Interactive Process Control Panel (a direct manipulation interface for process simulations)

by

Donald L. Miller

EDRC 05-25-88

Interactive Process Control Panel (a direct manipulation interface for process simulations)

Donald L. Millert

Engineering Design Research Center!
Carnegie Mellon University
Pittsburgh, PA 15212

ABSTRACT

The power and utility of dynamic process simulations can be enhanced by interactive interfaces which allow hands-on control of the model. Using modern software tools one can develop direct manipulation interfaces which attach to existing simulations. An example of such a system, the Interactive Process Control Panel, is presented.

June 1, 1988

t On sabbatical from Central Research and Development Department, E. I. du Pont de Nemours and Company Inc.

$ The Engineering Design Research Center is a National Science Foundation Engineering Research Center.

Interactive Process Control Panel (a direct manipulation interface for process simulations)

Donald L.Millert

Engineering Design Research Center^
Carnegie Mellon University
Pittsburgh, PA 15212

Introduction

Computer simulations of physical systems have become commonplace tools for engineers. Chemical engineers use process simulations for a variety of tasks ranging from process design to relief valve sizing. Simulation has changed significantly over the last two decades, beginning with programs punched on cards and submitted to mainframes, progressing through interactive computing on hardcopy terminals with line editors, and eventually to mini-computers which support graphics terminals and full screen editors. The most recent step in this evolution is the 32-bit desktop workstation which now offers sufficient computational power for many simulation tasks. At each step in this process the user interface has improved and using the computer to solve engineering problems has become easier, faster, and more enjoyable. The move to workstations with mouse based input and bit-mapped graphics screens offers exceptional promise in that it will allow us to redefine the traditional human-simulation interface. Over the next decade we may hope to change simulation in the way that word processing has changed document preparation.

The purpose of this note is to expound the power of hands-on process simulation and to encourage the chemical engineering community to exploit this technology. One good way to take advantage of this technology is to build advanced user interfaces into new simulation systems. An alternative is to develop interactive programs which are capable of serving as interfaces for existing simulations. As an example of this second approach, the *interactive process control panel,* a software system for hands-on control of dynamic simulations is presented here. This software represents a step towards the user interface of the next decade. I am happy to distribute the source code to the engineering community in the hope that this

t **On sabbatical from Central Research and Development Department, E. I. du Pont de Nemours and Company Inc.**

t **The Engineering Design Research Center is a National Science Foundation Engineering Research Center.**

system, or its derivatives, may grow into a general public domain tool. Parties interested in obtaining this software should contact the author.

Direct Manipulation and Simulations

Much recent research in man-machine interfaces is centered around *direct manipulation,* a term coined by Schneiderman [1]. With a direct manipulation interface a mouse is used to manipulate icons, windows, menus, etc. The Macintosh! operating system is a familiar example of direct manipulation, as are some of the popular spreadsheet systems. Hutchins et al. [2] provide an excellent discussion of direct manipulation. Readers who are interested in this area of research should also see diSessa [3]. Work by Hollan et al. [4,5] involving the *Steamer* system, is an impressive example of direct manipulation which will be of interest to chemical engineers. Steamer is a detailed, realistic control panel for a simulated steam propulsion plant.

The idea of an interactive interface for dynamic simulation is not new. Simulators have been used for training for quite some time, perhaps the most sophisticated examples being flight simulators used for pilot training. Many of these systems provide a realistic replica of the plant control room (or cockpit) with actual control hardware connected to a simulation of the process. The PICONS and G.2* systems provide similar capabilities in software and are designed to integrate Artificial Intelligence techniques with conventional process control hardware.

The Interactive Process Control Panel

The *interactive proces control panel* is not a simulator, but an *interface* for control of existing dynamic chemical process simulations. It is modeled after control panels found in chemical plant control rooms and is easily connected to sequential modular process simulations. Functioning graphical images of controllers and valves are manipulated using the mouse. The system has the following features:

- Images of PI controllers appear on the bit-mapped screen. These objects look and operate like real controllers and can be tuned or operated in manual or automatic mode using the mouse.
- With the mouse, the user can alter the controller parameters while the simulation is running. In addition to tuning parameters, the user can change the range of input and

t Apple Computer Corp.
***t* Gigamos Corp.**
• Gensym Corp.

output variables, effectively changing instrumentation or pump and valve capacities.

- The system is easily integrated with many existing simulations.
- New panels are constructed by selecting icons from a menu. When an object type is selected, a window appears presenting the properties to be filled in. When these properties are assigned, the object is added to the panel.
- The system is compatible with most procedural languages. Sequential modular simulations written in FORTRAN, C and Pascal have been interfaced using this tool.

The program alternates between calling the simulation and the window manager, and the system is easily integrated with sequential modular simulations in which time monotonically increases. It may be possible to use this system with equation based simulators, although it has not been done. With variable step size integrators the apparent speed of the simulation will change with step size, a minor inconvenience.

This system utilizes SunView† graphical objects (sliders, buttons, etc.) and thus requires a Sun workstation with the SunView windowing environment. A port of this system to X-Windows [6] may be done in the future. The control panel software is written in C [7] and makes use of data abstraction and computational objects. For an excellent review of object oriented programming see Winston and Horn [8]. Other languages, notably LISP and Smalltalk, offer much better facilities for object oriented programming. C offers some advantages over LISP in speed and ease of interfacing with FORTRAN, which remains a common language used in process simulation. If a major revision of this system is done (e.g. a port to X-Windows), C+ + [9] is a likely choice of language. The object oriented concept of classes (implemented in C+ +) may become a standard feature of C at some point in the future [10].

Using the Interactive Process Control Panel

This section describes the use of the interactive process control panel. As an example, consider the water-methanol distillation tower shown in Figure 1. The workstation screen in Figure 2 displays the control panel for the simulation of this tower. The top left and right windows are part of the control panel. The bottom left window contains strip charting which is part of the tower simulation program. The main simulation control panel is positioned along the top of the left window and has buttons which control the model execution: *abort, end, save panel, off* and *on*. The rest of this window contains objects for control of the distillation tower, two *valves*, followed by a *controller*, then an *indicator*, and three more *controllers*.

† SunView is a trademark of Sun Microsystems Corp.

The right window contains two *profile* objects which display the methanol composition and liquid flow rates throughout the tower.

Four types of objects are available; *valves* which take simple input from the user (e.g. setting feed flow), *controllers* which emulate PI controllers, *indicators* which display current values of process variables (these are just a digital version of dials), and *profiles* which use stacks of display bars to indicate profiles of variables (e.g. temperature along a reactor or distillation tower). The two valves in the left window allow the user to to set feed flow and composition. The indicator shows the value of the condensate flow. The four controllers regulate reflux, distillate, tails, and steam flow to control the top and bottom compositions and column inventory.

With the mouse the user can move the sliding bar on the valves adjusting the variable associated with the valve. Each controller has three sliding bars, *input, setpoint,* and *output.* The input slider continuously indicates the value of the controlled variable. The setpoint bar may be adjusted by the user. When in manual mode, the user controls the output bar using the mouse, in auto mode the controller manipulates this bar and hence the value of the manipulated variable. Controller mode is changed by pushing the man or auto buttons. The reset button removes controller windup, pushing it has the same effect as changing the mode from manual and back to automatic.

Both valves and controllers have tune buttons which allow on-line changes to their attributes. Pushing the tune button on a valve or controller causes an editing window to appear. With the mouse the user can choose a slot in this window, e.g. proportional band for a controller, and enter a new value. If the computer has trouble reading a data field the system identifies the errant slot. The maximum and minimum of the profiles can also be changed by typing in new values and pushing the read buttons in the profile windows. Indicators have no attributes and thus cannot be changed as the simulation runs.

Running a simulation, such as one shown in Figure 2, is quite unlike running a batch simulation. The user can introduce upsets, e.g. changing feed rate or composition, or the top composition setpoint. As the control system responds the user can take preemptive action, putting a control loop in manual and opening or closing a valve. The response of the process is available for immediate observation. When the reflux flow is increased a wave of increased liquid flow is seen moving down the column. One particularly interesting exercise is to begin the simulation with the distillation tower full of water running at minimal boilup and total reflux. At this point, startup of the column can be simulated with the gradual introduction of feed, and manual control up to the point that various control loops can be placed in automatic mode. Design of automatic startup procedures is greatly facilitiated by the ability to take action (which may be easily logged for later examination) during critical process transients.

Building an Interface

This section briefly outlines the methodology for attaching a sequential modular process simulation to the control panel. The model is first partitioned into three subroutines; startrun, endrun, and simulate. Startrun and endrun are called only at the beginning and end of the run and perform whatever initialization and file 10 are desired before and after simulation runs. Simulate makes one integration pass through the simulation and returns. In practice, simulate is usually just the existing batch run code with the statements causing the time loop commented out.

Next, the user must lay out a control panel for the process. For this purpose a mouse based tool called *builder* is provided. Builder presents the existing panel board (if one exists) along with several icons (valve, controller, indicator, or profile). When the user selects an icon an editor window appears presenting the appropriate attributes to be filled in. When the input is complete the user pushes a button causing the new object to be added to the panel.

The routines startrun, endrun, and simulate are compiled and linked with the panel library. One line of code per object is required to complete the interface. The C versions of the functions are shown below, FORTRAN callable versions exist as well.

C Versions of Control Panel Functions

```
float get_valve(n)               /* returns the current value of valve n */
  int n;
float pi_controller(n,input,DT)  /* returns output for controller n */
  int n;
  float input;                   /* input variable value */
  float DT;                      /* time step in hours */
set_indicator(n,value)           /* set the value of indicator n to value */
  int n;
  float value;                   /* new value for indicator n */
load_profile_yalues(n,v)         /* display the vector v in profile n */
  int n;
  float *v;                      /* vector of values for profile */
```

Discussion

Interactive simulation will change the way we interact with our computer models in the next decade. To take full advantage of this powerful concept we must develop tools which provide this type of simulation. Interactive simulation holds particular advantages in the following areas of chemical engineering simulation.

- Dynamic process simulation development
- Control system design

- Startup and shutdown procedures, batch processing
- Providing hands-on experience for students in process control

The software system presented here is useful for a variety of tasks including process simulation development and control system design. At du Pont, we have used it as an aid in control system design for several small processes, (e.g. an extruder system). We also have one complex simulation which is controlled by the panel, consisting of over 50 unit operations and 30 active controllers spread over five panel boards. One area where the present system is clearly useful is providing students with hands-on experience in process control. Changing tuning parameters and watching a system respond can enhance students' understanding of process and control system behavior.

The software tool described here is greatly lacking in many respects, but provides an example of better things to come. The list of improvements which could be made is long and includes a better indicator (a real dial), and a variety of different controller types and algorithms. Portability to other hardware and software environments (X-windows) would dramatically increase the system's utility. Other obvious improvements include the ability to alter not only the tuning, but control structure without programming changes, and a mouse based strip charting system. The primary reason for offering the software to the engineering community is the hope that it will evolve into, or at least inspire the development of, a general public domain tool.

References

1. B. Schneiderman, "The future of interactive systems and the emergence of direct manipulation," *Behavior and Information Technology,* 1, 237-256 (1982).
2. E. L. Hutchins, J. D. Hollan, and D. A. Norman, *User Centered System Design,* 87-124, Lawrence Erlbaum Associates, Hillsdale, NJ (1986).
3. A. A. diSessa, *User Centered System Design,* 125-152, Lawrence Erlbaum Associates, Hillsdale, NJ(1986).
4. J. D. Hollan, A. L. Stevens, and M. D. Williams, "STEAMER: An advanced computer assisted instruction system for propulsion engineering," *Proceedings of Summer Computer Simulation Conference,* 400-404 (1980).
5. J. D. Hollan, E. L. Hutchins, and L. Weitzman, "STEAMER: An interactive inspectable simulation-based training system," *A. I. Magazine,* 15-27 (Summer 1984).
6. R. W. Scheifler and J. Gettys, "The X-Window System"," *JACM Transactions of Graphics,* 5, 79(1987).
7. B. W. Kernighan and D. M. Ritchie, *The C Programming Language,* Prentice-Hall, Englewood Cliffs, NJ (1978).
8. P. H. Winston and B. K. P. Horn, *LISP,* 239, Addison-Wesley, Reading, MA (1984).
9. B. Stroustrup, *The* C+ + *Programming Language,* Addison-Wesely, Reading, MA (1986).
10. Narain Gehani, *C: An Advanced Introduction,* 153, Computer Science Press (1985).

Figure Captions

Figure 1

Process control diagram for water-methanol distillation tower. The tower has a steam reboiler and total condenser. Level indication is provided for the reboiler and reflux drum. Top and bottom compositions are analyzed on line.

Figure 2

Workstation screen showing control panel for distillation tower in Figure 1. The buttons at the top left of the screen control the model execution. Below these are the objects which control process parameters in the simulation. Two valves control feed flow and composition. Four controllers manipulate distillate, reflux, tails, and steam flow to control composition and inventory in the column. An indicator, below the first controller, displays the condensate flow. The strip charting is part of the process simulation.

Figure 1

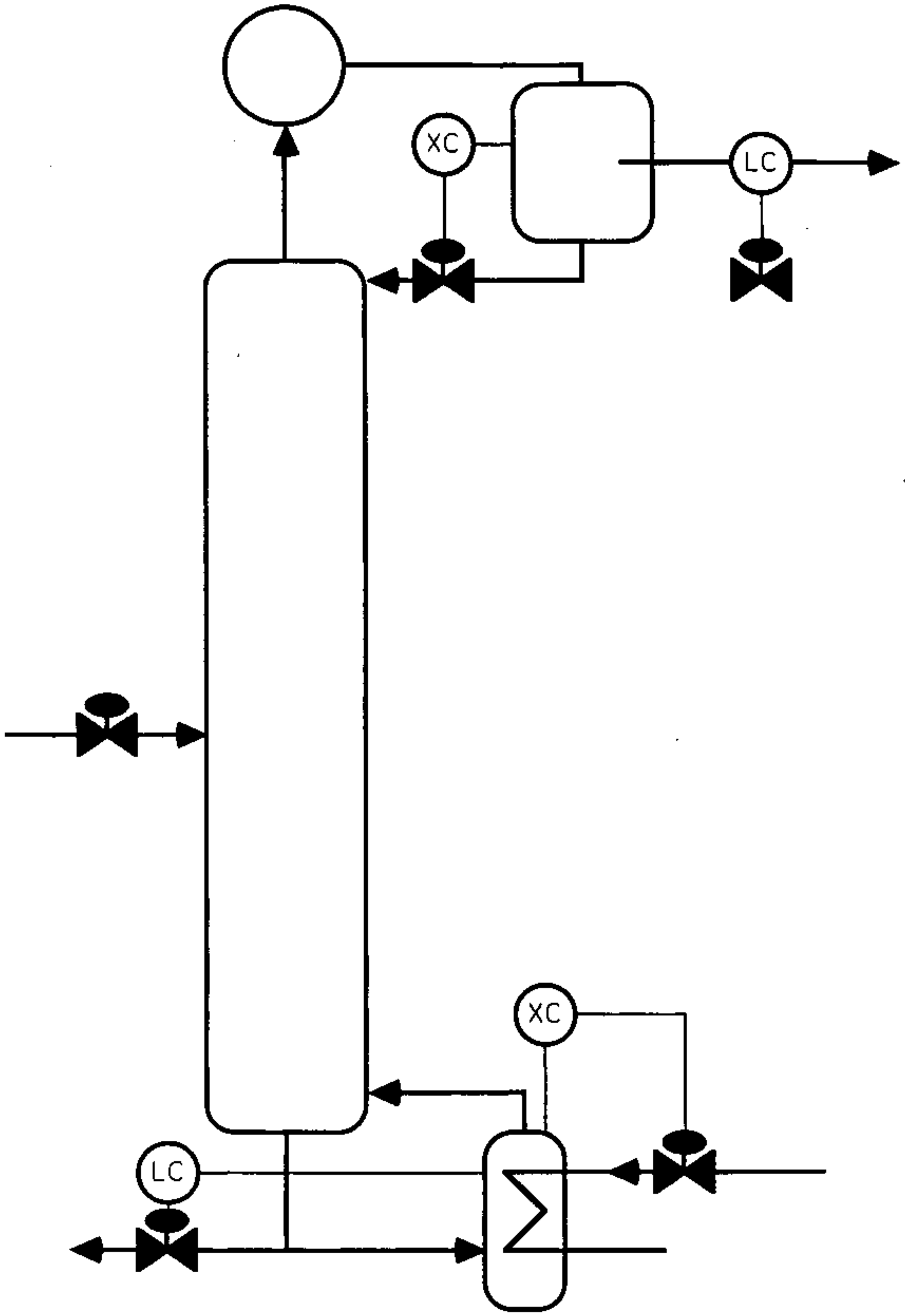
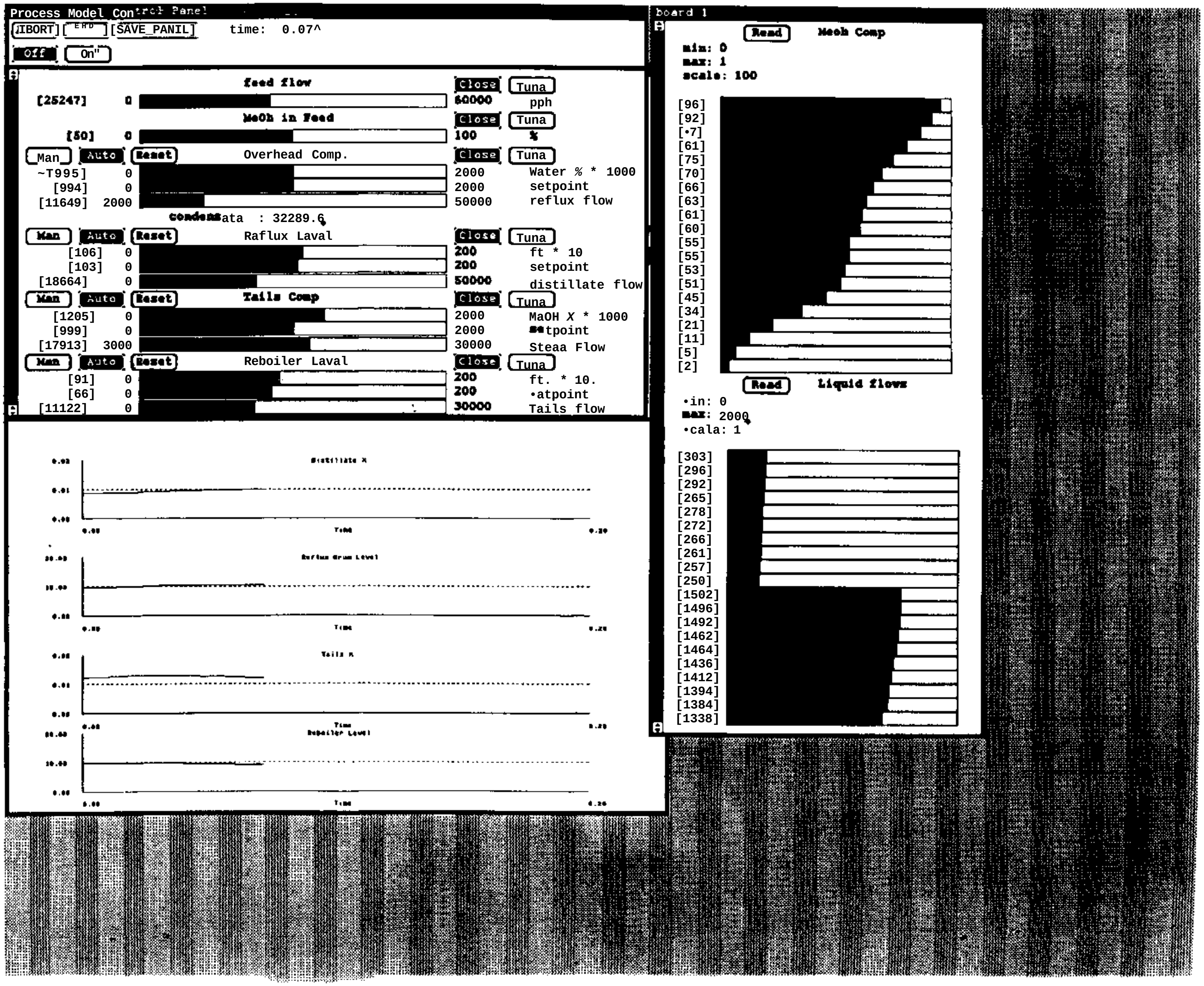


Figure 2



```
Process Model Control Panel
[ABORT] [END] [SAVE_PANIL]     time:  0.07^
[Off] [On"]

                              feed flow              [Close] [Tuna]
[25247]    0                                          60000   pph
                              MeOh in Feed           [Close] [Tuna]
   [50]    0                                          100     %
[Man] [Auto] [Reset]          Overhead Comp.         [Close] [Tuna]
~T995]     0                                          2000    Water % * 1000
 [994]     0                                          2000    setpoint
[11649] 2000                                          50000   reflux flow
          condensata  : 32289.6
[Man] [Auto] [Reset]          Raflux Laval           [Close] [Tuna]
 [106]     0                                          200     ft * 10
 [103]     0                                          200     setpoint
[18664]    0                                          50000   distillate flow
[Man] [Auto] [Reset]          Tails Comp             [Close] [Tuna]
[1205]     0                                          2000    MaOH X * 1000
 [999]     0                                          2000    setpoint
[17913] 3000                                          30000   Steaa Flow
[Man] [Auto] [Reset]          Reboiler Laval         [Close] [Tuna]
  [91]     0                                          200     ft. * 10.
  [66]     0                                          200     •atpoint
[11122]    0                                          30000   Tails flow
```

```
board 1
[Read]   Meoh Comp
min: 0
max: 1
scale: 100

[96]
[92]
[•7]
[61]
[75]
[70]
[66]
[63]
[61]
[60]
[55]
[55]
[53]
[51]
[45]
[34]
[21]
[11]
[5]
[2]

[Read]   Liquid flows
•in: 0
max: 2000
•cala: 1

[303]
[296]
[292]
[265]
[278]
[272]
[266]
[261]
[257]
[250]
[1502]
[1496]
[1492]
[1462]
[1464]
[1436]
[1412]
[1394]
[1384]
[1338]
```