UC ONLINE*

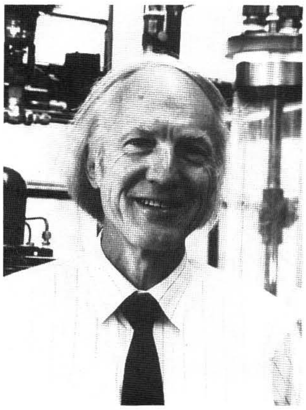
Berkeley's Multiloop Computer Control Program

ALAN S. FOSS
University of California
Berkeley, CA 94720

MULTILOOPS ABOUND. They are all around us. We invent these intricate control systems and apply them to distillation columns, fired heaters, reactors, steam systems. They are the "brains" and "nervous system" of chemical processes. Processes need them to operate and to operate safely.

That's news? Not at all. Everyone has known it for generations. Everyone, that is, except the students in our process control courses here in the States.**

That's got to change! And the change has to be made in the first course in process control—the *first* course, my colleagues, because there is seldom a second. Not everyone will agree with that of course, and those that do will ask, "How?"



Author **Alan Foss** writes that "after a quarter of a century of searching for ways to tell Californians about process control, I am still searching. The article published here reports one of the 'finds' along the way. Richer veins assuredly lie somewhere farther along the tunnel." Professor Foss came to the academic world after five years of industrial practice with the DuPont Company and studies at Worcester Polytechnic Institute and the University of Delaware.

HERE'S HOW

Imagine that you have a computer program that permits the user to configure any multiloop control system he desires for a particular process, say the system shown in Figure 1 for a distillation column. And suppose such a control system accepts process "measurements" from a dynamic simulation of the column and delivers its "commands" to that same simulation. With the keyboard command

YC, KP = 10, KI = 20, FREQ = 5

the user sets the proportional- and integral-gain pa-

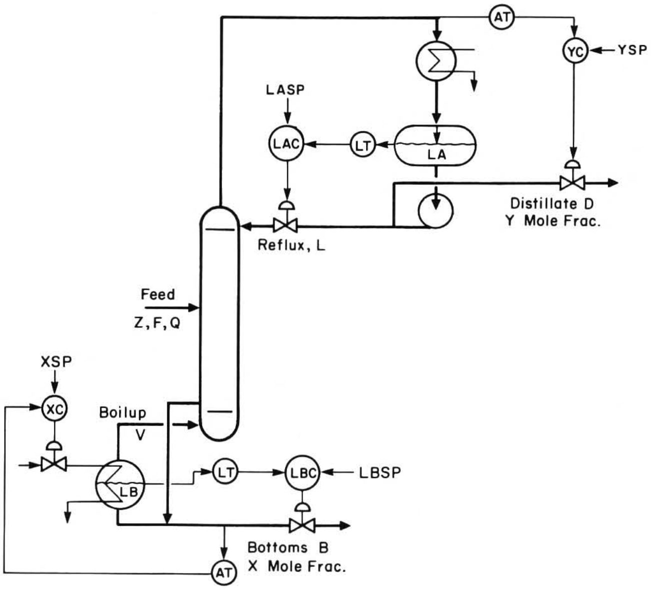


FIGURE 1. Control system for regulation of top and bottom product concentrations by manipulation of distillate and boilup flow rates. The relative gain for this control configuration is 0.68.

---

*©Copyright 1986 by P. H. Gusciora, C-H Mak, L. Poslavsky, and A. S. Foss

**There is probably a handful of departments to which this statement does not apply, and I know colleagues there will forgive this slight overstatement in the recognition that it is very close to the truth.

Imagine that you have a computer program that permits the user to configure any multiloop control system he desires for a particular process, say the system shown in Figure 1 for a distillation column. And suppose such a control system accepts process "measurements" from a dynamic simulation of the column . . .

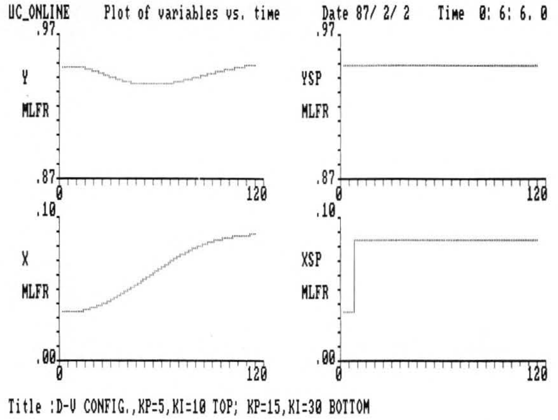


FIGURE 2. A page of the screen display of trends in top and bottom product concentrations. Response to a step increase in setpoint of bottom product concentration with control system of Figure 1.

rameters of the top-product concentration controller YC and the sample time to 5 seconds.

The command

$$\text{YC, CD} = \text{ON}$$

turns the controller to ON status and the user now has an operating process running in a real-time mode under the action of a simple control loop. He may commission the bottom-product concentration controller XC in the same way with the command

$$\text{XC, KP} = 10\text{, KI} = 20\text{, FREQ} = 5\text{, CD} = \text{ON}$$

at which time he will find himself in the land of multiloop control, control loop interactions, active input constraints, reset windup, and multiple alarms. Getting such a system to work when the column feed rate is varying will likely require a little tuning, all of which can be done "online" while the process and control system is running by simply typing commands for setting KP and KI similar to those above. The user knows how his system is shaping up by viewing periodically updated tabular data about the measured and manipulated variables and controller states or graphical displays of trends in any set of selected system variables, such as the group in Figure 2. The graphs shown there come from a full tray-by-tray calculation of a 39-tray column.

Now suppose the user is dissatisfied with the best performance he can squeeze out of this particular control system configuration and is curious about the claims found in Shinskey's book [4] for the superiority of the configuration shown in Figure 3. He is curious because the argument about the reduction in loop interaction given by Shinskey seems to be just what he is looking for, but he will have to see the performance improvement to believe it; Shinskey does not show performance. Some reconfiguration obviously needs to be done to convert the control system of Figure 1 to Figure 3.

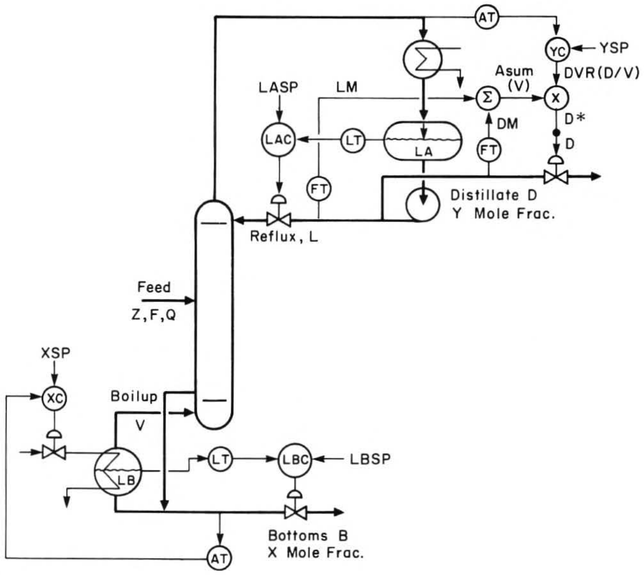


FIGURE 3. Control system using the ratio D/V and V as manipulated inputs. The relative gain is 1.92.

No problem, as we say. Turn off both controllers, define a new variable as the product of D/V and V in the overhead system, and redirect the output of controller YC to the multiplier. The sequence of commands for these changes, which can be made in a few minutes at the keyboard once one has decided what to try, are shown in Table 1. Turn both controllers to

TABLE 1
Keyboard Input to Convert the (D/V) configuration to (D/V,V)

```
VD,DVR,ASUM,D*
ASUM,AL=+,V1=DM,V2=LM,A0=1.0,A1=1.0,ML=0.0,MH=1200,UN=MPH,VA=600
DVR,MH=1.0,ML=0.0,SF=2,VA=.132
D*,AL=*,IO=24,V1=ASUM,V2=DVR,A0=1,A1=0,MH=300,ML=0,UN=MPH,VA=100.0
VREM,D
VD,D=24
D,IO=2,A0=0.0,A1=1.0,MH=300,ML=0.0,SF=2,AL=DA,UN=MPH,VA=100.0
YC,OD=DVR,OH=1.0,KP=.0083,KI=.0167,CD=ON
XC,CD=ON
```

Now, things are a little more involved than I have made out. I am sure that that is no news. There are a lot of details about maximum and minimum values of variables everywhere in the control system that need specification . . .

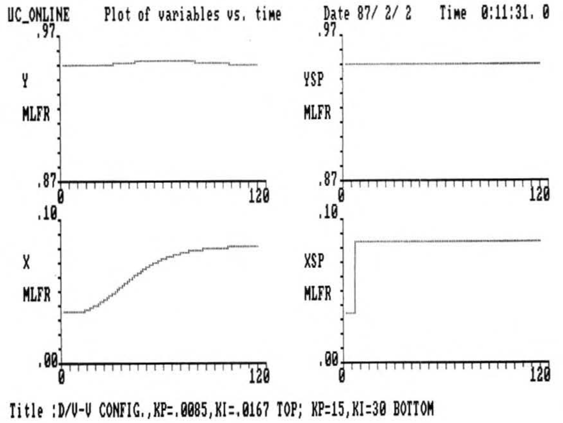


FIGURE 4. Top and bottom concentration responses to a step increase in setpoint of bottom product concentration. Control system of Figure 3.

ON and retune. Sure enough; performance is indeed better. The top product concentration is barely influenced by a change in the set point of the bottoms controller (see Figure 2 and 4).

"Professor, just a moment," you interject. "What about the need to recompile the control system program and to relink it to the process simulation?" That's not necessary these days. Everything about the control system is stored in tables. The program simply moves data from here to there when asked. All of that was worked out years ago by the computer scientists; we simply adopt the technique.

EASY CONFIGURATION

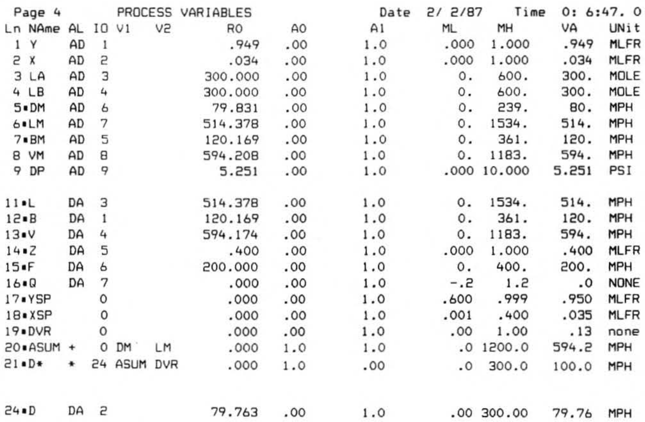
Now, things are a little more involved than I have made out. I am sure that that is no news either. There are a lot of details about maximum and minimum values of variables everywhere in the control system that need specification, else alarms announcing over-ranging would never reach the operator's eyes or ears. The declaration of such maximum and minimum values for all variables is an added chore for the user, to be sure, but is not the consideration of process limits important to safe process operation? It is, and we should expose students at least once to this component of the process control task.

The input of such information is easy. The screen display of the set of process variables for Shinskey's control configuration is shown in Figure 5. The variables named in the list are identified by the labels shown on the process diagram in Figure 3. All variables are free to be named by any 4-character symbol the user desires. So also may controllers be named. The command structure of ONLINE is designed so that the user need only type the name of the variable or controller followed by the attributes to effect whatever setting or change he desires to make in those attributes. Important information for the top product concentration y, for example, is that its input channel (IO) is l, that the type of algorithm (AL) is an analog-to-digital conversion (AD), and that the linear conversion of the concentration transducer signal to mole fraction has a slope of 1.0 (Al) and an intercept of zero (AO).

To set this information, the user simply types

$$y,\ IO = 1,\ AL = AD,\ Al = 1,\ AO = 0$$

This information appears immediately on the screen upon completion of typing this line. Error messages appear should there be any miskeying. Other variables in this example are seen to be identified as multipliers, summers, and digital-to-analog (DA) conversions. The use of input-output A/D and D/A data channels is a carry-over from a version of this program used with experimental apparatus. Their retention here serves two purposes: a decoupling of the simulation and control program is achieved, and students are made aware early that input and output channels must be specified when communicating with physical processes. Variables such as summers and multipliers have their inputs named under the VI and V2 columns. With such declarations, the user specifies that part of the control system configuration. The maximum and mininum values mentioned earlier are

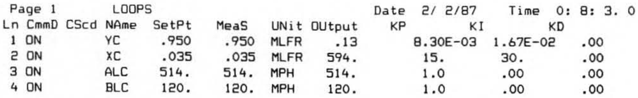


```
 Page 4        PROCESS VARIABLES                    Date  2/ 2/87    Time  0: 6:47. 0
Ln NAme AL  IO V1    V2         RO      AO          A1        ML     MH       VA   UNit
 1 Y    AD  1                 .949     .00         1.0      .000  1.000     .949  MLFR
 2 X    AD  2                 .034     .00         1.0      .000  1.000     .034  MLFR
 3 LA   AD  3              300.000     .00         1.0        0.   600.     300.  MOLE
 4 LB   AD  4              300.000     .00         1.0        0.   600.     300.  MOLE
 5▪DM   AD  6               79.831     .00         1.0        0.   239.      80.  MPH
 6▪LM   AD  7              514.378     .00         1.0        0.  1534.     514.  MPH
 7▪BM   AD  5              120.169     .00         1.0        0.   361.     120.  MPH
 8 VM   AD  8              594.208     .00         1.0        0.  1183.     594.  MPH
 9 DP   AD  9                5.251     .00         1.0      .000 10.000    5.251  PSI

11▪L    DA  3              514.378     .00         1.0        0.  1534.     514.  MPH
12▪B    DA  1              120.169     .00         1.0        0.   361.     120.  MPH
13▪V    DA  4              594.174     .00         1.0        0.  1183.     594.  MPH
14▪Z    DA  5                 .400     .00         1.0      .000  1.000     .400  MLFR
15▪F    DA  6              200.000     .00         1.0        0.   400.     200.  MPH
16▪Q    DA  7                 .000     .00         1.0       -.2    1.2       .0  NONE
17▪YSP      0                 .000     .00         1.0      .600   .999     .950  MLFR
18▪XSP      0                 .000     .00         1.0      .001   .400     .035  MLFR
19▪DVR      0                 .000     .00         1.0       .00   1.00      .13  none
20▪ASUM +   0 DM    LM        .000     1.0         1.0        .0 1200.0    594.2  MPH
21▪D*   *  24 ASUM  DVR       .000     1.0         .00        .0  300.0    100.0  MPH


24▪D    DA  2               79.763     .00         1.0       .00 300.00    79.76  MPH
```

FIGURE 5. Screen display of process variables for control system of Figure 3.

listed under the columns labeled MH (measurement high) and ML (measurement low). These limits represent the operable range of the measurement transducer.

The remaining part of the system configuration is established by naming the inputs and outputs of all the controllers. Figure 6 displays the video screen "page" that provides that information for each controller. The measurement source (MS), setpoint source (SS), and output destination (OD) for each controller is declared by variable name. These "connections" can be altered easily by the user through a few key strokes like those just mentioned. Such ease of reconfiguration is a feature indispensible to the efficient use of ONLINE in coursework. Students need to implement their conceptions in a matter of minutes, not days. Figure 6 also displays maximum and minimum declarations for the setpoint, the measured variable, and the controller output. The significance of these limits differs from those of the process variables just described. The limits on the measured variable, for example, are considered alarm limits, which when transgressed trigger an H or L message to the operator. The setpoint limits, normally set "inside" the limits on the measured variable, constrain the desired range of the controlled variable. Output limits reflect the rangeability of the process manipulatable variable driven by the controller. The ON-OFF-COND-FAIL status of the controller and the PID parameters are also displayed. All of the information for the process variables and controllers just described is "dynamic" and is updated in the data base and on the video display at intervals selectable by the user.



```
 Page 1        LOOPS                                     Date  2/ 2/87    Time  0: 8: 3. 0
Ln CmmD CScd NAme   SetPt    MeaS   UNit OUtput    KP          KI          KD
 1 ON         YC     .950    .950  MLFR    .13        8.30E-03  1.67E-02    .00
 2 ON         XC     .035    .035  MLFR   594.          15.        30.      .00
 3 ON         ALC    514.    514.  MPH    514.          1.0        .00      .00
 4 ON         BLC    120.    120.  MPH    120.          1.0        .00      .00
```

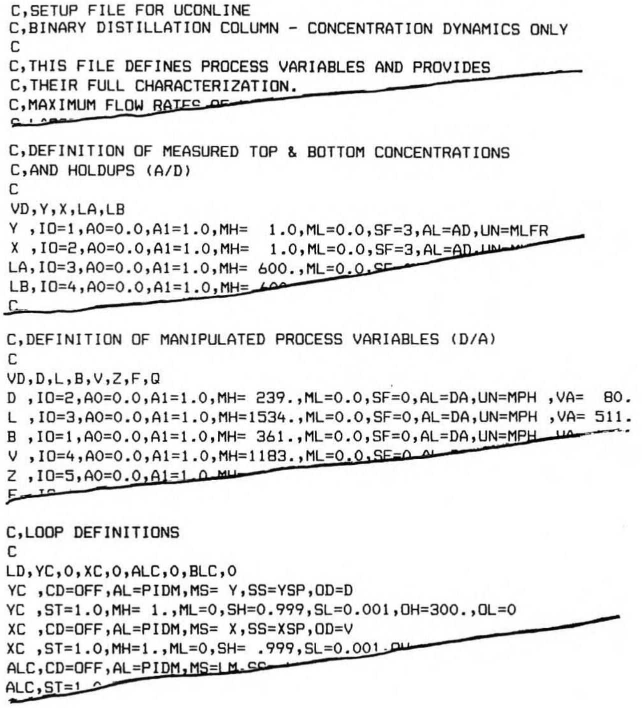
FIGURE 6. Screen display of controller page for system of Figure 3.

Suppose the excercise for the day concerns the tuning of a multiloop system already configured. The complete slate of information just described about system variables and controllers (excluding the PID parameters) can be prepared by the instructor ahead of time in a disk file. The user merely types

READ, SETUP

to load the entire configuration. Keyboard work is then necessary only for setting controller parameters. Portions of such a SETUP file are shown in Table 2. These files also serve as a permanent record and documentation of each control system.

TABLE 2
Key Portions of the File Used to Set Up the (D,V) Configuration for the Example of this paper



```
C,SETUP FILE FOR UCONLINE
C,BINARY DISTILLATION COLUMN - CONCENTRATION DYNAMICS ONLY
C
C,THIS FILE DEFINES PROCESS VARIABLES AND PROVIDES
C,THEIR FULL CHARACTERIZATION.
C,MAXIMUM FLOW RATES

C,DEFINITION OF MEASURED TOP & BOTTOM CONCENTRATIONS
C,AND HOLDUPS (A/D)
C
VD,Y,X,LA,LB
Y ,IO=1,AO=0.0,A1=1.0,MH=  1.0,ML=0.0,SF=3,AL=AD,UN=MLFR
X ,IO=2,AO=0.0,A1=1.0,MH=  1.0,ML=0.0,SF=3,AL=AD
LA,IO=3,AO=0.0,A1=1.0,MH= 600.,ML=0.0
LB,IO=4,AO=0.0,A1=1.0,MH=
C

C,DEFINITION OF MANIPULATED PROCESS VARIABLES (D/A)
C
VD,D,L,B,V,Z,F,Q
D ,IO=2,AO=0.0,A1=1.0,MH= 239.,ML=0.0,SF=0,AL=DA,UN=MPH ,VA=  80.
L ,IO=3,AO=0.0,A1=1.0,MH=1534.,ML=0.0,SF=0,AL=DA,UN=MPH ,VA= 511.
B ,IO=1,AO=0.0,A1=1.0,MH= 361.,ML=0.0,SF=0,AL=DA,UN=MPH
V ,IO=4,AO=0.0,A1=1.0,MH=1183.,ML=0.0
Z ,IO=5,AO=0.0,A1=1

C,LOOP DEFINITIONS
C
LD,YC,0,XC,0,ALC,0,BLC,0
YC ,CD=OFF,AL=PIDM,MS= Y,SS=YSP,OD=D
YC ,ST=1.0,MH= 1.,ML=0,SH=0.999,SL=0.001,OH=300.,OL=0
XC ,CD=OFF,AL=PIDM,MS= X,SS=XSP,OD=V
XC ,ST=1.0,MH=1.,ML=0,SH= .999,SL=0.001
ALC,CD=OFF,AL=PIDM,MS=LM
ALC,ST=1
```

If the exercise asks the user to invent his own control system, then the instructor merely lops off that segment of the SETUP file defining the control links, leaving only the process variables for loading. In Table 2, everything from the entry LD (loop definition) and below would be omitted in such a SETUP file. Creating that slate of information would constitute the exercise for the day. Alternatively, two SETUP files could have been prepared, one defining process variables only, the other the control links; the instructor supplies whichever file combination is appropriate.

Or "preanalysis" programs can be developed that prepare a complete SETUP file defining process variables. We have such a "front-end" program for binary distillation columns that calculates and displays the relative gains and steady-state operating conditions. A full set of information for the distillation simulation and the SETUP file for process variables, like that in Table 2, is written to disk upon user command and is read into ONLINE's data structure during the initialization phase. Such "front-end" programs, particularly for distillation columns, make it practical for students

Continued on page 154.

UC ONLINE

Continued from page 125.

to investigate control systems for a spectrum of situations.

SOPHISTICATED PID CONTROLLERS

The controllers implemented in ONLINE are sophisticated even though they excecute only a PID algorithm. Here are examples of what they can do.

Suppose in the cascade system shown in Figure 7, the operator decides to switch the flow controller to OFF. That action leaves the level controller ineffective because the link between the measured and manipulated variable has been "broken" It is essential that the level controller (and the process operator) be informed of any such break in the loop linkage whatever its genesis and that the level controller properly accommodate to the new situation. In ONLINE, information about the status of controllers and control system variables is communicated through the signal network in the direction opposite to that of the control signal propagation. In this particular circumstance, the receipt of information by the level controller about the OFF status of the flow controller will trigger an automatic switch in the level controller status from ON to CONDITIONAL. In the CONDITIONAL mode the level controller discontinues excecution of its algorithm until the flow controller status returns to ON. When the "downstream" link of a master controller is fully reestablished all the way to a valve or other process actuator, the controller status is switched automatically from CONDITIONAL to ON. This process of information transmission and status changes occurs no mater how deep the cascade or how branched the signal network [3]. We view these features as essential and are surprised to find them lacking in some commercially marketed packages.

A closely analogous situation arises when the valve driven by the flow controller is forced full open yet the level in the accumulator is still rising. In the event that integral action is used in the level controller, the integration should be suspended as soon as the valve saturation condition is known. That is accomplished by a digital counterpart of the feedback signal used in electronic controllers to achieve integral action. The controllers in ONLINE are equipped with a "TRACK" input variable (shown in Figure 7), which can be named as any control system variable previously defined. In the cascade system of Figure 7, the track variable is the output signal of the flow controller. As the controller module is processed, the value of the tracked variable is compared with high and low limits (such as the maximum and minimum flow through the valve in this example). If a transgression is detected, the integrating action is suspended as long as the saturated condition exists. Actually, the tracked variable is used to change the gains of all three modes to any values the user desires. Such a feature can also be used to fashion a 3-segment nonlinear controller. We use it in this way for gain scheduling in the temperature control of a laboratory heat exchanger over a range of flow rates. It adds a bit of spice to the life of students as they discover the interesting and useful embellishments that can be made to simple loops.

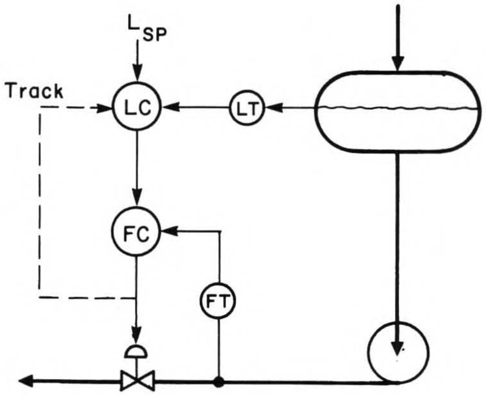


FIGURE 7. Cascade level control system with Track variable communicated to the master controller.

The use of a track variable is essential when high- or low-select operators are placed on the output of PI controllers because the integral calculation must be suspended for that controller that has been excluded from the active control link. Shinskey [6] gives several examples of this type. The track variable proves to be an extremely useful auxiliary input.

The user also has a choice of three different PID-controller configurations. The "classical" configuration employs the error between setpoint and measured signal in all three modes. Another uses the error in the integral mode only; the measured signal is used directly in the P and D modes. Such a configuration avoids "setpoint kick." A third uses error on P and I and measurement on D. All versions are implemented by the incremental algorithm and all employ a first-order filter on the derivative mode. The calculated output is checked against the maximum and minimum declared by the user and is not allowed to exceed those limits.

Other "signal transformations" available in ONLINE are: a lead-lag, addition, multiplication, and division of two signals, high and low selection of two signals, and square root evaluation. These operations permit the implementation of just about any multiloop

The availability of ONLINE has "shaken the earth" out here in Berkeley. Systems with similar capability are also causing a stir elsewhere. The possibilities for the treatment of process control problems beyond a stirred tank or a dead time are now without bound.

control system imaginable. Indeed, the limiting link in the control system configuration process seems to be one's imagination.

The number of controllers placed by designers on even simple and modest-sized processes can escalate to the point where tuning becomes a noisome chore, particularly when loops interact strongly. The incorporation of an automatic tuning procedure as one of the controller status states is an obvious way to relieve the user of such a task. We are working on it. The program structure of ONLINE permits the addition of such new features, new control algorithms, and new commands.

It is also obvious that "umbrella" procedures such as process optimization procedures and knowledge-based systems that monitor process operability could be interfaced with ONLINE. Those matters are also of pressing interest to chemical engineers, and our students could benefit from some practice on these topics.

UC ONLINE AT BERKELEY

Why use this "real-time" coupling of control system and process simulation to practice multiloop system operation? Why not instead use the "once-through" type of simulation starting from prescribed initial conditions and disturbances? The principal merit of a "real-time" system such as ONLINE is its interactive capabilities. The immediate observation of the cause-effect relation in complex processes as changes are made is a very effective teacher. Further, the interactive capability gives the student experience in running an operating process, which contrasts with the experience in calculating a "once-through" type of simulation. Making judgements about what should be done to halt a reactor runaway, for example, with information up to the moment, is distinctly different from making those judgements after the complete "once-through" response is available. "Once-through" calculations historically have been the unvaried fare of this subject. We now have a challenging new capability.

Why not use commercially available software packages for the teaching of multiloop control? Several are now available that run on microcomputer systems. Some of these systems, however, require special hardware. And some require "special" money. These commercial programs have been developed to meet the demands of the industrial workplace and understandably are much more sophisticated and complex than UC ONLINE. Those two attributes can be a liability for a university instructional environment, however, because there is more to master (by both student and instructor) and more time has to be invested in gaining that mastery. User manuals for commercial programs are very thick. That can easily deflect the focus of a course from one of learning principles and their application to one of searching for a route through the labyrinth of multiple screens, data files, and system conventions. UC ONLINE is not without these, but they were built keeping in mind the special needs of the chemical engineering student and the constraints imposed by the university instructional environment. Ten pages of description are all that's needed to inform students about UC ONLINE. They pick it up very fast.

The availability of ONLINE has "shaken the earth" out here in Berkeley. Systems with similar capability are also causing a stir elsewhere [1, 2]. The possibilities for the treatment of process control problems beyond a stirred tank or a dead time are now without bound. One can now ask students to invent those control systems that we have heretofore only talked about, to implement them, and to make them work. Attention can be focused on the synthesis of control systems addressing production rate control, control under constraints, local optimization, and variable structures. Interesting examples of problems in these areas are found in Shinskey's books [4, 5, 6], and UC ONLINE is up to handling all of them. With such capability we can now engage our students in an activity that, in the author's opinion, is the activity in which chemical engineers make their most significant contribution to process control systems: the invention and development of the control system configuration.

ACKNOWLEDGMENT

UC ONLINE was conceived and developed in 1984 by Paul H. Gusciora and Chi-Ho Mak, graduate and undergraduate students respectively. Their implementation was made in Fortran in the multitasking environment of the Data General RDOS and RTOS operating systems. Leonid Poslavsky, an undergraduate student, converted their version to the single-task version also in Fortran, described here; it runs on the IBM PC/XT and AT. The work of these young men holds my respect and admiration. The distillation simulation used as the example in this paper was developed by Professor Babu Joseph of Washington University while on sabbatical leave at Berkeley. Financial support for a portion of the conversion was provided by the chemical engineering department at Berkeley. The computing equipment used was donated by the IBM Corporation through the UC Berkeley/IBM joint project on distributed academic computing.

REFERENCES

1. Koppel, L. B., and G. R. Sullivan, "Use of IBM's Advanced Control System in Undergraduate Process Control Education," *CEE*, **20** (2), 70 (1986).
2. Lavie, R., Technion, Haifa, Personal communication, 1985.
3. Mak, C-H, and P. H. Gusciora, "ONLINE Control Program for Multiloop Control Systems." Internal report. Dept. Chem. Engr., Univ. of Calif., Berkeley, CA (1984).
4. Shinskey, F. G., *Distillation Control*, 2nd Ed. McGraw-Hill, New York, NY, 1984.
5. Shinskey, F. G., *Process Control Systems*, 2nd Ed. McGraw-Hill, New York, NY, 1979.
6. Shinskey, F. G., *Controlling Multivariable Processes*, Instrument Soc. Amer., Research Triangle Park, NC, 1981. ☐

MICROCOMPUTER GRAPHICS

Continued from page 121.

Subcooling and overheating of streams is allowed, but the user has the responsibility of ensuring the correct overall heat load.

A Help menu is brought up by special key F1, and gives information on the design menu options. It operates similarly to the data entry Help screens.

A minimum utility network for the four-stream problem of Figure 1b is shown in Figure 3. Seven units are used, which is two more than the absolute minimum. In Figure 4 the network has been evolved into a six-unit design with slightly increased utilities. The pinch lines have been removed, as the design violates the pinch decomposition.

In a design with more process streams and hence more units, the screen can become very busy. It may be necessary to do only part of the design at a time, as shown in Figure 5. For this nine-stream problem [14] only the hot end is shown, and the pinch lines moved far to the right.

A print-out of the network can be obtained at any time by pressing the <PrtSc> key.

SUMMARY

The HENS program relieves the student of a tedious target-setting hand calculation and allows rapid generation and change of networks in the graphical design stage. A major guideline during program development was that the program should allow the students to make the same mistakes as they could with pencil and paper. The objective of the program is to help students think about heat exchanger networks, not to think for them.

Student reception of the program has been good on limited exposure to fairly simple designs. Despite the on-line help material, several students had difficulty using the program efficiently. An in-class demonstration is planned for future classes that should alleviate this problem. Some improvements for the future might include allowing a split stream to be erased and allowing a design to be saved to disk, along with the problem data.

The program is available on disk at nominal cost from the author. Please indicate whether the AT&T version (640x400 screen) or the IBM version (640x200 screen) is wanted.

REFERENCES

1. Linnhoff, B. *et al.*, "User Guide on Process Integration for the Efficient Use of Energy," Inst. Chem. Engrs., Rugby, UK (1982).
2. Linnhoff, B., and J. A. Turner, "Heat Recovery Networks: New Insights Yield Big Savings," *Chem. Eng.*, (Nov. 2, 1981) p. 56.
3. Linnhoff, B. and E. Hindmarsh, "The Pinch Design Method for Heat Exchanger Networks," *Chem. Eng. Sci.* **38,** 745 (1983).
4. Linnhoff, B., and D. R. Vredeveld, "Pinch Technology Has Come of Age," *Chem. Eng. Prog.*, **80** (7), 31 (1984).
5. Challand, T. B., R. W. Colbert, and C. K. Venkatesh, "Computerized Heat Exchanger Networks," *Chem. Eng. Prog.* p. 65 (July, 1981).
6. Lipowicz, M., "Heat-Exchanger Software," *Chem. Eng.* (Aug. 4, 1986), p. 73.
7. Govind, R., D. Mocsny, P. Cosson, and J. Klei, "Exchanger Network Synthesis on a Microcomputer," *Hydrocarbon Processing*, p. 53 July, 1986.
8. SuperTarget™, Linhoff March, PO Box 2306, Leesburg, VA 22075.
9. Saboo, A. K., M. Morari, and R. D. Colberg, "RESHEX: An Interactive Software Package for the Synthesis and Analysis of Resilient Heat-Exchanger Networks: I. Program Description and Application," *Comp. Chem. Eng.* **10,** 577 (1986).
10. *ibid;* II. Discussion of Area Targeting and Network Synthesis Algorithms," *Comp. Chem. Eng.* **10,** 591 (1986).
11. Papoulias, S. A., and I. E. Grossmann, "A Structural Optimization Approach in Process Synthesis: II Heat Recovery Networks," *Comp. Chem. Eng.* **7,** 707 (1983).
12. Floudas, C. A., A. R. Ciric, and I. E. Grossmann, "Automatic Synthesis of Optimum Heat Exchanger Network Configurations,", *AIChE. J.* **32,** 276 (1986).
13. Wood, S., "Using Turbo Pascal™," Osborne McGraw-Hill, Berkeley, California (1986).
14. Tjoe, T. N., and B. Linnhoff, "Using Pinch Technology for Process Retrofit," *Chem. Eng.* (April 28, 1986) p. 47. ☐