig. 4. Graphical environment for programming the distributed control system.



Fig. 5. Environment for programming the operator's station.

able to realize quickly deviations attributed to interactions and act accordingly. The screen provided for configuring an operator's station is shown in Fig. 5.

At this stage the configuration and programming of the strategy of the normal mode of operation is completed and the student is now required to test and evaluate his implementation. He may perform the same functions that an ordinary operator carries out at this mode of operation. Observing the behaviour of the column the student may assess the correctness and timeliness of the DCS configuration and program. He may also change parameters, such as sampling period and see their influence on the system behaviour. By linking to the page of system operation he may follow a step by step procedure which advises him how to simulate the column operation. In order to implement this procedure, the student is required to describe the open loop dynamics of the considered distillation column in a software module built to simulate any process dynamics. A matrix of transfer functions or convolution coefficients is required to express the dynamics of the process. Also, the student is required to connect the process with the DCS simulator and run the combined simulation. The basic operator's functions, which the student is expected to perform are:

- set the control loops in the automatic operation, monitor the response of the controlled variables of the column by the use of the bar graphs of the operator's station,
- adjust manually the manipulated variables in order to drive the controlled variables close to their set points, when alarms are raised.

The typical operator's bar graph of the first group display is depicted in Fig. 6. The bar graphs present the dynamic changes of two of the controlled variables along with the manipulated and set-point variables, at an instance during the normal mode of operation. In this figure, the controlled variable (PV), the set point (SP) and the manipulated variable (MV) of the control loop of the distillate endpoint temperature are presented by the bar graph with the tag LCU1 S1. The same quantities of the control loop of the endpoint temperature of the first side stream are presented by the bar graph with the tag LCU1 S2. At the bottom of each bar graph the selection buttons for reverting from the manual to automatic modes of loop operation and vice versa are shown.

By means of this display the student makes all the necessary adjustments whereas the response curves of the variables over the time may be given and seen on a side window on the screen.

In a similar way, startup and emergency modes of distillation operation can be studied. The reader may execute the presented experiment by accessing the site in reference [14].

## CONCLUSIONS

This work has addressed the issue of teaching the implementation of industrial control functions by the use of an interactive simulator of distributed control systems. The simulator runs in a web environment, which integrates hypertext with the simulator and imposes an instructional model. The simulator can simulate the dynamic operation of a process under the control of a distributed control system, which is configured and programmed by

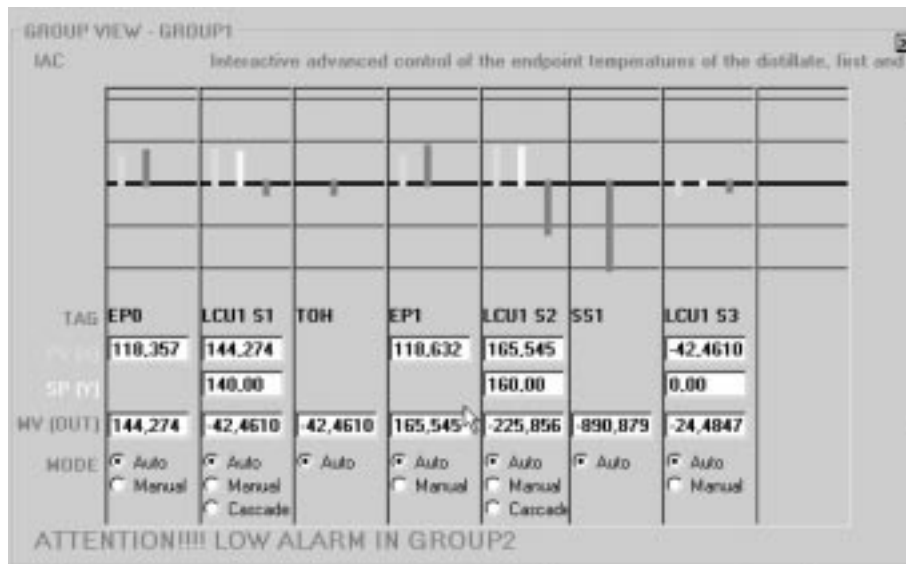


Fig. 6. Operator's display.

the student to perform the startup, normal and emergency control of the process. The instructional model supports an application-oriented teaching approach according to which the student carries out a simulation experiment during the execution of which he can switch to a tutorial-like

presentation of the underlying theory. This tutorial can be accessed at a depth and breadth that the individual needs of the student require. The first developed experiment concerns the Interactive Advance Control of a distillation column.

## REFERENCES

1. P. Antsaklis, T. Basar, R. DeCarlo, N. H. McClamroch, M. Spong, and S. Yurkovich, Report on the NSF/CSS Workshop on New Directions in Control Engineering Education, *IEEE Contr. Syst. Mag.*, 19, 1999, pp. 53–58.
2. D. Popovic, V. P. Bhatkas, *Distributed Computer Control for Industrial Automation*, Marcel Dekker, New York and Basel (1990).
3. J. C. Principe, N. R. Euliano, and W. C. Lefebvre, Innovating Adaptive and Neural Systems Instruction with Interactive Electronic Books, *IEEE Proceedings*, 88, 2000, pp. 81–95.
4. E. Bilotta, M. Fiorito, D. Iovane, and P. Pantano, An educational environment using WWW, *Comput. Networks ISDN Syst.*, 27, 1995, pp. 905–909.
5. J. Self, From constructionism to deconstructionism: anticipating trends in educational styles, *Europ. J. Eng. Educ.*, 22, 1997, pp. 295–307.
6. R. Kemm, D. Weaver, A. Dodds, G. Evans, D. Gartland, T. Petrovic, L. Delbridge, and P. Harris, Designing evaluating an interactive hypothesis testing tool to aid student understanding—gastric acid secretion and its regulation, in *ASCILITE'97 What Works and Why*, R. Kevill, R. Oliver, and R. Phillips, eds., (1997) pp. 324–330.
7. J. Allen, and C.J. Terman, An interactive learning environment for VLSI design, *IEEE Proceedings*, 88, 2000, pp. 96–106.
8. J. McClellan, R. Schafer, J. Schoforf, and M. Yoder, Multimedia and World Wide Web resources for teaching DSP, *Proc. IEEE Int. Conf. Acoustics Speech and Signal Processing*, (1996) pp. 1101–1104.
9. R. Harger, Introducing DSP with electronic book in a computer classroom, *IEEE Trans. Educ.*, 39, 1996, pp. 173–179.
10. R. Harger, Teaching in a computer classroom with a hyperlinked, interactive book, *IEEE Trans. Educ.*, 39, 1996, pp. 327–335.
11. D. Cooper and D. Dougherty, Control Station: An Interactive Simulator for Process Control Education, *Int. J. Eng. Educ.*, 17, 2001, pp. 276–287.
12. D. Shin, E. S. Yoon, S. J. Park, E. S. Lee, Web-based interactive virtual laboratory system for unit operations and process systems engineering education, *Computers & Chemical Eng.*, 24, 2000, pp. 1381–1385.
13. G. J. C. Copinga, M. H. G. Verhaegen, and M. J. J. M. van de Ven, Toward a Web-Based Study Support Environment for Teaching Automatic Control, *IEEE Control Sys. Mag.*, 20, 2000, pp. 8–19.
14. G. Hassapis, Industrial Informatics, <http://indinf.ee.auth.gr/myweb>

**George Hassapis** received his diploma in Electrical Engineering from the University of Patras, Greece and his M.Sc. and Ph.D. degrees from the Control Systems Centre of the University of Manchester, Institute of Science and Technology, U.K. in 1973 and 1978 respectively. After completing his military service he joined the Exxon corporation in 1980 where he was involved in the design and development of computer-based instrumentation and process control systems. From October 1985 he is employed by the Aristotle University of Thessaloniki, Dpt. of Electrical and Computer Engineering where he holds the position of the Associate Professor. His current research interests are focused on real-time computer systems, advanced computer architectures, hybrid systems and e-learning systems. He is a member of IEEE, the Technical Chamber of Greece and a chartered engineer of the Engineering Council of U.K. He is the author of more than 40 publications in scientific journals and conference proceedings.