Int. J. Engng Ed. Vol. 17, No. 3, pp. 276–287, 2001
Printed in Great Britain.

0949-149X/91 $3.00+0.00

Control Station: An Interactive Simulator for Process Control Education*

DOUG COOPER and DANIELLE DOUGHERTY
Chemical Engineering Department, University of Connecticut, Storrs, CT 06269–3222, USA.
E-mail: cooper@engr.uconn.edu

This paper explores the use and benefits of the Control Station® training simulator for process control education. A process control training simulator provides an interactive learning environment that can enhance learning by integrating the theoretical abstraction of textbooks with the tactile nature of the lab and plant. The primary objective of a training simulator is education. It can motivate, help with visualization, and provide hands-on practice and experience. Examples presented illustrate how the standard curriculum can be enhanced with a series of hands-on exercises and study projects.

INTRODUCTION

HANDS-ON challenges which demonstrate and reinforce abstract process control concepts benefit the learning process [1]. Such challenges can be motivating, promote critical thinking, facilitate understanding in the use and limitations of the theory, and help prepare students for the challenges of the professional world.

A process control training simulator offers an alluring method for providing students with the significant hands-on practice critical to learning the subject. The proper tool can provide virtual experiences much the way airplane and power plant simulators do in those fields. It can give students a broad range of focused engineering applications of theory in an efficient, safe and economical fashion. And it can work as an instructional companion by providing interactive challenges that track along with classroom lectures.

Process control is a subject area well suited to exploit the benefits of a training simulator [2, 3]. Modern control installations are computer based, so a video display is the natural window through which the subject is practiced. With color graphic animation and interactive challenges, a training simulator can offer experiences that literally rival those of the real world [4]. These experiences can be obtained risk free and at minimal cost, enabling students to feel comfortable exploring non-standard solutions at their desk. If properly designed as a pedagogical tool with case studies organized to present incremental challenges, we believe learning process control can be enormously enhanced with such a training simulator.

* Accepted 12 October 2000.

ABOUT CONTROL STATION

To this end, Control Station was developed as an instructional tool for process dynamics and control. Control Station is a point-and-click environment compatible with Microsoft Windows® and most computer networks. The development goal was a simulator that is visually appealing, easy-to use and accepted by students and practitioners alike. The software is designed so students will:

- learn how to collect and analyze process data to determine the essential dynamic behavior of a process;
- learn what 'good' or 'best' control performance means for a particular process;
- understand the computational methods behind the different control algorithms and learn when and how to use each one to achieve desired performance;
- learn how the different adjustable tuning parameters required for control algorithm implementation impact controller performance and how to determine values for these parameters;
- become aware of the limitations and pitfalls of each control algorithm and learn how to use this knowledge to their advantage.

Control Station is comprised of three modules: *Case Studies*, *Custom Process* and *Design Tools*. The *Case Studies* module provides real-world experience in modern methods and practices of process control through a collection of realistic processes to practice upon. The *Case Studies* processes available for study in Control Station include: level control of a tank, temperature control of a heat exchanger, concentration control of a reactor, and purity control of a distillation column. The controllers available for study on these processes include: P-only, PI, PD and PID controllers, cascade, feed forward, multivariable decoupling, model predictive (Smith predictor), and discrete sampled data control.

The *Custom Process* module is a block oriented environment that lets students construct a process and controller architecture to there own specifications for a wide range of custom control analyses. Students can investigate the benefits and drawbacks of different control architectures, tunings sensitivities, loop performance capabilities, and a host of other issues important to the practitioner.

The *Design Tools* module is used to fit linear dynamic models to process data and to compute PID controller tuning values. The models from *Design Tools* can also be used to construct advanced control strategies that use process models internal to the controller architecture such as feed forward and model predictive control. Because data can be imported from real operating processes, *Design Tools* can solve a wide range of challenging problems for controller design, analysis and tuning.

A CHEMICAL PERSPECTIVE

Each discipline views process control from a different perspective. To help orient the reader, consider these typical examples drawn from chemical process control:

- *Process variables:* temperature, pressure, pressure drop, level, flow, density, concentration;
- *Final control elements:* solenoid, valve, variable speed pump or compressor, heater or cooler;
- *Control algorithms:* on/off, PID, cascade, ratio, feed forward, multivariable decouplers, model predictive;
- *Process applications:* reactors, separators, distillation columns, heat exchangers, furnaces.

Many chemical engineering processes are literally one of a kind. Consequently, their associated control system will be unique in design and implementation.

Additionally, chemical processes can be nonlinear and nonstationary, and can have long time constants, significant dead time, and/or noisy measurement signals. Disturbances occur from numerous sources including loop interaction from other controllers in the plant.

EXAMPLE LESSONS

The following lessons have been drawn from the Control Station process control training simulator to illustrate the value such software provides to learning. The reader can download a free Control Station demo at: www.engr.uconn.edu/control/.

P-only controller performance

The computer graphic display for the gravity drained tanks process, shown to the right in Fig. 1, consists of two vessels stacked one above the other. Liquid drains freely through a hole in the bottom of each tank. The controller output signal manipulates the flow rate of liquid entering the top tank. The measured process variable is the liquid level in

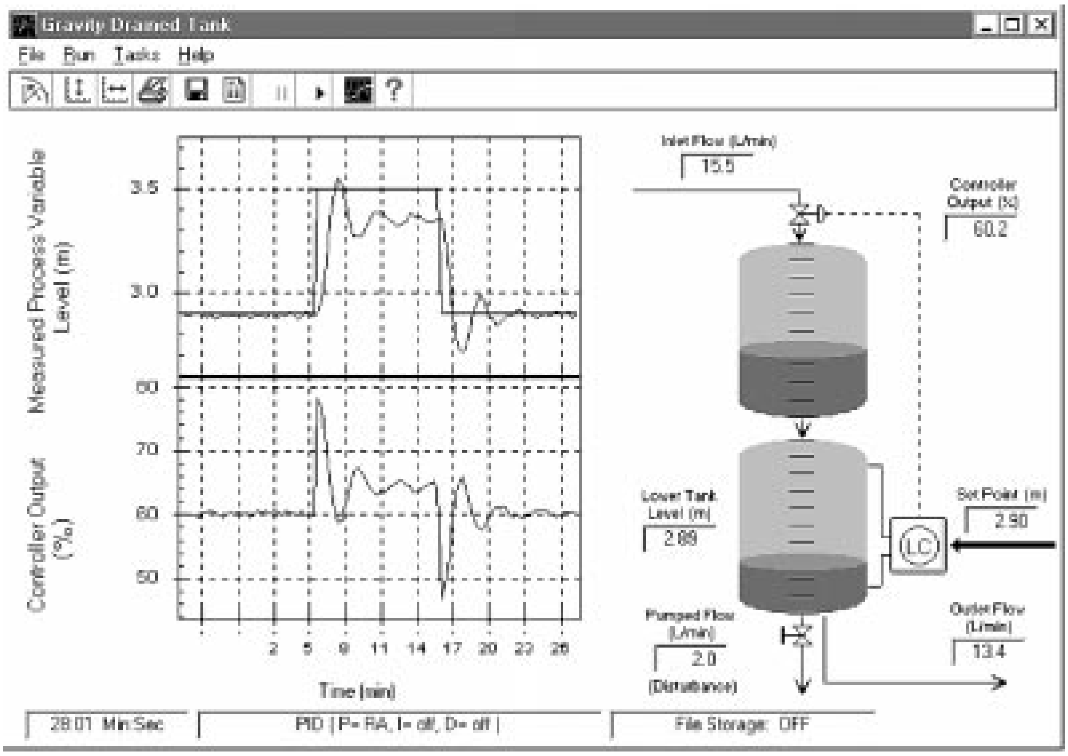

Fig. 1. Gravity drained tanks graphic.

the lower tank. The disturbance variable is a secondary flow out of the lower tank from a positive displacement pump, so it is independent of the liquid level.

The left of the screen shows moving strip charts with 'live' process data. The upper strip chart tracks the measured liquid level in the lower tank while the lower strip chart tracks the controller output.

The traditional place to begin a course is with the study of process dynamics. Students start with this process because its dynamic behavior is reasonably intuitive. If they increase the liquid flow rate into the top tank, the liquid levels rise. If they decrease the inlet flow rate, the levels fall.

To study dynamic behavior, students generate a step test plot and compute by hand the first order plus dead time (FOPDT) model parameters: steady state process gain, K_P, overall time constant, τ_P and apparent dead time, θ_P. After they have gained mastery with hand calculations, they use *Design Tools* to automate the model fitting task so they can explore more practical issue. A fit of process data from a double pulse test using *Design Tools* is shown in Fig. 2 for the gravity drained tanks.

Students then move on to controller design. Tuning a controller from the PID family follows a standard procedure applicable to real processes in the plant or lab as well as to the *Case Studies* simulations available in Control Station:

1. Move the process to the design level of operation.
2. Step, pulse or otherwise perturb the controller output signal so that the measured process variable response clearly dominates the noise in the measurement signal. Record the process response data.
3. Fit a first order plus dead time (FOPDT) model to this controller output to measured process variable data. As shown in Fig. 2, *Design Tools* is well-suited for this modeling task.
4. Use the resulting FOPDT model parameters in a correlation to compute initial controller tuning values. *Design Tools* also performs this task if desired.
5. Implement your controller on the actual process and perform final tuning by trial and error until your control objectives are satisfied.

In this example, students use their FOPDT model parameters in tuning correlations to compute a P-only controller gain, K_C. Figure 3 displays a Control Station plot showing set point tracking performance for the gravity drained tanks under P-only control. The K_C tuning value used to generate the set point response to the left in the figure is computed from the integral time weighted absolute error (ITAE) correlation [5, 6] using the FOPDT model parameters from Fig. 2.

With this as a starting point, the students now turn to what-if studies. The set point response to the right in Fig. 3 explores how K_C impacts offset and oscillatory behavior for set point tracking under P-only control. Students also explore disturbance rejection under P-only control. Is the best tuning for set point tracking the same as for disturbance rejection? And how is 'best' tuning defined?

For this and all Control Station processes, the

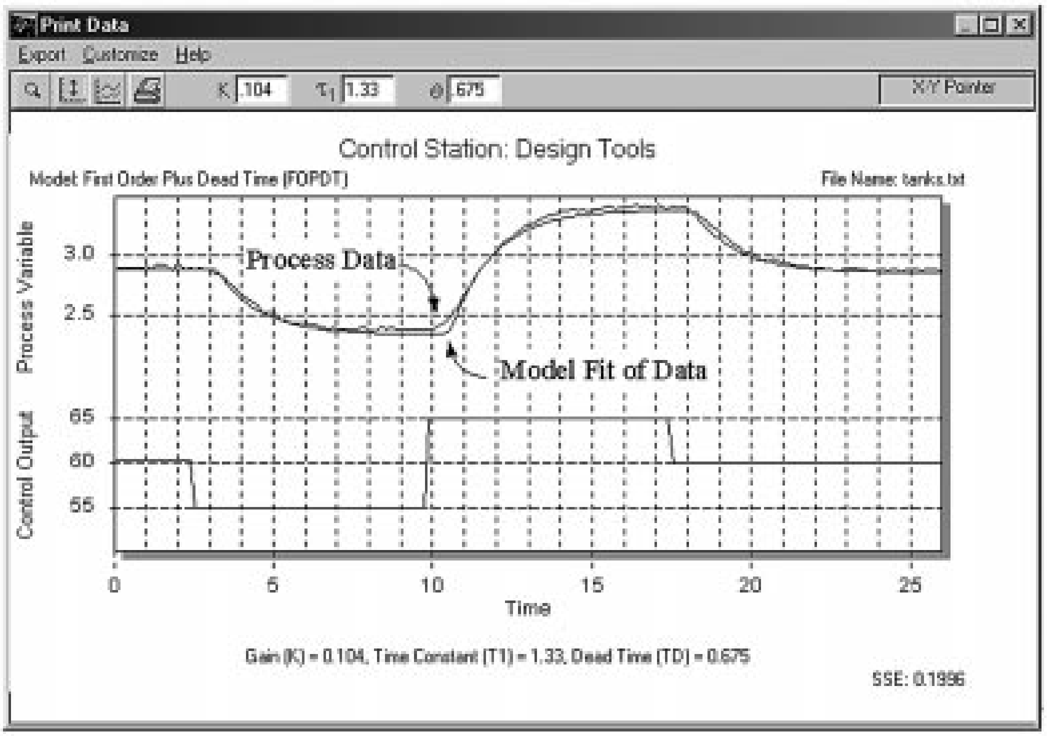

Fig. 2. A Design Tools analysis of process data yields a FOPDT model.

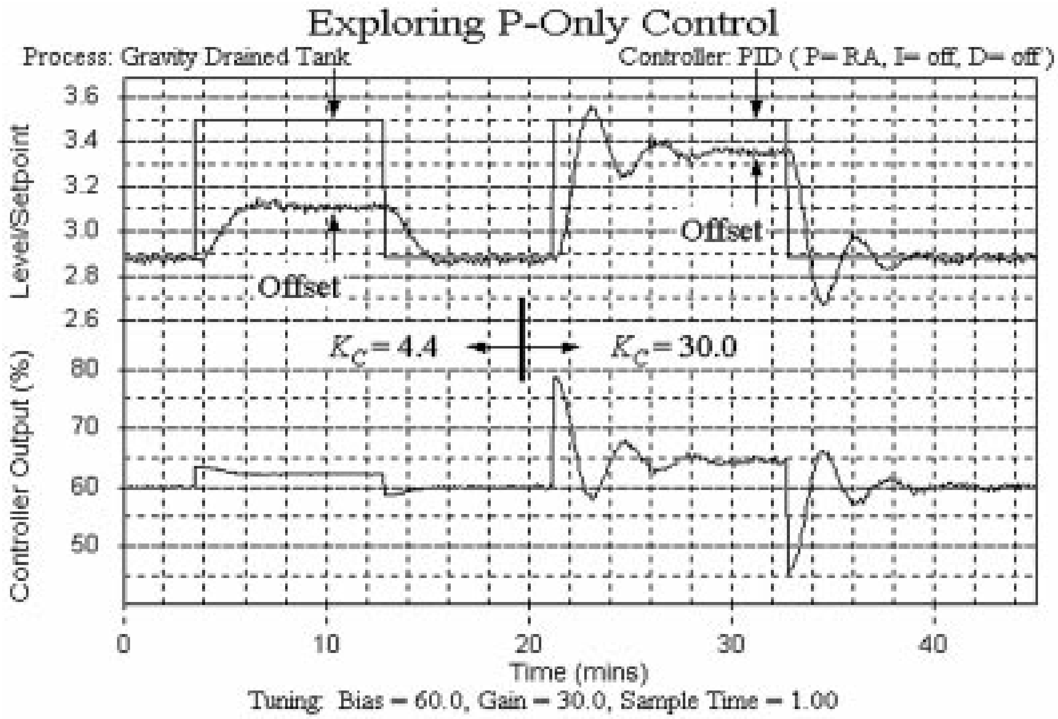

Fig. 3. P-only set point tracking performance as K_C changes.

students can change the level of random noise in the measured process variable. They can also manipulate the controller output signal, set point and disturbance variables using a step, oscillating, ramp, or pseudo-random binary sequence (PRBS) signal sequence. The current version of Control Station offers only one disturbance variable for each process, and this disturbance can be changed at will by the student. We note that this is not realistic since a real plant can have many disturbances from a variety of sources that will affect the process, and as disturbances, they are generally not available for manipulation by the engineer. The students are made aware of this during class.

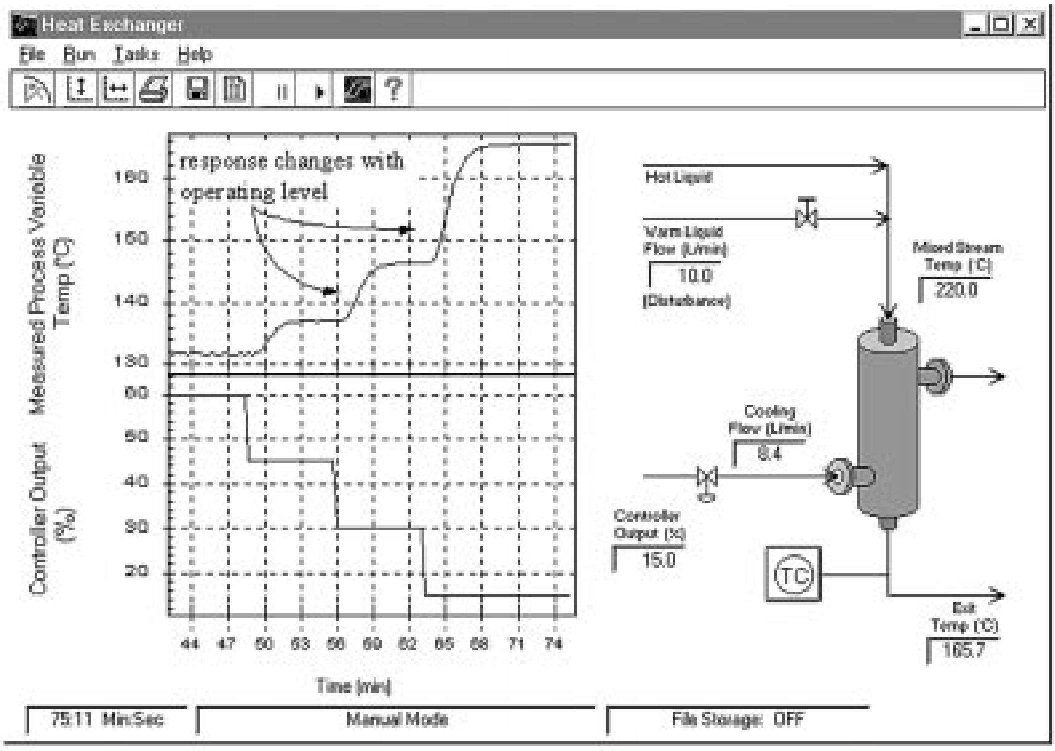

Fig. 4. Heat exchanger displays nonlinear behavior.

PI control and nonlinear behavior

The process graphic for the counter-current, shell and tube, lube oil cooler (a kind of heat exchanger) is shown to the right in Fig. 4. The controller output signal manipulates the flow rate of cooling liquid on the shell side. The measured process variable is the lube oil temperature exiting on the tube side.

Students learn an important lesson about process dynamics by studying the nonlinear character of this process as shown in the strip charts to the left of Fig. 4. The steady state gain of the process clearly changes as the operating level changes. Less obvious is that the time constant of the process also changes.

For processes that have such a nonlinear character, students learn that the performance of a controller will change as the process moves across operating levels. Figure 5 illustrates this point. The exchanger is under PI control and as the set point is stepped to different operating levels, the nonlinear behavior of the process clearly impacts set point tracking performance. Thus, students learn that a controller is designed for a specific or design level of operation, and the best practice is to collect dynamic test data as near as practical to this design operating level.

Figure 5 also shows that the heat exchanger has a negative steady state gain. Students learn that a complete design includes specifying the action of the controller (reverse vs. direct acting) [5–8]. They learn this concept because if they enter it wrong, the controller output will quickly drive the valve to either full open or full closed and it will remain there until the correct controller action is implemented.

For what-if studies, students explore how PI controller tuning parameters interact and affect set point tracking performance. Figure 6 shows a tuning map that they develop from an ordered tuning investigation using an ideal linear transfer function process that is available in the *Custom Process* module. An analogous study can explore the impact of controller tuning on disturbance rejection performance.

PID control and measurement noise

Derivative action can decrease the process settling time because it resists rapid movement in the measured process variable [5]. In Control Station, the PID controller algorithm can be implemented with a choice of derivative action either on controller error or process measurement [5–8]. Students learn how derivative action impacts controller performance with studies similar to that shown in Fig. 7, which focuses on the derivative time tuning parameter.

The center trace of Fig. 7 shows the set point tracking performance of a PID controller tuned using the IMC (internal model control) tuning correlation [5, 6]. For all responses in Fig. 7, the proportional and integral tuning parameters K_C and τ_I remain constant. Also, the measurement noise has been set to zero. The trace to the left shows how the oscillating nature of the response increases as derivative action is cut in half. The trace to the right shows that when derivative action is too large, it inhibits rapid movement in the

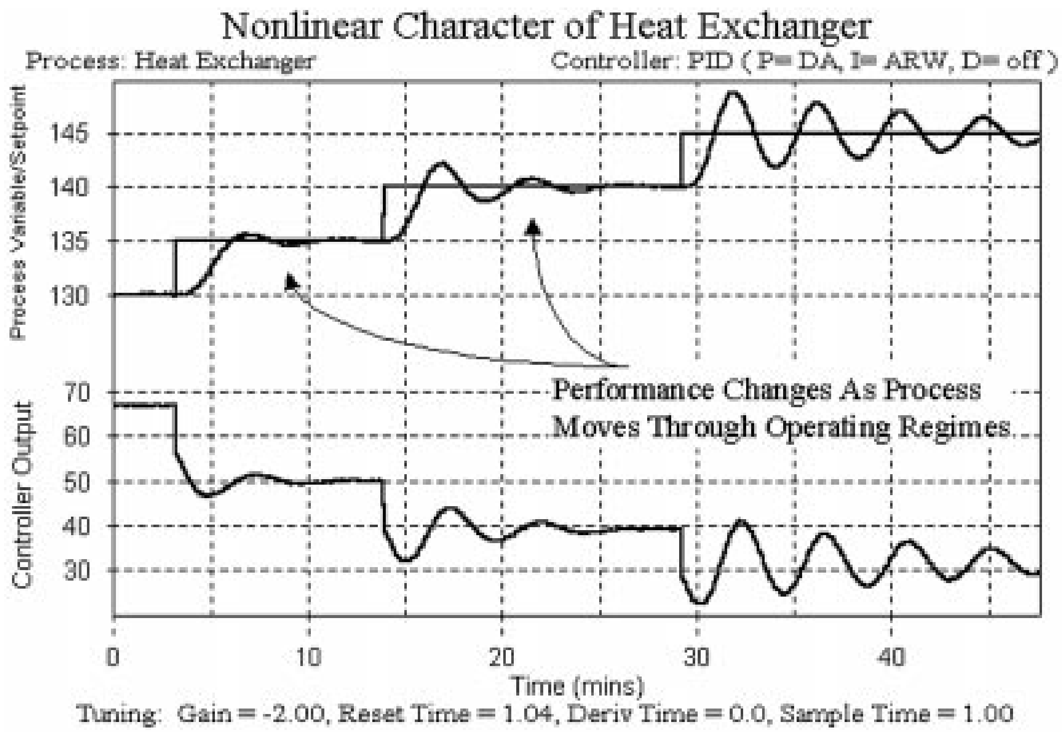

Fig. 5. Nonlinear behavior impact performance.

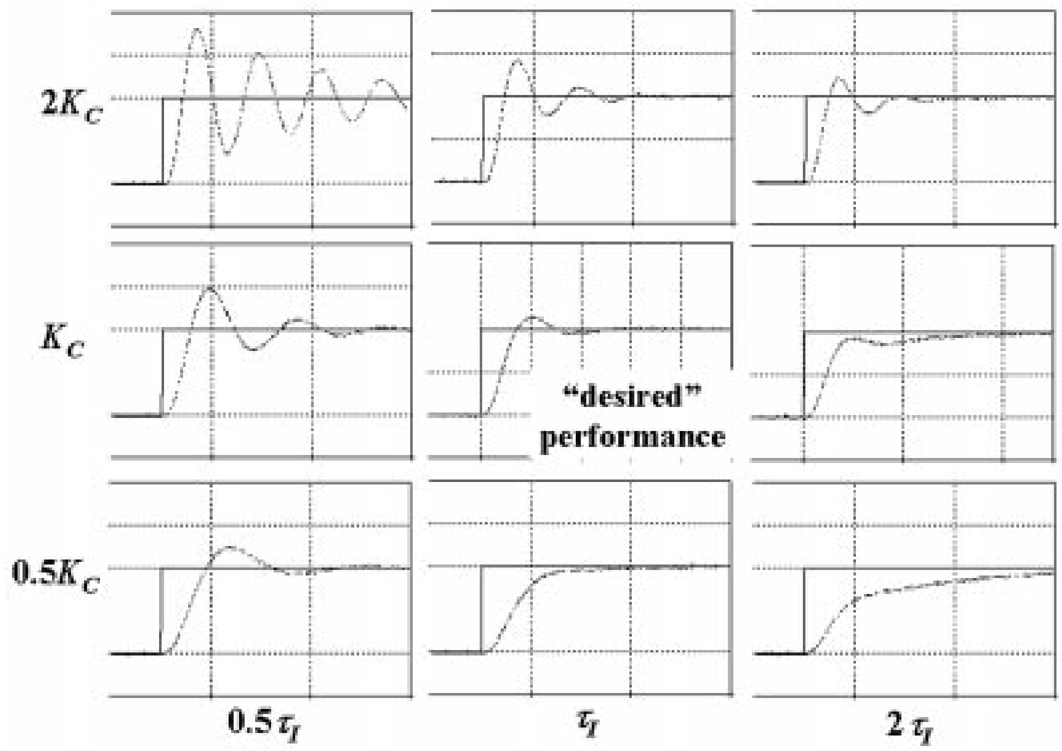

Fig. 6. PI controller tuning impacts performance.

measure process variable, causing the rise time and settling time to lengthen.

When noise is added to the measured process variable, students learn that derivative action causes the noise to be amplified and reflected in the controller output signal. Figure 8 shows this quite clearly with a side-by-side comparison of a PI and PID controller. For this comparison, the same amount of measurement noise was used throughout the experiment. This study helps students visualize that a PI controller is not impacted by noise while the derivative action of the PID controller reflects and amplifies it in the controller output signal.

Control loop interaction and decoupling

The graphic shown to the right in Fig. 9 is a binary distillation column based on the model of McCune and Gallier [9]. The column has two measured process variables and two manipulated variables. The reflux rate is used to control distillate purity and steam rate is used to control the purity of the bottoms stream.

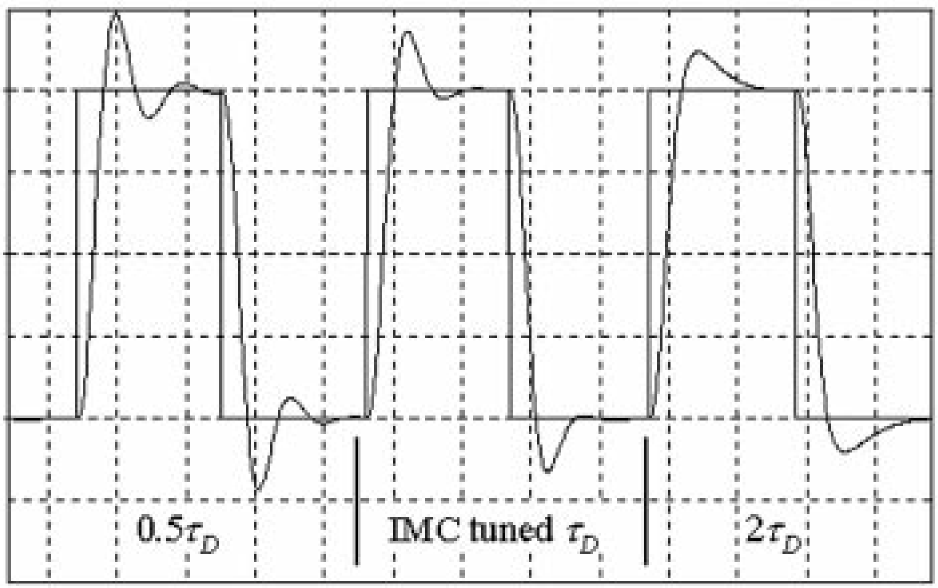

Fig. 7. Derivative action impacts oscillatory behavior.

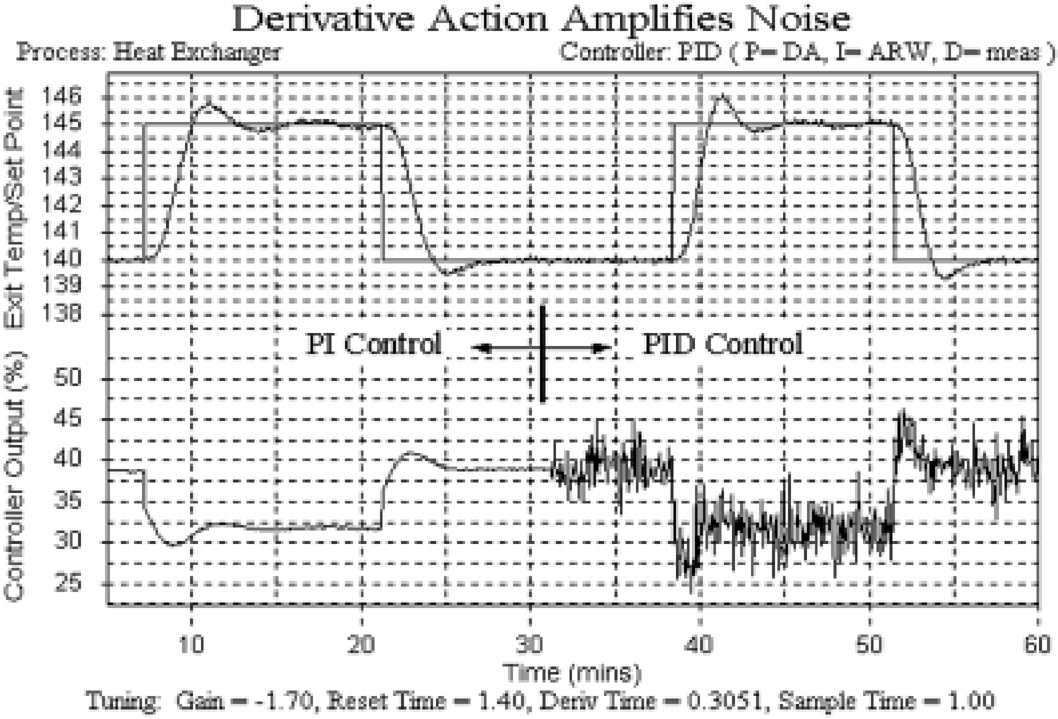

Fig. 8. Measurement noise is amplified and reflected in controller output signal with derivative action.

Students use this process to explore the interactions that can occur in multivariable applications. Control loop interactions occur because when the distillate purity out of the top of the column is too low, the top controller compensates by increasing the flow of cold reflux into the column. This increased reflux flow will indeed cause an increase in the distillate purity. However, the additional cold reflux will work its way down the column trays and eventually begin to cool the bottom of the column. This cooling causes the purity of the bottoms stream to move off its set point and produce a controller error.

The bottom controller compensates by increasing the flow of steam into the reboiler. This produces an increase in hot vapors traveling up the column, which eventually causes the top of the column to begin to heat up. The result is that the

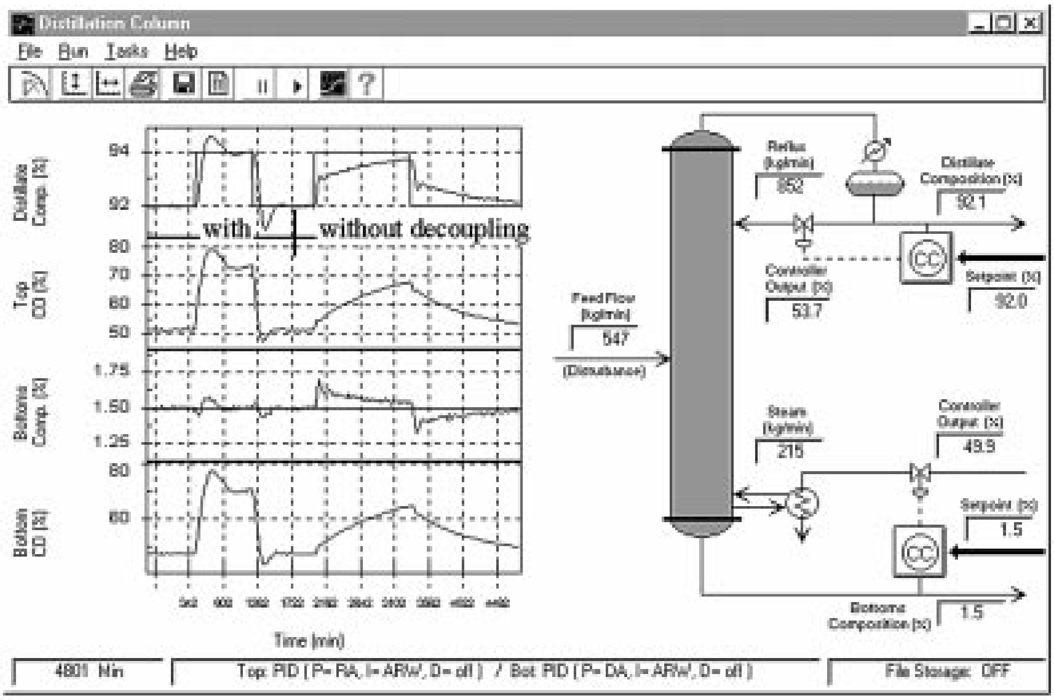

Fig. 9. Distillation column graphic.

distillate purity again becomes too low. In response, the top controller compensates by again increasing the flow of cold reflux into the column.

The strip charts to the right in Fig. 9 show the distillate composition responding to two set point steps. The controller 'fight' when no decouplers are used is shown on the right side of the strip charts. Controller interaction causes the set point response to be quite slow since both controllers are working at cross purposes.

Decouplers are feed forward elements where the measured disturbance is the controller output signal of another loop on the process. Two decouplers are required to compensate for loop interaction, one for each controller [6]. Like a feed forward element, each decoupler requires identification of a process and disturbance model. The left side of the strip charts in Fig. 9 shows that with decouplers in place, this loop interaction is dramatically reduced.

Students explore different controller modes, loop tunings, model structures and many other design issues. With two controllers and four models for complete decoupling, students also learn how important bookkeeping is to the control designer.

STUDENT PROJECT: REJECTING PROCESS DISTURBANCES

To illustrate how Control Station can be used in a more substantial project, we consider here the implementation of control strategies designed to reject disturbances that upsets steady process operation. The most popular strategies for improved disturbance rejection are cascade control and the feed forward with feedback trim architecture. Both strategies trade off additional complexity in the form of instrumentation and engineering time in return for a controller better able to reject the impact of disturbances on the measured process variable. Neither strategy benefits nor detracts from set point tracking performance.

This project focuses on Control Station's jacketed reactor process. Students begin by establishing a base case of disturbance rejection using a conventional single loop feedback controller. Then they detail the implementation of cascade and feed forward controllers and compare disturbance rejection performance and capability.

Base case—conventional feedback control

Figure 10 shows the screen display for the jacketed reactor process under single loop feedback control. The jacketed reactor simulation models a continuously stirred vessel in which an exothermic (heat producing) reaction occurs. Residence time is constant in this well mixed reactor, so the conversion of reactant feed to desired product can be inferred from the temperature of the reactor exit stream.

To control the reactor's exit stream temperature (the measured process variable), the vessel is enclosed with a jacket through which a cooling liquid flows. The controller manipulates a valve

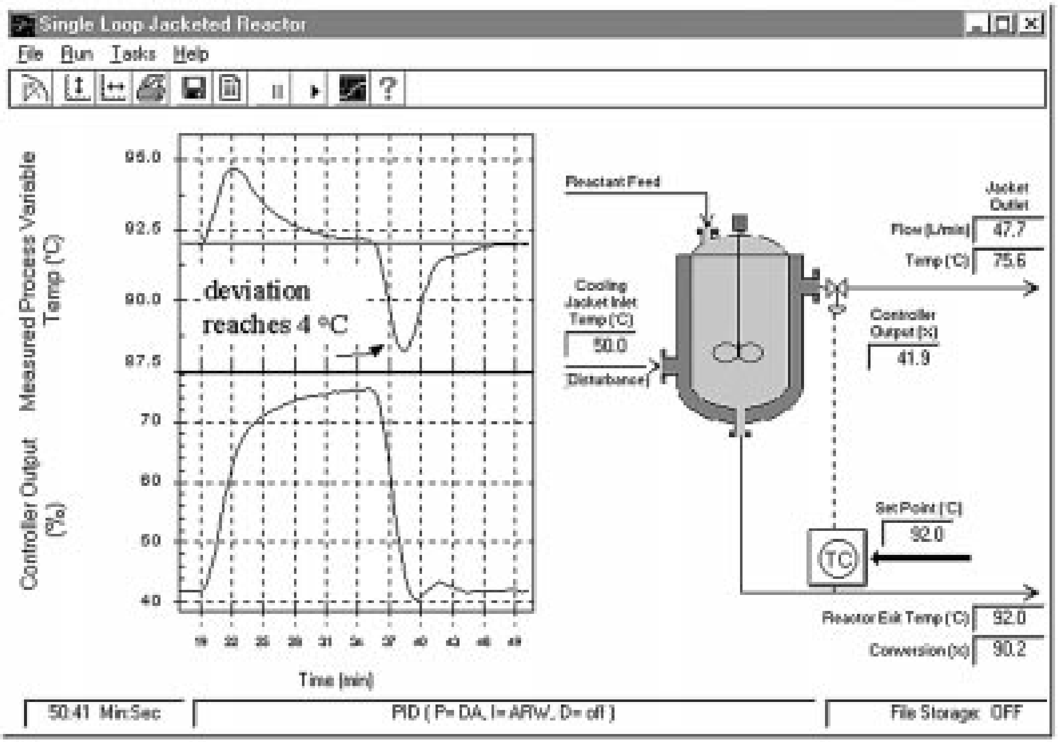

Fig. 10. Disturbance rejection in jacketed reactor with single loop feedback control architecture.

which adjusts the cooling jacket liquid flow rate. If the exit stream temperature (and thus conversion) is too high, the controller opens the valve. This increases the cooling liquid flow rate, which cools off the reactor and causes the heat producing reaction to slow. Ultimately, the measured temperature of the stream exiting the reactor decreases in response to the control action. The disturbance variable of interest for this process is the temperature of cooling liquid entering the jacket.

The control objective for this project is to maintain the reactor exit stream temperature at its set point while rejecting disturbances caused by changes in the temperature of the cooling liquid entering the jacket. Though the user is free to change operating conditions, the design level of operation for this investigation is a reactor exit stream temperature of 92°C. The cooling jacket's inlet temperature is normally 50°C but on occasion is known to unexpectedly spike as high as 60°C.

Following the P-only controller design procedure detailed above, the controller output is perturbed while in manual mode to cause the measured reactor exit stream temperature to exhibit a clear response that dominates any measurement noise. The resulting dynamic process data is read into the Control Station's *Design Tools* module where a FOPDT model is fit to the process data. *Design Tools* then uses the model parameters in tuning correlations to recommend PI controller tuning values.

With tuning values in hand, students return to the jacketed reactor process and enter them into the controller design menu. The disturbance rejection performance of the single loop PI controller is tested. As shown in the strip charts to the left in Fig. 10, the measured reactor exit stream temperature is initially steady at the design set point value of 92°C.

To test the controller, the cooling jacket's inlet temperature is stepped from its design value of 50°C up to 60°C and back again (this disturbance trace is not shown but can be seen using software plotting options available in Control Station). The single loop PI controller works to maintain an exit stream temperature near the constant set point of 92°C, but the disturbance transient produces deviations that reach about 4°C during the event as displayed in the figure.

Cascade control disturbance rejection

The open loop behavior of the cascade jacketed reactor is also identical to the single loop base case. The process graphic for the cascade controller architecture is shown to the right in Fig. 11. A cascade controller requires a *secondary* process variable for its design. The main process variable associated with original control objective is called the *primary* variable. This secondary process variable has specific requirements:

- it must be measurable with a sensor;
- the same valve used to manipulate the primary variable must also manipulate the secondary variable;
- the same disturbances that are of concern for the primary variable must also disrupt the secondary variable;

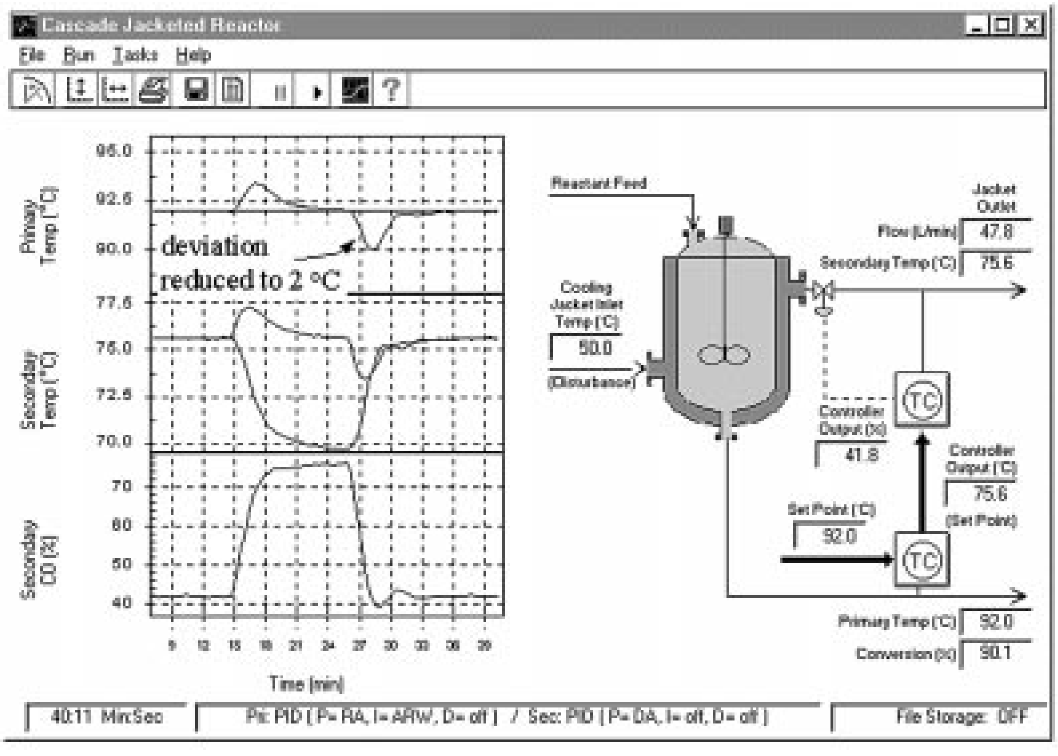

Fig. 11. Jacketed reactor process with cascade control architecture.

- the secondary variable must be *inside* the primary process variable, which means it responds well before the primary variable to disturbances and valve manipulations.

With a secondary process variable identified, a cascade architecture is constructed with the secondary loop literally nested inside the primary loop.

Because the control objective for this project is disturbance rejection, it is indeed appropriate to consider a cascade architecture. The primary process variable remains the reactor exit stream temperature. As shown in Fig. 11, the cooling jacket's outlet temperature is used as the secondary process variable. Hence, like all cascades, there are two measurement sensors, two controllers and one valve; the same valve as in the single loop case.

Again, the design level of operation is a reactor exit stream temperature of 92°C and the concern is the cooling jacket's inlet temperature, which is normally at 50°C but can spike as high as 60°C. The cascade design starts with the secondary controller. Set point tracking is the design objective for the secondary controller because its function is to track set point changes computed by the primary controller. The students follow the traditional approach of using a P-only controller because it provides a quick response and rapid settling of the secondary variable to set point changes. The primary loop remains in manual mode while the primary controller is being tuned. After testing, the secondary loop to ensure desirable performance, it is left in automatic and students turn their attention to the primary controller.

For a cascade architecture, the primary controller output is the set point of the secondary controller. In order to generate dynamic process data for the controller design, the secondary set point is perturbed and the response of the primary measured process variable is recorded. A PI controller is then tuned and tested following the procedures previously discussed.

The strip charts in Fig. 11 show the cascade performance for the same disturbance steps as used in the base case of Fig. 10. The cascade architecture performs better than the single loop case, reducing the reactor's exit temperature deviations to about 2°C during the disturbance rejection event. This improvement did not come free, however, as the cascade architecture requires an additional sensor, controller and tuning effort.

Feed forward disturbance rejection

A feed forward controller is an alternative architecture for improved disturbance rejection. Feed forward implementation requires the purchase and installation of a sensor to directly measure the offending disturbance, and the development and programming of a feed forward control element. This element is comprised of a linear disturbance model and a linear process model. The computation performed by the feed forward element may be thought of as a two step procedure:

1. The *disturbance* model receives the disturbance measurement and predicts an 'impact profile,' or when and by how much the measured process variable will be impacted.

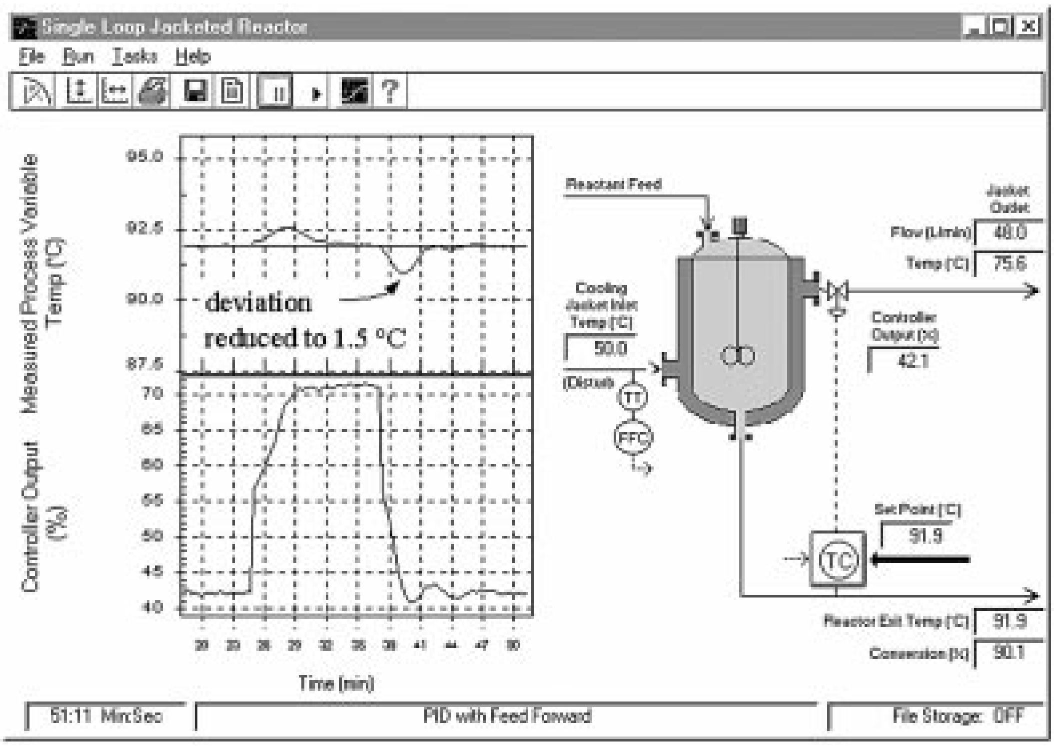

Fig. 12. Jacketed reactor with disturbance sensor for feed forward control.

2. Given this predicted sequence of disruption, the *process* model then back calculates a series of feed forward control actions that will exactly counteract the disturbance as it arrives so the measured process variable remains constant at its set point.

Implementation requires that the two models be programmed into the control computer. Linear models will never exactly describe the behavior of real nonlinear processes, so a feed forward element can dramatically reduce the impact of a disturbance but will never provide perfect disturbance rejection.

To account for model inaccuracies, the feed forward signal is combined with a traditional feedback control action to create a total controller output signal. The feedback controller provides *trim*. That is, it rejects those unmodeled portions of the measured disturbance that make it past the feed forward element and reach the measured process variable. The feedback controller also works to reject all other unmeasured disturbances affecting plant operation and provides set point tracking capabilities as needed.

The control objective for this study is the same as that used in the previous two investigations. To construct the feed forward controller, the inlet temperature of the cooling jacket (the disturbance) is measured with a sensor as shown in Fig. 12. The signal from this disturbance temperature sensor is sent to a feed forward element comprised of a process model and disturbance model.

The process model describes the controller output to measured process variable behavior. The same FOPDT model fit of the dynamic data for controller tuning can serve as the feed forward process model. Generating disturbance driven data can be problematic for real processes if the disturbance variable cannot be manipulated at will. Control Station permits such disturbance manipulations so students can explore and learn. Here, the cooling jacket's inlet temperature is stepped from the design value of 50°C, up to 60°C and back again. The resulting disturbance driven data is fit with a FOPDT model fit in *Design Tools* to yield the required feed forward disturbance model.

With a process and disturbance model identified, the students can implement the feed forward controller. This is achieved by using the convenient parameter entry forms in the controller design menu of the jacketed reactor simulation. The feed forward element is combined with the same PI feedback controller used in the base case to yield a feed forward with feedback trim architecture.

Figure 12 shows the disturbance rejection performance of this controller. As in the previous demonstrations, the exit stream of the reactor (measured process variable) is initialized at the design set point value of 92°C. To test the controller, the cooling jacket inlet temperature is stepped from its design value of 50°C up to 60°C and back again.

As shown in the figure, the feed forward with feedback trim architecture performs much better than the single loop base case and similar to the cascade controller in rejecting the disturbance. Specifically, while the reactor exit temperature deviations for the single loop PI controller range as high as 5°C, this advanced architecture limits the maximum deviation to about 1.5°C. Like the cascade architecture, the improved performance required an additional sensor, controller and tuning effort.

As shown in Fig. 12, the feed forward controller initiated rapid compensating controller action just after the disturbance event to minimize its impact on the measured process variable. Perfect disturbance rejection was not achieved because the FOPDT models only approximate the higher order and nonlinear behavior of the jacketed reactor process.

CONCLUDING REMARKS

Presented here are a few examples of the lessons and challenges that the Control Station training simulator can provide to students. While we do not believe a training simulator is better than or a replacement for real lab experiences, we do believe that Control Station provides students with a broad range of meaningful experiences in a safe and efficient fashion. These experiences can be obtained risk free and at minimal cost, enabling students to feel comfortable exploring nonstandard solutions at their desk. We also believe if a training simulator is properly designed, it can bridge the gap between textbook and laboratory, enabling significantly enhanced learning for process control theory and practice. If the reader would like to learn more, they are encouraged to contact Doug Cooper at cooper@engr.uconn.edu, or visit www.engr.uconn.edu/control.

Acknowledgements—The authors would like to acknowledge the contributions of David Stauffer to this effort.

REFERENCES

1. M. J. Rabins and C. E. Harriss, Jr., Controls, risk and educational responsibility: the ethical/professional links, *Control Eng. Practices*, **3** (1995) p. 711.
2. V. J. VanDoren, Simulation simplifies 'What-If' analysis, *Control Eng.*, **45** (1998) p. 68.
3. T. E. Marlin, Software laboratory for undergraduate process control education, *Computers & Chemical Eng.*, **20**, S1371 (1996).

4. C. M. Cooper, DIST: A computer supplement used in the teaching of a graduate distillation course, *Proc. 1991 Annual Conf. American Society for Engineering Education*, **2**, (1991) p. 1449.
5. D. E. Seborg, T. F. Edgar and D. A. Mellichamp, *Process Dynamics and Control*, Wiley, New York, NY (1989).
6. C. A. Smith and A. B. Corripio, *Principles and Practice of Automated Process Control*, Wiley, New York, NY (1985).
7. M. L. Luyben and W. L. Luyben, *Essentials of Process Control*, McGraw-Hill, New York, NY (1997).
8. B. A. Ogunnaike and W. H. Ray, *Process Dynamics, Modeling, and Control*, Oxford, New York, NY (1994).
9. L. C. McCune and P. W. Gallier, Digital Simulation: A Tool for Analysis and Design of Distillation Controls, *ISA Transcations*, **12**, (1973) p. 193.

Doug Cooper is Professor of Chemical Engineering the University of Connecticut. His research focuses on developing control methods that are both reliable and easy for practitioners to use. Education of students and training of industrial practitioners has assumed a central role in his work. Over the years he has studied the control of fluidized bed incineration, heat exchange, distillation, injection molding, and catalytic reactors.

Danielle Dougherty received a BS in chemical engineering from Widener University in 1997 and is currently working toward her Ph.D. in chemical engineering under the direction of Doug Cooper at the University of Connecticut. Her research thesis focuses on adaptive multivariable model predictive control.