A LABORATORY TO SUPPLEMENT COURSES IN PROCESS CONTROL*

BABU JOSEPH, CHAO-MING YING, DEEPAK SRINIVASAGUPTA
Washington University • St. Louis, MO 63130

Laboratory exercises reinforce theoretical concepts taught in classrooms. Such exercises also offer an opportunity to put theory into practice, to realize practical limitations in real-life situations. Typical students remember 5% of what they hear, 10% of what they read, and 75% of what they do.[1] Open-ended laboratory exercises allow room for creativity, tailor the course to the needs of the student, and provide a sense of accomplishment for the student.

The Process Control Laboratory (PCL) at Washington University was established in 1980 and serves as the primary teaching laboratory for undergraduate and graduate courses in process control. The laboratory comprises eight bench-scale experiments representative of control problems encountered in industrial practice. The experiments were designed to give students practical experience in chemical process control such as exposure to hardware and software, control-system design, implementation, and testing. The computer hardware and software have changed substantially every five years, but the experimental setups have not. The new hardware and software are immensely more powerful and enable students to focus on process-control issues as opposed to program development or debugging. Minimal training in software is now needed. The net result is that the students now focus on learning control concepts rather than debugging program code.

The current generation of hardware uses Pentium-III computers with Windows 98 as the operating system. These computers are interfaced to the experimental setups using data acquisition boards. Wherever necessary, the drivers supplied with the board have been rewritten as Dynamically Linked Libraries (DLLs) accessible from MATLAB. This permits the user to read from and write to the board directly from the MATLAB/SIMULINK interface. We have taken advantage of the high-speed computers to run the experiments directly from the SIMULINK interface. This removed a major block in transitioning from the simulated experiments used in the early part of the course to the real experiments used in the second half of the course.

There are a number of reasons for using simulated processes in the first half of the course. They are easy to duplicate. This makes it easy to disseminate copies to every student for use with their own computers so they don't have to come to the laboratory to do the experiments. The simulations are realistic and hence the students are exposed to control principles in a setting that is difficult to provide in a laboratory. For example, we use the simulation of a naphtha cracker to introduce basic control concepts. Simulations are reproducible, so it is possible for the instructor to check if the control implementation worked properly. Simulations can be run as fast as the computer speed allows; this allows the student to repeat a run, if necessary, to correct any mistakes.

COURSE CONTENT

The outline of the 3-credit-hour process control course is shown in Table 1. At the beginning of the course the students are introduced to concepts of block diagram representation, transfer functions, response characteristics of first-, second-, and higher-order systems, and feedback control using PID control. In the first two laboratory sessions, the students are introduced to MATLAB/SIMULINK software, and during

Babu Joseph *is the Edward C. Dicke Professor of Chemical Engineering at Washington University. He received his BTech (1971) from the Indian Institute of Technology, Kanpur (1971), and his MS and PhD from Case Western Reserve University (in 1974 and 1975, respectively). His research interests are in the area of process engineering with emphasis on process control. He can be reached at <joseph@che.wustl.edu>.*
Chao-Ming Ying *graduated with a DSc in Chemical Engineering from Washington University in May of 1999. He received his BS (Honors) degree in 1993 from the Zhejiang University, Hangzhou, and his MS degree from Washington University in 1998, all in chemical engineering. He currently works at i2 Technologies in Cambridge.*
Deepak Srinivasagupta *graduated from the Indian Institute of Technology, Kharagpur, with BTech (Honors) in chemical engineering in May 1998. He is currently a doctoral student under Professor Joseph at Washington University. His research interests include polymer composites manufacturing, process control, fault detection, and process identification.*

* *Based on a presentation given at the ASEE meeting in St. Louis, June 2000.*

the next few weeks, process identification, feedback control, PID control, and controller tuning concepts are covered in the lectures. Simultaneously in the laboratory the students use the simulation of a naphtha cracker to implement and test concepts of identification, feedback control loop setup, and controller tuning (see Figure 1). The naphtha cracker simulation also introduces them to the economic benefits of good process regulation since the reactor temperature is directly tied to the yield of the ethylene and propylene production rate.

The multiloop control concepts of feedforward, cascade, and override (selective) control are introduced next. The students are then asked to implement these on the naphtha cracker simulation and to evaluate the effectiveness of such multiloop control strategies. Again, the economic significance of improved control is made clear by providing a measure of the cumulative profitability of the cracker operation. The concepts of decentralized control and interaction in multivariable processes are also introduced during this time.

In the remaining part of the course, students are introduced to the experimental laboratory and are asked to choose a project on one of the experimental setups. The course project is usually an open-ended process control problem. Experiments are designed to permit implementation of controllers of varying complexity. The students must identify the control issues, choose control strategies, implement and test those strategies, and if necessary, improve them. Since the experiments usually take longer than the one-hour laboratory formally assigned, we have made the labs accessible 24 hours a day, seven days a week, without any supervision. A project presentation at the end of the semester achieves two important goals: it gives a valuable exercise in communication skills while also enabling other students to learn novel applications of the material learned in class.

DESCRIPTION OF THE EXPERIMENTS

Table 2 (next page) summarizes the eight experiments available in the laboratory; they are shown in Figures 2 through 9. All these experiments introduce the basic concepts of process identification, noise filtering, and controller tuning. In addition, each of the experiments offers the opportunity to design, implement, and test advanced control strategies.

In the pressure- and level-control experiment (Figure 2), the objective is to control the water level and air pressure in a tank by manipulating the water inlet valve and air outlet valve. It has been used to expose the students to ideas regarding controller tuning of integrating processes, multivariable control, and override control. The objective for the temperature and level control in a heated-tank experiment (Figure 3) is to control inlet flow, water level, and water temperature in the tank by manipulating the water valve and heater input. It introduces concepts in control of unstable processes, cascade flow control, and feed-forward control. The objective of the air-temperature control experiment (Figure 4) is to control temperature of the air by manipulating the heater power. It introduces concepts in nonlinear control via gain scheduling and adaptive control.

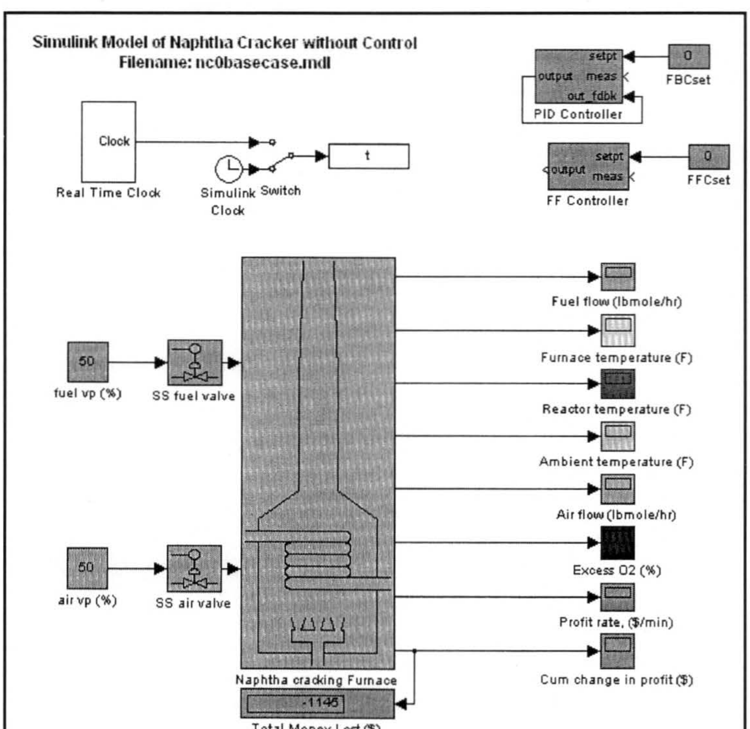

Figure 1. *Simulink model of naphtha cracker.*

TABLE 1
Process Control Course at Washington University

One semester/3 credit hours
Two hours lecture/1 hour lab/ week

Lecture	*Lab*
Introduction (Block Diagrams, Laplace Transforms, Control Hardware)	Introduction to MATLAB/SIMULINK
Process identification, PID controller and controller tuning	Simple feedback control of the naphtha cracker simulator
Multiloop controllers	Implement and Test FFC/Cascade/Override Control on the naphtha cracker simulator
Multivariable control	Tune and test an MV controller on naphtha cracker simulator
Analysis of closed-loop systems	Lab project
Stability	Lab project
Frequency response analysis	Lab project
Control hardware and process safety	Lab project
Project presentations	Project presentations

The objective in the temperature and level control in a mixing-tank experiment (Figure 5) is to control the water level and temperature by manipulating inflow of cold and hot water. The concepts introduced are multivariable control and noninteracting control by decoupling. The control of a batch reactor (Figure 6) educates students in typical batch operation procedures such as startup, operation, and shutdown procedures, in dealing with emergencies, in dealing with process operation faults, and in batch process control logic.

For the control of empty and packed bed tubular reactors experiment (Figure 7), the objective is to control the exit temperature in a packed or empty bed. It allows comparison of cascade control, feed-forward control, and empty vs. packed beds concepts. It turns out that cascade control is worse than direct feedback control for the packed bed system. The motive in the control of heated bar temperature experiment (Figure 8) is to control the temperature in a bar heated at one end. It permits the exploration of cascade control and dead-time compensation.

Finally, the double-pipe heat exchanger experiment (Figure 9) involves a highly interacting multivariable control problem with a wide choice of control objectives and control strategies. It permits exploration of various concepts such a Relative Gain Array (RGA), Model Predictive Control (MPC), and multi-loop control.

For a more complete description of the experiments and for downloading the virtual experiments and software, see the website[2] cited at the end of this article. See also the book by Brosilow and Joseph.[3]

TABLE 2
List of Experiments

Fig.	Experiment	Concepts Introduced
2	Pressure and Level Control	Control of a nonself-regulating process, override control of level in a tank
3	Temperature and Level Control in a Heated Tank	Cascade flow control, level control
4	Air Temperature Control	Control of a nonlinear process via gain scheduling
5	Temperature and Level Control in a Mixing Tank	Multivariable control, interaction, decoupling via steady-state equations
6	Control of a Batch Reactor	Control of discrete event systems, building software safety interlocks
7	Control of Empty and Packed Bed Tubular Reactor	Feedback versus cascade control
8	Control of Heated Bar Temperature	Feedback versus cascade control
9	Double Pipe Heat Exchanger	Multivariable control and interaction, MPC, decentralized versus centralized control

MATLAB/SIMULINK INTERFACE

There are a variety of data acquisition and control software, such as Labview, MATLAB, Program CC, Picles, and LabTech Control. MATLAB (and SIMULINK) has strong support for control education[4] through supporting textbooks and software. SIMULINK provides MATLAB with a graphical user interface (GUI) capability that permits a rich variety of control-system methods to be effectively designed and tested with relative ease. Advanced control algorithms such as MPC and multiloop control can be easily implemented and tested. MATLAB also provides convenient add-on toolbox packages for system identification, digital signal processing, and control systems analysis. A fully functional MATLAB/

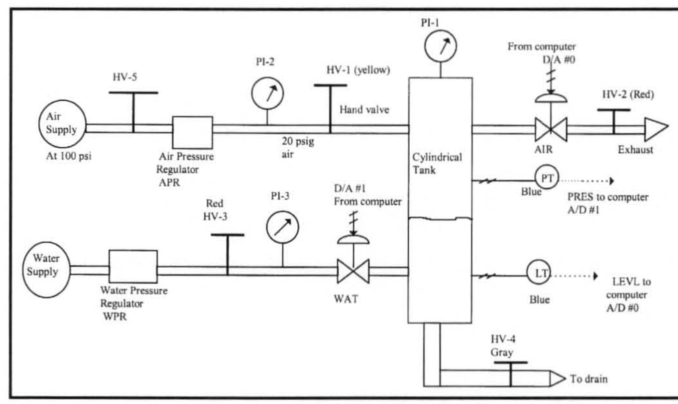

Figure 2. Pressure and level control experiment.

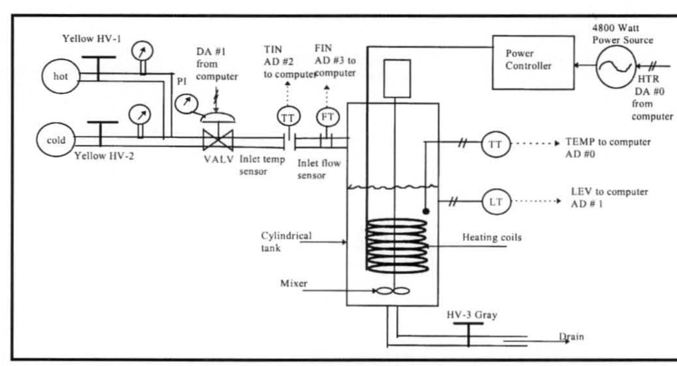

Figure 3. Temperature and level control in a heated-tank experiment.

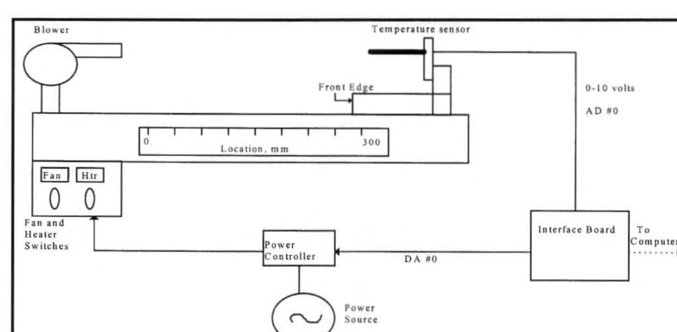

Figure 4. Air-temperature control experiment.

SIMULINK version is available as an inexpensive student edition.

SIMULINK is a block-diagram-based dynamic simulator that provides one-to-one correspondence with control theory. Requiring no programming, the focus is on learning and not on debugging. Changes can be done in real-time and there is instant feedback on success

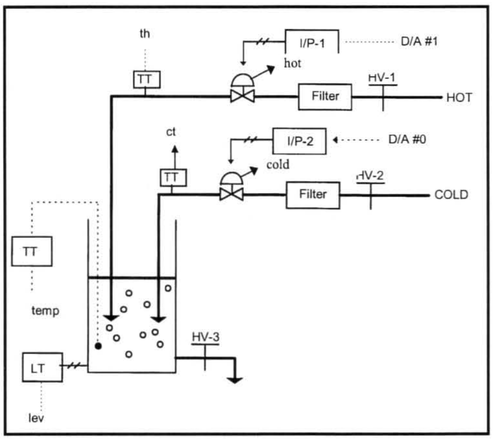

Figure 5. *Temperature and level control in a mixing tank experiment.*

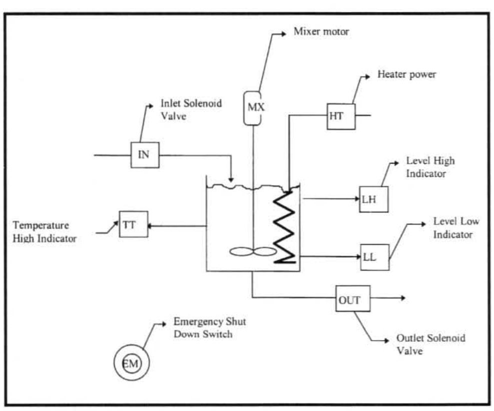

Figure 6. *Control of a batch-reactor experiment.*

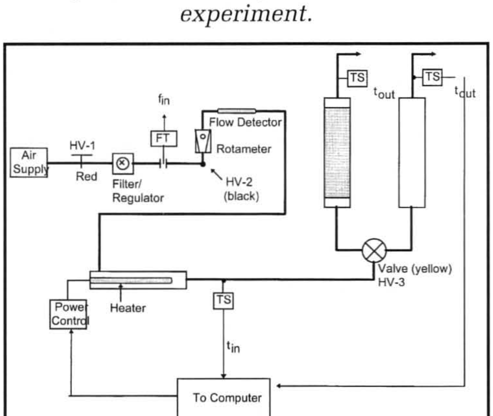

Figure 7. *Control of empty and packed tubular bed reactors experiment.*

or failure. Additional tools that have been developed at Washington University for this platform include a realistic PID block with a self-tuning capability. These additional software modules are shown in Table 3. SIMULINK also supports many of the toolboxes supplied with MATLAB in a block-diagram form.

A SIMULINK subsystem block encapsulates the SIMULINK S-function block that communicates with the MATLAB driver DLL. The subsystem block (Figure 10 shows an example) has the manipulated variables as inputs and outputs the measured variables. This block reads real-time experimental measurements and delivers control action values to the process. We took advantage of the fast computers available and effectively slowed down the execution of SIMULINK to achieve real-time synchronization with the experiments, in accordance with a user-defined sample time (with a default value of 1 second). It is only recently that MATLAB started providing tools for accessing data acquisition hardware and real-time capability. The self-tuning controller can automatically perform identification tests for first-order plus dead time (FOPDT) and integrating processes and determine the PID parameters. Alternatively, the user can manually perform the identification tests, build a process model via the polynomial based POLYID identification software,[5] and use PIDTUNER, a MATLAB graphical user interface program that has been developed here to interactively design the controller.

LABORATORY SAFETY

All students have an access code to work in the laboratory during off hours. Since the laboratory is run without supervision during those times, the experiments are designed to be intrinsically safe: All the experiments use only air and/or water; all tanks have overflow capability to prevent accidental spills;

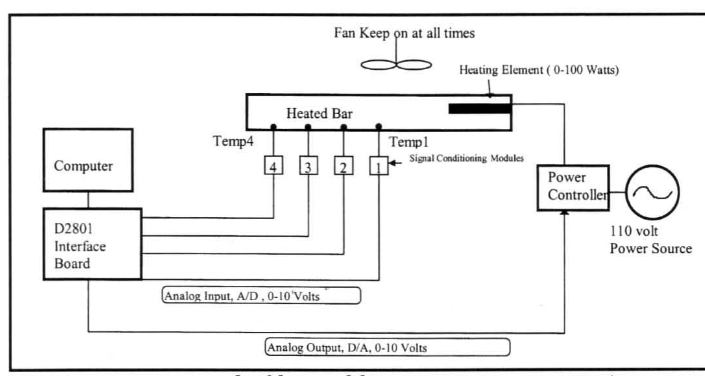

Figure 8. *Control of heated bar temperature experiment.*

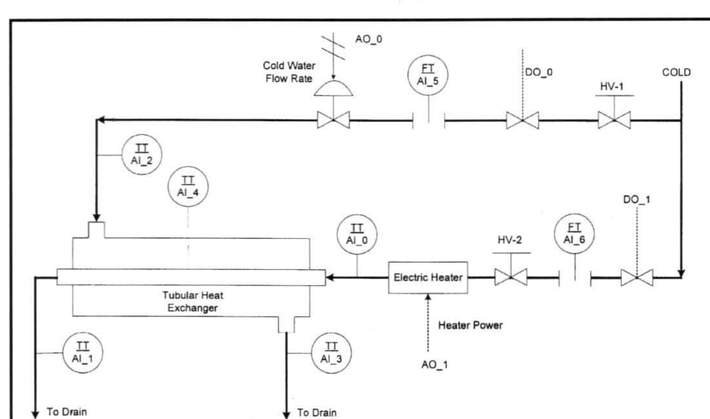

Figure 9. *Double pipe heat exchanger experiment.*

all of the heaters have safety built-in cut-offs to prevent heater burnout and fires; and as a precaution, the students are asked to turn off all power to heaters and to shut off water and air before they leave the laboratory.

VIRTUAL EXPERIMENTS

We have designed the experiments to be fast responding, with time constants of the order of a few minutes or less. Still, the time required to run tests on the real experiments can be a problem. The availability of simulated models can mitigate this to some extent. Some of the experiments described here are also available as virtual experiments. This permits the students to safely explore and test control strategies on a fast simulator before implementing it on the actual process. In addition to the laboratory experiments, we have provided a few virtual experiments (see Table 4).

Real experiments have a limited range of options. It is difficult to reproduce all problems that arise in practice. The experiments must be kept simple and safe—such experiments also consume significant time. Virtual experiments can help overcome such drawbacks by enabling simulations of industrial processes that are too complex, unsafe, or expensive to be built at laboratory scale. We have built additional virtual experiments, including the Shell fractionator column simulation and the Tennessee Eastman (TE) problem. The Shell fractionator column control problem has been used as a unifying example for comparison of feedforward, feedback, cascade, multiloop, inferential, and multivariable control structures. The TE problem is a process control challenge problem provided by industrial practitioners. The challenge is to come up with a multivariable control strategy for fast and safe transitions of the process between various operating modes. The advantage of these case studies is that a wealth of literature is available from leading research in the field.

INTERNET-ACCESSIBLE EXPERIMENTS

Development of Internet-accessible experiments is also currently in progress.[6] Internet-accessible experiments allow laboratory exercises to be assigned as homework. It would be possible for the teacher to run a demonstration from the classroom, resulting in immediate translation of theory into practice. Such a facility could be run anytime from anywhere. With greater bandwidth expected to be widely available in the future, a webcam would allow the students to see the operations remotely. These ideas were tested by making the heat-exchanger and the mixing-tank experiments accessible over the Internet. The data-acquisition systems were connected to a server and network communication was achieved through TCP/IP programs. The client-end has the same SIMULINK interface as before, allowing the remote user to manipulate the control structure while passing measurements/control signals back and forth. Safety is an important issue in such experiments, as the remote user is unable to see what is happening in the laboratory. Our studies[7] have shown that the performance of most process control systems is largely unaffected by the random transmission delays and data loss.

PEDAGOGY

It is important to design the experiments so the student can focus on the control issues. Fast dynamics are important to

TABLE 3
Other MATLAB Tools

Tool	*Description*
1. PID Controller Block	This SIMULINK block incorporates concepts found in industrial controllers such as • Controller face plate • Bumpless transfer • Reset windup protection • Auto/manual switch • Self-tuning capability • Noise filtering
2. MODELBUILDER	A toolbox to generate first-order plus dead-time models from plant test data. Uses least-square curve fitting method.
3. MPC Toolbox	A toolbox for implementing model predictive control.
4. PIDTUNER	A toolbox that implements various PID controller tuning rules.

TABLE 4
Virtual Experiments

Name	*Description*	*When Used*
1. Simple level control (MATLAB and SIMULINK with animation GUI)	A self-regulating tank level control system with PID controller	Beginning of course to introduce MATLAB and SIMULINK
2. Naphtha cracker simulator	A simulation of naphtha cracking reactor in a furnace	Middle of course to introduce identification, feedback, cascade, override, and multivariable control concepts.
3. Tennessee-Eastman Process (SIMULINK implementation)	A large-scale process-control problem proposed by industry.	As a project during the last month of the course.
4. Shell fractionator column (SIMULINK implementation)	The benchmark process-control problem proposed by Shell.	As a project during the last month of the course.

minimize the time taken to complete an experiment. This requires careful choice of flow rates, heat capacities, and tank sizes. All experiments must be intrinsically safe and forgiving of mistakes made in the control systems. Finally, it is important to ensure that the equipment is reliable over the duration of the course to prevent student frustration.

The use of simulated experiments at the beginning of the course has some significant advantages—they can be assigned as homework exercises since accessibility and availability are not issues. Simulations are very predictable, smooth, fast, and repeatable, so the students can focus on the concept at hand. At the beginning of the course, the students are still unfamiliar with the subject and a common application platform for all students facilitates classroom discussion and peer-to-peer learning.

MATLAB and SIMULINK are chosen as the medium for teaching control because of their widespread use in engineering education and research. A rich variety of textbooks, problems, tools, and toolboxes are already available for this software environment. Our students have already been exposed to MATLAB in earlier courses, providing an added incentive to use this software. SIMULINK has the advantage that control-system synthesis can be accomplished in a straightforward and convenient form through the graphical user interface and block-diagram notation. Since the block-diagram notation is used extensively in class to describe control systems, a one-to-one correspondence between lab exercises and lecture room material is possible. The SIMULINK block-diagram structure allows easy development of the control system architecture independent of the process and data-acquisition hardware/software. Our past experience with the use of commercial packages for process control, such as Distributed Control System (DCS) packages, indicates that mastering these complex environments can be time consuming and takes time away from learning the basic concepts. The one-to-one correspondence between the concepts covered in the classroom lecturees and the laboratory sessions, as well as the reliance on a common block-diagram front-end for all of the real experiments, virtual experiments, and Internet-accessible experiments, tremendously simplified the translation of classroom theory into practice.

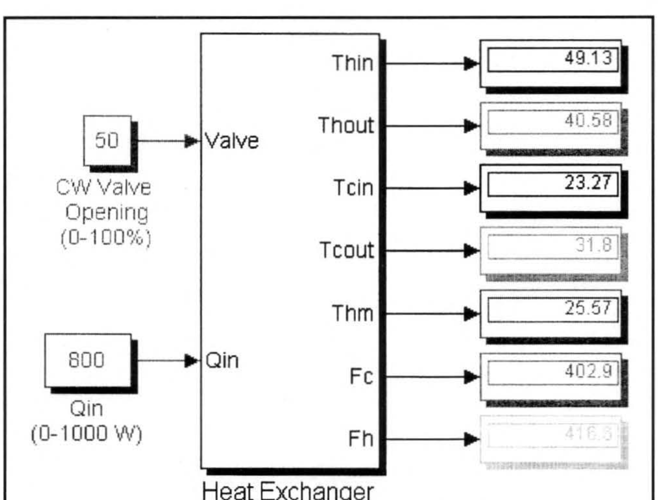

Figure 10. *SIMULINK interface for the double-pipe heat exchanger.*

CONCLUSIONS

The combination of lecture/laboratory provides a seamless environment for learning and applying process-control concepts. By providing a simple, yet powerful user interface using the familiar block-diagram notation of representing and analyzing control systems, the transition from classroom to the laboratory is made efficient, allowing the students more time to focus on learning process control as opposed to technology or software.

Real experiments motivate students and increase their enthusiasm for learning. By providing a sufficiently powerful tool, the students can become creative and develop a sense of ownership for their work, which in turn improves their confidence and interest in learning. There is a sense of accomplishment when the control system they design works well and achieves the set objectives.

The main drawback associated with running such a real-time physical laboratory is the need to keep up with the fast pace of change in technology. One must invest constantly in learning new technology and incorporating it into the hardware and software used in the laboratory. Fortunately, each new generation of hardware and software has made usability simpler and more powerful.

ACKNOWLEDGMENT

We would like to acknowledge the financial support provided by MEMC Electronic Materials Inc., and The Camille and Henry Dreyfus Foundation. We wish to thank Patrick Underhill and David Kettler, who developed the SIMULINK models for the heat exchanger and the naphtha cracker, respectively.

REFERENCES

1. Edgar, T.F., "Information Technology and Chemical Engineering Education: Evolution or Revolution?", University of Missouri-Rolla Academy of Chemical Engineers Lectureship (1999)
2. ChE 462: Process control course website <http://it.che.wustl.edu/che462/>
3. Brosilow, C.B., and B. Joseph, *Techniques of Model-Based Control,* Prentice Hall, Inc., Upper Saddle River, NJ; to be published (2002)
4. Bequette, B.W., and B.A. Ogunnaike, "Chemical Process Control and Practice," *IEEECont. Sys. Mag.,* **21**(2), 10 (2001)
5. Ying, C., and B. Joseph, "Linear System Identification Using Polynomial Models," *Ind. Eng. Chem. Res.,* **38**, 4712 (1999)
6. Srinivasagupta, D., "An Internet-Mediated Process Control Laboratory," submitted to *IEEE Cont. Sys. Mag* (2001)
7. Srinivasagupta, D., B. Joseph, and H. Schättler, "Effect of Random Communication Delays on Network Mediated Control Systems: A Case Study," submitted to *J. Process Control* (2001) ❐