APPLICATION OF AN INTERACTIVE ODE SIMULATION PROGRAM IN PROCESS CONTROL EDUCATION

N. Brauner, M. Shacham,[1] M.B. Cutlip[2]
Tel-Aviv University
Tel-Aviv, 69978, Israel

In a paper titled "Process Control Education in the Year 2000,"[1] strong emphasis was put on the importance of mathematical modeling and computer simulation with interactive graphics as key pedagogical tools in both the present and the future of process control education. Since computer simulation has been used in control education for at least twenty years now, it is valid to ask what has changed and what additional roles an interactive simulation package can play in process control education.

In the past the most commonly used packages have been control-oriented packages such as ACS[2] or industrial control systems.[3] These packages are appropriate for demonstrating the behavior of practical control systems and are quite suited for use as "add ons" for a traditional control course. A major deficiency, however, is that these programs behave as a black box, giving results when input is provided but hiding the mathematical model from the user.

There are now available some new interactive simulation packages which accept the mathematical model of the control system as input in addition to the numerical data of the process. The user must provide the model, thus creating the desirable connection between control theory and practical application. Using this type of package can become an integral part of the control course and not just an add-on as it has been in the past with the older packages.

In order to take full advantage of the many desirable capabilities of the new simulation tools, however, the content of the traditional undergraduate control course should be substantially revised. One of the needed revisions, for example, is a reduced emphasis on linear systems theory. Most process control textbooks were written before the advent of user-friendly, interactive simulation packages, and as a result many of them put too much emphasis on linear systems and linearization methods. Most current mathematical and control packages employ numerical solution methods which can solve simultaneous nonlinear ordinary differential equation (ODE) systems as easily as they solve linear ones. That means that the traditional dependence on linearization could and should be reevaluated and substantially reduced.

Another curriculum revision would be in the required use of block diagrams within the control package. Such diagrams were absolutely necessary when analog computers were used, and they can be very helpful in demonstrating the behavior of linear systems; but their importance should be carefully reevaluated in light of the new simulation packages. The differential equations (which are the basis for the block diagrams) can now be inserted directly into the simu-

***Neima Brauner** received her BSc and MSc from the Technion, Israel Institute of Technology, and her PhD from the University of Tel Aviv. She is currently Associate Professor in the Fluid Mechanics and Heat Transfer Department. She teaches courses in Mass and Heat Transfer and Process Control. Her main research interests include two-phase flows and transport phenomena in thin films.*

***Mordechai Shacham** is Professor and Head of the Chemical Engineering Department at the Ben Gurion University of the Negev, Beer Sheva, Israel. He received his BSc and DSc from the Technion, Israel Institute of Technology. His research interests include applied numerical methods, computer-aided instruction, chemical process simulation, design, and optimization, and expert systems.*

***Michael B. Cutlip** received his BChE and MS from The Ohio State University and his PhD from the University of Colorado. He has taught at the University of Connecticut for the last twenty-five years, serving as Department Head for nine years. His research interests include catalytic and electrochemical reaction engineering, and he is co-author of the POLYMATH numerical analysis software.*

[1] *Ben-Gurion University of the Negev, Beer-Sheva, 84105 Israel*
[2] *University of Connecticut, Storrs, CT 06269*

© *Copyright ChE Division of ASEE 1994*

lation program, and the required conversion to block diagrams becomes unnecessary.

Modifying and reorganizing an existing control course to embrace the new tools is an evolutionary process and can present an interesting challenge for the instructor. In this paper we will offer several practical examples for using an interactive simulation package in different sections of an undergraduate control course.

There are several interactive simulation packages which can be used as a learning tool in the control course, but it is not our intent to review all of them. We will demonstrate some applications using the POLYMATH software (which was developed by two of the authors, Shacham and Cutlip), but we want to emphasize that other software (such as the widely used MATLAB package) can also be used for the same purposes.

The POLYMATH software package was originally developed for the mainframe Plato education computer system.[4] The current version of POLYMATH (2.1.1.PC) is distributed by the CACHE (Computer Aids for Chemical Engineering Education) Corporation, a non-profit organization that disseminates educational computer programs to chemical engineering departments. This version runs on the IBM Personal Computer, PS/2, and most compatibles.

Various forms of POLYMATH have been in use for almost a decade in support of chemical engineering education. Some important features are:

- It is a general purpose program now in use in over one hundred chemical engineering departments. In several departments the students are introduced to POLYMATH in their first chemical engineering course, so that when they reach the control course it is a familiar calculational tool for them. Students can also put this software on their own personal computers for easy access and use.
- The user works directly with the model equations which provide a direct link between the physical phenomena and the control system. This is in contrast to many control-systems simulator programs where the user only provides parameters to "black box" models (such as ACS[2] or UC Online[5]) or the user is required to convert the equations into block diagrams prior to solution (such as Tutsim[6, p.532] or UCAN II[7]).
- Problem set-up, solution, and modification times are very short. This is especially important in educational use where a long wait for the result often discourages exploration and curiosity.

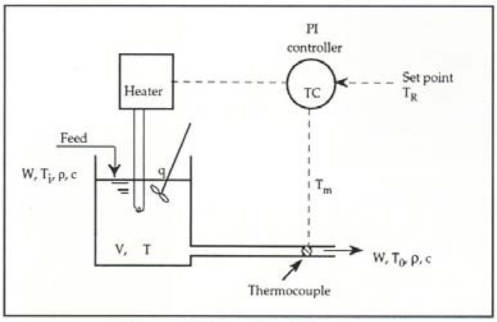


Figure 1. *Stirred tank heater*

EXAMPLE 1
Control of a Stirred Tank Heater

The dynamics and control of a stirred tank heater are discussed in several popular textbooks.[6,9] This simple system includes the stirred tank and a PI controller and is depicted in Figure 1.

The feed stream at constant rate (units: W kg/min) flows into a stirred tank equipped with a heating device; we want to heat this stream to a higher temperature T_R(°C). The outlet temperature is measured by a thermocouple, and the required heat supply, q, is adjusted by a PI temperature controller. The control objective is to maintain $T_o = T_R$ in the presence of a load due to an inlet temperature, T_i, which differs from the design value, T_{is}.

The model equations are:

Energy balance on the stirred tank:

$$\rho VC\frac{dT}{dt} = WC(T_i - T) + q; \quad T_o(0) = T_R \tag{1}$$

The thermocouple dynamics as described by first-order lag + dead time:

$$T_o(t) = T(t - \tau_d) \tag{2a}$$

$$\tau_m\frac{dT_m}{dt} + T_m = T_o; \quad T_m(0) = T_o(0) = T_R \tag{2b}$$

The heat supply as manipulated by the PI controller and actuator can be defined as

$$q(t) = q_s + K_c(T_R - T_m) + K_R\int_0^t (T_R - T_m)dt \tag{3a}$$

where q_s is the heat supply in design condition

$$q_s = WC(T_R - T_{is}) \tag{3b}$$

The numerical values of the parameters are

$$\rho VC = 4000.0 \text{ KJ/°C}$$
$$WC = 500 \text{ KJ/(min °C)}$$
$$T_{is} = 60°C$$
$$T_R = 80°C$$

This simple process can be used to demonstrate various concepts in different sections of the control course. Three possible applications are:

1. ***Closed loop dynamics***

Demonstrate stable and unstable regions for PI control using

K_c = 1000 - 10,000, K_r = 0 - 5,000 without (τ_m, τ_d = 0) and with (τ_m = 0 min, τ_d = 1) measurement deadtime.

2. *Controller Tuning*

Tune the PI controller using Aström's "ATV" method[8] and the Ziegler-Nichol's[6, p.233] settings.

3. *Reset Windup*

Investigate the controller behavior if the output from the heating tank is limited to twice the design value (q 20,000 kJ/s) and the inlet temperature reduced to half of its design value and then is restored to the steady state value after thirty minutes.

Solutions

Most of the equations needed to solve this problem can be typed directly into POLYMATH without any modification. But since POLYMATH is a general-purpose software program, it does not have functions which are specific to the control area, such as step, ramp, time delay, etc. Most of these functions can be generated, however. The generation of a step change at t = 1, for example, is accomplished by the equation

$$\text{step} = \frac{(t-1) + \text{abs}(t-1)}{2(t-1) + 0.000001} \tag{4}$$

This equation generates: step = 0 for t 1; step ~ 1 for t > 1. The value 0.000001 is added to the denominator in order to prevent division by zero when t = 1.

The integral of the error, required in Eq. (3a), is obtained by solving the differential equation

$$\frac{d(e_{sum})}{dt} = T_R - T_M; \quad t = 0,\ e_{sum} = 0 \tag{5}$$

Padé approximation[6, p.103] can be employed for representation of time delay. For instance, the first-order Padé approximation

$$e^{-\tau_d s} \cong (1 - \tau_d s/2)/(1 + \tau_d s/2)$$

yields in the time domain a first-order differential equation for the measured temperature

$$\frac{dT_o}{dt} = \left[T - T_o + \frac{\tau_d}{2}\left(\frac{dT}{dt}\right)\right]\frac{2}{\tau_d}; \qquad t = 0,\ T_o = T_R \tag{6}$$

Nonlinear and nonideal aspects can be demonstrated using the limits on the operation of the controller. The basic PI controller may require negative or inaccessibly high positive values of heat input, q, for some combinations of controller setting and magnitude of the step change in the input temperature. Limits can be put on the variables using equations similar to Eq. (4). For example, the operation

$$q_1 = \frac{q + \text{abs}(q)}{2} \tag{7}$$

gives q_1 = q if q 0; q_1 = 0 otherwise.

1. Closed loop dynamics of the stirred tank heater

Figure 2 shows the mathematical model, numerical constants, and initial values as they were entered into the POLYMATH ODE solver program for the case where τ_d = 1, K_c = 10,000, K_R = 0 (P-only controller) and a step change of -20°C in the feed is introduced at t = 1 sec. The options available to the user at this point are also shown: they include solution or modification of the problem, storage in a library, request for additional information regarding solution methods used, etc. If the "solve the problem" option is selected, the equations are numerically integrated, and the program selects either the explicit Euler or the 4th-order Runge-Kutta method, according to the required accuracy. For stiff systems, the user may ask to use the implicit Euler method. All of these methods include algorithms for estimating the integration error and changing step size if necessary. Solution times may vary from several seconds (for a PC without a math co-processor) to less than one second

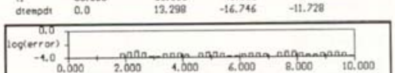
```
The equations:
    d(temp)/d(t)=(uc*(ti-temp)+q)/rhovc
    d(errsum)/d(t)=tr-tm
    d(tm)/d(t)=(temp-tm-(1/2)*dtempdt)*2/1
    uc=500
    rhovc=4000
    kc=10000
    tr=80
    kr=0
    q=10000+kc*(tr-tm)+kr*errsum
    step=((t-1)+abs(t-1))/(2*(t-1)+0.000001)
    ti=60-step*20
    dtempdt=(uc*(ti-temp)+q)/rhovc
Initial values:  t0= 0.0, temp0= 80.000, errsum0= 0.0, tm0= 80.000
Final value:  tf = 10.000

                         PROBLEM OPTIONS
                     ⇧F7 to solve this problem.
  ⇧←┘ to alter the problem.          F6 for helpful information
  F9 for file and library options.   ⇧F10 for the MAIN MENU.
```

Figure 2. *Mathematical model input to POLYMATH ODE solver for Example 1.*

Partial results

Variable	Initial value	Max. value	Min. value	Final value
t	0.0	10.000	0.0	10.000
temp	80.000	85.514	72.450	72.450
errsum	0.0	9.8250	-0.2722	5.5925
tm	80.000	85.892	73.696	84.069
uc	500.00	500.00	500.00	500.00
rhovc	4000.0	4000.0	4000.0	4000.0
kc	0.100×10^5	0.100×10^5	0.100×10^5	0.100×10^5
tr	80.000	80.000	80.000	80.000
kr	0.0	0.0	0.0	0.0
q	0.100×10^5	0.730×10^5	-0.489×10^5	-0.307×10^5
step	0.0	1.000	0.0	1.000
ti	60.000	60.000	40.000	40.000
dtempdt	0.0	13.298	-16.746	-11.728



F9 to display results. (q,t,d active)
⇧←┘ to make changes.   ⇧F8 for new problem.   F6 for help

Figure 3. *Partial results for Example 1.*

(for a computer with a co-processor).

Figure 3 shows a display of partial results which includes a table of initial, minimal, maximal, and final values of all the variables. Observing this table shows immediately that the model is unrealistic since the heat input, q, becomes negative at a particular point.

The bar chart near the bottom of the screen shown in Figure 3 gives the history of the integration error. The information in this chart can be used to assess the accuracy of the results and reduce the final time if more accurate results are needed. User options shown at the bottom include display as well as change, storage, and retrieval options. The display options include graphical ("g") or tabular ("t") presentation and output of the results to a DOS file ("d"). If graphical display of the temperature is selected, the graph shown in Figure 4a appears, indicating that for the specified parameter values the response is indeed unstable.

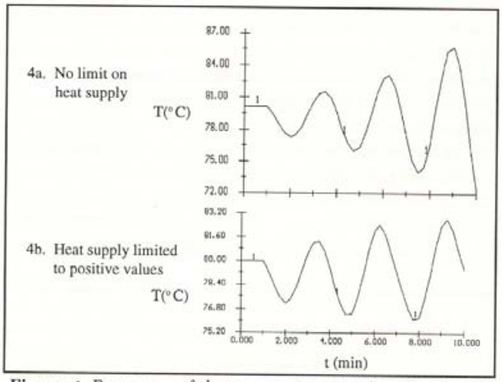


Figure 4. *Response of the temperature in the stirred tank to -20°C step change in feed temperature.*

TABLE 1
Controller Tuning Using Aström's "ATV" Method

```
(1)  d(temp)/d(t)=dtempdt
(2)  d(tm)/d(t)=(temp-tm-(tau/2)*dtempdt)*2/tau
(3)  wc=500
(4)  rhovc=4000
(5)  err=81-tm
(6)  h=4000
(7)  m1=(err+abs(err))/(2*err+0.000001)
(8)  m2=(err-abs(err))/(-2*err+0.000001)
(9)  q=10000+h*m1+h*m2
(10) dtempdt=(wc*(60-temp)+q)/rhovc
(11) tau=1
     t(0)=    0, temp(0)=    80, tm(0)=      80
     t(f)=   10
- This set of equations will generate the limit
  cycle in the measured temperature using the
  above method.
- To observe the response with proportional
  control when kc is set to the ultimate gain
  change equations 5-12 as follows:
(5)  d(errsum)/d(t)=tr-tm
(6)  kc=8450
(7)  tr=80
(8)  kr=0
(9)  q=10000+kc*(tr-tm)+kr*errsum
(10) ti=60-20
(11) dtempdt=(wc*(ti-temp)+q)/rhovc
(12) tau=1

  and set errsum(0)=0.
- To check the response with different kc and kr
  settings change equations (6) and (8).
```

The mathematical model can be made more realistic by introducing Eq. (7) into it to prevent the heat input from becoming negative. The growth rate of the oscillations is more moderate in this case, as shown in Figure 4b, but the system is still unstable.

This first part of the example problem can be used as an introductory example in an undergraduate process control course. Students can introduce changes to the system and observe for the first time the difference between systems with and without control, P vs. PI controller, effect of system parameters (time constants, dead time) and can familiarize themselves with the concepts of offset, stability, etc. Most of these concepts are shown in the textbooks, but the fact that the student can introduce the desired change and immediately observe the results can contribute considerably to an understanding of the material.

2. Controller tuning using Aström's "ATV"[8] method

When using this method, a relay of height, h, is inserted as a feedback controller. This nonlinear controller will cause the system to produce limit cycle of the controlled variable. The relay type change of the manipulated variable is achieved by two equations similar to Eq. (7) which generate (1,0) and (-1,0) values according to the sign of the error. The equations typed into POLYMATH for this assignment are shown in Table 1 for parameter values (t_d = 1; t_m = 0). A small change in the controller set-point is introduced (T_R is increased to 81°C). The behavior of the manipulated and controlled variable during the "ATV" procedure is shown in Figure 5. The period of the limit cycle is the ultimate period (P_u). Thus, the ultimate frequency is

$$\omega_u = \frac{2\pi}{P_u} \tag{8}$$

and the ultimate gain is

$$K_u = \frac{4h}{a\pi} \tag{9}$$

where a is the amplitude of the primary harmonic of the output.

The ultimate period and gain, as found above, can be used with the standard tuning formulas. The process response to a 33% step change in the inlet temperature obtained with a PI controller tuned using the Ziegler-Nichols controller settings[6,p.223] is shown in Figure 6.

3. Reset Windup

The model equations for the case where the output from the heater is limited and there is a substantial drop in the inlet temperature are very similar to the system shown in Figure 1, except that an equation similar to Eq. (7) has to be added to limit the heater's output.

The simulation results show that the PI controller on the heating coil will cause the heat output to reach its maximal value shortly after the inlet temperature is reduced. Since the heat output is not enough for reaching the set-point temperature, the error term in the integral part of the controller continues to increase until the inlet temperature is restored to its steady-state value. Because of this accumulated error term, the controller keeps the heat supply at its maximum long after the restoration of the inlet temperature. This causes the outlet temperature to reach a much higher value than the set point, as shown in Figure 7a.

Many industrial controllers have anti-windup provisions. This feature can be demonstrated in this example by switching off the error accumulation when the required heat supply exceeds the bounds. The outlet temperature response is shown in Figure 7b. In this case the outlet temperature will rapidly reach the set-point value, after the inlet temperature is restored to the steady-state value.

EXAMPLE 2

Dynamics of a Nonlinear Liquid-Level System

The liquid-level control system is frequently used in process control textbooks to demonstrate the difference between linear and nonlinear systems[i.e.,6,p.72] where emphasis is put on linearization of the nonlinear system around the steady state.

For this example, consider the system, shown in Figure 8, which consists of a tank of constant cross sectional area, A, into which a valve with flow resistance characteristics, $q_o(t) = ch^{1/2}$, is attached, where h is the liquid level in the tank and c is a constant. The flow rate into the tank, q, varies with time.

The following numerical and steady-state values are appropriate:

$$A = 1\,\text{ft}^2; \quad c = 20\,\text{ft}^{2.5}/\text{min}; \quad q_s = 60\,\text{cfm}; \quad h_s = 9\,\text{ft}$$

Using these numerical values, the response of the system to small and large (up to 90%) step changes in the inlet flowrate should be observed and the response using the nonlinear and linearized model should be compared.

Solution

The equation representing the liquid-level system is

$$q - ch^{1/2} = A\frac{dh}{dt} \tag{10}$$

The equation can be linearized around the steady state

$$(q - q_s) - \frac{h - h_s}{R_1} = A\frac{dh}{dt} \tag{11}$$

where $R_1 = 2h_s^{1/2}/c$.

Equations (10) and (11) can be introduced into the POLYMATH ODE solver with only slight modification. The response to reduction of the inlet flow to 10 cfm is shown in Figure 9.

We know that linearlzation is likely to yield close approximation of the dynamics of the system near the state around which the linearization is done. Indeed, when there is a 10% change in the inlet flow, responses of the nonlinear and linearized systems are very similar. The initial slope is the same, and the difference between the process gains that are calculated using the two models is only 5%. But using the linearized model far from the steady state may give very unreasonable results. If, for example, the tank's wall is much higher than the steady-state level and one tries to predict the maximal inlet flowrate that can be used without tank overflow, the difference between the predictions by the two models can be considerable. An even more interesting result occurs when the inlet flowrate is drastically reduced—the linearized model may predict a negative level at the new steady state, which is of course impossible. Such is the

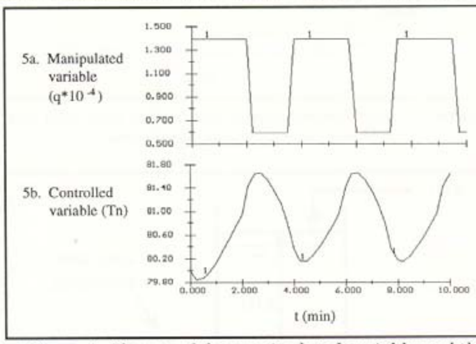


Figure 5. *Change of the manipulated variable and the controlled variable in "ATV" tuning.*

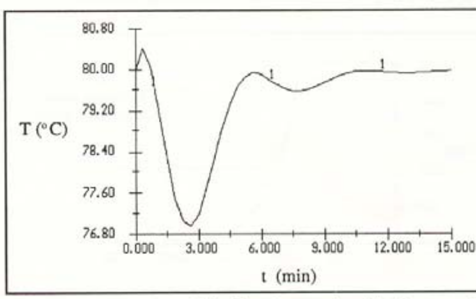


Figure 6. *Response of the heating tank with PI controller and Ziegler-Nichols settins.*

situation in Figure 9. The nonlinear model predicts the new steady-state level as 0.25 ft and the linearized model predicts -6 ft as the new level.

It should be noted that reducing the flowrate even further may cause difficulties with even the nonlinear model. Because of integration errors, h may become a small negative number, which makes it impossible to calculate the $h^{1/2}$ term. This can be prevented by putting a limit on h by applying an equation similar to Eq. (7). The same method can be used when the linearized model is solved by numerical simulation, but not when it is solved analytically.

A comparison of the nonlinear and linearized solutions by students should reinforce the following conclusions:

- It is important to remember the difference between a system which can be represented by a linear model and linearization of a nonlinear model. Linearization can represent the system well only near the point of linearization.
- It is always advisable to compare results from the nonlinear and linearized models in order to be able to appreciate the magnitude of error introduced by linearization.
- Results obtained from computer solution must always be carefully checked. Equations used outside the bounds of their validity, or numerical integration errors, may lead to incorrect or even absurd results.

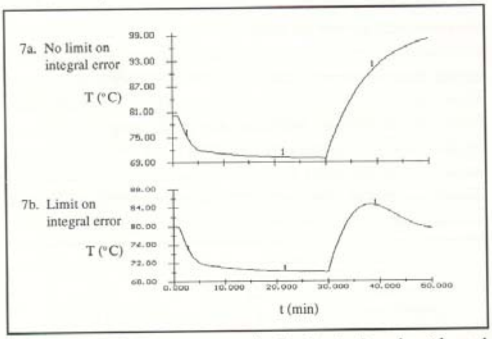


Figure 7. *Outlet temperature in the heated tank with and without limit on the integral error.*

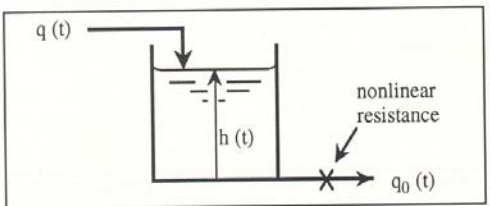


Figure 8. *Liquid-level system with nonlinear resistence.*

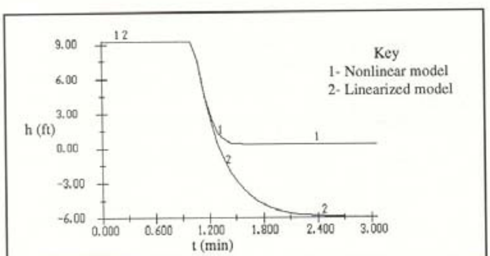


Figure 9. *Response of liquid level to reduction of the inlet flow rate to 10 cfm.*

CONCLUSIONS

We have demonstrated several interesting applications of an interactive ODE simulation program in this paper. Experience has shown the following important benefits of using such programs in process control:

1. *There are many aspects of dynamic process behavior that can be studied only by using nonlinear models that include, for example, limits on variables.*
2. *Interactive simulation complements analytical methods very nicely by ensuring better understanding and allowing more realistic problems to be considered.*
3. *The strengths and weaknesses of analytical solutions and numerical simulation can be clearly demonstrated. This is important in particular when linearizing nonlinear equations where the restrictions of the linearized model must be well understood.*

The examples and exercises given in Figure 1 and Table 1 can be put into immediate use in the classroom. Additional examples of applying an ODE solver for comparing analytical and numerical solutions and for more complex phenomenon could not be included in this paper because of space limitations. Information on these examples can be obtained from any one of the authors.

REFERENCES

1. Edgar, T.F., "Process Control Education in the Year 2000," *Chem. Eng. Ed.*, **24**, 72 (1990)
2. Koppel, L.B., and G.R. Sullivan, "Use of IBM's Advanced Control System in Undergraduate Process Control Education," *Chem. Eng. Ed.*, **20**, 70 (1986)
3. Buxton, B., "Impact of Packaged Software for Process Control and Chemical Engineering Education and Research," *Chem. Eng. Ed.*, **19**, 144 (1985)
4. Shacham, M., and M.B. Cutlip, "A Simulation Package for the PLATO System," *Computers and Chem. Eng.*, **6**, 209 (1982)
5. Foss, A.S., "UC ONLINE: Berkeley's Multiloop Computer Control Program," *Chem. Eng. Ed.*, **21**, 122 (1987)
6. Coughanowr, D.R., *Process Systems Analysis and Control*, McGraw-Hill Book Co., New York (1991)
7. Hittner, P.M., and D.B. Greenberg, "We Can Do Process Simulation: UCAN-II," *Chem. Eng. Ed.*, **14**, 138 (1980)
8. Aström and Hagglund, Proceedings of the 1983 IFAC Conference, San Francisco, CA (1983)
9. Smith, C.A., and A.B. Corripio, *Principles and Practice of Automatic Process Control*, John Wiley & Sons, New York (1985) ❐