

PICLES

A Simulator for Teaching the Real World of Process Control

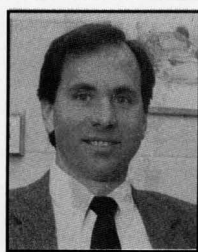
DOUGLAS J. COOPER
*University of Connecticut
Storrs, CT 06269-3222*

Process control classes often become more like abstract mathematics courses as the semester proceeds. Many instructors rightly believe that there is a need for students to experience the application of classroom theory to real processes so they may appreciate not only the nuances but also the main points of the lectures.

Having spent three years in the real world of process control with Chevron Research Company, I became frustrated when I began teaching at the university level and discovered that (outside of the lab) few tools were available to me to teach many of the lessons I considered important.

Too many important concepts are lost when the bulk of assignments begin, "Start with this transfer function and . . ." For example, students must learn the serious implications that arise because transfer functions disregard that real processes are nonlinear and have measurement noise and other nonideal behaviors. They must learn to quickly and reliably perform identification studies (real-world production people can be downright ornery if one asks to experiment with their process). If they succeed in obtaining data from the process, students must learn to use it to reasonably approximate the local process behavior with a linear model—and that only then do they have the transfer function for use with their classroom design theory.

When their analysis is complete, students must



Douglas J. Cooper is Associate Professor of Chemical Engineering and has been teaching process control at the graduate and undergraduate level for the past eight years. He received his BS from the University of Massachusetts in 1977, his MS from the University of Michigan in 1978, and his PhD from the University of Colorado in 1985 after three years of industrial experience with Chevron Research Company

© Copyright ChE Division of ASEE 1993

learn that their controller design, no matter how sophisticated the approach, is only an initial approximation—that it must be fine-tuned on the real process. In the real world, this fine-tuning proceeds by trial-and-error and must consider both set point tracking and disturbance rejection.

The best instruction concerning the real world (short of the school of hard knocks) is obtained through carefully constructed laboratory experiences. Although we have several nice process control experiments in our laboratory at the University of Connecticut, the reality is that each study can take several hours to perform. As such, it is not reasonable to have the students explore more than the most major issues in the lab.

To teach these important lessons, the Process Identification and Control Laboratory Experiment Simulator (PICLES) was developed. The contribution PICLES brings to an existing course is that it enables the students to quickly explore many of the lessons by following the same procedures they would have to follow if working with a real process.

WHAT IS PICLES?

Let me begin by pointing out that PICLES is *not* a control system analysis or design package. Quite the opposite, this software provides realistic processes that students can use to practice the analysis and design methods they are taught. Students say that PICLES is easy, and even fun, to use. Most commands can be executed with simple key-strokes. Colorful graphics help the students follow the action on the screen as the results of their decisions unfold.

The processes in PICLES encompass a variety of behaviors. The processes have varying degrees of nonlinearity so students can explore how process behavior can change with the operating regime. This also lets them practice compromising controller tunings to maintain stability over a wide range of nonlinear operation.

The processes range from low to high order and

Chemical Engineering Education

Having spent three years in the real world of process control with Chevron Research Company, I became frustrated when I began teaching at the university level and discovered that (outside of the lab) few tools were available to me to teach many of the lessons I considered important.

have different process gains, time constants, and apparent dead times, so students can investigate how these phenomena affect process behavior and controller stability. The processes have noise in the sampled data so students can see that, in practice, the difference between a 10% overshoot and a 15% overshoot can sometimes be indistinguishable.

In the current release (version 2.1), available controllers are all PID, and with PICLES it is easy to explore all combinations from P-only to full PID control. Because each process has colorful, dynamic graphics, after performing a controller design students can implement their solution and obtain immediate visual feedback on system performance.

There is a PID Velocity algorithm and a PID Position algorithm, so students can observe the consequences of reset windup. They can select "Derivative on Measurement" or "Derivative on Error" so they can see what "derivative kick" is all about. Some controllers require the student to enter the bias or null value, while others have a bumpless feature where the bias is automatically set in a fashion similar to what they would encounter with some commercial controllers.

There are also model-based controllers in PICLES.

A Smith predictor enables students to observe how dead time affects controller performance and that dead time compensation offers real benefits. A Feed Forward element permits them to see how disturbance rejection works using both static and dynamic compensators. Decouplers enable them to explore methods for minimizing loop interaction on the distillation column.

COMPUTER SYSTEM REQUIREMENTS

PICLES is designed to run on IBM-compatible personal computers. The computer must have at least EGA graphics, although VGA graphics provides better resolution. For rapid execution, a computer with a '386 or '486 processor should be used. A math coprocessor is not required, but it adds additional speed to program execution.

THE PICLES PROCESSES

Gravity Drained Tanks • This process, shown in Figure 1, is two non-interacting gravity-drained tanks in series (see assignment 1b later in this article for more about the figure). The manipulated variable is the flow rate of liquid entering the first tank. The measured/controlled variable is the liquid level in the second tank. This process displays a nonlinear behavior because the drain rate from each tank is proportional to the square root of the hydrostatic head (liquid level in the tank). The disturbance, or process load, is a flow out of the second tank due to a positive displacement pump. Hence the disturbance is independent of level except that it loses suction at extremely low liquid levels in the second tank.

Heat Exchanger • This process, shown in Figure 2, is a counter-current lube oil cooler (see assignment 3e later in this article for more about this figure). The manipulated variable is the flow rate of cooling water on the shell side. The measured/controlled variable is lube oil temperature exiting the exchanger on the tube side. An interesting characteristic of this nonlinear process is that distur-

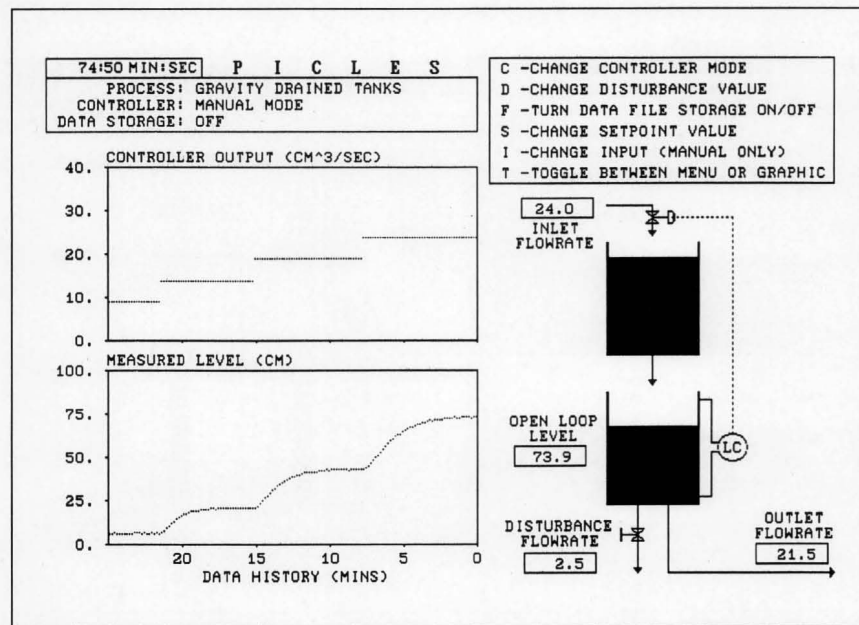


Figure 1. Gravity Drained Tanks shows nonlinear behavior.

bances, generated by changing the flow rate of warm oil that mixes with the hot oil entering the exchanger, display an inverse or nonminimum phase behavior. The process also has a negative steady state gain.

Design a Process • Design a Process has a display, shown in Figure 3, that is similar to that found on commercial controllers (see assignment 6a for more about this figure). It permits students to input a transfer function and obtain a visual appreciation when studying problems found in textbooks. The student can specify a steady state process gain, an apparent dead time, up to three process time constants, and a valve time constant. It is also possible to specify a "linearity factor" if a nonlinear process is to be designed.

Mystery Processes • These processes are not really mysterious. Rather, they are simply Design a Process with a fanciful name and with all parameters pre-specified and hidden from the student. Thus each Mystery Process displays a behavior that ranges from first to fourth order and has different overall process gains, time constants, apparent dead times, and degrees of nonlinearity. Because there is no *a priori* indication of expected process behavior, the student must rely strictly on process identification studies for controller design. This simulates the disassociation that is often felt when tuning controllers from a remote control room and makes the simulations perfect for project work later in the semester. All of the mystery processes use the same graphic as shown in Figure 3.

Pumped Tank • This process is a surge tank. The manipulated variable is brine flow rate out of the bottom of the tank and is adjusted with a throttling valve at the discharge of a constant pressure pump. This approximates the behavior of a centrifugal pump operating at relatively low throughput. The measured/controlled variable is the liquid brine level. This surge tank presents an interesting control challenge because of the integrating nature of the process. The disturbance variable, or process load, is the flow rate into the tank.

Distillation Column • The Distillation Column, shown in Figure 4, is a binary distillation column that separates water and methanol (see assignment 8a for more about this figure). The column dynamics are simulated using a model published by Wood and Berry.^[1] There are two controlled variables and two manipulated variables. The reflux rate controls the distillate composition and the rate of steam to the reboiler controls the bottoms composition. The feed rate to the column is the dis-

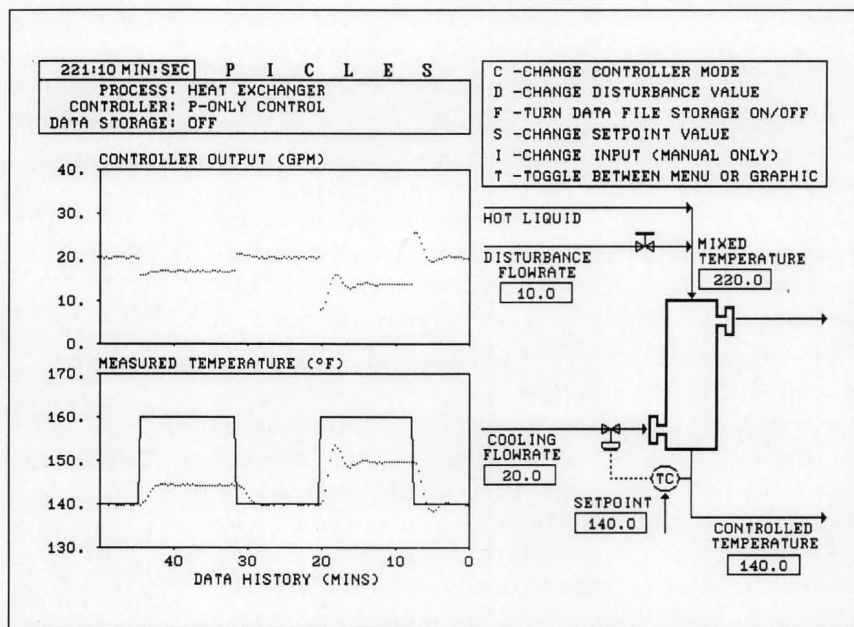


Figure 2. Heat Exchanger under P-Only control with different controller gains.

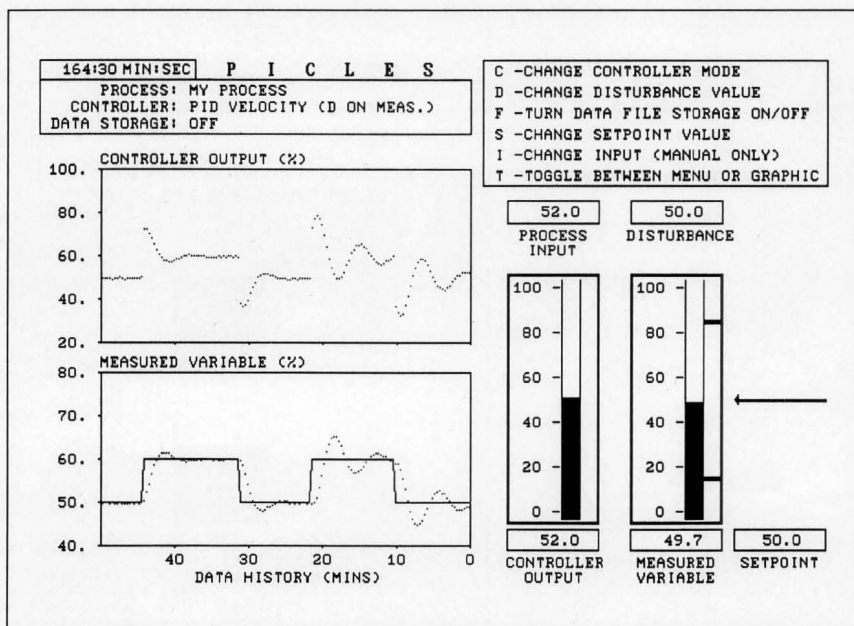


Figure 3. Design a Process under PI control with differing amounts of dead time.

turbance variable. This process illustrates interaction between two controllers.

AVAILABLE CONTROLLER MODES

The control algorithms in the current version of PICLES are all PID and include

- Manual Control
- P-Only Control (Manual Bias)
- Velocity PID Control (Derivative on Measurement)
- Velocity PID Control (Derivative on Error)

- Position PID Control (Bumpless)
- Velocity PID with Smith Predictor
- Velocity PID with Feed Forward
- Velocity PID with Decoupler (Distillation Column Only)

Version 3, which will be available in 1994, will include I-Only control and a discrete controller algorithm.

Figure 5 shows the Design Menu used to specify controller parameters. The process being simulated in this figure is the Heat Exchanger process (see assignment 5a for more about this figure). Note that the simulation noise level can be changed if it is appropriate for an assignment. Also, in the spirit of the "real world," high and low alarms can be set to provide additional challenge in using the program.

Although the limitation to PID algorithms is viewed as a serious limitation by some, I try to exploit this fact within the classroom lectures. For example, I establish that the PID controller is a special case of the Internal Model Control design method. Also, I show how the Smith Predictor is a limiting case of some predictive controller design methods. Thus, PICLES can be used to explore certain aspects of these newer design standards.

Using the limitation to PID algorithms is viewed as a serious limitation by some, I try to exploit this fact within the classroom lectures. For example, I establish that the PID controller is a special case of the Internal Model Control design method. Also, I show how the Smith Predictor is a limiting case of some predictive controller design methods. Thus, PICLES can be used to explore certain aspects of these newer design standards.

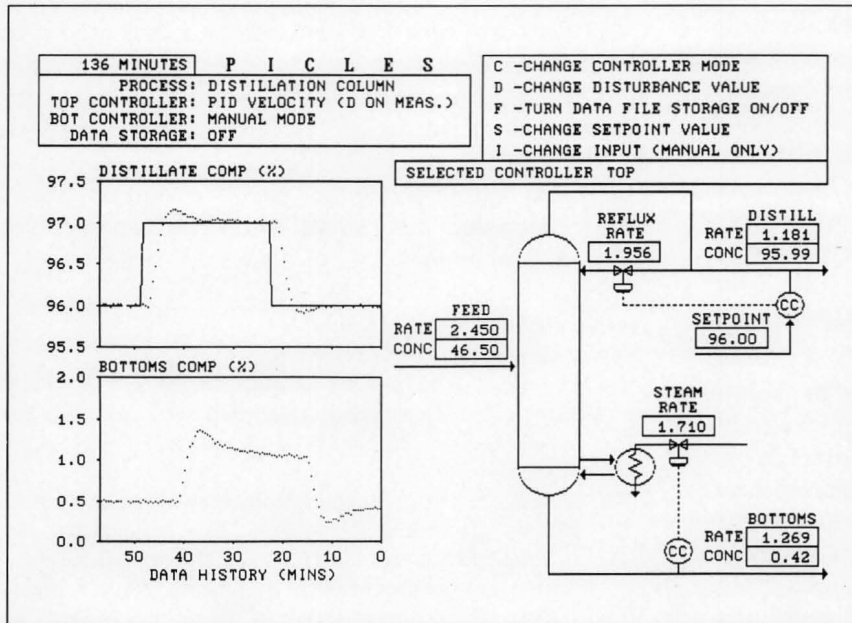


Figure 4. Distillation Column with Distillate under PI control and Bottoms in Manual

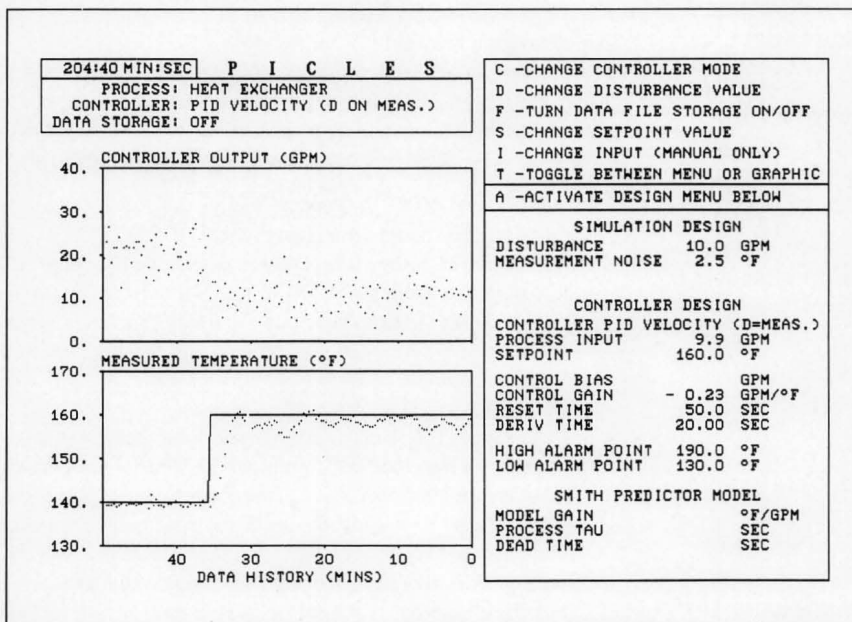


Figure 5. Design Menu of Heat Exchanger under PID control with measurement noise.

USING PICLES IN THE COURSE

I start with the Gravity Drained Tanks process. The model can be easily derived in class, it behaves intuitively, and the nonlinear behavior is modest. The simulation graphics also provide realism to help give the students an understanding of the dynamic behavior of the process.

Since I believe that some practice in programming is important, I also have the students code up their own Gravity Drained Tanks process based on the equations derived in class. I then have them determine the process parameters which cause their simulation to approximate the dynamics of the PICLES simulation.

After several assignments in process dynamics and process identification, I move on to the Heat Exchanger.

It is a slightly more complicated process, but it still behaves intuitively. It has a higher degree of nonlinearity and also has a negative steady state gain, which reinforces my lecture that gains not only have magnitude and units, but also a sign. The nonminimum phase or inverse dynamics of the disturbance response provides another new twist.

After they have explored several investigations of process dynamics, some identification methods, and explored a few controllers and design techniques using Gravity Drained Tanks and Heat Exchanger, I use the Mystery Processes for project work. I assign a different Mystery Process to each group of students and let them tie things together by doing an identification, preliminary controller design, and finally determining a single "best" tuning for both set point tracking and disturbance rejection, all as one assignment. Because the processes are nonlinear, each student can have his or her own project by specifying different ranges of operation for each problem (*i.e.*, Amy must design for an output range of 20-30%, etc.).

I use Design a Process intermittently to isolate specific process behaviors. For example, I ask the students to implement a true first-order process under P-Only control and let them demonstrate that such a process is unconditionally stable for all values of controller gain. They then show that a second-order process under P-Only control can approach the limit of stability, and finally, that higher order processes under P-Only control can go unstable. When combined with a class discussion on system stability using root-locus, the students benefit from relating theory to practice while the subject is being taught.

As another example of using Design a Process later in the course, I assign a set of time constants and a process gain and ask the students to design and validate a controller. Then, keeping those process variables and tuning parameters fixed, they add dead time to the process and discuss their observations on the effect dead time has on closed-loop performance. Finally, they design and implement a Smith predictor to relate our in-class derivations and discussions with actual application to assist them in understanding the benefits of dead-time compensation.

When the students start feeling confident, I give them the Pumped Tank process. The integrating nature of the process really surprises them and requires me to do a lot of explaining ("Why is there no offset with a P-Only controller?" "How come with a PI controller, this process goes unstable when the

controller gain is too high *and* too low?").

Finally, the Distillation Column lets the students see what can happen when more than one controller is operating on the same process. The interactions show them that optimizing controllers individually does not necessarily produce an optimum solution when the controllers begin to interact. Also, the students can investigate how model-based decouplers can work to minimize this interaction.

EXAMPLE HOMEWORK ASSIGNMENTS

To illustrate how PICLES can be integrated into an existing course in process control, this section lists sample homework assignments. These assignments follow the order of development used in most textbooks and let students visually appreciate these important concepts. The five figures previously discussed also serve as partial answers to selected problems.

Assignment on Process Dynamics

1. Using Gravity Drained Tanks in Manual mode:
 - a. Plot and discuss how the measured level responds to step changes in the manipulated inlet flow rate and the disturbance flow rate. Comment on the natural stability of the process.
 - b. Starting from three different steady state operating regimes, plot how the measured level responds to manipulated inlet flow rate perturbations of fixed size. Based on these plots, discuss the nonlinear nature of the process. (*Answer*: Figure 1 shows the nonlinear nature of the process because the measured level responds differently for three changes of equal size in the manipulated inlet flow rate.)

Assignment on Process Identification

2. Using Design a Process:
 - a. Generate an open loop input/output step response curve for a true first-, second-, and third-order process. How does the time to 63.2% of change compare to the time constants assigned? Discuss your conclusions.

Assignment on P-Only Control

3. Using the Heat Exchanger in P-Only mode:
 - a. For a design operating temperature, determine the value and units of the controller bias.
 - b. Obtain a FOPDT (first order plus dead time) model describing process dynamics around this design temperature and use it to compute a P-Only controller gain using ITAE, direct synthesis, IMC, etc.
 - c. Starting with this controller gain and bias value, use trial-and-error to find the "best" gain, where for this assignment "best" is defined as a 10% overshoot for set point steps of a specified size. Now show the ability of this "best" controller to reject step disturbances.
 - d. Starting from the design operating temperature and using your "best" tuning, make set point step changes of various sizes in both directions. Discuss your observations on offset.

- e. Pick a specific set point change and plot the response of the process when using your "best" controller gain, half of that gain, and twice that gain. Discuss your observations on the relationship between controller gain and offset. (*Answer:* Figure 2 shows set points steps with two different controller gains, and that offset and the oscillatory nature of the response changes as controller gain changes.)

Assignment of PI Control

4. Using Gravity Drained Tanks or Heat Exchanger in PI velocity mode:
 - a. Explain why no bias is necessary for this controller.
 - b. Obtain FOPDT model describing process dynamics around a design point of operation and use it to compute a PI controller gain and reset time using ITAE, direct synthesis, IMC, etc.
 - c. With these parameters as a starting point, use trial-and-error to find the gain and reset which provide a "best" performance. Here, "best" performance is defined as a 10% overshoot and a 25% decay ratio to a set point step of specified size. Why can we design for two performance criteria with a PI controller, but only for one with a P-Only controller?
 - d. Plot a matrix of process responses for the same set point step where this matrix includes all combinations of your "best" tuning, a gain that is double and half of your "best," and a reset time that is double and half your "best." Use your observations to explain the roles of gain and reset time on controller performance.

Assignment of PID Control

5. Using the Heat Exchanger in PID velocity mode:
 - a. Design and implement a PID controller and compare its performance to PI control. For this comparison, test a number of set point and disturbance scenarios and show where the derivative action really pays off. Plot the distinctive scenarios and use them to explain why or why not any performance benefit occurred. (*Answer:* Figure 5 shows that derivative action can produce very poor performance when employed on a noisy measured variable.)

Assignment on Dead Time and the Smith Predictor

6. Using Design a Process in PID with Smith Predictor mode:
 - a. For the assigned process gain and set of time constants, design and validate a PI or PID velocity mode controller that gives a 10% overshoot and a 25% decay ratio for a given set point step. Keeping the process variables and tuning parameters constant, add dead time to the process and discuss your observations on the effect dead time has on closed-loop performance for this same set point step. (*Answer:* Figure 3 shows two sets of set point steps. PI controller tuning is fixed throughout. Controller performance is markedly different for the first set point steps where the process possesses no dead time compared to the second set of steps where the process possesses thirty seconds of dead time.)
 - b. Keeping the same process gain, set of time constants, and including the dead time, tune your controller to again produce a 10% overshoot and a 25% decay ratio. Compare this plot and tunings to the case where no dead time was present. Discuss your observations.
 - c. Now design and implement a Smith Predictor and again tune the controller to produce a 10% overshoot and a 25% decay

ratio. Compare this plot and tunings with the previous two cases and discuss the pros and cons of dead-time compensation.

Assignment on Disturbance Rejection and Feed Forward

7. Using Gravity Drained Tanks or Heat Exchanger in PID with Feed Forward mode:
 - a. For the design point of operation, develop a FOPDT model of the disturbance-to-output dynamic relationship. Using this model, compare a static and a dynamic feed-forward compensator for step changes in the disturbance variable.
 - b. For the Gravity Drained Tanks, the disturbance immediately impacts the measured variable while there is a lag before input variable manipulations can compensate. Explain how this influences your comparison of the static and dynamic compensators.
 - c. For the Heat Exchanger, there is a reasonable lag before a disturbance impacts the measured variable. Discuss how this influences your comparison of the static and dynamic compensators.

Assignment of Multivariable Control and Decoupling

8. Consider controller design for the Distillation Column when given specified design operating concentrations for the distillate and bottoms.
 - a. While the bottom controller remains in Manual mode, design and implement a PI controller for the top controller. Plot the performance of the controller for distillate concentration set point steps both up and down. (*Answer:* Figure 4 shows one possible solution to this question.)
 - b. While the top controller remains in Manual mode, design and implement a PI controller for the bottoms controller. Plot the performance of the controller for bottom concentration set point steps both up and down.
 - c. Using the controller tuning parameters from a and b above, implement PI controllers on both loops. Make set point changes for both controllers and discuss loop interaction.
 - d. Now design and implement both static and dynamic controllers. Perform the same set point changes as in part c and discuss the impact of model-based decoupling.

FINAL NOTE

For more information about PICLES and available teaching materials, write to the author at the Chemical Engineering Department, University of Connecticut U-222, Storrs, CT 06269-3222.

ACKNOWLEDGMENTS

I would like to thank the students without whom PICLES would not exist. These include *Architects:* Allen Houtz, Robert Schlegel, and Adam Lalonde, and *Builders:* Scott Ferrigno, Ralph Hinde, Jr., Larry Megan, C. Steven Micalizzi, Phil Pearson, and Yan Wan.

REFERENCES

1. Wood, R.K., and M.W. Berry, "Terminal Composition Control of a Binary Distillation Column," *Chem. Eng. Sci.*, **28**, 1707 (1973) □