Publication No. 02-103-132

MODELING OF THE PHOSPHATE BENEFICIATION PROCESS AS A FORERUNNER TO ADAPTIVE CONTROL

Prepared by
BCD Technologies & University of Alabama

under a grant sponsored by

FIPR
Florida Institute of
Phosphate Research

November 1996

The Florida Institute of Phosphate Research was created in 1978 by the Florida Legislature (Chapter 378.10 1, Florida Statutes) and empowered to conduct research supportive to the responsible development of the state's phosphate resources. The Institute has targeted areas of research responsibility. These are: reclamation alternatives in mining and processing, including wetlands reclamation, phosphogypsum storage areas and phosphatic clay containment areas; methods for more efficient, economical and environmentally balanced phosphate recovery and processing; disposal and utilization of phosphatic clay; and environmental effects involving the health and welfare of the people, including those effects related to radiation and water consumption.

FIPR is located in Polk County, in the heart of the central Florida phosphate district. The Institute seeks to serve as an information center on phosphate-related topics and welcomes information requests made in person, by mail, or by telephone.

Research Staff

Executive Director
Paul R. Clifford

Research Directors

G. Michael Lloyd Jr.	**-Chemical Processing**
Jinrong P. Zhang	**-Mining & Beneficiation**
Steven G. Richardson	**-Reclamation**
Gordon D. Nifong	**-Environmental Services**

Florida Institute of Phosphate Research
1855 West Main Street
Bartow, Florida 33830
(941) 534-7160
Fax:(941)534-7165
http://www.fipr.state.fl.us

MODELING OF THE PHOSPHATE BENEFICIATION PROCESS AS A FORERUNNER TO ADAPTIVE CONTROL

FINAL REPORT

Bernard J. Scheiner[1], Charles L. Karr[2], and Donald A. Stanley[1]
Principal Investigators

[1] BCD TECHNOLOGIES
Tuscaloosa, AL 35406

[2] THE UNIVERSITY OF ALABAMA
Tuscaloosa, AL 35487-0280

Prepared for
FLORIDA INSTITUTE OF PHOSPHATE RESEARCH
1855 West Main Street
Bartow, Florida 33830

Contract Manager: Patrick Zhang

November 1996

DISCLAIMER

The contents of this report are reproduced herein as received from the contractor.

The opinions, findings and conclusions expressed herein are not necessarily those of the Florida Institute of Phosphate Research, nor does mention of company names or products constitute endorsement by the Florida Institute of Phosphate Research.

PERSPECTIVE

Patrick Zhang, Research Director - Beneficiation & Mining

It is a ever present need for the phosphate industry to continually improve productivity, resources utilization and overall efficiency. One of the most effective approaches to achieve a significant improvement in efficiency is the real-time control of each of the unit operations. Process control systems routinely used in mineral processing were born with the invention of the microprocessor. It is expected that control systems of future will continue to develop towards user friendly computer based systems which use powerful software analysis and control techniques.

Process control in phosphate beneficiation plants is still very primitive even though the industry ages over a century. Three major factors have attributed to this situation: lack of on-stream, chemical/mineralogical analysis, lack of reliable unit operation models, and dramatic fluctuation of the flotation feed. FIPR has directed extensive research efforts at evaluating on-stream analytical techniques, two of which look promising. The next logical step would be to develop process models in such a way that the instant on-line analytical information may be utilized by a process control system.

An optimizing, adaptive process controller combines the process control capabilities of fuzzy logic with the adaptive capabilities of genetic algorithms. Fuzzy logic is a technique through which a computer controller is implemented using subjective, linguistic concepts similar to human decision making. Genetic algorithms are search algorithms based on processes found in nature. They rapidly locate near-optimal solutions to difficult search and optimization problems. The adaptive control technology has two major advantages over the conventional stabilized control strategy: It has the learning capability, and does not require a lot of sensory information.

This project involved three major components. In phase 1, six specific models of unit processes were selected and appropriate software were developed. These models were tested using both literature data and plant data collected as the project progressed. Phase 2 combined unit process models to form circuit models. The final phase evaluated the models by comparing the predictions of the model with plant data.

Another product of the project is a graphical user interface with the following capabilities: 1) to arrange icons on a desktop computer to build a visual representation of a processing plant, 2) to tune a process model against a list of targets, and 3) to evaluate the effectiveness of a tuned model.

TABLE OF CONTENTS

List of Figures

EXECUTIVE SUMMARY

Phosphate flotation plants in the United States are not currently operating at peak performance; this inadequacy must be rectified if America is to continue competing in the world market. The biggest improvements in the efficiency of existing phosphate processing plants in the near future are likely to be made by improving process control. Many of the most effective process control systems currently being developed and implemented in industry are model-following control systems. Development of these model-following control systems require fast and accurate computer models of the system to be controlled. This project was undertaken to provide a suite of computer models that can be used to develop intelligent, adaptive controllers for phosphate flotation systems.

The computer modeling software was written in the C++ computer language and is object-oriented for maintainability. The system includes a graphical user interface so that the user does not have to be intimately familiar with the computer modeling methods, search algorithms, and error reduction techniques used in the models. The modeling software and the proposed control software are both very complex. A blackboard system was developed to manage the complexity. This blackboard system coordinates the efforts of the various software modules.

Six computer models have been developed to simulate a phosphate flotation system. These models have been tested on plant data obtained from the Florida phosphate industry. The six computer models include both first principle and empirical models. First principle models are based on the actual physics of the system being modeled. They allow the user to investigate the effects of altering parameters outside of the region for which data is available. Empirical models are not necessarily concerned with the physics of the system being modeled. They are concerned only with accurately modeling the response of the system.

The first computer model discussed is a first principle model originally developed by Jordan and Spears (1990). This model is in effect a standard model for flotation in conventional cells because it is based on a few decades of world-wide research. This computer model uses particle sizes and densities, bubble sizes, relative velocities, and induction time to compute three probabilities generally associated with flotation: (1) the probability of a bubble-particle collision, (2) the probability of a particle adhering to a bubble, and (3) the probability of a particle remaining attached to a bubble. These probabilities are used in conjunction with two additional measured probabilities to compute an expected recovery.

The second and third computer models are statistical models. These models utilize various statistical tools to analyze the data obtained from the flotation system. These tools include regression analysis, factor analysis, cluster analysis, and projection to latent structures. The statistical models provide very little insight into the physics of the flotation system. However, they are very useful in exploring the structure of data. They can identify different physical mechanisms contributing to the data elements and allow insight into the extent and nature of the error associated with the data.

The fourth computer model is a neural network model. Neural networks are approximate models of the human brain in which computational elements called neurons are placed in a network of weighted connections called synapses. These networks are then trained on existing data to accurately reproduce the response of the system. Neural networks produce very powerful and very flexible models. Unfortunately, they are the ultimate “black box”; they give no indication why a particular input produces a given output.

The fifth computer model is based on fuzzy logic and genetic algorithms. Fuzzy logic is a technique that allows computers to manipulate subjective, linguistic concepts like those commonly used in human decision-making. Genetic algorithms are search algorithms based on the mechanics of genetics. They are robust algorithms that have been used successfully to solve a wide range of problems. Together fuzzy logic and genetic algorithms can accurately model phosphate flotation plants from relevant plant data. Although the performance of this model is similar to that of the neural network model, it is not a black box. The fuzzy model gives a clear explanation of its reasoning in the form of linguistic rules.

The last computer model is a rate constant model. The rate constant model, like the first principles model, consists of a collection of simple first-order differential equations. But instead of basing the computation of the rate constants for the differential equations on a model of the underlying physical processes, the rate constant model fits them statistically from plant operating data. The only independent variables in this model's differential equations are the concentrations of mineral species in the feed slurry. This model requires the plant data observations to be partitioned into clusters, each representing the same kind of operating conditions. Rate constants are then computed for each cluster. Thus, the rate constant model consists of several sets of rate constants along with a rule for choosing the appropriate set for any given operating state for the plant.

During the development of the models it became apparent that all of the parameters which are important for the system were not being measured. Unmeasured factors such as clay content of the material going to the conditioner and/or particle size dramatically affect the short range response of the system. For example, for an identical recovery, fatty acid requirements may vary by a factor of two. BCD researchers have developed a model following control system based on fuzzy logic and genetic algorithms which can overcome a partial lack of sensory data from the system. This method has been tested on helicopter flight control, the formation of hexamine, and pH control by titration. Based on this experience we believe that the six models can be used to model existing plants from operational data. If the data is inadequate for the control methodology, the models can indicate precisely what additional data is needed.

The programs which implement the various models and any additional control software used to control the plant or a model of the plant are of necessity complex. In addition, the programs rely on a large number of concepts which may be unfamiliar to most users. For this reason, we have developed a prototype interface which stands between the user and the program complexity. The interface can be used to design and test circuits using the models embedded in the interface.

MODELING OF THE PHOSPHATE BENEFICIATION PROCESS AS A FORERUNNER TO ADAPTIVE CONTROL

1 ABSTRACT

A first principles model of froth flotation and five empirical models based on rate constants, statistical theory, expert systems, fuzzy logic, neural networks, and regression were developed. A software library contains the computer routines necessary to implement a blackboard system used to coordinate the various software components needed to represent a phosphate processing plant. Software has been developed for a user friendly graphical interface needed to combine various unit process models into a model of a specific plant.

2 BACKGROUND

The objective of this project was to develop and apply an object oriented mineral processing software library initially conceived at the U.S. Bureau of Mines, Tuscaloosa Research Center to modeling phosphate flotation. In a few cases the appropriate components of the software library were available from the Bureau of Mines. In many other cases new system components were developed, tested, and implemented by personnel now associated with BCD Technologies.

Programs implementing a number of unit process models can become very complex, especially when a circuit or an entire plant is being modeled. Computer blackboard systems originally developed for the military are well suited for handling such complexity. The software components for implementing a blackboard system have been developed and assembled in the software library.

The initial military blackboard system was developed to allow computers to understand and respond to speech for a 1,000 word vocabulary (Hayes-Roth, 1985 and 1993). The blackboard system analyzed the signal from a microphone, recognized syllables, words, and then the meaning of sentences in order to search for and find a requested library reference. This required that a large amount of diverse knowledge be brought to bear on a complex task. The blackboard system designed for speech understanding proved to be applicable for a variety of other difficult problems. One significant use of a blackboard system is as a pilot's advisor. Modern fighter aircraft have a variety of sensors that provide more information than a pilot can easily assimilate. A blackboard based information manager has been developed that can recognize information relevant to a current situation and present a clear concise summary to the pilot. These tasks are similar in complexity to a model and control system of a minerals beneficiation plant, suggesting that blackboard systems can handle the computational demands of complex, real-time models of phosphate beneficiation.

Computer software has been written that implements a blackboard system. The blackboard system is an integral part of the interface and modeling software. If this work is extended to process control the blackboard system will coordinate the control process. Although the blackboard is an integral part of the software, its complete description is lengthy and it is never visible to the user. The blackboard will not be described in detail in this report.

Data for mineral processing systems comes from a large number of different sensors and has varying degrees of accuracy. Such sensory information must be integrated into a consistent process model. The modeling process is greatly handicapped by error present in the observed data or by significant unmeasured variables. Techniques of error reduction, materials balance optimization, and the estimation of unmeasured variables are essential to the development of high quality models and are included in the current software system. Calculation of unmeasured variables has been successfully accomplished in the case of a model for column flotation when induction time was a significant but unmeasurable variable (Yeager, et al., 1994). Programs to implement these techniques are available in the class library.

After this preliminary data processing, there are a number of situations in which various parameters of a model must be estimated and adjusted. This common problem of parameter estimation can be reduced to a search problem. Thus, the modeling system requires a robust search algorithm. Genetic algorithms are search algorithms based on the mechanics of evolution. They have proven successful in wide variety of problems. An object class has been written that effectively implements a genetic algorithm; the class is used to solve many optimization problems that arise in the attempt to accurately model the phosphate flotation system.

The remainder of this report summarizes the six computer models and the user interface developed in this effort.

3 THE COMPUTER MODELS

There are two basic types of computer models. The first type, a first principles model mimics the physics associated with the real system being modeled. The second type, the empirical model, duplicates the output of a physical process without concern for the physical details of the process being modeled. In this section, six individual models are described. One is a first principles model and five are empirical models. In each of the six cases, the technique used to model the flotation system is described and results are presented depicting the performance capabilities of the model.

3.1 First Principles Model

The first principles model considered in this research effort is due to Jordan and Spears (1990). In this model, the probability of flotation of a particle is expressed as the product of five different quantities: (1) the probability of collision (P_c), (2) the probability of attachment (P_a), (3) the probability of the particle staying attached to the bubble (P_s), (4) the frequency probability of each bubble size within the unit volume (P_b), and (5) the number of unit volumes passing through this portion of the flotation cell per second (P_u). The probability of collision depends on the size of the particle and the bubble and the hydrodynamics of the flotation pulp and hence is related to the number of collisions per unit time and unit volume. The velocities of bubbles and particles are computed using Kolmogoroff 's theory of turbulence. The hydrodynamics within the flotation cell determine the dissipation energy of the fluid. In this study, the actual bubble size distributions, power consumption, and fluid velocity profiles have been used to provide an accurate mathematical representation of the flotation process. The probability of attachment depends upon the surface characteristics of the mineral which determine the induction time required to attach the hydrophobic particle to the bubble. The probability of a particle remaining attached depends upon the degree of turbulence in the system. The probability of the particle remaining attached to the bubble is computed using an empirical equation.

Mathematical models of this type are important to our control development because they are not restricted by the availability of existing plant data. Because of this they can be used to search for reasons for the inadequacies of existing plants. For example if sensory information is inadequate they can be used as a guide to what measurements are needed. Because of the power and versatility of these models we will explore one particular model in depth.

The mathematical model of flotation due to Jordan and Spears (1990) has been implemented and used to model phosphate flotation data. This model makes the use of the following assumptions, all of which are valid in the flotation of phosphate:

1. The flotation cell is a mixed reactor.
2. The turbulent velocities are isotropic.
3. For small unit times and unit volumes, a unit volume of the pulp remains intact.
4. The bubble size distribution is evenly distributed throughout the flotation cell.

Given the above assumptions, the rate of flotation is computed by balancing the existing unattached particles within the flotation cell according to the formula

$$\frac{dN_{up}}{dt} = K_{bpa}\,[(\frac{d_p}{d_{\max_p}})^{\frac{3}{2}} - 1]$$

where $\mathbf{N_{up}}$ is the number of unattached particles, $\mathbf{d_p}$ is the particle diameter, is the rate constant for bubble-particle attachment, and $\mathbf{d_{maxp}}$ is the maximum particle size that can be floated. Once the hydrophobic particle becomes attached, the flotation rate is determined by balancing the existence of the attached hydrophobic particles in the flotation system according to the following formula

$$\frac{dN_{ap}}{dt} = K_{bpa}\,[1-(\frac{d_p}{d_{\max_p}})^{\frac{3}{2}}] - (\frac{Q_a}{f_a V})\,N_{ap}$$

where $\mathbf{N_{ap}}$ is the number of attached particles, V is the volume of the cell, $\mathbf{Q_a}$ is the volumetric flow rate of an- into the cell, and $\mathbf{f_a}$ is the equilibrium volume fraction of air in the cell.

In computing the flotation rate, there are numerous intermediate calculations made. These include the following.

1. Collision efficiency $(\mathbf{P_c})$:

$$P_c = C_c\,(\frac{d_p}{d_b})^{C_e}$$

where $\mathbf{C_c}$ and $\mathbf{C_e}$ are collision efficiency constants, and $\mathbf{d_p}$ and $\mathbf{d_b}$ are particle and bubble sizes.

2. Fluid dissipation energy (ε):

$$\varepsilon = \frac{E}{\Sigma(P_{U_f}\ U_f^3)}\ U_f^3$$

where E is the mean dissipation energy, $\mathbf{P_{Uf}}$ is the probability of the fluid velocities for a given particle size, and $\mathbf{U_f}$ is the fluid velocity.

3. Kinematic viscosity (υ):

$$\upsilon = C_v \frac{\kappa^2}{\varepsilon}$$

where $\mathbf{C_v}$ is the kinematic viscosity constant and κ is the kinetic energy of the fluid.

4. Induction time (t_i):

$$t_i = C_i d_p^{\frac{3}{2}}$$

where $\mathbf{C_i}$ is the induction time constant.

5. Particle-bubble relative velocity (U_{rbp}) :

$$U_{rbp} = \frac{0.4\ \varepsilon^{\frac{4}{9}}}{\upsilon^{\frac{1}{3}}} [(\frac{\rho_f - \rho_b}{\rho_f})^{\frac{4}{3}} (d_b)^{\frac{14}{9}} + (\frac{\rho_p - \rho_f}{\rho_f})^{\frac{4}{3}} d_p^{\frac{4}{3}}]^{0.5}$$

where ρ_f, ρ_b, and ρ_p are the fluid, bubble, and particle densities.

6. Probability of collision (P_c):

$$U_i = \frac{N_b}{V}\frac{N_p}{V} [P_c\ 5\ (r_b + r_p)^2\ U_{rpb}\frac{1.0}{U_{f^c}}]\frac{U_f}{1.0} * \frac{V}{N_p^-}$$

where $\mathbf{N_b/V}$ is the number of bubbles per unit volume of the cell, $\mathbf{N_p/V}$ is the number of particles per unit volume, and $r_b + r_p$ is the collision radius of the bubble and particle.

7. Probability of attachment (P_a):

$$P_a = \operatorname{sec}h^2 \left(\frac{3\ U_{rpb}\ t_i}{2\ d_b}\right)$$

where h is the height of the cell and U_{rpb} is the velocity of the average bubble-particle group.

8. Probability of remaining attached (P_s):

$$P_s = \left(1 - \frac{d_p}{d_{\max_p}}\right)^{\frac{3}{2}}$$

9. Probability of flotation (P_f):

$$P_f = \sum_{db}\sum_{Uf} P_c\, P_a P_s P_b P_u$$

Now, the probability of collision, times the probability of attachment, times the probability of a stable attachment are multiplied by the frequency probability of each fluid velocity, the frequency probability of each bubble size within the unit volume, and the number of unit volumes passing through this portion of the flotation cell in one second.

Summing up all the possible attachment and collision probabilities calculated from different combinations of fluid velocities and bubble sizes gives the flotation attachment probability of a particular size particle within the unit volume. After achieving stable attachment, the bubble particle aggregate will eventually reach the froth and leave in the flotation concentrate. This model assumes that the froth only transports material from the cell.

The discrete time varying rate constant model is given by the following equations:

$$N_{up}(t) = (1 - P_f)\, N_{up}(t - 1)$$

$$N_{ap}(t) = \left(1 - \frac{Q_a}{V f_a}\right) N_{ap}(t-1) + P_f N_{up}(t-1)$$

$$N_{apc}(t) = \frac{Q_a}{V f_a}\ N_{ap}(t)$$

where $N_{ap}(t)$, $N_{up}(t)$, and $N_{apc}(t)$ are the number of attached, unattached, and attached particles in the concentrate, respectively.

The first principles model was used to model phosphate flotation data for 6 hours of smoothed plant operating data. The results from the program correlated well with the data. When the computer model was tested on actual, unfiltered plant data, the results were not as accurate. Under these conditions, the computer model did not consistently predict the performance of the flotation system to within 10% accuracy. However, when statistical analysis was used to filter the noise from the plant data, the computer model was able to predict flotation performance to within 10% accuracy.

In conclusion, the first principles computer model due to Jordan and Spears is an adequate representation of phosphate flotation systems. However, the model only performs acceptably when the data from the plant is filtered using statistical methods. These methods could routinely be applied to real-time plant data. The results from this model are worth pursuing further because a first principles model has several advantages over empirical models. These advantages include but are not limited to the following: (1) unlike empirical models, first principles models are generically extensible to flotation systems for which they were not originally developed; (2) first principles models are not based on empirical coefficients which require a trial-and-error tuning process; and (3) first principles models allow one to play the "what if" games (e.g., to ask questions such as "what happens if a certain flow rate is reduced by 10% ") generally necessary in the development of effective control strategies. The interface discussed later in this report will allow the user to ask questions about changing model parameters and to see the results of such changes.

3.2 Standard Multivariate Regression Model

Standard multivariate regression models were investigated using test data from a phosphate rougher circuit, measured eight times an hour. Most modeling was done using phosphate recovery as the dependent variable, since the measured grade had little variation in this data set. Independent variables investigated included mass rate and BPL assay of the raw material flow, flow rates for fatty acid and fuel oil, and conditioner section pH levels. These variables were combined in various ways (scaling reagent flows by total mass flow and by phosphate mass flow, for example), and various exponential powers, reciprocals, and logarithms of the variables were computed. The goal was to find combinations of the original and transformed variables which effectively model the flotation system recovery. Many combinations of these terms were tried, using a variety of criteria for selecting and rejecting terms. The best models closely fit the larger time-scale structure of the data. But single regression models did not effectively reproduce the short time scale variations in the data.

Instead of asking a single regression equation to model all of the plant's operating modes, a "moving window" approach was tried. A new regression model was fitted periodically, using several hours of observations. The regression used a stepwise method for selecting the best of the dependent variables, usually stopping at three or fewer. This approach proved effective, achieving correlations better than 0.9. Further investigation with this method showed that essentially the same results could be achieved without complex data transformations, and that the resulting models could do an acceptable job of predicting - not just reproducing - the dependent variable.

Figure 1 shows the result of applying this adaptive stepwise regression approach: fifty hours of test data were analyzed, using a modeling window five hours wide and re-computing the model once each hour. The figure shows the original data, smoothed using a moving average of three observations, and predicted recovery values: each five-hour model was used to compute recovery values for the following hour's data. The correlation between the predicted and observed values shown in the plot is just over 0.7.

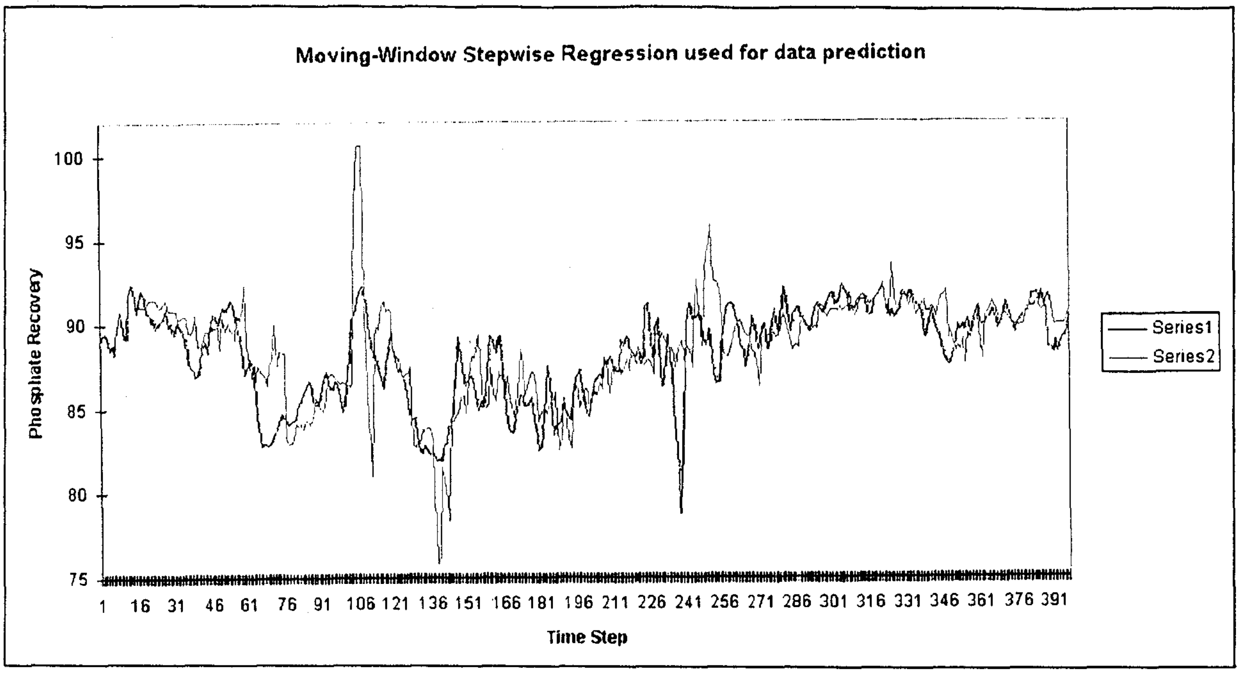


Figure 1 - Adaptive stepwise regression is applied to fifty hours of test data (400 observations). The correlation between predicted values (Series 2) and measurements (Series 1) is 0.71. The correlation between modeled values and measurements for this example is 0.93 (using at each data point the regression model computed for the five-hour window which contains the point in the middle hour).

3.3 Partial Least Squares Regression

The second of the two statistical computer models uses the technique of Projection to Latent Structures (also known as Partial Least Squares Regression) (Geladi and Kowalski, 1986). This method has been used by Kester et al (1991) to compress data from chemical processing plants into a few simple components that reproduce most of the information collected by a large number of sensors.

For the PLS method, the independent (X) and dependent (Y) variables are typically all centered and scaled to mean=0 and variance=1. Then the regression proceeds rather like a principal-components analysis, finding a series of orthogonal vectors that carry the maximum amount of information about the X-Y space. But in PLS the X and Y spaces are decomposed separately, and the criterion at each step is to find the one-dimensional X vector and Y vector whose covariance is maximum. This identifies the best-predictor combination of independent variables and the best-predicted combination of dependent variables. Then the residual parts of the X data and of the Y data are found -- that is, the variation not yet used for prediction and the variation not modeled by earlier components -- and the process is repeated. Usually most of the data is modeled in just a few principal components, with higher-order components representing noise. Figure 2a shows the effectiveness of the Partial Least Squares Regression Model. This figure shows the results of a simulation in which the recovery was modeled.

A PLS model has the same form as a standard multivariate regression model. But the goal of the PLS method is to reduce the dimensionality of the model to two or three independent variables, each of which is a linear combination of the X variables, as in the standard regression model. Furthermore, in actual processing plants the Y variables are generally available infrequently and only after a delay, while the X variables are measured continuously. If the PLS analysis is performed on observations taken when the plant is in good control then tracking the values of the handful of reduced independent variables identified by the PLS model has been found to provide a powerful means of monitoring chemical processes. This means that common frequently made measurements can be used with confidence because they are in effect calibrated by more expensive and less frequently made measurements.

Control charts are produced by plotting the reduced X components against each other as standard scatter plots. The in-control observations used for the PLS analysis cluster in a compact group, and as long as continuing measurements lie within the original cluster, the plant remains in good control. When the measurements fall outside the cluster, the exact way in which they deviate from the cluster can be used to deduce why the process went out of control.

In an application of PLS to phosphate flotation data, points were considered to represent in-control operating conditions if both recovery and grade exceeded a threshold, generally chosen as half a standard deviation above the mean; both mean and standard deviation for each variable were computed either from the entire data set or from some time block of data. Flotation data was processed using different selection criteria for the control reference sets as well as different scaling factors, then reduced to three X components. These were plotted pairwise to produce three control charts shown in Figure 2b.

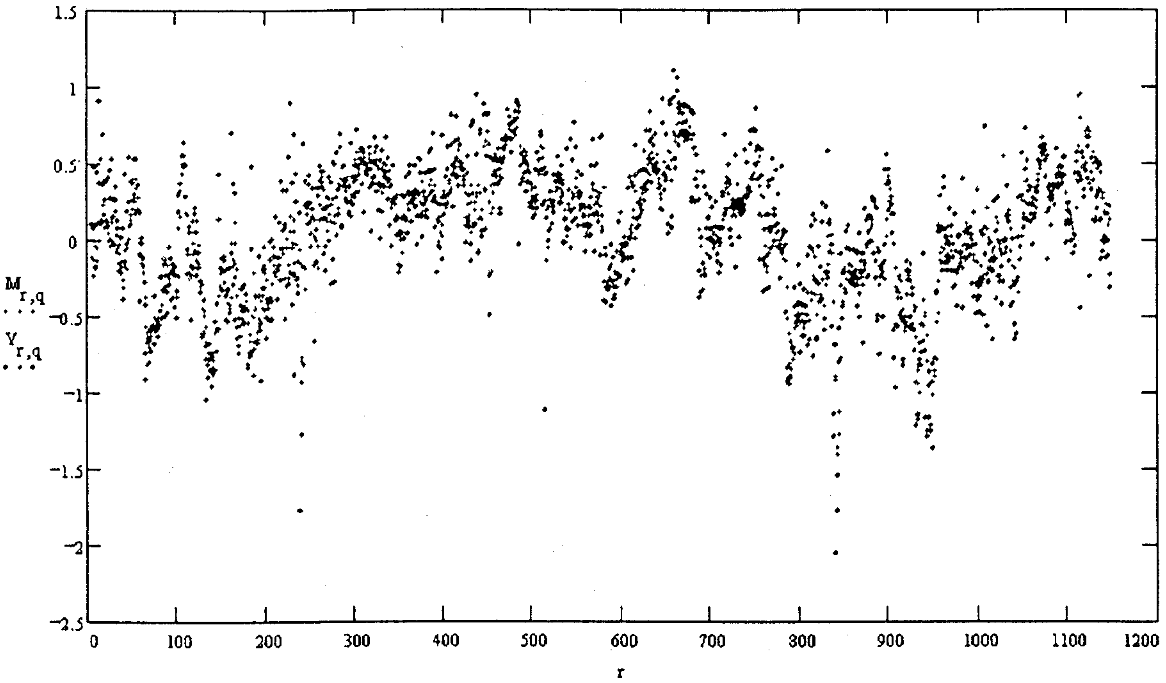
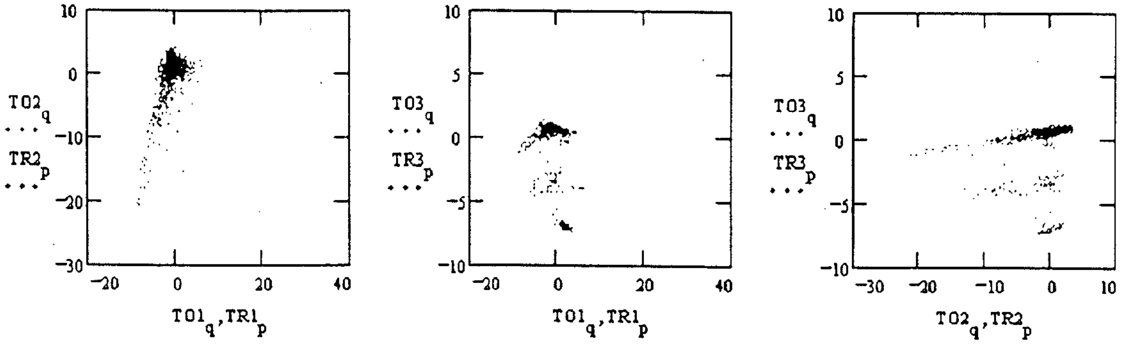


Figure 2 - PLS results
(a) A single multivariate regression model derived from Projection to Latent Structures is used to model a week of phosphate flotation data (1140 observations). The modeled variable (shown centered and scaled) is phosphate recovery from a rougher circuit. The correlation between measured values (Y, shown in blue) and modeled values (M, in red) is 0.77 for this example analysis.



Blue: in control
Red: out of control CONTROL CHARTS based on Projection to Latent Structures Analysis

(b) The control charts are based on the test data used for Figure 2a. In this PLS analysis, grade and concentrate mass flow as well as phosphate recovery were used as dependent variables.

3.4 Neural Network Model

Artificial neural networks are computational systems that mimic the biological neural networks of the mammalian brain. The human brain contains about 100 billion neurons (neural cells), interconnected in a complex network via synapses (a junction between axons and dendrites). Artificial neural networks are grossly simplified models of living neural systems.

A wide variety of neural network architectures (the arrangement and connection of neurons) have been used successfully to solve numerous problems. However, despite the differences in the architectures, neural networks are based on a common component, the neuron which is the basic processor in neural networks. In their most basic form, neurons are computational units that receive multiple signals, sum these signals, and compute an output based on an activation function (generally nonlinear) that depends on the sum of the signals received. Each neuron has one output, which is generally related to the state of the neuron (its activation). The output of each individual neuron may well be passed on to other neurons in various positions in the network via weighted connections. Each neuron receives inputs over these connections, called synapses, from other neurons. The inputs are the activations of the incoming neurons multiplied by the weights of the synapses along which they travel. The activation of the neuron is computed by applying a threshold function to this product. A nonlinear threshold function is used to introduce nonlinearities into the neural network. An abstract model of the basic neuron is shown in Figure 3.

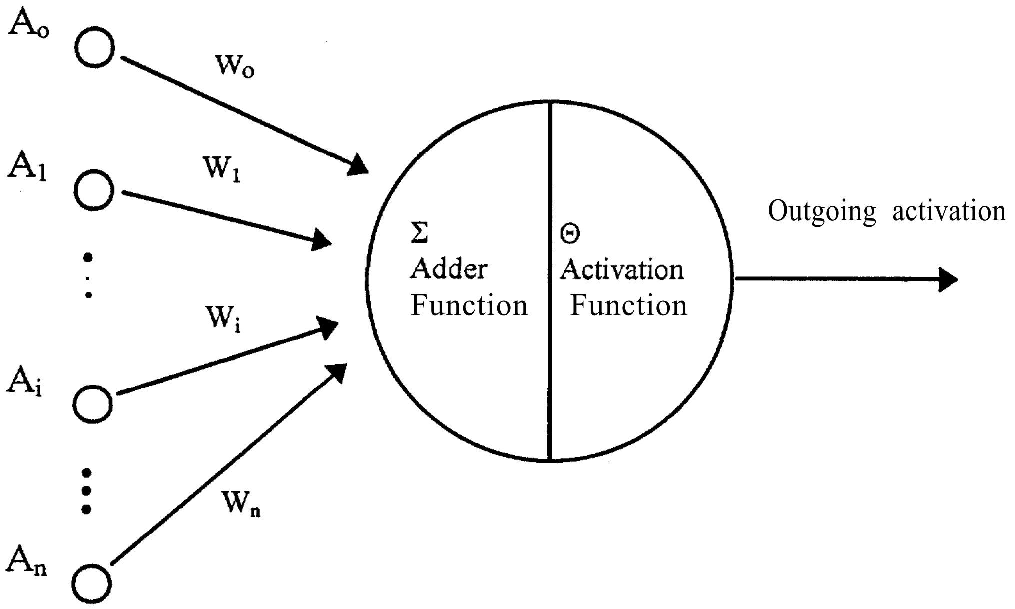


Figure 3 - The basic neuron receives input in the form of activations from other neurons, processes the information, and provides its own outgoing activation which is then sent to other neurons in the neural network.

The activation or threshold function is generally a nonlinear function. Nonlinearity is an extremely important attribute to neural networks because it allows them to model the complex relationships existing between input and output parameters. The most commonly used nonlinear function that is appropriate for neural networks is the sigmoidal function defined by

$$f(x) = \frac{1}{1 + e^{-x}}$$

where x is the sum of the input signals to the neuron.

The individual neurons of neural networks are placed in various layers. The neurons in the layers are interconnected by weighted synapses. Figure 4 shows a schematic of a typical backpropagation network consisting of three layers: (1) input layer, (2) hidden or middle layer, and (3) output layer. The input layer neurons receive an activation signal from the physical environment (data from the system to be modeled). These input layer neurons then sum the input signals and compute an activation based on their activation or sigmoidal functions. The resulting activation is passed to each of the neurons in the hidden layer (for a fully connected network each neuron in the input layer is connected to each neuron in the hidden layer, and each neuron in the hidden layer is connected to each neuron in the output layer). Along the way, the activation signals are multiplied by the weights associated with each of the synaptic connections along which the signals are passed. These weighted activation signals serve as inputs to the hidden layer neurons. And, selecting the weights in the network such that it gives the correct answers is the whole problem.

The hidden layer neurons perform calculations with the signals similar to the calculations performed by the input layer neurons resulting in activation signals that are passed to the output layer neurons. The output neurons use these weighted input activation signals coming from the hidden layer neurons to compute their output activation signals. These activation signals coming from the output neurons are the output of the neural network. This process by which an input signal is received from the physical world, multiplied by weights associated with the connections between input and middle layer neurons, nonlinearized with activation functions, multiplied by a second set of weights associated with the connections between middle and output layers, and nonlinearized again with activation functions, is called a forward pass. This pass is used both in training a neural network and in using a trained neural network to model a system. It is, however, the backward pass which gives neural networks their adaptive capabilities. In the backward pass, the network weights are adjusted using a derivative-based algorithm so that the network accurately models the training sets it is presented.

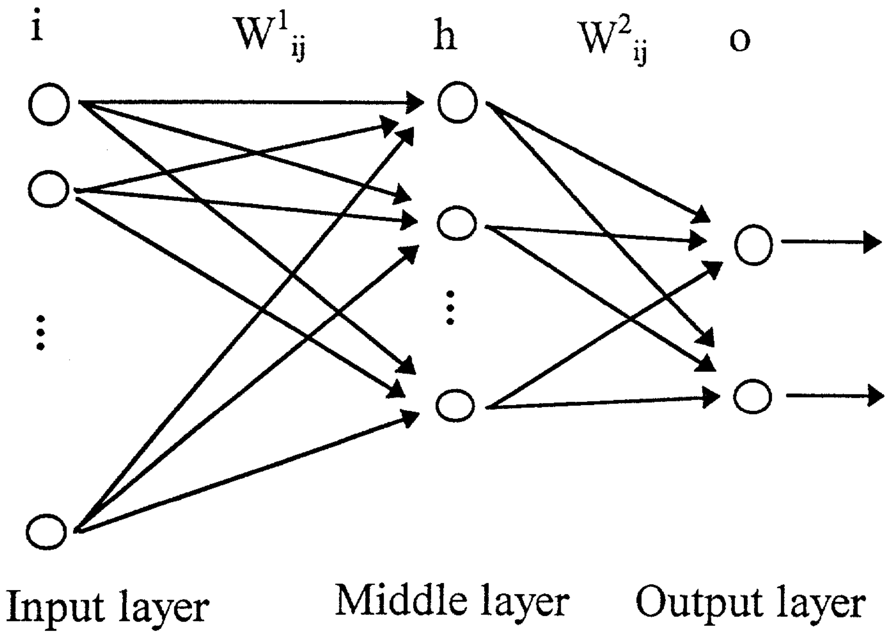


Figure 4 - Typical backpropagation neural networks consist of three layers of neurons interconnected by weighted synapses.

As described above, neural networks have associated with them an architecture in which numerous neurons are connected into layers and a mechanism for accepting an input signal and converting said input signal into an output signal. The relationship between the network's output signal and its input signal is determined by the weighted connections which are adjusted in a backward propagation of errors. The backward propagation of errors through the network represents the training phase of a neural network while the forward pass is 'the actual implementation of the neural network. Generally, the implementation is easy and fast while the training can be complex and time-consuming.

Backpropagation neural networks take a training example (data from the system being modeled), compute an output signal, and then compute an associated error based on the appropriate known solution it is provided. Initially, its errors are due mainly to the fact that the weighting values on the synapses are not accurate. Thus, a mechanism is needed by which the weights can be adjusted in a manner so that the network "will get the correct answer." The backpropagation of errors is a relatively generic concept. There are many ways to do this. However, the classic backpropagation algorithm uses a derivative based approach.

Neural networks have the ability to learn any arbitrarily complex nonlinear mapping. This is due to the nonlinear activation functions and the introduction of a middle or hidden layer. However, there are potential drawbacks to neural networks. First, one of the most cited disadvantages of neural networks is that they can involve extremely long training times. Second, neural networks can "memorize" the training data but fail to extend generalizations to situations for which they have not been specifically trained. Often, this occurs when the neural network has too many hidden nodes; a delicate problem since too few hidden nodes and the neural network will not train while too many nodes causes the neural network to memorize. Third, neural networks do not give any indication as to why they do what they do. In other words, when a neural network is used as a computer model, it receives input data and produces an output value. There is no easy way of understanding the complex relationships between the input values and output values that are contained within the neural networks weight matrices. Thus, neural networks are the quintessential "black box" device.

The neural network software used in the current research effort can be used to model a system about which the developer has no special knowledge. However, the more a model developer knows about the system being modeled, the more efficiently the model can be developed. Additionally, there are some decisions that must be made in implementing a neural network, and an experienced neural network practitioner will generally make these decisions more efficiently than a novice. These decisions include: (1) how many neurons to put in the middle layer, (2) what kind of learning rate to employ, (3) whether or not to include momentum terms which allow the system to overcome local optima, and (4) what kind of activation function to use.

The neural network computer model of the phosphate flotation circuit was two input, one output model. It had ten hidden nodes, was trained using 125 data sets, ran for 100 iterations, and took approximately 7 minutes to train. The model performed acceptably both on training data and on test data that was comprised of data on which the neural network had not been trained. Figures 5 and 6 indicate the performance of the neural network. Additionally, Figure 7 shows a time series comparison of the neural network model as compared to flotation data. A correlation coefficient of 0.9593 was obtained using the neural network model of the flotation system.

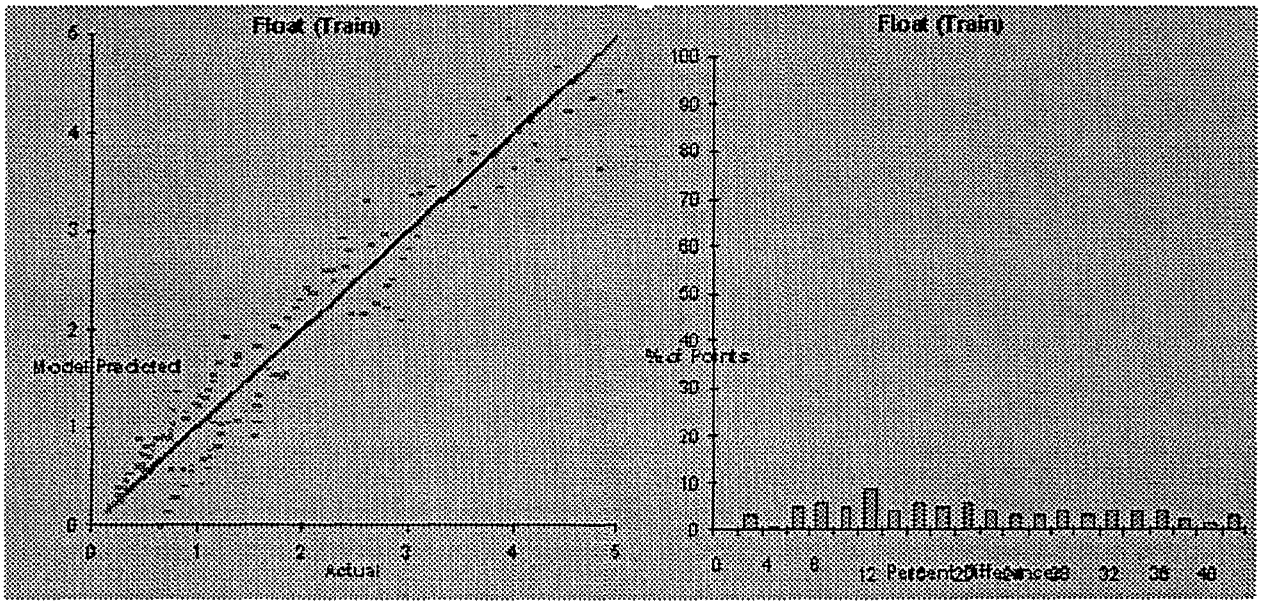


Figure 5 - Neural Network Training Data - (a) The neural network model of a flotation circuit accurately predicts grade for the training data as shown in the plot of model predicted vs. actual values. (b) The histogram demonstrates that error associated with the model predicted values is generally within acceptable engineering error values.

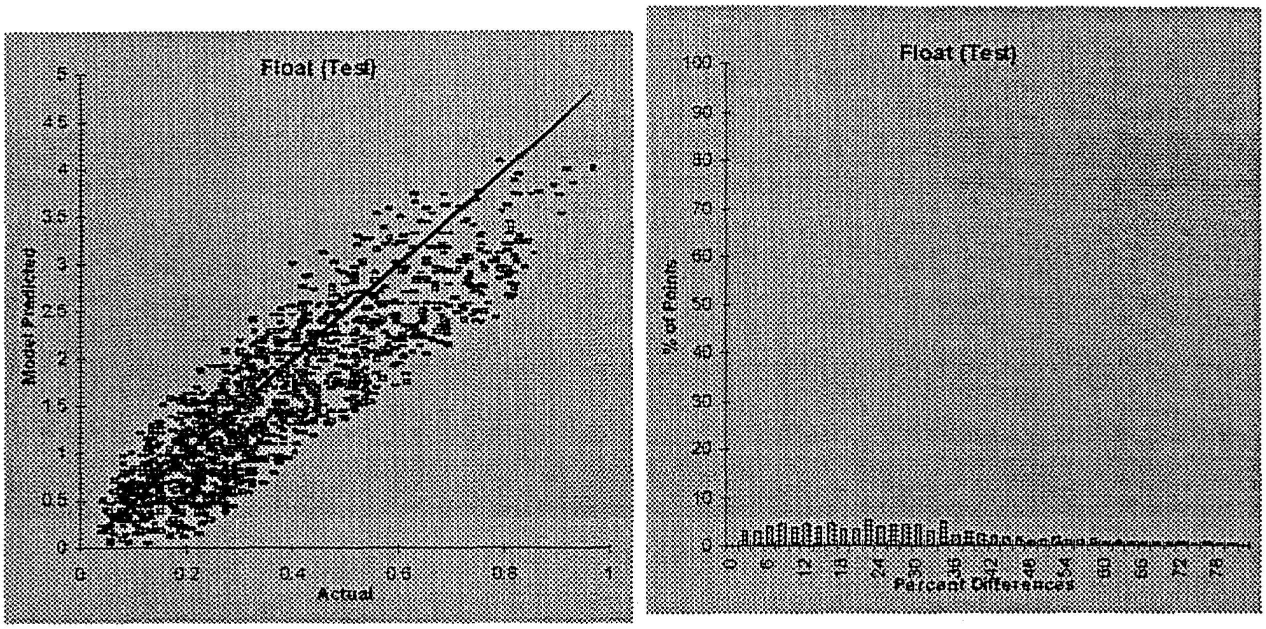


Figure 6 - Neural Network Test Data - (a) The performance of the neural network model of a flotation circuit is acceptable on the test data. (b) the histogram depicts the acceptable model performance.

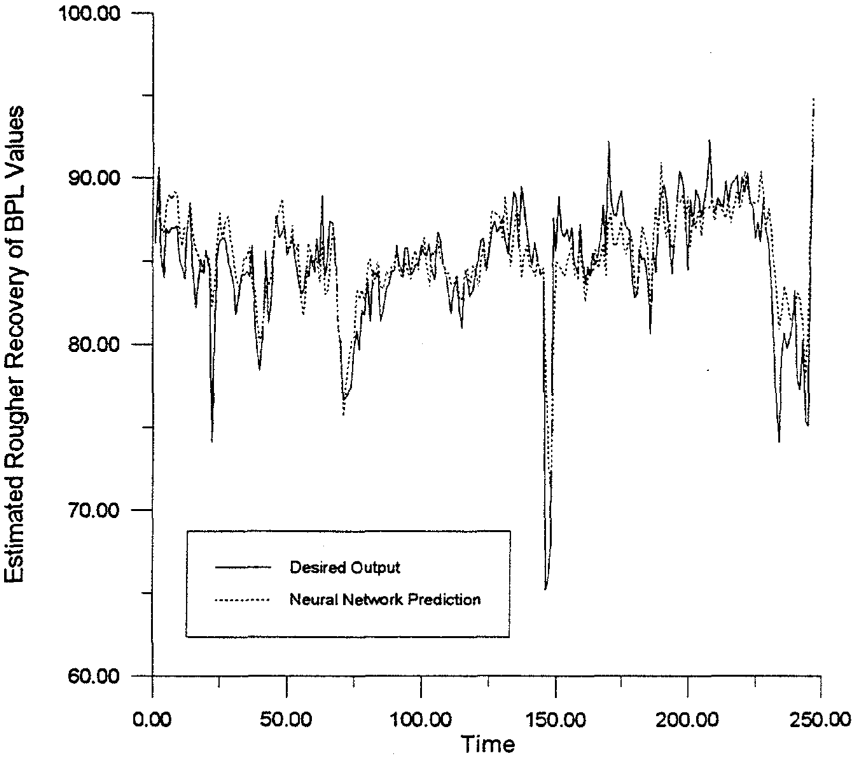


Figure 7 - Neural Network Time Series Data - The neural network model accurately predicts the recovery of a rougher circuit.

3.5 Fuzzy-Genetic Model

Unlike neural networks which provide no insight into their decision-making, fuzzy models describe relationships between input and output variables in linguistic terms. Like conventional rule-based systems (commonly called expert systems), fuzzy linguistic models use a set of production rules which are of the form

IF *{condition)* THEN *{action}.*

The left-hand-side of the rules (the *condition* side) consists of combinations of the input variables. The right-hand side of the rules (the *action* side) consists of combinations of the output variables. Unlike conventional expert systems, fuzzy linguistic models discussed here employ rules that utilize fuzzy terms like those appearing in human rules-of-thumb. For example, a reasonable rule for a fuzzy linguistic model used for a grinding circuit might be:

IF {percent solids is *“small”* AND ball size is “very *small” AND* mill speed is *"fast”* AND dispersant addition is *“medium”*}
THEN {fineness is 57.87 AND energy is 16.35 AND distribution modulus is 0.3789 AND viscosity is 0.3227}.

This rule describes a relationship between the input and output parameters using linguistic terms like *“small”, “medium”, "large",* and *"fast"* to represent the input parameters and numeric values to represent the output values (Karr, 1993a).

The linguistic terms are conceptually easy for humans to comprehend. But, these terms are actually subjective terms -- they can mean different things to different human “experts.” Such subjective concepts are not readily manipulated by computers. Thus, a mechanism is needed for transforming subjective, linguistic concepts into a form that can be manipulated by computers. Linguistic, or fuzzy, terms are assigned concrete meaning via fuzzy membership functions. And, the fuzzy membership functions work in concert with the fuzzy rule base to produce the fuzzy linguistic model.

In fuzzy mathematics, membership functions are used to assign a degree of membership to each input variable for a given condition existing in the physical system. A degree of membership can be viewed as the degree to which the fuzzy descriptor accurately describes the actual variable value. A membership value of 1.0 indicates the descriptive term describes the variable value with complete confidence; a membership value of 0.0 indicates the descriptive term definitely does not describe the variable value. Thus, fuzzy membership function values provide a mechanism for an input variable, say percent solids, to be described (to some extent) by more than one subjective term.

While providing a mechanism for allowing expert systems to work with subjective terms, fuzzy membership functions generate a need to deviate from the approach used in traditional expert systems. In traditional expert systems, only one rule is eligible to be implemented, or “to fire”, for every condition that could exist in the physical system. However, in fuzzy systems, each rule can be thought of as being eligible to fire for every condition in the physical system. This leads to a dilemma since the various rules prescribe different values of the output variables. This dilemma is solved via the weighted averaging of values of the output variables prescribed by each of the individual rules. The weighting factors used are simply the minimum degrees of membership associated with each rule; these minimum membership function values are determined using "classic" fuzzy techniques (Karr, 1991). This approach makes practical sense since each rule is allowed to play some role in the eventual magnitude of the output values; those rules in which there is a great deal of confidence (those with large degrees of membership) have large weighting factors while those rules in which there is a small degree of confidence (those with small degrees of membership) have small weighting factors. This approach is commonly adopted in the development of fuzzy systems (Karr, 1993a). It has been demonstrated to be a powerful tool in the realm of process control, but can also be effectively used in computer modeling.

There are two aspects to producing a fuzzy linguistic model that are particularly difficult: (1) selecting the rule set (i.e., selecting the values for the output variables for each combination of input parameters) and (2) tuning the membership functions so that the definitions of the linguistic terms derived from the input values works in conjunction with the rule set to accurately model the physical environment (i.e., selecting fuzzy membership functions that properly "define" the linguistic terms used in the rule set). These tasks are not independent operations; the way the membership functions are defined directly affects the definition of an effective rule set, and vice versa. Fortunately, these tasks can be efficiently done simultaneously using a genetic algorithm.

When applying a genetic algorithm to any particular search problem, there are basically two major decisions to be made: (1) how to represent or code potential solutions to the problem as character strings and (2) how to determine the fitness function needed to drive the search. The coding issue is effectively addressed by using concatenated, unsigned binary, linearly mapped representations (Karr, 1991). Genetic algorithms require the natural parameter set of the problem to be coded as a finite string of characters. Although many character sets have been used in the coding of real-world examples, the parameter sets in this study are coded as strings of zeros and ones. For example, a parameter Θ_1 is quite easily represented as a binary string. M bits are allotted for defining the constant and are interpreted as a binary number (1001010111 is the binary number 599). This value is mapped linearly between some user determined minimum (Θ_{min}) and maximum (Θ_{max}) values according to the following:

$$\Theta_1 = \Theta_{min} + \frac{b}{2^M - 1}(\Theta_{max} - \Theta_{min})$$

where b is the integer value represented by an M bit string. The values of Θ_{min} and Θ_{max} in a given problem are selected by the user based on knowledge of the problem or actual physical constraints. Each search parameter is represented similarly, and the bit-substrings are concatenated to form a single string that represents each parameter for which a value is sought.

In the problem of generating a fuzzy model, there are two different types of search parameters to be considered. The first type of parameter that must be considered "defines" the triangular membership functions (see paper by Karr, 1991 for details). The second type of parameter defines the rule set. Consider again the example rule for grinding presented earlier:

IF {percent solids is *"small"* AND ball size is "very *small" AND* mill speed is "fast" AND dispersant addition is *"medium"*}
THEN {fineness is 57.87 AND energy is 16.35 AND distribution modulus is 0.3789 AND viscosity is 0.3227}.

There are four values (57.87, 16.35, 0.3789, and 0.3227) that must be declared to make the rule complete (This *real-valued consequent* approach to defuzzification is described in the paper by Karr, 1993a). As these values are altered, the performance of the model changes.

The fitness function issue is easily solved by employing a curve fitting approach using the data from the physical system being modeled (Karr, Stanley, and Scheiner, 1991). Although any of a number of curve fitting criteria can be used, the objective selected in the current effort is to minimize the maximum error associated with the data points used for training the fuzzy-genetic algorithm model.

A self-generating fuzzy linguistic modeling system has been developed that employs a genetic algorithm to select membership functions and to define rule sets. The system is capable of predicting the performance of nonlinear processes and mechanisms by considering only a data set that contains the input/output relationships.

As with neural network modeling, the process of developing fuzzy-genetic algorithm models is more efficient when the model developer has some insight into the system being modeled. And, as with neural network modeling, the automated fuzzy-genetic algorithm modeling software requires some implementation decisions that are generally made more efficiently by experienced practitioners. For instance, the model developer must determine: (1) how many descriptive terms are needed to adequately describe the parameters incorporated into the model, (2) how large the genetic algorithm populations should be, (3) an efficient genetic algorithm coding scheme, and (4) typical genetic algorithm parameters such as the probabilities of crossover and mutation.

A fuzzy-genetic model of phosphate flotation has been developed. The computer model was trained using bit strings of length 875, employed populations of size 200, and ran for 40 generations. The computer model accurately predicted the performance of the flotation system as depicted in Figure 8. A correlation coefficient of 0.9102 was obtained using the computer model.

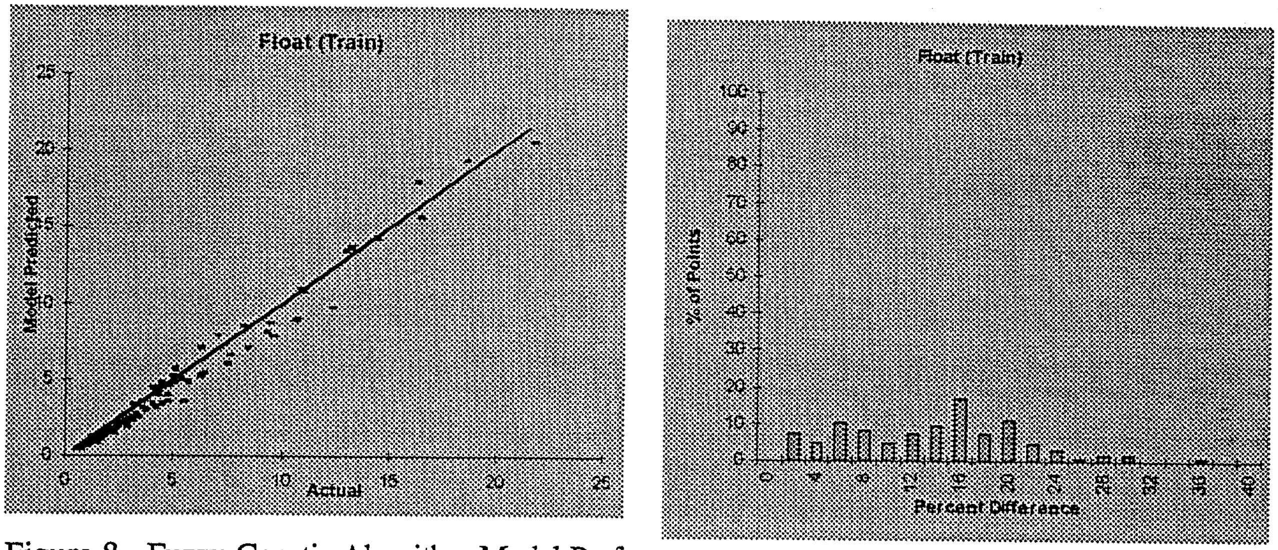


Figure 8 - Fuzzy-Genetic Algorithm Model Performance

(a) The fuzzy-genetic algorithm model of a flotation circuit provides an accurate prediction of grade as shown in the plot of model predicted vs. actual values. (b) The histogram also bears out the acceptable performance.

3.6 Rate Constant Model

The mathematical form of the first principles model of Jordan and Spears is a simple first-order differential equation, in which the number of attached particles leaving in the concentrate is directly proportional to the number of attached particles in the flotation cell. The simplest possible model with this essential form is the following differential equation:

$$Q_c = kcV$$

where c is the concentration (i.e. volume fraction) of a mineral in the slurry, V is the volume of slurry in the unit (excluding air), k is the rate constant associated with the mineral, and $\mathbf{Q_c}$ is the volumetric flow rate of the mineral to the concentrate.

If perfect mixing is assumed, the concentration of a mineral in the tails is equal to its concentration in the flotation unit. The volumetric flow rate of the mineral to the tails $\mathbf{Q_t}$ is then given by:

$$Q_t = cQ_{slt}$$

where $\mathbf{Q_{slt}}$ is the total volume of the tailing slurry. These equations can be solved for a given feed flow by assuming that the feed is constant long enough for the flotation cell to reach an equilibrium concentration, and then iterating as follows:

$$c_{i+1} = c_i - (Q_c + Q_t - Q_f)Dt/V$$

where Q_f and Q_t are the volumetric rates of the floatable species in the feed and tailings, respectively, and Dt is a small interval of time. The initial concentration, c_0, is assumed to equal its concentration in the feed, i.e., $c_o = c_f = Q_f / Q_{slf}$.

This simple model gives a rate constant for a situation where all of the process parameters such as ore type, conditioning, flotation reagent flow, and bubble size remain constant. This is of course unrealistic. Yet this simplest model can still be effective if it is applied only within a restricted set of operating states. If a number of such states are identified, and a simple model is tuned to perform well for each of those restricted conditions, then the collection of all such mini-models, taken together and used as a single model, may perform as well as or better than one complex model tuned over the entire range of process states. The generic rate constant model is designed as a coordinator for multiple copies of a very simple, general model, fitted well to a fairly homogeneous set of data. This is similar to an approach used by Reuter and van Deventer (1992).

In order to properly construct and tune the general rate constant model, it is necessary to perform statistical clustering on the relevant data set. Given a large set of data describing the actual performance of a process, a cluster analysis technique based on work by Ginsberg and Whiten (1991) is used to break the data up into fairly homogeneous subsets. This technique produces ellipsoidal clusters, which are appropriate for mineral processing data. It is notoriously difficult to define a statistic that corresponds well to the idea "visually appealing assignment to clusters," so the data viewer was initially designed as a tool for checking the outcome of the clustering operation; the viewer has since acquired other functions as well.

Figure 9 shows the ability of the general rate constant model to reproduce grade and recovery values for a phosphate flotation system.

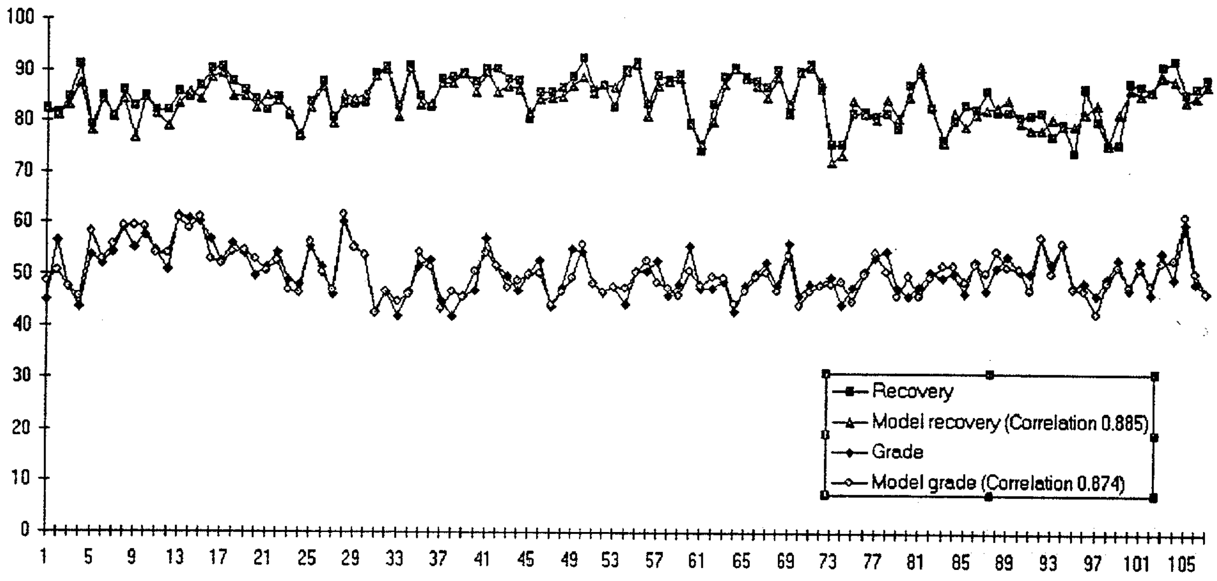


Figure 9 - Rate Constant Model Time Series Data - The rate constant model effectively predicts the grade and recovery of a rougher circuit.

3.7 Summary

Six different computer models of phosphate flotation have been developed. These computer models can be used to simulate the performance of flotation systems. However, as will be seen in the next section, they have much greater potential.

4 GRAPHICAL USER INTERFACE

In the previous section, six computer models of phosphate flotation were presented. Each of these models can provide an accurate prediction of phosphate recovery.

It was apparent early in the development of these computer models that a graphical user interface would be needed so that users not well versed in the intricate details of the models could use them effectively. In addition, the graphical user interface could enable the user to employ the models in a variety of ways that would prove quite difficult if the models occurred as stand-alone applications. The work which actually integrates a model with the graphical interface has been done as a proof-of-concept demonstration using the rate constant model. However, the graphical user interface is not a good environment for developing complex models. For this reason, all of the models have been developed as stand-alone programs. The remaining models will be integrated into the interface after they have been tested further.

There are a number of tasks for which the interface and the models of phosphate flotation can be used. One is to simulate actual plant behavior, for a single process or for an entire circuit. A second is to experiment with model parameter values in order to investigate potential consequences of changing aspects of the plant's operation. Another is to examine the parameters of models tuned to match data taken under different operating conditions in order to gain insight into unmeasured variables. Yet another, of course, is automated control. Before any of these tasks can be accomplished, the user must be able to represent the plant. A complete user's guide will give a detailed description of the operation of the interface and its associated stand-alone programs.

4.1 Stand-Alone Components for Data Processing

Plant operating data is presented to the modeling system in a standard file format. The data is expected to be recorded as columns of numbers in spreadsheet or similar form. The modeling system expects to see an extract of such a file, preceded by a text prefix declaring what each column actually means. The prefix language is very flexible, to accommodate the wide variety of possible measurements. See the system's user manual for a full description of this language.

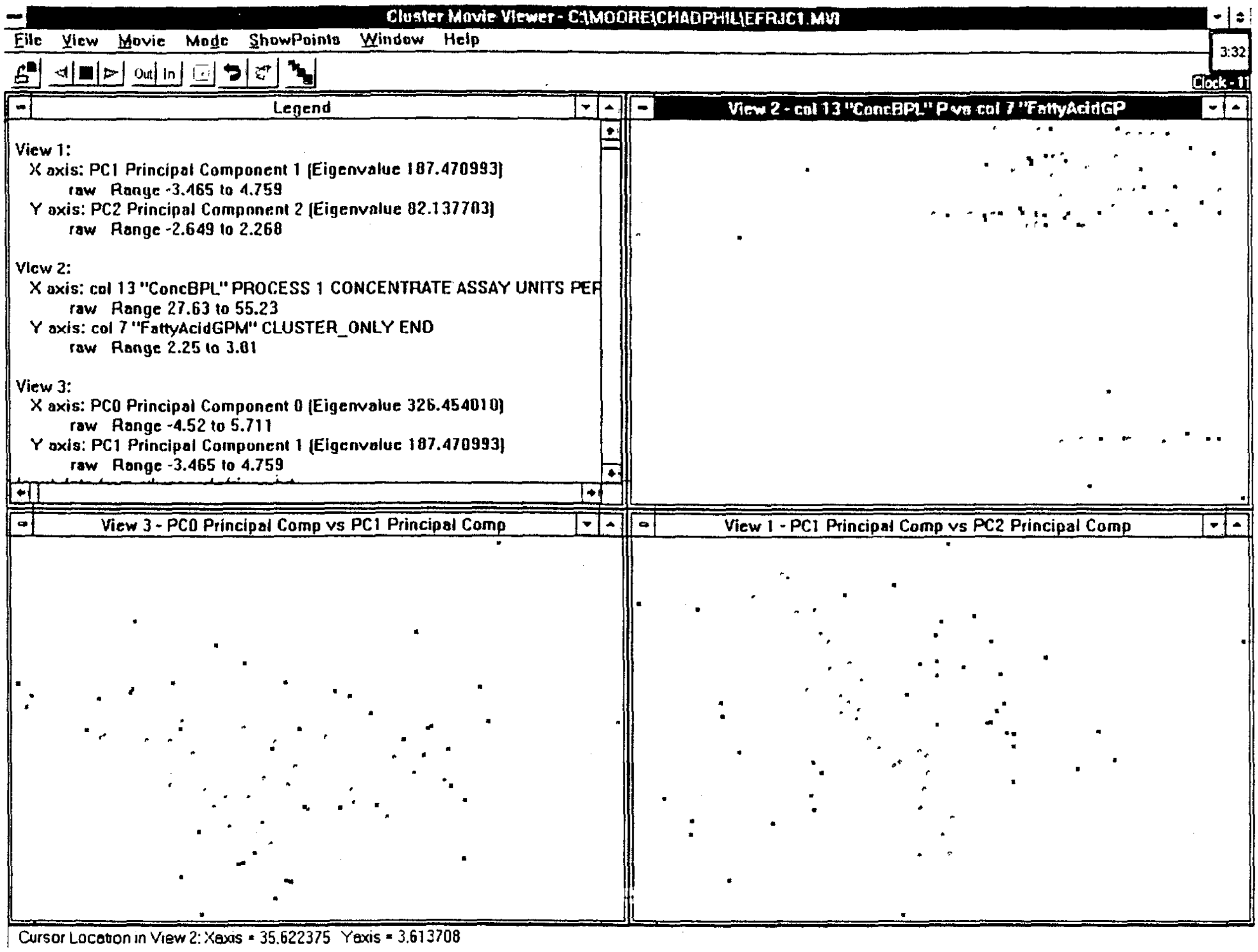
4.1.1 Cluster Analyzer

The clustering technique adapted for the modeling system is based on the work of Ginsberg and Whiten (1991). Engineering data typically produces long thin clusters rather than compact ones; this technique adapts the clustering rule to the local shape of the clusters it finds. Since clustering outcomes can be very sensitive to choices about scaling and centering, the cluster analyzer offers several scaling choices, and also offers the option of clustering in principal-components coordinates and of scaling using data which has been corrected for noise by reducing its higher-dimension components. The cluster analyzer performs the principal-components analysis used subsequently by the data viewer and by the model-tuning operations of the graphical interface.

4.1.2 Data Viewer

The viewer displays scatterplots of the data in coordinate systems derived from the principal-components analysis: each axis may show a measured variable, a measured variable scaled as requested for the cluster analysis, a corrected measured variable after error reduction (or the corresponding computed error) based on any number of dimensions, or a principal component. Any number of scatterplots can be displayed at the same time. The Viewer uses scatterplot dot color (changing with time) to show the aggregation of points into clusters. Since a point is shown in the same color in every scatterplot, the effect is to highlight data relationships. The Viewer can also step directly through all the points (which will normally be presented in order by time, so this animation traces plant operation through time). The user can run an animation frame-by-frame or continuously (at varying speeds), forwards or backwards. The user can also use the mouse to select one point or a clump of points in one scatterplot; these points are then highlighted in all the views, and can be saved to a separate data file for further analysis. Since each plot is small when many are displayed at once, it is possible to zoom in and out and pan over each display in order to make it easy to see any area of interest. The user also has some control of plotting-dot symbols and colors.

Figure 10 shows data from the same test data set as Figure 9. Three scatterplots are shown, with a legend window identifying the coordinate axes for each plot. Points which were selected from view 1 are highlighted in all views.


Cluster Movie Viewer - C:\MOORE\CHADPHIL\EFRJC1.MVI
File View Movie Mode ShowPoints Window Help
3:32
Out In
Clock - 11
Legend
View 2 - col 13 "ConcBPL" P vs col 7 "FattyAcidGP
View 3 - PC0 Principal Comp vs PC1 Principal Comp
View 1 - PC1 Principal Comp vs PC2 Principal Comp

View 1:
  X axis: PC1 Principal Component 1 (Eigenvalue 187.470993)
      raw Range -3.465 to 4.759
  Y axis: PC2 Principal Component 2 (Eigenvalue 82.137703)
      raw Range -2.649 to 2.268

View 2:
  X axis: col 13 "ConcBPL" PROCESS 1 CONCENTRATE ASSAY UNITS PER
      raw Range 27.63 to 55.23
  Y axis: col 7 "FattyAcidGPM" CLUSTER_ONLY END
      raw Range 2.25 to 3.81

View 3:
  X axis: PC0 Principal Component 0 (Eigenvalue 326.454010)
      raw Range -4.52 to 5.711
  Y axis: PC1 Principal Component 1 (Eigenvalue 187.470993)
      raw Range -3.465 to 4.759

Cursor Location in View 2: Xaxis = 35.622375 Yaxis = 3.613708

Figure 10 - The Data Viewer is shown as the user explores phosphate-flotation data.

4.2 User Interface Capabilities

The graphical user interface modeling system provides a desktop with the general capabilities described below.

The user can arrange icons on the desktop in order to build (and edit, save, and restore) a visual representation of a minerals-processing plant. This representation looks very much like a traditional flow diagram. Figure 11 demonstrates this aspect of the modeling system which resembles a commercial drawing package but is specialized for producing models of flotation plants.

The desktop icons can be attached to data structures that model the processes and flows of the plant. The model for a flow consists of a list of mineral constituents (ores, gangue, water) and corresponding quantities (rates of flow); the ore constituents are characterized by mineral type, particle size, and (for some contexts) induction time. The model for a process determines its mineral constituents from its input flows; its parameters may be provided by the user, or may be determined automatically by the modeling system.

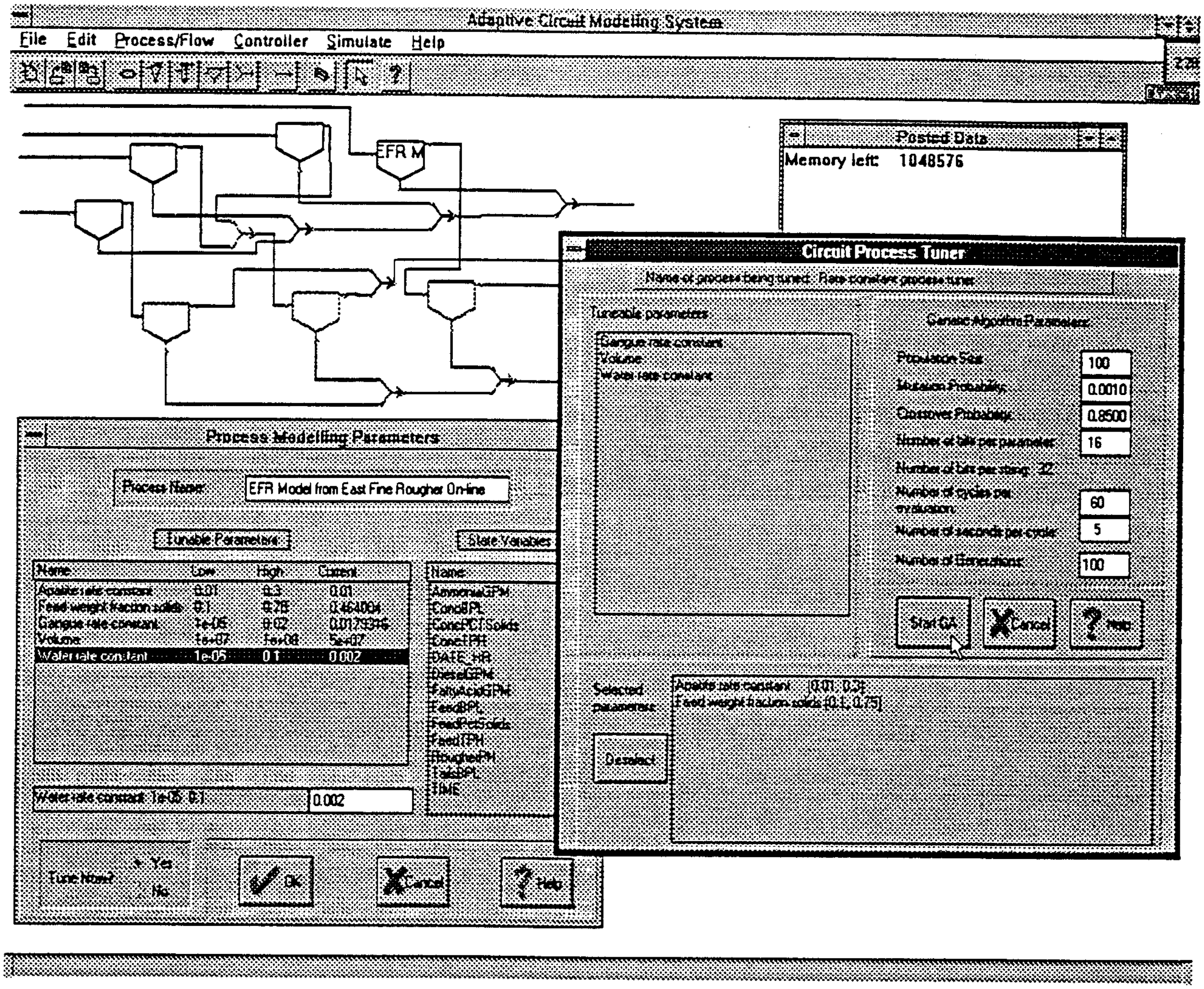
"Tuning" is the modeling system's term for choosing process-model parameters that do the best job of matching model simulation outcomes to some desired target. Tuning in all cases is done using a genetic-algorithm search strategy. Targets are flow states: they may correspond to actual observations of plant operating data, or may represent desired conditions. A single tuning target for a flow may be entered directly from the graphical interface The modeling system can also optimize parameters against a large collection of operating data organized in a data file. The rate constant model has been fully implemented in the interface for tuning with such data.

To tune a process model against a list of targets, the modeling system evaluates each potential set of model parameters as follows. For each observation (i.e. row of a data file), the modeling system adjusts the state of the process's input flows to match the observation targets; simulates the process (using the candidate process parameters) till it reaches equilibrium; then checks how closely the process's output flows match the observation targets.

To evaluate the effectiveness of a tuned model, the user needs to be able to repeat this process on both the data used for tuning and on new data, and then to examine the results. The modeling system can read a standard-format file of testing data and compute corresponding model outcomes, then write a comparison-data file with the target and actual values side-by-side and in a format suitable for reading into a spreadsheet. The spreadsheet can then be used to compute statistics and draw graphs. This capability is currently implemented for the rate constant model.

The entire state of the modeling system desktop (the graphic representation of the plant, all models and targets, links to the data files) can be saved to a data file and restored later for further processing. The saved file is an ordinary text file, and can be examined using any text editor. Some information the graphical user interface does not yet display can be found (and carefully updated) in this file.

Figure 11 shows the modeling system's dialogs for editing and tuning a rate constant model. The toolbar shows the choice of icons available for building the visual representation of a plant. The example plant has models for most flows (shown as heavy lines rather than light lines) but only one flotation process (labeled "EFRM"); the user is modifying the process in the dialogs shown.



Memory left: 1048576

Process Name: EFR Model from East Fine Rougher On-line

Name	Low	High	Current
Apatite rate constant	0.01	0.3	0.01
Feed weight fraction solids	0.1	0.75	0.464004
Gangue rate constant	1e-05	0.02	0.0179346
Volume	1e+07	1e+08	5e+07
Water rate constant	1e-05	0.1	0.002

Water rate constant 1e-05 0.1 0.002

Name
AmmoniaGPM
ConcBPL
ConcPCTSolids
ConcTPH
DATE_HR
DieselGPM
FattyAcidGPM
FeedBPL
FeedPctSolids
FeedTPH
RougherPH
TailsBPL
TIME

Tune Now? ● Yes ○ No

Name of process being tuned: Rate constant process tuner

Tuneable parameters
Gangue rate constant
Volume
Water rate constant

Genetic Algorithm Parameters

Population Size: 100
Mutation Probability: 0.0010
Crossover Probability: 0.8500
Number of bits per parameter: 16
Number of bits per string: 32
Number of cycles per evaluation: 60
Number of seconds per cycle: 5
Number of Generations: 100

Selected parameters:
Apatite rate constant [0.01, 0.3]
Feed weight fraction solids [0.1, 0.75]

Figure 11 - The Modeling System desktop is shown as the user prepares to ask the system to tune parameters for a rate constant model.

4.3 Potential Improvements to the User Interface

The Neural Net and Regression models are not currently part of the interface for the modeling system. An early version of a neural net model was partly integrated with the modeling system, and the current model could be incorporated relatively easily.

As the modeling system is presently designed, data observations are handled quite independently; it is not easy to express interactions between observations. The Jordan Spears first principles, the rate constant, and the neural network models are not time series models and fit comfortably into this design (although effective prediction with the neural network model may require providing in each observation some system state for recent bits of material flow in addition to measurements for the current bit of material flowing through the plant).

Operations that require a global view of operating data through time are segregated in the cluster analysis program.

The regression model research has been very different, always treating the data as an explicit time series and applying a wide variety of series filtering techniques to condition the data and evaluate the modeling outcomes. These operations fit far less comfortably into the existing interface framework. It is not clear how much of this analysis may be needed when a regression model is integrated with the modeling system. In any case some combination of separate processing (as with the cluster analyzer) and "widening" of the data observation to include prior values (as for the neural network model) should be sufficient to satisfy the model's requirements. But if regression modeling is to be central to the modeling system, it may be a better plan to redesign the system's representation of the data to incorporate a time-series viewpoint. So adding the regression model will be more time-consuming task than adding the neural network model.

To add commands so the user can easily simulate a subcircuit rather than just one process at a time will not be difficult. The logic for such simulation is already present; what is needed is a careful extension to the user interface to define manipulations for assembling and disassembling subcircuits and for querying the modeling system to operate on them, while making it easy to see the results yet impossible to direct the system to do anything unsafe (such as to delete a process which is part of a subcircuit).

The modeling system currently offers models only for flotation processes (and the trivial parameterless mixer process). The editing procedure that builds a graphic representation of a plant understands about ball mills and hydrocyclones, but the simulating and tuning procedures do not. These features are provided to permit the development of a complete plant model. Some code was written for ball mill and hydrocyclone models but this code has not been integrated into the modeling system.

At present, the user can simulate model operation for a single flow input state, or validate model parameters using a file of state data. The interface needs to provide an additional command asking the modeling system to interpret the observations in the data file as a description of a time-varying input flow, for a more complex and useful simulation of plant behavior.

Unlike later models, the first models to be implemented (the first principles and rate constant models) exhibit no direct dependence on reagent flow rates. Instead the dependence is indirect: reagent levels affect induction time in the first principles model, and the rate constant model expects data clustering to distinguish among operating regimes with different reagent levels. Since reagent flow rates provide the only effective point of control for the real-world processes to be modeled, the modeling system needs to represent them explicitly, as it does minerals, in its flow structures. Expanding the first principles and rate constant models to incorporate direct dependencies on reagent flow rates would make it easier to use these models for automated control.

A user working seriously to tune a model to a real plant is likely to want (at least) two capabilities that are not now present in the modeling system but that would be straightforward to add. The first is an easy way to save and restore small bits of the desktop -- for example the state of one process after tuning a model using one set of parameters for the tuner genetic algorithm, in order to compare it with the model tuned using different genetic algorithm parameters. At present the only option is to save the entire desktop. The second is the ability to set or examine a single model tuning target or flow state using the data description language mentioned previously (corresponding to plant measurements). The current user interface always shows target and flow state values in a format which corresponds to the modeling system's internal representation of flows in terms of mass flow rates of mineral species.

5 CONCLUSIONS

A comprehensive modeling system for simulating phosphate flotation has been developed. The system includes a graphical user interface that can dramatically reduce the time needed to produce accurate computer models of flotation, and a blackboard system to coordinate system actions. The software system contains six different computer models of flotation. In addition, it has been designed so that computer models of other unit processes such as grinding, columns, and mixers can be easily incorporated into the system.

It is important to note that the accuracy of each of the computer models developed in this effort could be improved with better operating data. For instance, data from phosphate plants in which the feed has been sized would reduce some of the error in the models. In addition, information on the effectiveness of the conditioning phase would be valuable in the development and tuning of the models.

Although none of the six computer models is adequate to accurately model all of the situations that occur in phosphate flotation plants, taken collectively they can provide an adequate depiction of the response of the systems to be considered. Since control of the plant is the ultimate objective, even though these computer models are not perfect, they do provide an adequate framework for adaptive control.

In working with all of the models it was apparent that not all of the parameters which are important for the system were being measured. Factors such as clay content of the material going to the conditioner and/or particle size dramatically affect the short range response of the system. For example, for an identical recovery, fatty acid requirements may vary by a factor of two. The first principles model can be an important tool in exploring the effect of missing parameters because the values of these parameters can be assumed, and then the assumptions can be checked against data from the system being modeled.

Past experience with adaptive control technology provides examples of control systems (for helicopter flight, hexamine reaction, and pH titration) developed using fuzzy logic and genetic algorithms which overcame the lack of sensory data from the system. The knowledge obtained in developing the six computer models, suggests that these models will allow the development of on-line, adaptive control systems which can be used in industrial systems. Control systems can adapt to unmeasured changes if these changes have a time scale of several hours. In the phosphate flotation industry, this is likely to be at least partially true. Given the above, it is likely that an effective controller can be developed from the existing computer models and measurements that is concerned only with the long term nature of the processes. An adaptive control system should to some extent be able to accommodate the fact that not all variables are measured. However, if additional sensory information becomes available, an even more effective control system can be produced.

LITERATURE CITED

Hayes-Roth, B. (1983). The blackboard architecture: A general framework for problem solving? Technical Report KSL-87-31, Stanford University.

Hayes-Roth, B. (1985). Blackboard architecture for control. Journal of Artificial Intelligence, 26, 251-321.

Jordan, C. E., and D. R. Spears (1990). Evaluation of a turbulent flow model for fine-bubble and fine-particle flotation. Minerals & Metallurgical Processing. May: 65-73.

Geladi, P., and Kowalski, B. R. (1986). Partial least-squares regression: A tutorial. Analytica Chirnica Acta, 185, 1-17.

Ginsberg, D. W., and Whiten, W. J. (1991). Cluster analysis for mineral processing applications. Transactions of the Institution of Mining and Metallurgy. 100 (Sept-Dec): 139-146.

Karr, C. L., and B. Weck (1996). Linguistic modeling of grinding using approximate reasoning. Fluid Particle Separation Journal. 9(2): 119-128.

Karr, C. L., B. Weck, and K. Nishita (1996, submitted). Comparison of fuzzy and neural network based computer models of mineral processing equipment. Fluid Particle Separation Journal.

Kresta, J. V., MacGregor, J. F., and Marlin, T. E. (1991). Multivariate statistical monitoring of process operating performance. The Canadian Journal of Chemical Engineering, 69: 35-47.

Reuter, M. A., and van Deventer, J. S. J. (1992). The simulation and identification of flotation processes by use of a knowledge based model. International Journal of Mineral Processing, 35, 13-49.

Stanley, D. A., C. L. Karr, and B. J. Scheiner (1995). Using principal component analysis for data interpretation. Proceedings of the Annual Meeting of the Fine Particle Separation Society: 792-800.

Yeager, D. P., Brown, M. E., and Stanley, D. A. (1994). An object-oriented software library for minerals beneficiation and separation processes. Fluid/Particle Separation Journal, 7(2) 65-69.

Publication No. 02-103-132

ADAPTIVE CIRCUIT MODELING SYSTEM

USER GUIDE

ATTACHMENT TO FINAL REPORT OF PROJECT

MODELING OF THE PHOSPHATE BENEFICIATION PROCESS AS A FORERUNNER TO ADAPTIVE CONTROL

Prepared by
BCD Technologies & University of Alabama

under a grant sponsored by

FIPR
Florida Institute of
Phosphate Research

February 1997

The Florida Institute of Phosphate Research was created in 1978 by the Florida Legislature (Chapter 378.101, Florida Statutes) and empowered to conduct research supportive to the responsible development of the state's phosphate resources. The Institute has targeted areas of research responsibility. These are: reclamation alternatives in mining and processing, including wetlands reclamation, phosphogypsum storage areas and phosphatic clay containment areas; methods for more efficient, economical and environmentally balanced phosphate recovery and processing; disposal and utilization of phosphatic clay; and environmental effects involving the health and welfare of the people, including those effects related to radiation and water consumption.

FIPR is located in Polk County, in the heart of the central Florida phosphate district. The Institute seeks to serve as an information center on phosphate-related topics and welcomes information requests made in person, by mail, or by telephone.

Research Staff

Executive Director
Paul R. Clifford

Research Directors

G. Michael Lloyd Jr.	**-Chemical Processing**
Jinrong P. Zhang	**-Mining & Beneficiation**
Steven G. Richardson	**-Reclamation**
Gordon D. Nifong	**-Environmental Services**

Florida Institute of Phosphate Research
1855 West Main Street
Bartow, Florida 33830
(941) 534-7160
Fax:(941) 534-7165
http://www.fipr.state.fl.us

ADAPTIVE CIRCUIT MODELING SYSTEM

USER GUIDE

ATTACHMENT TO FINAL REPORT OF PROJECT

MODELING OF THE PHOSPHATE BENEFICIATION PROCESS AS A FORERUNNER TO ADAPTIVE CONTROL

Bernard J. Scheiner[1], Charles L. Karr[2], and Donald A. Stanley[1]
Principal Investigators

Dorian P. Yeager, Patricia S. Moore, and Marcus Brown[2]
Software Designers

[1] BCD TECHNOLOGIES
Tuscaloosa, AL 35406

[2] THE UNIVERSITY OF ALABAMA
Tuscaloosa, AL 35487-0280

Prepared for
FLORIDA INSTITUTE OF PHOSPHATE RESEARCH
1855 West Main Street
Bartow, Florida 33830

Contract Manager: Patrick Zhang

February 1997

TABLE OF CONTENTS

TABLE OF FIGURES

1. INTRODUCTION

Achieving effective long-term control of a minerals processing plant is an extremely complex problem. The number of input sensor signals and the number of possible output control actions is large, and they have many interdependencies. Those interdependencies include internal ones, such as the relationship between gas rate and recovery in a flotation column, and external ones, such as the relationship between a control action in a rougher unit and a corresponding control action in a cleaner unit from the same circuit. Thus it is imperative that the user of the system be given effective tools for managing that complexity. The Modeling System described here is an example of such a tool.

USERS: Users of this tool (and descendants of this tool) are expected to fall into two classes, which will be referred to as Designers and Operators. Designers will chiefly use the software to simulate hypothetical plants; will use the tool intermittently, in an exploratory mode; and will need a system that is flexible, easy to learn and easy to remember. Operators will use the software for well-patterned tasks to monitor and control an existing plant; will use the tool frequently; and will need a system that is robust, fast to respond, and concise to direct. The user interface developed for this tool is intended to meet the needs of both classes of users.

TASKS: There are a number of purposes for which the Modeling System is intended to be used. One is to simulate actual plant behavior, for a single process or for an entire circuit. One is to experiment with model parameter values in order to investigate potential consequences of changing aspects of the plant's operation. Another is to examine the parameters of models tuned to match data taken under different operating conditions in order to gain insight into unmeasured variables. Yet another, of course, is automated control. All of these require the software to make similar operations available to the user: to select a model type and to set or examine its parameters; to establish the state of input flows, then compute and examine the resulting output flows; to attach a list of input-flow states and then use them for simulation or (in conjunction with corresponding output-flow states) for automatically selecting or evaluating model parameters. And before any of these operations are possible, the user must be able to represent the plant to be described by the models. The Modeling System's graphical user interface is designed to enable and to coordinate all these activities.

INTERFACE: This graphical user interface provides a visual representation of a minerals beneficiation circuit with all its flows and processes. Standard symbols are used to represent various types of processes, connected by lines representing flows as in traditional materials-flow diagrams. The user interacts with these symbols by direct manipulation, as if on a drawing board, using a mouse to place, select, and move symbols. Toolbar buttons and menu options identify new-symbol types and editing operations such as *Erase.* Once the structure of the visual representation is established, this interface also enables the user to build a parallel network of mathematical models for the circuit's flows and processes. When the network of process and flow models has been constructed, this modeling network can be used for simulation and tuning experiments as described above.

STRUCTURE: The Modeling System is constructed using the Object-Oriented Library for Modeling and Control of Minerals Beneficiation Processes, a library of C++ classes that began its development in 1993 at the Tuscaloosa Research Center of the U.S. Bureau of Mines. Fundamental classes in this library have been described in some detail elsewhere; in particular, see References 7 and 9. One group of classes in this library is the Graphical Representations: the visual representation of a processing plant is composed of instances of this class type. These know their category, their location on the desktop, their connections to other Graphical Representations, and little else. Useful work with the Modeling System is mainly done by the second group of classes, the ones that implement mathematical models: Flow and Circuit Process and their descendants. But the Graphical Representation instances are fundamental to the user interface's operation: they form the skeleton supporting Model flesh. A Graphical Representation object can exist without a corresponding Model, but a Model object must have a Graphical Representation. Various kinds of process Model classes have very different internal structures and operations, but all process Models “cycle”: that is, look at their input flows and internal states and compute new output-flows and internal states. All more complicated model actions such as tuning are built from this basic step. User requests and Modeling System actions are coordinated by a third group of classes, the SupervisorWindow and Blackboard classes. A SupervisorWindow object handles the desktop itself and manages the networks of Model and GraphicalRepresentation objects; the Blackboard classes are designed to schedule and coordinate not only Model-object operations such as simulation but also input from external sensors, decision-making by software Controller objects, and so on.

CURRENT STATUS: The Modeling System as it exists today is still a research object, not a production tool. As the research project of which it is part has shifted direction, some parts of the software have fallen out of use; other parts prove to need more flexibility than their early design can provide. Limited resources have often been spent on establishing new capabilities, rather than maintaining old ones or ensuring that the new ones are fully integrated into the Modeling System. An example of disuse is software process control: one of the first capabilities built into the Modeling System, when its only model was of column flotation, was fuzzy process control: see Reference 8. But the focus turned to modeling conventional flotation cells and to building an effective user interface; the column flotation model at present is broken, and control capability has not yet been built into the newer models. An example of a research design decision, made when the focus was on first-principles models, that proved less than clairvoyant is the omission of conditioning reagents from the definition of a Flow object. An area in which the Modeling System’s capabilities are at present incomplete is handling models for circuits with several processes: a user of the current system who wants to explore real plant data will work on one process at a time, and may prefer a graphical representation in which that process is not connected to others.

GUIDE: The Modeling System has associated standalone programs for preprocessing actual plant operating data, and a standard file form for such data, all described later. This User Guide is organized into several parts:

- a hands-on example session, which provides an overview of the Modeling System environment and its capabilities;
- reference sections describing menu and toolbar selections, mouse operations, dialog screens, standard format for files of plant operating data, and the standalone Cluster Analyzer and Viewer programs;
- some annotated example files.

REFERENCES:

1. Finch, J. A., and G. S. Dobby (1990). *Column Flotation.* Pergamon Press, Oxford.
2. Ginsberg, D. W., and W. J. Whiten (1991). Cluster analysis for mineral processing applications. Transactions of the Institution of Mining and Metallurgy. 100 (Sept-Dec): 139-146.
3. Jordan, C. E., and D. R. Spears (1990). Evaluation of a turbulent flow model for fine-bubble and fine-particle flotation. Minerals & Metallurgical Processing. May: 65-73.
4. Scheiner, B. J., C. L. Karr, and D. A. Stanley (1996). Modeling of the phosphate beneficiation process as a forerunner to adaptive control: Final Report to the Florida Institute of Phosphate Research, Bartow, FL.
5. Stanley, D. A., C. L. Karr, and B. J. Scheiner (1995). Using principal component analysis for data interpretation. Proceedings of the Annual Meeting of the Fine Particle Separation Society: 792-800.
6. Yeager, D. P., Brown, M. E., and Stanley, D. A. (1994). An object-oriented software library for minerals beneficiation and separation processes. Fluid/Particle Separation Journal, 7(2) 65-69.
7. Yeager, D. P., C. L. Karr, and D. A. Stanley, (1995). “Column Flotation Tuning Using a Genetic Algorithm,” presented at the 1995 Annual Meeting of the Society of Metallurgical Engineers, Denver, Colorado.
8. Yeager, D. P., D. A. Stanley, and Alexander May (1995). An Object-oriented Framework for Adaptive Fuzzy Circuit Control. Proceedings of the IEEE Conference on Computational Intelligence.

9. Yeager, D. P. (1995). Final Report to the United States Bureau of Mines, Tuscaloosa Research Center, for contract P3250030.

2. HANDS-ON I: EXPLORE THE GRAPHICAL INTERFACE

The Adaptive Circuit Modeling System desktop lets a plant designer or engineer explore a variety of questions about plant operation. This exploration must begin from a representation of the plant. To create this representation, first start the program. You will see an empty desktop, with a window labeled "Posted Data" in the middle. Later, that window will display output from tuning and simulating plant processes; for now, grab its title bar and move it out of the way. You will need it to be bigger too, so if there is room on your screen you can resize it now too.

The representation of your plant will consist of symbols ("icons") for plant processing units, connected by lines representing flows of material. The desktop toolbar has five buttons for the five kinds of processing units it knows about: for ball mill, for hydrocyclone, for flotation column, for conventional flotation cell, and for mixer. Click one of these buttons, then click anywhere on the desktop to place an instance of the icon you selected. You can move an icon: grab it by depressing the left mouse button while the cursor is positioned in the icon's interior, then (still holding the mouse button down) move the mouse to drag the icon where you want it; button-up drops the icon in its new location. If you do the drag-and-drop motion starting outside any icon interior, you will see a dotted rectangle which disappears when you let the mouse button up. This is a selection rectangle: any icon inside the rectangle when you release the button turns red, to signify "selected". Single-click the icon to un-select it again. Selection will be used in later versions of the desktop to identify collections of processes ("subcircuits") for simulating or tuning together, and for saving and restoring bits of desktop state (for example, a process tuned to match each of several different days' operating data).

To build your plant representation, you will need two more buttons from the toolbar. First is the erase button: after you click the erase button, clicking on any desktop icon erases that icon. Second is the flow button. which requires you to use the same drag-and-drop motion as *move* and *select.* Click the flow button, then position the cursor at one end of the flow you want to draw, press the left mouse button, drag the cursor to the other end of the flow, and release the button. If one end of the flow is close enough to a connection point on a process icon, a bright green dot will be drawn to show that the modeling system recognizes that the flow is an input or output of the process. If you intended a connection but didn't get quite close enough, draw a small selection box around both the flow end and the process connection point: this asks to make a connection if there is none, or to break a connection that has already been made. The desktop also checks for possible connections whenever you move the endpoint of a flow. The direction of a flow is determined by its process connections if it has any (the feed flow connections are always on the left side of the icons). If the flow is not connected to a process, its direction is determined by the way you drew it (button-down at the source end, button-up at the destination end). The desktop does not let you connect one flow directly to another: either the parts should be drawn as a single flow, or you need to use one or more mixers to combine the parts into one resulting flow.

You can grab a free end of a flow, or one of its stairstep line segments, and move it in the same way you moved a process icon. The desktop gives you more flow segments as needed, and collapses segments when they partly coincide or define a single line, so you can pull your flow lines around however you want to make a pleasing picture of your plant's connections.

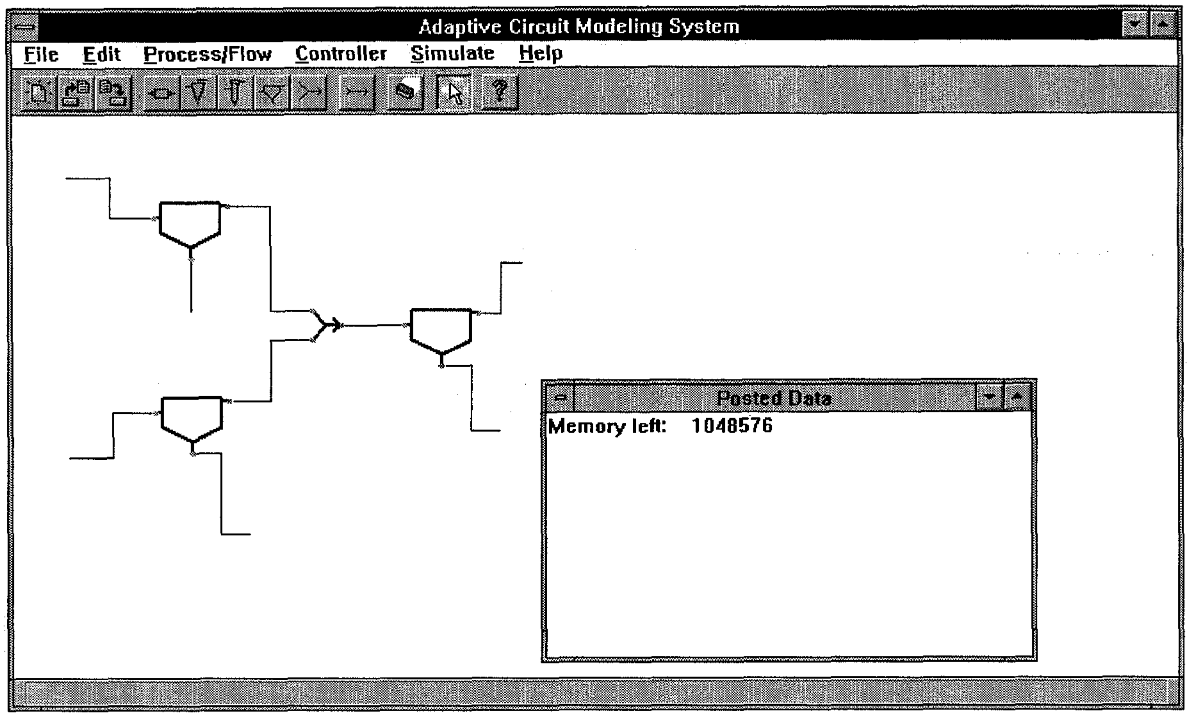
All the toolbar buttons let you take a single action: if you want to erase four icons you must press the erase button, then click on a desktop item, four times, The default action state is represented by the arrow button: click this if you press the erase button by mistake. You can clear the entire desktop with the leftmost button, or by using the menu selection **File / New Circuit.** Save the entire current contents of the desktop to a disk file with the save button or the menu choice **File / Save Circuit.** Restore a saved desktop from a file with the load button or the menu choice **File / Load Circuit.** The modeling system uses a standard Windows dialog to get the name of the disk file to contain the desktop, but by default it expects a filename with extension ".cir" (for circuit). One disk file can contain just one saved desktop. The saved file is an ordinary text file: you can read it, and even edit it if you want.

The examples that follow are based on a circuit with three flotation cells, two rougher cells feeding an amine separator. To draw this circuit, clear the desktop (erase each process icon and flow line, or press the New Circuit button at the left of the toolbar. Then:

1. Click, move cursor to left top, click to put icon on desktop. This icon represents North Rougher.
2. Click, move cursor to left middle, put icon on desktop. This icon represents South Rougher.
3. Click, move cursor to center, put icon on desktop. This icon represents the mixer combining both rougher concentrate flows into the Amine feed flow.
4. Click, move cursor to right center, put icon on desktop. This icon represents Amine.
5. Now draw nine flows: feed, tailings, and concentrate flows for each flotation-cell process. All will be connected to at least one process, so you can draw flows starting from either end. For each one,

   a. Click.
   b. Move the cursor to one end of the flow (perhaps to a process connection point).
   c. Press the left mouse button.
   d. Move the cursor to the other end of the flow.
   e. Release the mouse button.
   f. Check for the green dot(s) showing that the flow is properly connected to its process(es). Move the flow ends as necessary to help the Modeling System recognize the connection.

Your screen should now look something like Figure 2-1. Save the desktop.



Memory left: 1048576

Figure 2-1. Desktop shows graphical representation of example plant.

3. HANDS-ON II: ESTABLISH THEN USE FLOW AND PROCESS MODELS

3.1 Establish Flows

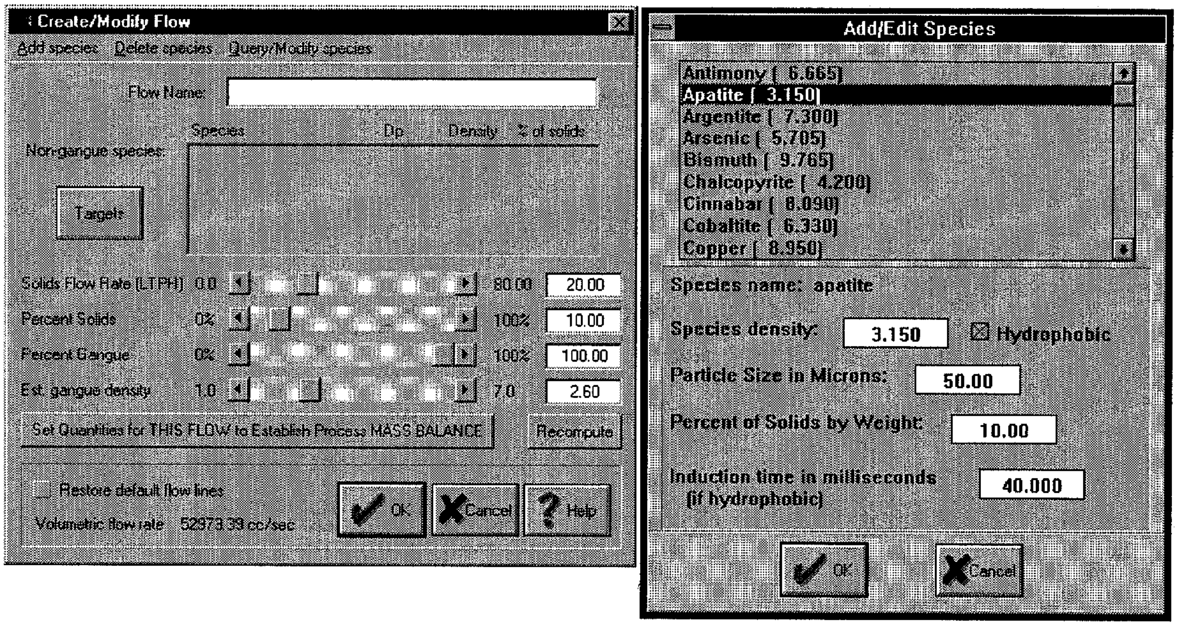
In order to do anything interesting with the plant whose layout you have just created, you must describe the contents of the flows and establish models for the processing units. The Modeling System demands that all flows in the plant have the same list of potential mineral constituents, and it assumes that two of the flow constituents are called "gangue" and "water". Add one unsized apatite constituent to the North Rougher feed flow:

- Double-click the left mouse button on any segment of the North Rougher feed flow. You will see a screen like the one at the left of Figure 3-1. Fill in the *Flow Name* field with "N Rougher Feed".
- Next click on the menu choice Add Species to bring up a list of minerals the system has data for. Select *apatite*. Your dialog screen will look like the one at the right of Figure 3-1.

Leave the default values in place to follow the subsequent example. Or you can adjust apatite density to match an ore you are familiar with, or change particle size or induction time as appropriate to experiment with the first-principles model.

- When you click *OK*, you will return to the flow-editing screen. There you will see that *% of Solids* in the *Species* display box is the value from the Add/Edit Species screen, and *Percent gangue* has been decreased so that percents for all solids add to 100.

You can use the sliders or type values directly into the edit fields to change gangue density, weight fraction solids, and total solids mass flow rate. You can make the system recalculate *Volumetric flow rate* after you have changed any of these values by clicking the Recompute button. To edit a mineral, double-click on its entry line in the *Species* display box, or single-click to highlight its entry and then use the menu.



Flow Name:

Non-gangue species:

Solids Flow Rate [LTPH] 0.0 – 80.00 20.00

Percent Solids 0% – 100% 10.00

Percent Gangue 0% – 100% 100.00

Est. gangue density 1.0 – 7.0 2.60

□ Restore default flow lines

Volumetric flow rate 52973.39 cc/sec

Antimony [6.665]
Apatite [3.150]
Argentite [7.300]
Arsenic [5.705]
Bismuth [9.765]
Chalcopyrite [4.200]
Cinnabar [8.090]
Cobaltite [6.330]
Copper [8.950]

Species name: apatite

Species density: 3.150 ☒ Hydrophobic

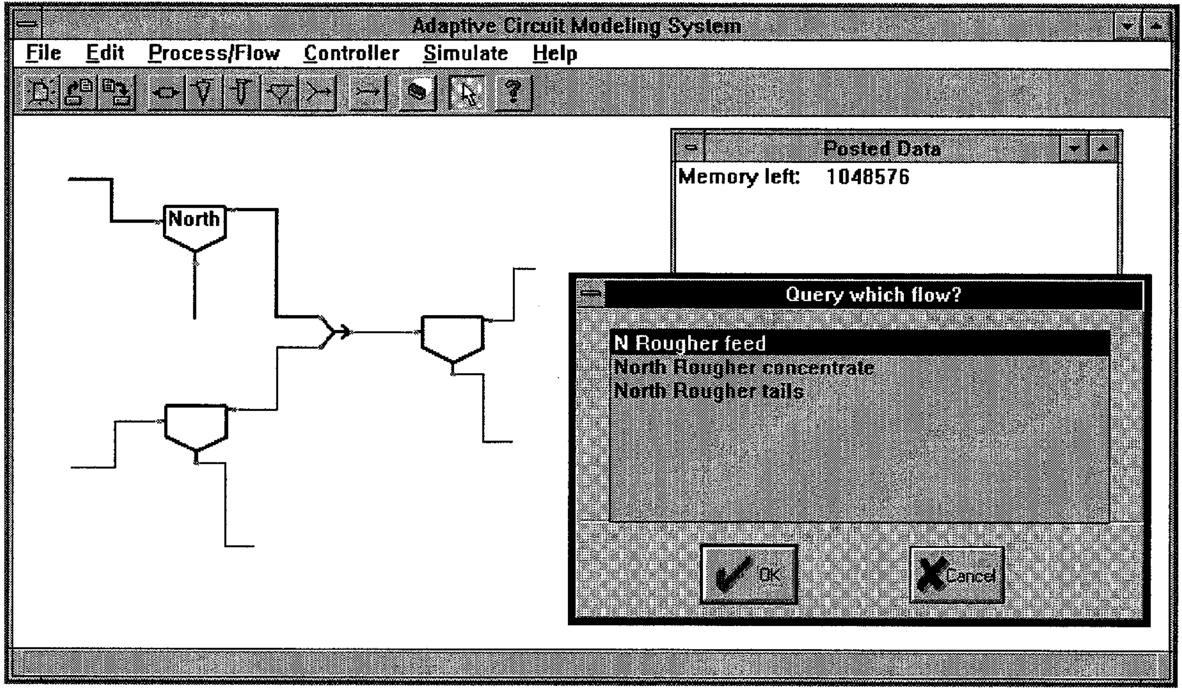
Particle Size in Microns: 50.00

Percent of Solids by Weight: 10.00

Induction time in milliseconds (if hydrophobic) 40.000

Figure 3-1. Screens for creating and editing a Flow model.

Checking the box *Restore default flow lines* asks the modeling system to redraw the flow on the desktop using its default three line segments. This can be useful if moving flows or process icons produces a result you don’t like. The *Targets* button will be described later. When you click *OK* to exit the dialog, you will see that the flow is drawn on the desktop with a dark rather than a light line: this tells you there are values (that is, a model) associated with that flow. You can examine or edit any flow model either by double-clicking the flow’s line representation on the desktop or by selection the menu items **Process/Flow / Query/modify a Flow Model.** You will see a screen like Figure 3-2 with one name “N Rougher Feed” in the list of possible flows. If you now click OK you will find yourself right back at the dialog of Figure 3-1 that you used when creating the flow, and the N Rougher feed flow should look like the screen in Figure 3-3.

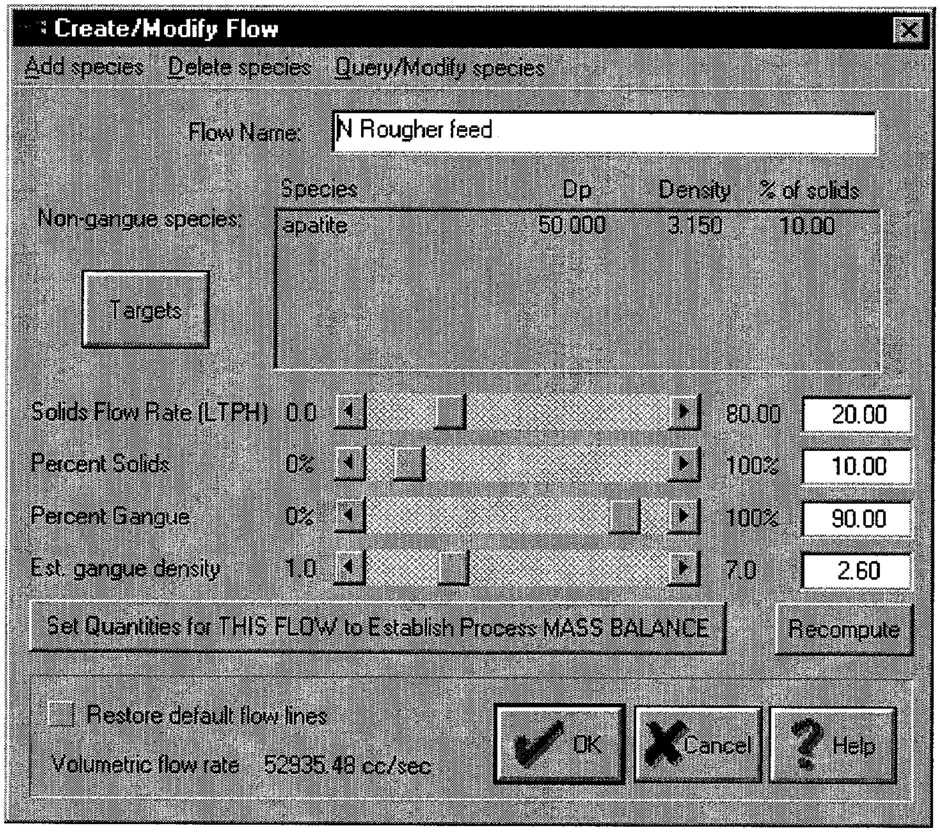

Adaptive Circuit Modeling System
File Edit Process/Flow Controller Simulate Help
North
Posted Data
Query which flow?
OK
Cancel

Memory left: 1048576

Query which flow?

- N Rougher feed
- North Rougher concentrate
- North Rougher tails

Figure 3-2. Selection screen asks user to choose from a list of Flow models.


Create/Modify Flow
Add species Delete species Query/Modify species
Targets
Set Quantities for THIS FLOW to Establish Process MASS BALANCE
Recompute
OK
Cancel
Help

Flow Name: N Rougher feed

Non-gangue species:

Species	Dp	Density	% of solids
apatite	50.000	3.150	10.00

Solids Flow Rate (LTPH)	0.0	80.00	20.00
Percent Solids	0%	100%	10.00
Percent Gangue	0%	100%	90.00
Est. gangue density	1.0	7.0	2.60

□ Restore default flow lines

Volumetric flow rate 52935.48 cc/sec

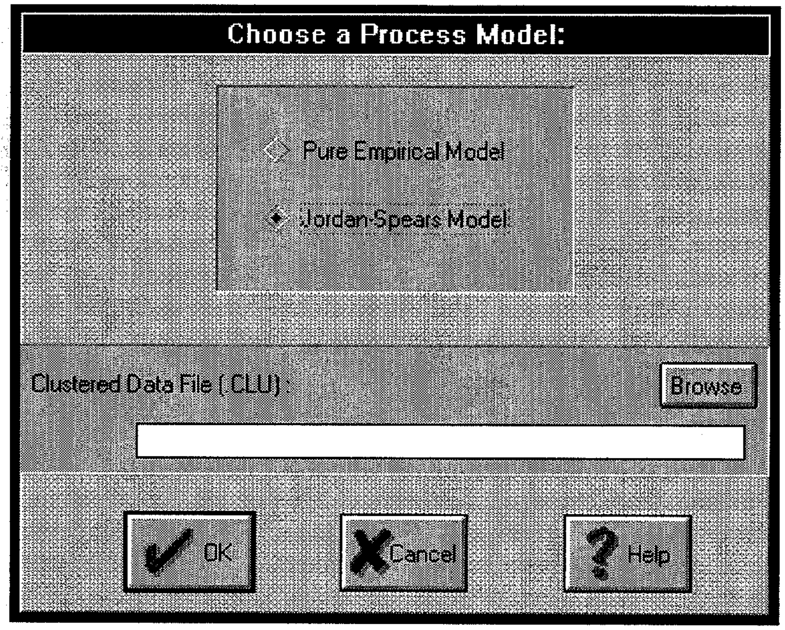
Figure 3-3. Flow Model for feed to North Rougher flotation process.

In the same way, establish the South Rougher feed flow. Now you can repeat this work for all seven other flow icons in the circuit, or you can let the system create the output flows for each processing unit as you create models for the three flotation cells and for the mixer.

3.2 A First-Principles Flotation-Cell Model

Now you are ready to create a model for the North Rougher. Double-click the upper-left ▽ icon on the desktop. If you have not created flow models for the output flow icons attached to this process icon, the system first asks if you really want to create a new flow for the concentrate and then one for the tailings. You do, so click the *OK* button in answer to each question. The system will automatically assign names to these flows which may not be appropriate, but you can edit the flows later to change the names (or any other values). Just remember that all flows must consist of the same species, so don't add or delete minerals.

In this version of the modeling system, there are two possible models for a conventional flotation cell: the first-principles model following the work of Jordan and Spears, and a Pure Empirical Rate Constant model. As it is now implemented in the system, the Rate Constant model demands a data file containing plant operating data to be used in determining model parameters; so the first model you create will be a first-principles model. When the output flows are created, the system will present a screen like Figure 3-4; check the box next to *Jordan-Spears* Model, then click *OK*. You will see a screen like Figure 3-5. This screen is described in some detail in Section 7.1.


Choose a Process Model:

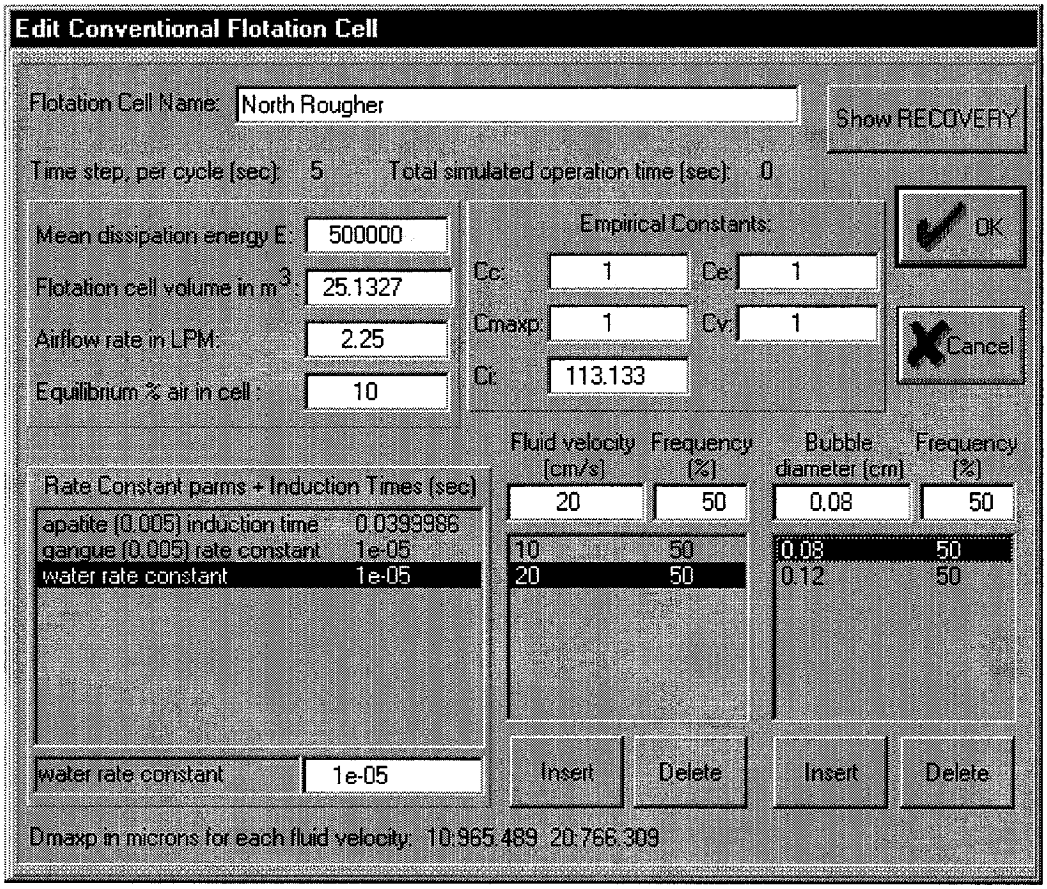
○ Pure Empirical Model

● Jordan-Spears Model

Clustered Data File (.CLU) :

Figure 3-4. Screen used to specify model type for conventional flotation cell.

Fill in the process name ("North Rougher"). Then you can set the Jordan-Spears constants as you want (for this model, the default values for the empirical constants are unlikely to be satisfactory). To adjust the fluid-velocity and bubble-size distributions, use the edit fields above and the buttons below the distribution display boxes. When you click on a line in one of the display boxes, the line's values appear in the box's edit fields. Click the box's *Delete* button to delete the selected line, or click the box's *Insert* button to insert a line whose values match the numbers in the edit fields. If the bubble diameter (or fluid velocity) in the edit field matches an existing entry, that entry's frequency will be updated; otherwise a new line will be added to the distribution list. To modify a rate constant parameter, select its line in the listbox; a new value can then be entered in the edit line below the listbox. When you select another line, you will see the updated value entered in the list. The displayed induction times are computed from particle diameters and the model induction time parameter C_i; you cannot modify these values directly, but they will be recomputed whenever the model parameter is changed. As in updating a rate constant parameter, you must select another field in order to tell the Modeling System that you have finished entering a new value.



Edit Conventional Flotation Cell

Flotation Cell Name: North Rougher

Time step, per cycle (sec): 5 Total simulated operation time (sec): 0

Mean dissipation energy E: 500000

Flotation cell volume in m^3: 25.1327

Airflow rate in LPM: 2.25

Equilibrium % air in cell : 10

Empirical Constants:

Cc: 1 Ce: 1

Cmaxp: 1 Cv: 1

Ci: 113.133

Rate Constant parms + Induction Times (sec)

```
apatite (0.005) induction time    0.0399986
gangue (0.005) rate constant      1e-05
water rate constant               1e-05
```

water rate constant 1e-05

Fluid velocity (cm/s)	Frequency (%)
20	50
10	50
20	50

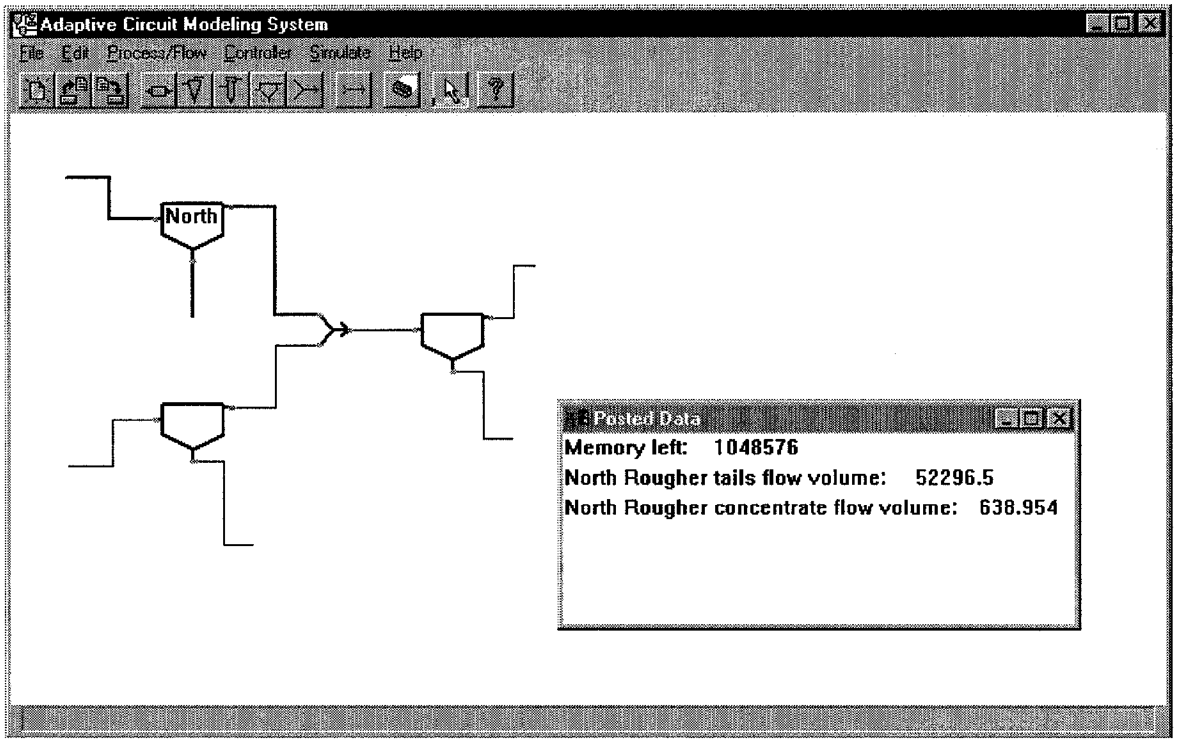
Bubble diameter (cm)	Frequency (%)
0.08	50
0.08	50
0.12	50

Dmaxp in microns for each fluid velocity: 10:965.489 20:766.309

Figure 3-5. Screen used to specify model parameters for first-principles (Jordan-Spears) model for conventional flotation cell.

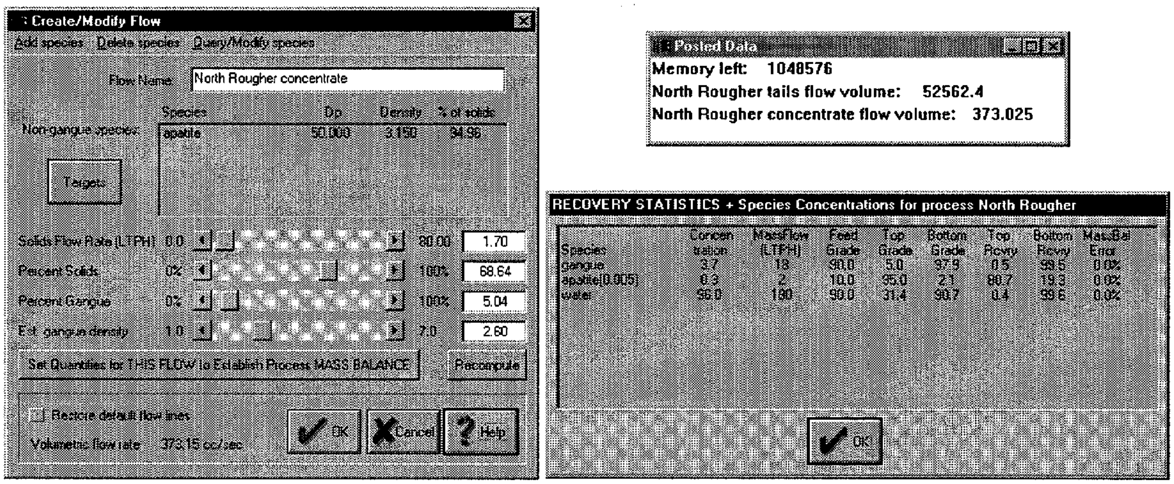
When the process parameters have been set, click *OK* to exit the process-edit dialog. Notice that the process's icon on the desktop is now labeled with the first few characters of the process name. This lets you tell which processes have models and which do not. Even if the model name is empty, the icon will be marked: in this case the label will be "*".

Now you can watch the process operate. Choose menu items **Simulate / Begin Simulation, instantaneous Cycling;** the system will ask you what time step to use for the simulation (how many seconds in each simulation cycle) and will ask you to choose which process to simulate. Right now you have only one possible process, so just click OK. The modeling system will assume that the feed flow to the process is constant and has the values you entered for “N Rougher Feed”; it will then compute tails and concentrate flows and resulting process state using the process parameters you established. While the simulation runs, the display box labeled Posted Data will contain two new lines reporting the computed volume for the tails and concentrate flows, as shown in Figure 3-6. At first these values will change rapidly, but when they stop changing the simulation has found the equilibrium state for the system for the given feed. You can view the solution by clicking the Show RECOVERIES button, and by looking at the values attached to the output flows. First stop the simulation (menu choice **Simulate / End Simulation,** then the process-choice screen). Then double-click on each output flow to see its resulting state. For the system default values for all flows and process variables, the simulation results are as shown in Figure 3-7. Now you can try modifying process parameters to see their effect on the computed process equilibrium state.


Adaptive Circuit Modeling System
File Edit Process/Flow Controller Simulate Help
North
Posted Data

Memory left: 1048576
North Rougher tails flow volume: 52296.5
North Rougher concentrate flow volume: 638.954

Figure 3-6. Posted Data window shows output flow volumes while process is being simulated.


Create/Modify Flow
Add species Delete species Query/Modify species
Targets
Set Quantities for THIS FLOW to Establish Process MASS BALANCE
Recompute
OK
Cancel
Help
Posted Data
OK

Flow Name: North Rougher concentrate

	Species	Dp	Density	% of solids
Non-gangue species:	apatite	50.000	3.150	34.96

Solids Flow Rate (LTPH)	0.0		80.00	1.70
Percent Solids	0%		100%	68.64
Percent Gangue	0%		100%	5.04
Est. gangue density	1.0		7.0	2.60

□ Restore default flow lines

Volumetric flow rate 373.15 cc/sec

Memory left: 1048576
North Rougher tails flow volume: 52562.4
North Rougher concentrate flow volume: 373.025

RECOVERY STATISTICS + Species Concentrations for process North Rougher

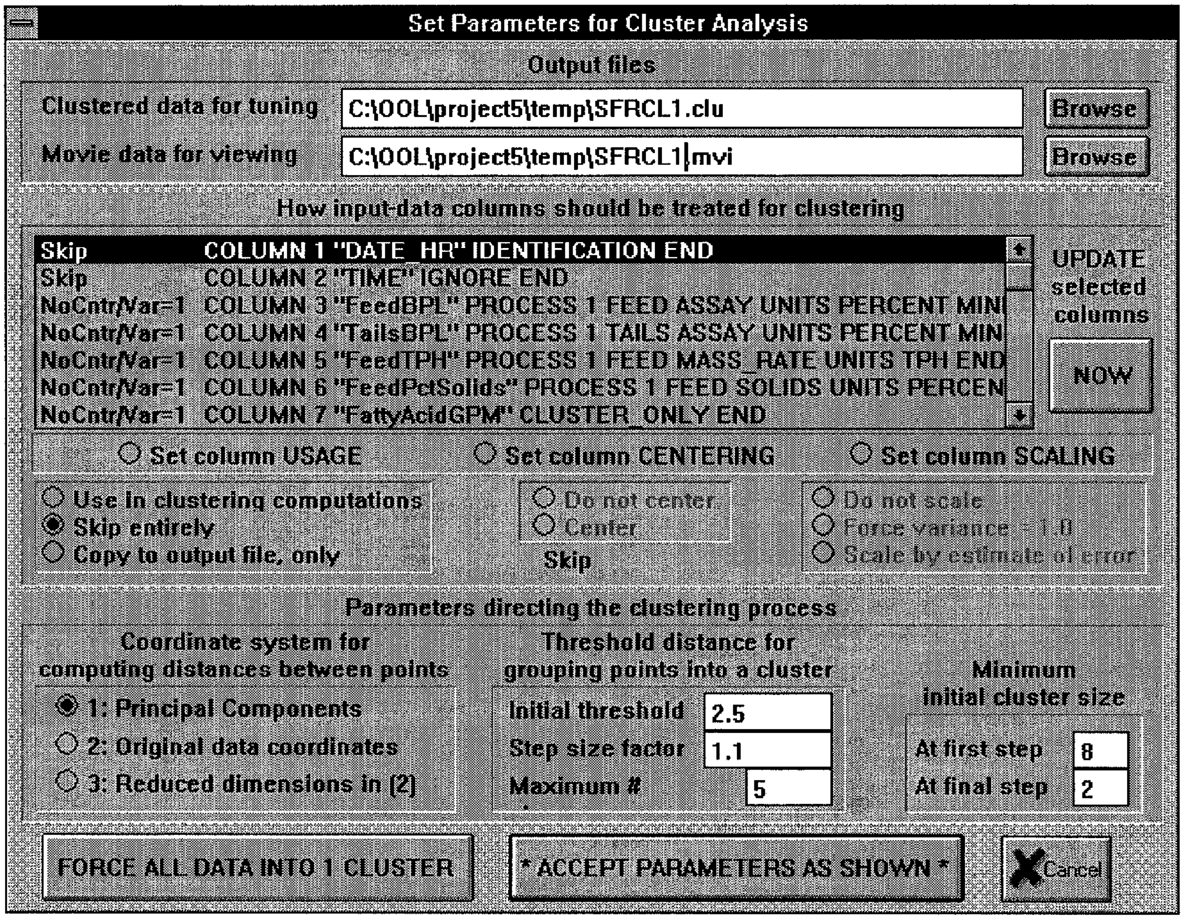
Species	Concentration	MassFlow (LTPH)	Feed Grade	Top Grade	Bottom Grade	Top Rcvry	Bottom Rcvry	MassBal Error
gangue	3.7	18	90.0	5.0	97.9	0.5	99.5	0.0%
apatite[0.005]	0.3	2	10.0	95.0	2.1	80.7	19.3	0.0%
water	96.0	180	90.0	31.4	90.7	0.4	99.6	0.0%

Figure 3-7. State of process and Concentrate flow after simulating default first-principles flotation cell model.

3.3 A Pure Empirical Rate Constant Model

Establishing a Pure Empirical Rate Constant model requires more steps than creating a Jordan-Spears model. A Pure Empirical Rate Constant model must be based initially on a file of empirical data, and the model must be tuned before it can be used. Standard data file format is described in Section 8.1, and this example will make use of the annotated file SFR.TXT shown in section 9.1. This tiny data set consists of 20 records each describing what happens to an hour of feed entering a fine rougher flotation cell. The input file is composed of measured and computed values plus a descriptive prefix. The tuning-data file to be created from it by the Cluster Analyzer has most of the same information plus decisions about how each data column should be used; plus results from the principal components analysis and cluster analysis; plus some identifying comments; and less outlier points. An example clustered-data file based on SFR.TXT is SFRCLl.CLU in section 9.2.

To create the clustered-data file, run the Cluster Analyzer. It has just three menu choices; select the leftmost one, **Cluster.** The Cluster Analyzer will ask you to identify its input file, using a standard Windows file-selection dialog; then you will see a screen like Figure 3-8, whose fields are described in Reference Section 8.2. Besides the clustered-data output file, the Cluster Analyzer will write a Movie file so you can examine the outcome of the cluster analysis, see how the search for clusters was actually performed, and examine the results of the principal components analysis of your data.



Set Parameters for Cluster Analysis

Output files

Clustered data for tuning: C:\OOL\project5\temp\SFRCL1.clu

Movie data for viewing: C:\OOL\project5\temp\SFRCL1.mvi

How input-data columns should be treated for clustering

```
Skip           COLUMN 1 "DATE_HR" IDENTIFICATION END
Skip           COLUMN 2 "TIME" IGNORE END
NoCntr/Var=1   COLUMN 3 "FeedBPL" PROCESS 1 FEED ASSAY UNITS PERCENT MIN
NoCntr/Var=1   COLUMN 4 "TailsBPL" PROCESS 1 TAILS ASSAY UNITS PERCENT MIN
NoCntr/Var=1   COLUMN 5 "FeedTPH" PROCESS 1 FEED MASS_RATE UNITS TPH END
NoCntr/Var=1   COLUMN 6 "FeedPctSolids" PROCESS 1 FEED SOLIDS UNITS PERCEN
NoCntr/Var=1   COLUMN 7 "FattyAcidGPM" CLUSTER_ONLY END
```

○ Set column USAGE ○ Set column CENTERING ○ Set column SCALING

○ Use in clustering computations
● Skip entirely
○ Copy to output file, only

○ Do not center
○ Center
Skip

○ Do not scale
○ Force variance = 1.0
○ Scale by estimate of error

Parameters directing the clustering process

Coordinate system for computing distances between points
● 1: Principal Components
○ 2: Original data coordinates
○ 3: Reduced dimensions in [2]

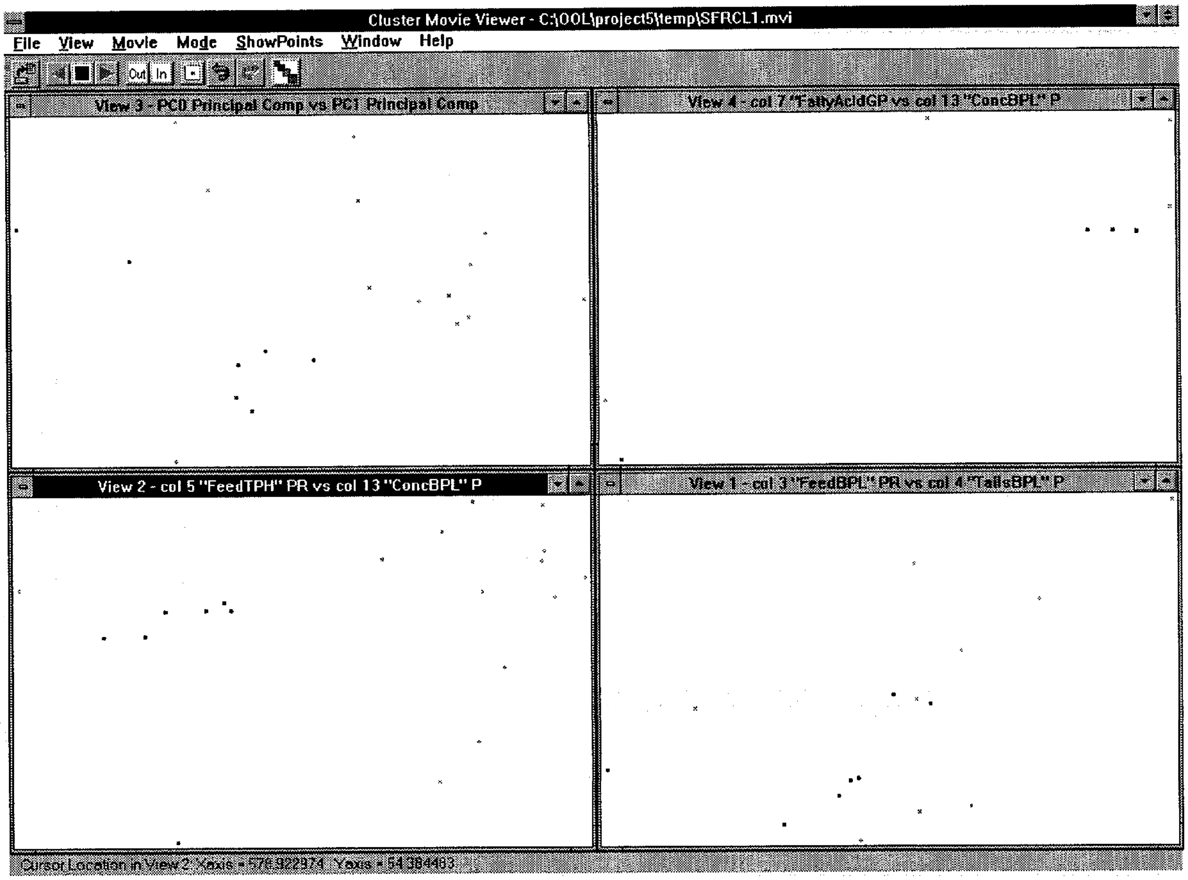
Threshold distance for grouping points into a cluster
Initial threshold: 2.5
Step size factor: 1.1
Maximum #: 5

Minimum initial cluster size
At first step: 8
At final step: 2

Figure 3-8. Cluster Analyzer output files, input-column handling, and clustering controls are all declared here.

Fill in names for the output files, as shown in Figure 3-8. Now, following the procedure described in the reference section, set usage for the IDENTIFICATION column to *Copy,* and set usage for all other columns to *Center* and *ForceUnitVariance;* then override parameter settings in the bottom section by clicking on the **FORCE ALL DATA INTO 1 CLUSTER** button at the bottom of the screen. This is an OK button: the analysis will begin, and messages about it will be posted to the main window. Clustering will be done in principal-components coordinates; and the Cluster Analyzer will ask you how many dimensions to use for your data points. A usual way to answer that question is to look at the eigenvalues of the principal components; choose a number that includes all the large eigenvalues but none of the tiny ones. In this case, all answers lead to the same results.

When the analysis is complete, use the Data Viewer to look at the results. Reference Section 8.3 describes the Viewer menu choices, toolbar buttons, and mouse operations. The animation of the search for clusters will not be very interesting, since you asked to force all points into a single cluster; but the first animation in the movie file, which flashes each data point once in sequence, can be very useful. Click the Open toolbar button and select your newly-created movie file; then use the View / New Movie View menu item to open several scatterplot subwindows. Making each subwindow in turn the active one, select a different set of plot axes for each subwindow with View / New Axes for View. The default subwindow title bar does not list the plot axes, but selecting axes fills the bar; so you may want to do it even for your first subwindow. Window / Tile lets you see all the scatter-plots at once, as in Figure 3-9. Now when you run the animation in single-step mode (see Mode menu choice and triangle toolbar buttons) you will see each point highlighted in all the scatterplots at once.

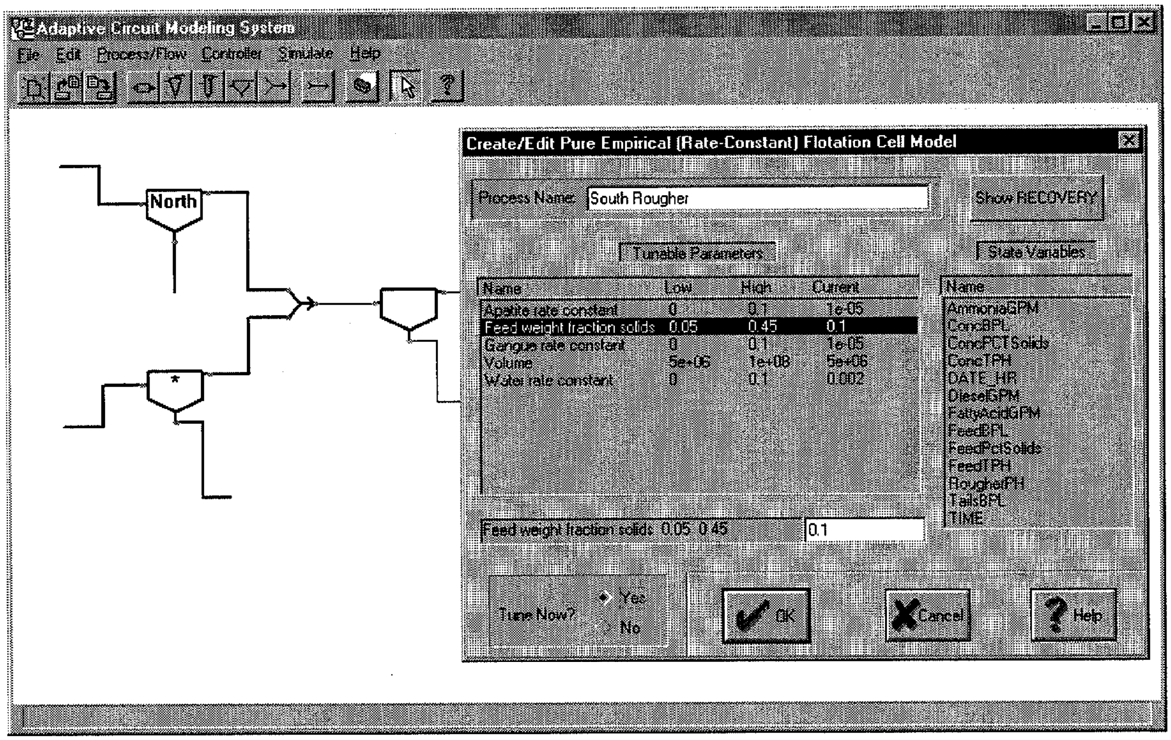

Cluster Movie Viewer - C:\OOL\project5\temp\SFRCL1.mvi
File
View
Movie
Mode
ShowPoints
Window
Help
Out
In
View 3 - PC0 Principal Comp vs PC1 Principal Comp
View 4 - col 7 "FattyAcidGP vs col 13 "ConcBPL" P
View 2 - col 5 "FeedTPH" PR vs col 13 "ConcBPL" P
View 1 - col 3 "FeedBPL" PR vs col 4 "TailsBPL" P

Cursor Location in View2: Xaxis = 576.922974 Yaxis = 54.384483

Figure 3-9. Data Viewer shows four scatterplots based on tiny example data file.

Now that you have a clustered data file, you can create a Pure Empirical model for the South Rougher flotation cell on your desktop. You ask to create the model for this process by double-clicking on the process icon, just as before. If you have not yet created models for the concentrate and tailings flows, dialog boxes will ask you for permission to create them. You will again be asked to specify which model you want to use for this process: the Jordan-Spears model or the Empirical model. This time, choose Empirical model and select the clustered-data file you just created, either by typing its name in the edit line or by browsing through the appropriate directories, just as most Windows software allows you to do when opening files.

The Modeling System will now present a large dialog window as in Figure 3-10, listing default process parameters and proposing a process name. To modify a parameter, select its line in the listbox; a new value can then be entered in the edit line below the listbox. When you select another line, you will see the updated value entered in the list. You can also decide whether to tune the model immediately to the data-file targets. If you do not choose to tune it now, you can tune it later by choosing that option from the menus at the top of the window.


Adaptive Circuit Modeling System
File Edit Process/Flow Controller Simulate Help
North
*
Create/Edit Pure Empirical (Rate-Constant) Flotation Cell Model
Show RECOVERY
Tunable Parameters
State Variables
Tune Now?
Yes
No
OK
Cancel
Help

Process Name: South Rougher

Name	Low	High	Current
Apatite rate constant	0	0.1	1e-05
Feed weight fraction solids	0.05	0.45	0.1
Gangue rate constant	0	0.1	1e-05
Volume	5e+06	1e+08	5e+06
Water rate constant	0	0.1	0.002

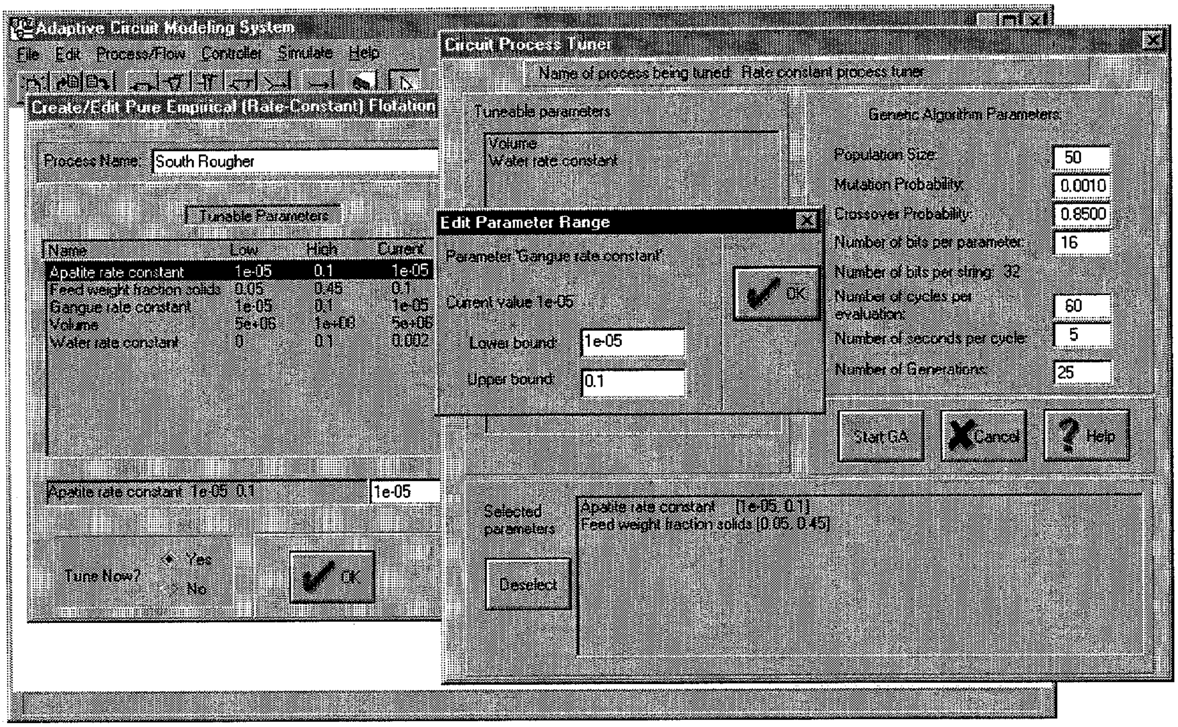
Name
AmmoniaGPM
ConcBPL
ConcPCTSolids
ConcTPH
DATE_HR
DieselGPM
FattyAcidGPM
FeedBPL
FeedPctSolids
FeedTPH
RougherPH
TailsBPL
TIME

Feed weight fraction solids 0.05 0.45 0.1

Tune Now? ● Yes ○ No

Figure 3-10. Process parameters for Pure Empirical flotation cell model.

Once you begin tuning the process, you will see a dialog window listing the tunable parameters, as in Figure 3-11. These parameters will include attributes such as “Volume” and “Gangue Rate Constant.” You must select which of these you want to use to tune the model. Double click on a desired parameter to select it; this will bring up a small dialog box listing upper and lower bounds for that parameter. These must be present, so adjust them as needed: it is impossible to set a parameter to a value outside this range. You can move a parameter from the selected list back to the unselected list: highlight its line and click the *Deselect* button. You must do this, then select the parameter again, in order to re-edit the parameter's range.


Adaptive Circuit Modeling System
File Edit Process/Flow Controller Simulate Help
Create/Edit Pure Empirical (Rate-Constant) Flotation
Tunable Parameters
OK
Circuit Process Tuner
Edit Parameter Range
OK
Start GA
Cancel
Help
Deselect

Process Name: South Rougher

Name	Low	High	Current
Apatite rate constant	1e-05	0.1	1e-05
Feed weight fraction solids	0.05	0.45	0.1
Gangue rate constant	1e-05	0.1	1e-05
Volume	5e+06	1e+08	5e+06
Water rate constant	0	0.1	0.002

Apatite rate constant 1e-05 0.1 1e-05

Tune Now? ● Yes ○ No

Name of process being tuned: Rate constant process tuner

Tuneable parameters

Volume
Water rate constant

Parameter 'Gangue rate constant'

Current value 1e-05

Lower bound: 1e-05

Upper bound: 0.1

Genetic Algorithm Parameters:

Population Size: 50

Mutation Probability: 0.0010

Crossover Probability: 0.8500

Number of bits per parameter: 16

Number of bits per string: 32

Number of cycles per evaluation: 60

Number of seconds per cycle: 5

Number of Generations: 25

Selected parameters

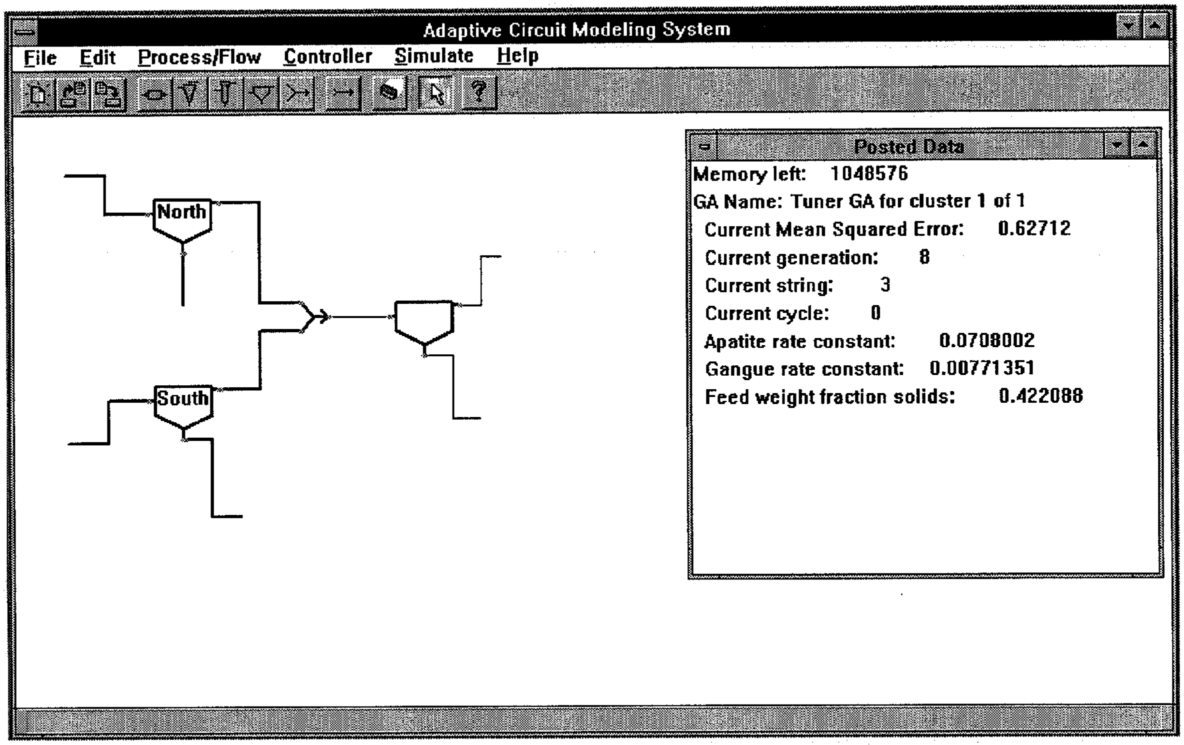
Apatite rate constant [1e-05, 0.1]
Feed weight fraction solids [0.05, 0.45]

Figure 3-11. Tune a Pure Empirical flotation cell model using clustered data file from Section 9.2.

Initially you should also adjust the genetic algorithm's Population Size and Number of Generations parameters, since the default values specify an analysis that may take too long to complete. The time required depends on the number of generations multiplied by the population size; for an initial demonstration choose 5 or 10 rather than 25 or 50 for each. This will not produce good answers, but it will illustrate the tuning process. The Modeling System will tune one set of model parameters for each cluster in the data file. It evaluates each potential set of model parameters as follows. For each observation (i.e. row of the data file), the Modeling System adjusts the process's input flows to match the observation values; cycles the process (using the candidate process parameters) till it reaches equilibrium; then checks how closely the process's output flows match the observation's outputs. The candidate parameter set is given a score which is the average of its performance for all observations. The genetic algorithm then directs the search for the best candidate parameter set: starting with Population-Size parameter sets, generated at random, it evaluates them all, selects the most successful ones and shuffles their parameters, then repeats these steps Number-of-Generations times.

To start the tuning search, click on the "Start GA" button. The Posted Data window illustrated in Figure 3-12 shows which cluster is being tuned, and which "individual" (which parameter set) within which generation is being evaluated. It also shows the current best mean squared error found for this cluster, and the values for the tunable parameters which yield that value. The flag "🖫–" or "–✇" next to a parameter value indicates that its current value is very close to its lower or upper boundary. Since the search cannot extend beyond the boundary, tuning again with different boundary values may well find better parameters.

When the tuning search is complete, the subwindow will be cleared of the messages relating to the clusters, the error values and the parameters, and will only have the "Memory Left:" message. You must still use the menu command **Process/Flow / Process tuning / Quit tuning process model** to terminate all tuning activity. When you do this, the last tuned parameters are displayed along with the final error value. This can be useful when you let a tuning search run overnight.


Adaptive Circuit Modeling System
File Edit Process/Flow Controller Simulate Help
North
South
Posted Data

Memory left: 1048576
GA Name: Tuner GA for cluster 1 of 1
Current Mean Squared Error: 0.62712
Current generation: 8
Current string: 3
Current cycle: 0
Apatite rate constant: 0.0708002
Gangue rate constant: 0.00771351
Feed weight fraction solids: 0.422088

Figure 3-12. Posted information shows progress as Pure Empirical model is tuned.

To evaluate the effectiveness of a tuned Pure Empirical model, you need the same kind of data you used for tuning. The Modeling System needs to do the same sort of analysis for the test data as it did for tuning data. For each test observation, it must first decide which submodel should be used; then it must adjust the process's input flows to match the observation, cycle the process (using the appropriate submodel's parameters) till it reaches equilibrium, and compare the process's output flows to the observation values. You ask the Modeling System to do this with either of the menu commands **Simulate / Begin simulation** (these commands are equivalent, and do not behave at all as they do for the first-principles model). The Modeling System will read a standard-format file of testing data, compute model outcomes, then write a comparison-data file with the target and actual values side-by-side and in a format suitable for reading into a spreadsheet. The spreadsheet can then be used to compute statistics and draw graphs. When you ask to **Simulate** a Pure Empirical model, the Modeling System will ask first for the name of the test data file to *OPEN*, and next for the name of a comparison file to *SAVE*. Then all observations are processed, without pausing to let you schedule other work at the same time (as you can do while tuning a model) and without posting progress information. The format of a comparison file is illustrated in Section 9.5. The first useful test data file for a model is usually the standard-format file on which the clustered tuning data file was based.

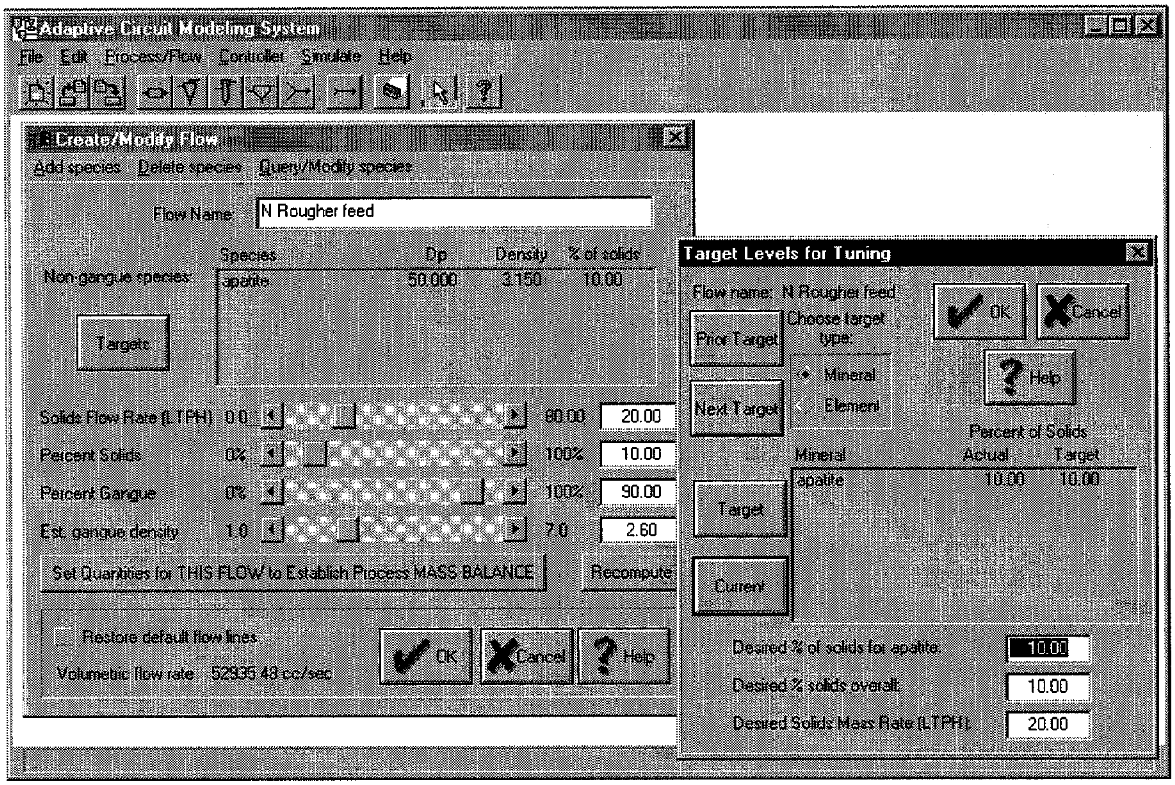
3.4 Establish Remaining Models

There is still no model for the mixer output flow, or for the output flows of the Amine process. A mixer model has no parameters; create it now by double-clicking on the process icon. You will be asked to supply a name for the mixer's output flow. Notice that the model label on the mixer icon is just a sequence number. Then create another first-principles (Jordan-Spears) model for the Amine process. You can see the resulting graphical representations in the desktop pictured in Figure 3-16.

3.5 Tune a First-Principles Flotation-Cell Model

You can ask the Modeling System to tune other models as well as the Pure Empirical model. But this tuning will be based on a single desired process feed state and corresponding desired output flow states. These desired flow states are called *flow targets,* and are described in Section 6.3. They are not stored as observations in a data file but are entered directly from the Modeling System interface, using the *Targets* button on the *Create/modify Flow* screen.

You can ask, for example, how much larger the values for the gangue and water rate constants in the North Rougher model must be to produce a more reasonable distribution of water and sand between the concentrate and tails flows than is shown in Figure 3-7. One way to answer this question is to try some values to see what happens. Another way is to let the tuning process look for better values. Double-click on the N Rougher feed flow line to see its *Create/modify Flow* screen, then click the *Targets* button. In the *Target Levels for Tuning* screen, click the *Current* button to create a target which matches the current flow state. The result should look like Figure 3-13. Do the same for the concentrate flow, but then change the desired % of solids for apatite and desired solids mass rate as shown in Figure 3-14; finally click the *Target* button to make the flow state match this modified target. Now you can make a target for the tails flow. First use the button *Set Quantities for THIS FLOW to Establish Process MASS BALANCE* to compute the appropriate flow state, which is shown in Figure 3-15. Then make a target from this flow state. These targets are in mass balance.


Adaptive Circuit Modeling System
File Edit Process/Flow Controller Simulate Help
Create/Modify Flow
Add species Delete species Query/Modify species
Targets
Set Quantities for THIS FLOW to Establish Process MASS BALANCE
Recompute
OK
Cancel
Help
Target Levels for Tuning
Prior Target
Next Target
Target
Current
OK
Cancel
Help

Flow Name: N Rougher feed

	Species	Dp	Density	% of solids
Non-gangue species:	apatite	50.000	3.150	10.00

Solids Flow Rate (LTPH)	0.0	80.00	20.00
Percent Solids	0%	100%	10.00
Percent Gangue	0%	100%	90.00
Est. gangue density	1.0	7.0	2.60

□ Restore default flow lines

Volumetric flow rate: 52935.48 cc/sec

Flow name: N Rougher feed

Choose target type:

● Mineral

○ Element

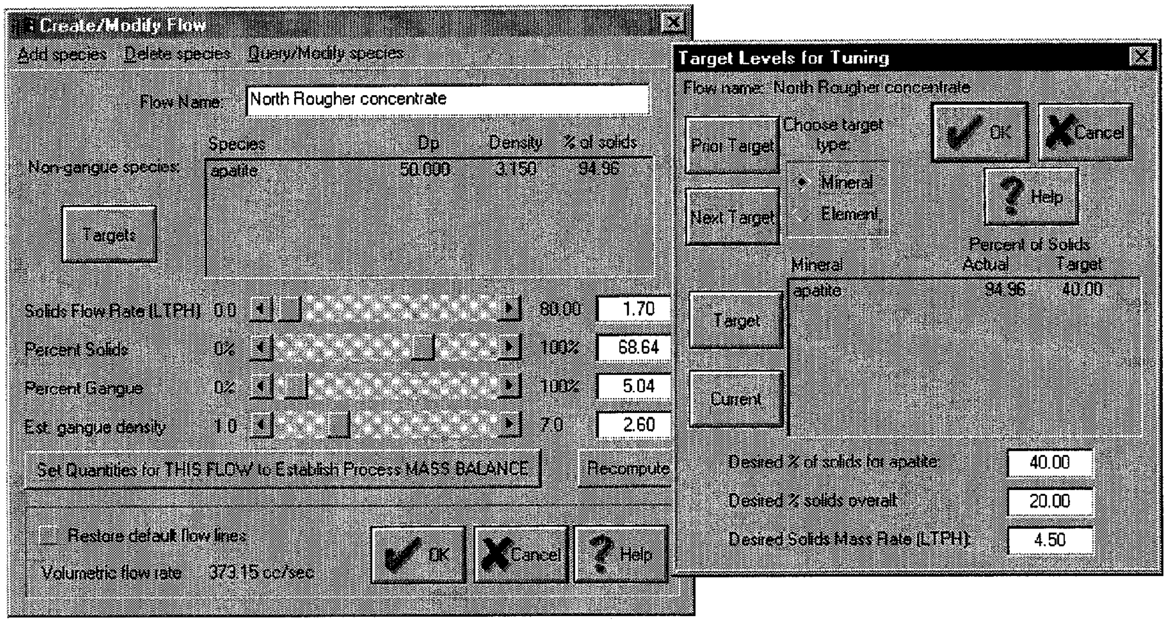
	Percent of Solids	
Mineral	Actual	Target
apatite	10.00	10.00

Desired % of solids for apatite: 10.00

Desired % solids overall: 10.00

Desired Solids Mass Rate (LTPH): 20.00

Figure 3-13. Flow Target for N Rougher Feed Flow for tuning first-principles model.



Flow Name: North Rougher concentrate

Non-gangue species:	Species	Dp	Density	% of solids
	apatite	50.000	3.150	94.96

	Min		Max	Value
Solids Flow Rate (LTPH)	0.0		80.00	1.70
Percent Solids	0%		100%	68.64
Percent Gangue	0%		100%	5.04
Est. gangue density	1.0		7.0	2.60

□ Restore default flow lines

Volumetric flow rate 373.15 cc/sec

Flow name: North Rougher concentrate

Choose target type:
● Mineral
○ Element

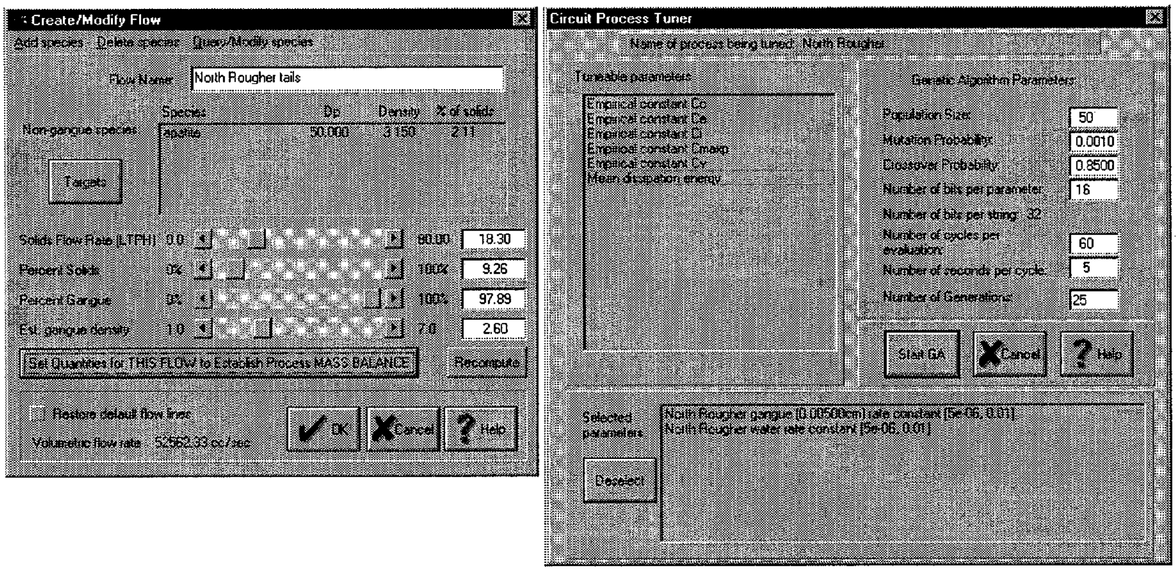
Mineral	Percent of Solids Actual	Percent of Solids Target
apatite	94.96	40.00

Desired % of solids for apatite: 40.00

Desired % solids overall: 20.00

Desired Solids Mass Rate (LTPH): 4.50

Figure 3-14. Flow Target for North Rougher concentrate flow for tuning first-principles model.



Flow Name: Noith Rougher tails

Non-gangue species:	Species	Dp	Density	% of solids
	apatite	50.000	3.150	2.11

	Min		Max	Value
Solids Flow Rate (LTPH)	0.0		80.00	18.30
Percent Solids	0%		100%	9.26
Percent Gangue	0%		100%	97.89
Est. gangue density	1.0		7.0	2.60

□ Restore default flow lines

Volumetric flow rate 52562.33 cc/sec

Name of process being tuned: North Rougher

Tuneable parameters:

Empirical constant Co
Empirical constant Ce
Empirical constant Ci
Empirical constant Cmaxp
Empirical constant Cy
Mean dissipation energy

Genetic Algorithm Parameters:

Population Size:	50
Mutation Probability:	0.0010
Crossover Probability:	0.8500
Number of bits per parameter:	16
Number of bits per string:	32
Number of cycles per evaluation:	60
Number of seconds per cycle:	5
Number of Generations:	25

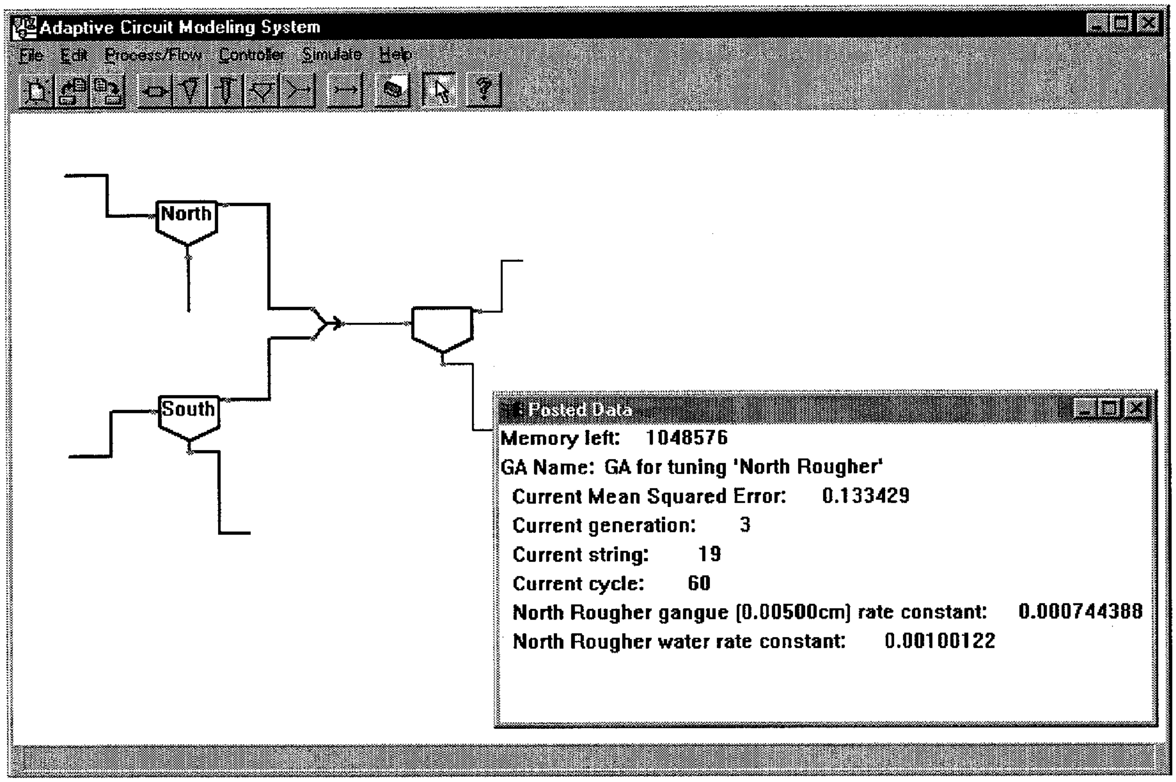
Selected parameters

North Rougher gangue [0.00500cm] rate constant [5e-06, 0.01]
North Rougher water rate constant [5e-06, 0.01]

Figure 3-15. Flow Target for North Rougher tails flow. Tune four empirical constants from flotation cell model..

The targets are now complete, so you can use menu choice **Process/Flow | Process tuning | Tune process model** to start the tuning process. You will first see a parameter-selection screen closely related to Figure 3-11; make it look like Figure 3-15, then press the Start GA button to begin tuning. Figure 3-16 shows posted progress information.

Double-click on the process and flow icons to examine the models to see the effect of the tuning process.


Adaptive Circuit Modeling System
File Edit Process/Flow Controller Simulate Help
North
South
Posted Data

```
Memory left:   1048576
GA Name:  GA for tuning 'North Rougher'
  Current Mean Squared Error:      0.133429
  Current generation:      3
  Current string:        19
  Current cycle:       60
  North Rougher gangue [0.00500cm] rate constant:    0.000744388
  North Rougher water rate constant:      0.00100122
```

Figure 3-16. Posted Data window reports progress as first-principles model is tuned.

4. MENU AND TOOLBAR OPERATIONS REFERENCE

4.1 File MENU OPERATIONS

The Modeling System provides a desktop where you can draw a plant representation, attach models to the flows and processes of that representation, and edit the internal state of those models. You can save your current desktop, or restore a prior desktop, at any time. At present, the Modeling System gives you only one desktop at a time. In standard Windows style, these operations are collected under the File heading on the main menu; and standard file-choosing dialogs are used.

File | New circuit

Erase all current contents of the desktop. Same as button on the toolbar.

File | Load circuit

Replace all current contents of the desktop with saved desktop contents written earlier to a disk file. The Modeling System expects the saved file to have extension *.cir*. Same as button on the toolbar.

File | Save circuit

Write all current contents of the desktop to a disk file; the desktop contents are not modified. Same as button on the toolbar.

File | Exit

Exit the Modeling System. Same as closing the window from the button on the caption bar.

4.2 *Edit MENU OPERATIONS*

The Modeling System does not support Cut and Paste or Undo editing operations. Selection status does not currently affect any operations, but this capability is expected to be used for defining subcircuits and partial desktop save/restore operations.

Edit / Select All

Mark all icons "Selected"; they will be drawn in red. Click on any icon to deselect it. Using the mouse to enclose an icon in a selection box also marks the icon "Selected".

Edit / Deselect All

Mark all icons "Unselected"; they will be drawn in black.

4.3 Process/Flow MENU OPERATIONS

All menu choices relating to the visual representation of a plant are found under this heading; all such menu items have corresponding toolbar buttons. In addition, menu choices for viewing and changing process and flow models are also grouped here. The task "Create a process or flow model" is performed by double-clicking the mouse on an icon that has no model: it is not found in the menu system. You can tell whether or not a process icon or flow line has an associated model. Flow icons which have no models are drawn using thin lines, and icons with models are drawn with heavy lines. Process icons are always drawn with heavy lines, but a process icon without a label is not decorated and one that has a model always also has a short label. The label is taken (usually) from the first characters of the process name; if no better label can be found, "*" is used.

Process/Flow / Place new icon on the desktop SUBMENU

The entries under this submenu correspond exactly to buttons on the toolbar. For a new process icon, choose a menu item or click a tool button, move the cursor on the desktop to the place you want the icon, then click the left mouse button: your selected icon will appear under the cursor. After the click, the mouse reverts to its default move-and-select (arrow) mode: you can immediately grab the new icon and adjust its position as needed. For a new flow line, choose menu item "Place a new Flow icon" or click the Flow tool button; move the cursor on the desktop to the location for one end of the flow, depress the left mouse button, move the cursor to the location for the other end of the flow, and release the mouse button. See Section 5, MOUSE OPERATIONS REFERENCE, for information about interpreting and adjusting the appearance of a plant's graphical representation.

Button	Menu Choice
↣	Process/Flow | Place new icon | Place a new Flow icon
⊟	Process/Flow | Place new icon | Place a new Ball Mill icon
∇	Process/Flow | Place new icon | Place a new Hydrocyclone icon
⊤	Process/Flow | Place new icon | Place a new Flotation Column icon
▽	Process/Flow | Place new icon | Place a new Conventional Flotation Cell icon
≻	Process/Flow | Place new icon | Place a new Mixer icon

Process/Flow / Create a Subcircuit

The subcircuit will consist of all process models that can be reached from a starting process or flow model, plus the corresponding input and output flows. This menu command was designed to create a circuit for fuzzy control. The only operation that can be performed on such a subcircuit is simulation. If you make a subcircuit whose only process is a conventional flotation cell with a Pure Empirical model, you can use it to do the same kinds of simulation operations as with the first-principles model ("simulation" means "model validation" with a Pure Empirical model).

Process/Flow / Query/modify a Process Model

The Modeling System will present the names of all process models; choose one from this list. This operation is the same as double-clicking on the corresponding desktop icon. The Modeling System presents the same screen for creating, editing, or inspecting each kind of process model. These screens are shown and described in Section 7.

Process/Flow / Query/modify a Flow Model

Select a flow model from the list presented by the Modeling System. This operation is the same as double-clicking on the corresponding flow line on the desktop. will list the names of all flow models. The Modeling System presents the same screen sequence for creating, editing, or inspecting a flow; it is shown and described in Section 6.

Process/Flow / Delete a Process Model

Select a model from the list presented by the Modeling System. The process model will be deleted, but the corresponding icon will not be erased: so this operation is not the same as the toolbar Eraser button.

Process/Flow / Delete a Flow Model

Select a model from the list presented by the Modeling System. A flow model cannot be deleted if it is connected to a process model; the process model must be deleted first. When the flow model is deleted, the corresponding flow line is not erased: so this operation is not the same as the toolbar Eraser button.

Process/Flow / Process tuning SUBMENU

By "tuning" the Modeling System means using a genetic algorithm to select process model parameters that do the best job of matching model simulation outcomes to some desired target. For flotation column and Jordan-Spears models, you must establish a single set of targets on the process's flows, using menu item **Process/Flow / Query/modify a Flow Model** (if you do not explicitly set targets, the current flow states will be used). If you do not know that these targets are consistent with each other (i.e. the same amount of material flows into and out of the process), use the **Balance Targets** command in this submenu. Before you use the current flow states for a process as the basis for flow targets, you can force them to be in mass balance by computing one flow as the resultant of all the others: this is the operation performed by the button "Set Quantities for THIS FLOW..." in the Create/Modify Flow screen. For the Pure Empirical model, the Modeling System needs many flow targets; it assumes these have been edited into a standard data file and preprocessed by the Cluster Analyzer (see Section 8.2), so you will be asked to identify a cluster file (expected extension *.clu)* for data input.

Process/Flow / Process tuning / Tune process model

The Modeling System will ask you to select a model from a list; identify the parameters you want to tune and set ranges of possible values; and set control values for the genetic algorithm that performs the tuning. Then the Modeling System launches the genetic algorithm. You can watch its progress in the Posted Data window. You can work in other parts of the desktop while tuning is in progress, but both tuning and your other work will be slowed.

Process/Flow / Process tuning / Quit tuning process model

Whenever you ask the Modeling System to begin tuning a model, you will also have to ask it to stop tuning. This is true even after the tuner genetic algorithm has completed its search. Until tuning and simulation operations have been explicitly stopped, the Modeling System will not permit you to update flow models. When you stop a tuning search, the Modeling System displays the best parameters it found and the corresponding error value. A genetic-algorithm search process can require a lot of time; you can use this command to abandon the tuning process before it finishes.

Process/Flow / Process tuning / Balance targets

The Modeling System will ask you to select a process model from a list. An explicit flow target must be present for each of the process's input and output flows; unlike the tuning operation, this operation will not build default targets. After a brief genetic-algorithm search, the Modeling System will replace these targets with ones that are in mass balance and that are as close as possible (in a least-squares sense) to the original ones.

4.4 Controller MENU OPERATIONS

Control based on fuzzy logic was one of the first capabilities built into the Modeling System. This implementation was designed to be used with a first-principles model (for effective training of the controller) and a randomly-varying feed flow (to exercise the controller; code is present in the Modeling System but not integrated with the user interface, which came later). The controller is based on a selection of *sensor readings* and *control adjustments* made available by the process model to be controlled. Only the flotation column model (which does not currently work) is fully designed to work with a controller; the Jordan-Spears first-principles flotation-cell model has the necessary superficial structure but defines no readings or adjustments.

Controller | Create controller

Controller | Delete controller

Controller | Activate controller

Controller | Deactivate controller

Controller | Train controller

Controller | Query/modify controller

Controller | Cease training

4.5 Simulate MENU OPERATIONS

Like process tuning, simulation is performed differently for different models. For the Pure Empirical model, the Modeling System performs model validation rather than a simple simulation: it expects to read a file of test data in exactly the same format as the file presented to the Cluster Analyzer, it computes the equilibrium output flows corresponding to each input flow state in the file, and it writes a comparison file (formatted for input to a spreadsheet such as Excel) containing headers and columns for test data and for the corresponding model outcomes. It is straightforward to generate statistics and graphs from this comparison file. For all other models, the simulation operation assumes an unchanging feed flow (equal to the flow model's current state, not to its current target); the Modeling System continues to compute new process outputs until you tell it to stop. You can see the results of the simulation in the Posted Data window and by inspecting the process and its output flows You can force the Modeling System to simulate a Pure Empirical model for a single feed flow by defining a subcircuit containing just the Pure Empirical process. You must edit the process yourself to make sure it contains the right set of model parameters for the given feed; then simulate the subcircuit process rather than the Pure Empirical process. When you stop simulating, the simulated state remains in the Pure Empirical process and in the flows.

Simulate / Begin simulation, Instantaneous cycling

Choose from a list of process models, then run simulation or validation. Computations will be done as fast as your computer can perform them.

Simulate / Begin simulation, Real-time cycling

Choose from a list of process models, then run simulation or validation. If the process model is not Pure Empirical, match simulated time as closely as possible to real time (wait as needed for the real clock to catch up to the simulated one).

Simulate / End simulation

Simulation computations continue until you terminate them with this command. Model validation operations are not at present designed to be interrupted.

4.6 Help MENU OPERATIONS

Help text has been adapted from this User Guide. The Modeling System uses the standard Windows Help facility.

HELP / Help on use Of Adaptive Circuit Modeling System

This Help menu command, function key F1, and the question-mark toolbar button all perform the same action: they display the online Help Table of Contents screen. This screen has links to Help for all Modeling-System operations, including related stand-alone programs. Start here, and follow the links from screen to screen as needed, by clicking on words and phrases displayed in green. Press the Index button in the Help-screen toolbar to see an alphabetical list of all the topics for which Help has been constructed. You can jump directly from this list to whatever Help screen you select, then follow links from that screen. You can also use the Browse toolbar buttons (labelled >> and <<) to step through all Help topics. The topics will be listed in an order which corresponds roughly to their sequence in the User Guide.

HELP / Help on use Of the Help System

The Using Help screen has links to online information about the standard Windows Help facility.

HELP / About

The About screen tells what version of the Modeling System you are running.

4.7 TOOLBAR BUTTONS

FILE-OPERATION TOOLBAR BUTTONS

See **File** menu operations **New Circuit, Load Circuit, Save Circuit.**

ICON-SELECTION TOOLBAR BUTTONS

See **Process/Flow / Place New Icon on Desktop** submenu: icons represent ball mill, hydrocyclone, flotation column, conventional flotation cell, mixer, and flow.

ERASER TOOLBAR BUTTON

Click this button, then click an icon on the desktop: the icon will disappear. If a model is associated with the icon, the model disappears too. After the click the mouse reverts to its default move-and-select (arrow) mode, just as it does after the icon buttons.

ARROW TOOLBAR BUTTON

Click this button to return the mouse to its default mode. This button is useful if you press an icon button or the eraser button and then change your mind.

HELP TOOLBAR BUTTON

This button corresponds exactly to menu choice **Help / Help on use of the Adaptive Circuit Modeling System** and to pressing the Fl function key: all three actions display the initial Table of Contents Help screen, and you can follow links to all Help screen from there.

5. MOUSE OPERATIONS REFERENCE

Only one function is assigned to the right mouse button in the Modeling System. Whenever this Guide says “click” it means “click using the left mouse button.” Clicking the right mouse button makes the Modeling System show activity on the Blackboard (the structure that coordinates all Modeling System actions). This display is the only expression of the Blackboard architecture that can be seen directly from the user interface, and its purpose is diagnostic only. Clicking the right mouse button again hides the display. If you leave the display active, updating it will consume most of the Modeling System’s time.

After you press one of the process icon buttons on the toolbar, the next mouse click anywhere on the desktop will place a new process icon on the desktop under the cursor. The Modeling System does not check to make sure process icons do not overlap, since you can move or erase icons as needed. After you press the flow icon button, you must identify both source and destination points for the new flow: move the cursor to one end of the flow, press the left mouse button, move the cursor to the other end of the flow, and then release the mouse button. If the flow is (or becomes) attached to a process, the attachment determines the direction of the flow; until then, the mouse-up end is the destination. For all process icons, attachments can be made only at standard points; the feed attachment is always on the left. If the modeling system recognizes that a flow and process are attached, a small green square will appear at the connection point.

After you press the eraser button on the toolbar, the next mouse click will delete the closest icon *if* one is close enough: you must click pretty much on top of an icon to delete it.

The rest of the time, the mouse is in its default mode. In this mode, clicking the left mouse button once on an icon (either process or flow) deselects it. Clicking once on an empty area of the desktop asks to redraw the desktop. Double-clicking on an icon asks to examine (and perhaps edit) the associated model if there is one, or else to create a new model. You can move a process icon by grabbing it (move the cursor over it and press the left mouse button), dragging it where you want it (move the mouse, keeping the button depressed), then dropping it (release the button). The same action can be performed on a flow line segment or an unattached end of a flow line. If the same action is applied at a point which does not belong to any icon, the Modeling System instead draws a selection rectangle: icons inside the rectangle when the button is released are marked “selected” (drawn in red). A tiny rectangle of this sort drawn to contain a process connection point and the nearby end of a flow asks to make or unmake the attachment between the process and the flow (indicated by a green dot). See also Section 2, HANDS-ON I: EXPLORE THE GRAPHICAL INTERFACE, for a more extended description of typical mouse actions.

6. DIALOG SCREEN REFERENCE: SCREENS FOR FLOWS

6.1 CREATE/MODIFY FLOW Screen

Reach this screen from menu entry **Process/Flow / Query/modify Flow** or by double-clicking on a flow icon on the desktop.

What is a Flow?

A flow is a model of a multi-species slurry which feeds a process, connects one process with another, or constitutes an output slurry of a process or circuit. A Modeling System Flow object characterizes this slurry using volumetric flow rates for a list of constituents: water, gangue, and ore minerals. A mineral constituent is characterized by particle size, density, and induction time as well as mineral composition; the same mineral may appear as more than one species in a flow. The user view of the flow has a very different vocabulary of measurements, as this screen shows.

For process tuning, it is necessary to be able to express *desired* as well as *actual* states for a flow. Such a desired state is called a Flow Target, and it is stored with the flow it describes. A user specifies a Target using the same kind of measurements as in the user view of a flow. Because a target may be incomplete (some assays or flow percent solids may be missing), the Modeling System stores a Target just as it is entered. When a Flow is changed to conform to an incomplete Target, the result sometimes reflects the difference in these measurement types.

Flow Name

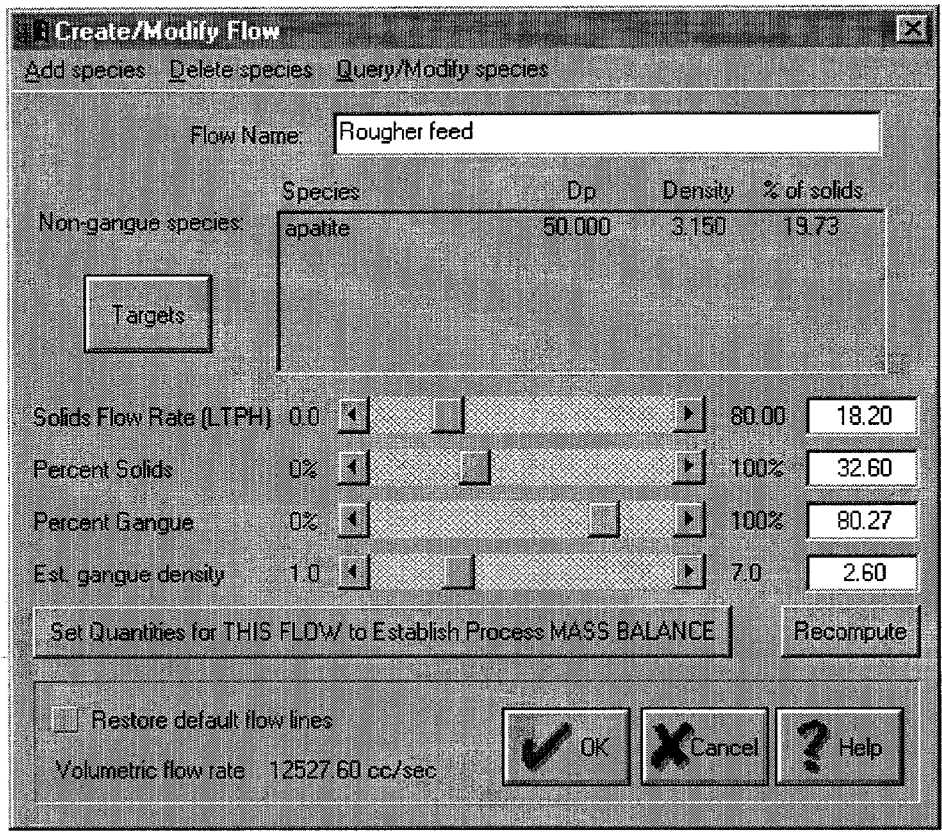
Use the tab key or mouse to place cursor in this window, then enter any desired text. Names must be unique; menu operations Delete Flow and Query/modify Flow identify the targeted flow by name.

Nongangue mineral species

Every flow has by default the components gangue and water. This list shows additional flow components. Use the entries in this screen's menu bar to add or delete minerals from the list. To view a species entry, you can double-click on a list item, or you can click on it once to highlight it and then select menu entry Query/Modify Species. The same screen, titled Add/Edit Species, is used for all species operations. When editing, you can not only change descriptive parameters such as particle size, you can in fact replace one mineral species by another. The value for % of Solids will be copied to this list from the Add/Edit Species screen, and Percent Gangue will be adjusted so that all percents add to exactly 100. If this would require a negative Percent Gangue, the gangue value is set to zero and the other percents are rescaled. If the constituent minerals in a flow change, existing flow targets no longer make sense and so are deleted.

Targets button

Click the Targets button to bring up the screen Target Levels for Tuning. This screen lets you examine any target attached to this flow; you can also modify the current target (if there is one) or create a new target (if none exists). You can also use the Target Levels screen to force the flow to match a target: this is an easy way to set % of Solids for all species in a mixed-ores flow.



Flow Name: Rougher feed

Non-gangue species:

Species	Dp	Density	% of solids
apatite	50.000	3.150	19.73

Solids Flow Rate (LTPH)	0.0	80.00	18.20
Percent Solids	0%	100%	32.60
Percent Gangue	0%	100%	80.27
Est. gangue density	1.0	7.0	2.60

□ Restore default flow lines

Volumetric flow rate 12527.60 cc/sec

Figure 6-1. Create/Modify Flow Screen

Solids Flow Rate (in long tons per hour, for all ore minerals plus gangue)
Percent Solids (ratio of solids to total flow, by weight not volume, as a percent: remainder of flow is water)
Percent Gangue (percent of total solids, by weight, made up of non-ore species)
Estimated Gangue Density (relative to the density of water, which is 1.0)

Each of these four fields may be set in two ways. You can use its slider (grab the thumb button in the middle and drag it to the left or right; or click near the end arrows for a large step; or click on the arrows for a small step). You can also enter a number directly in the corresponding edit box.. The minimum and maximum values are shown at the left and the right of each slider. It is possible to enter a Solids Flow Rate value larger than the maximum, but not to transfer such a value to a Target; if you construct a target from such a flow, then make the flow conform to the target, you do not recover the same flow. Percent Gangue is treated as a residual value when non-gangue species % of Solids values change: its value is increased or decreased as needed to keep the total exactly 100. When Percent Gangue is changed, the other percents are adjusted proportionally.

Set Quantities for THIS FLOW to Establish Process MASS BALANCE

The Modeling System will examine all the other flows attached to the same process, and (when possible) set values for this flow so that the process input and output flows are exactly balanced. If the flow is attached to two processes, the Modeling System will ask which one you want to balance.

Recompute

Press this button to force an update of the flow display after you have changed any of the values controlled by sliders. Volumetric flow rate and percent solids for non-gangue species may change.

Restore Default Flow Lines

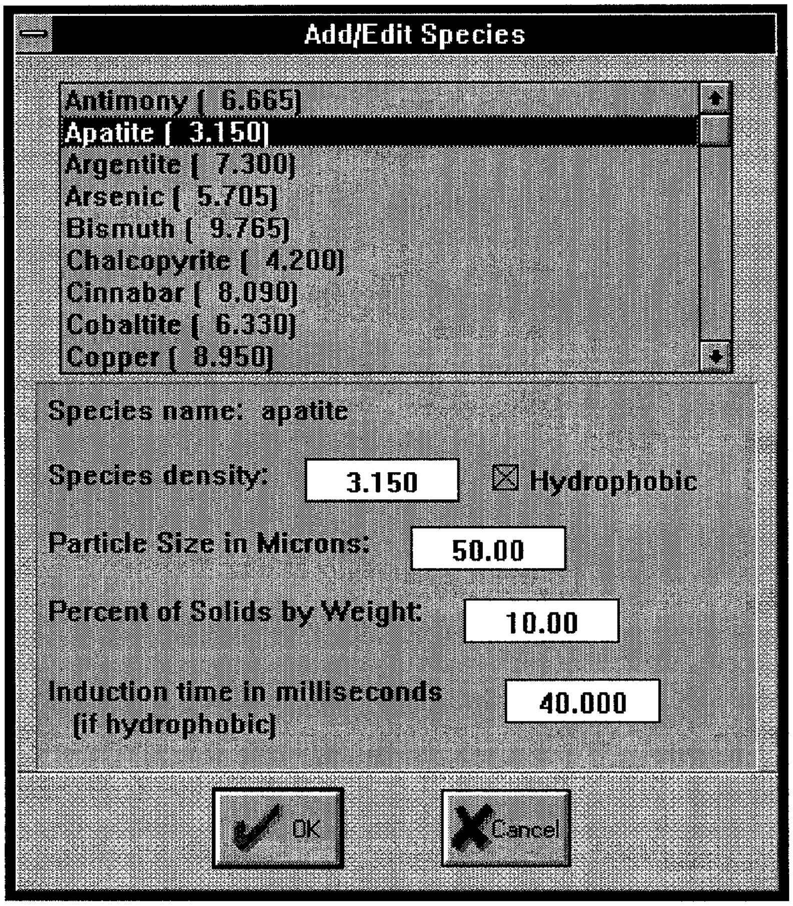
If you are unhappy with the layout of this flow's stairstep line on the desktop, you can recover its default appearance (usually three line segments long) by exiting this screen with the Restore Default box checked. Clicking the box toggles its setting. This request will be honored even if you exit using the CANCEL button.

OK and CANCEL buttons

Until you click one of these buttons, all changes (including any possible Flow Target updates) have been made to a **copy** of the flow you are editing. When you click OK the changes are applied to your actual flow; if you click CANCEL the changes are discarded. If the flow is being used in a tuning or simulation operation which is still in progress, the OK button is disabled so you cannot modify the in-use flow, but you can freely update the displayed flow copy.

6.2 ADD/EDIT SPECIES Screen

Reach this screen from the Create/Modify Flow screen, by selecting menu item Add Species or Query/Modify Species or by double-clicking an item in the Non-gangue Species List.



Add/Edit Species

Antimony (6.665)
Apatite (3.150)
Argentite (7.300)
Arsenic (5.705)
Bismuth (9.765)
Chalcopyrite (4.200)
Cinnabar (8.090)
Cobaltite (6.330)
Copper (8.950)

Species name: apatite

Species density: 3.150 ☒ Hydrophobic

Particle Size in Microns: 50.00

Percent of Solids by Weight: 10.00

Induction time in milliseconds (if hydrophobic) 40.000

Figure 6-2. Add/Edit Species Screen

The Modeling System has a library of ores and mineral species for which it can supply default values of density, induction time, and particle size. The Modeling System also knows (but does not display) the chemical composition of these minerals; this information is used to translate between by-element assays and by-mineral assays in flow targets (see the description of the screen Target Levels for Tuning). For this reason you cannot define a new mineral here, but must select from the existing list. (At present, changing the library contents requires recompiling the Modeling System rather than just updating an initialization file .)

Once you select a mineral, you can change any of its displayed values:

Species density (relative to the density of water, which is 1.0)
Particle size (diameter Dp, in microns)
Percent of Solids by Weight (see description below)
Hydrophobic flag (checked means yes)
Induction time (in milliseconds)

You can set induction time for a non-hydrophobic species, even though the value is not meaningful. This value is not used by either conventional flotation cell model. New values entered here will not change any default values but will apply only for this species in the current flow model (and to any flow model copied from it, as will happen if this flow feeds a process and you let the Modeling System create the models for the process's output flows: see the beginning of Section 3.2 in the hands-on example). If you make complex changes to the definition of a flow, it is a good idea to use the Modeling System to copy them, rather than trying to repeat the same changes for each flow model: process models require that their input and output flows have the same structure.

A value in the field Percent of Solids by Weight will be copied directly to the Non-gangue Species list in the prior Create/Modify Flow screen, unless the sum of percents in that list would exceed 100; in that case, Percent Gangue is set to 0.0 and the other percents are scaled to add to 100.

OK and CANCEL buttons

As is true for all the process and flow dialog screens described in this section, what is displayed is a copy of a model associated with a desktop icon: no changes are committed to the actual flow model until you choose OK for both this screen and for the Create/Modify Flow screen from which this screen was called.

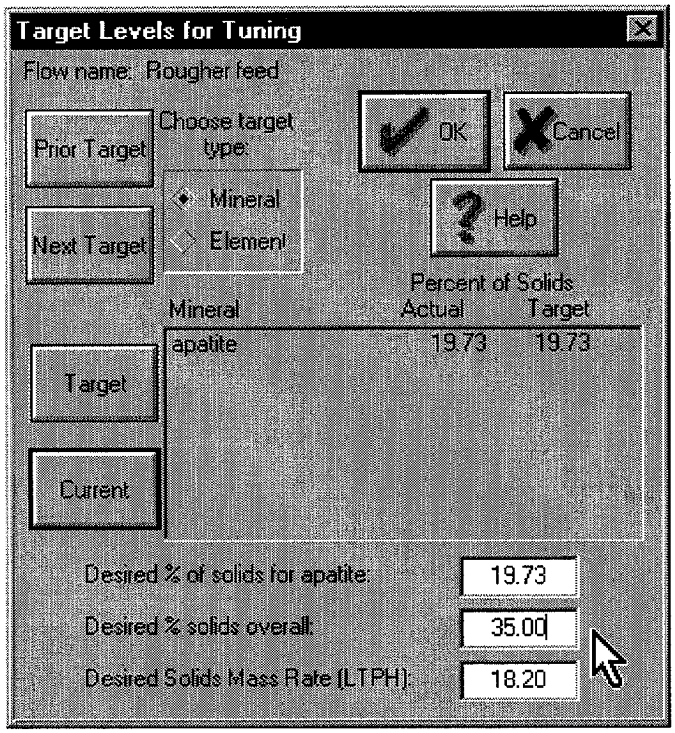
6.3 TARGET LEVELS FOR TUNING Screen

Reach this screen by clicking the Targets button on a Create/Modify Flow screen.

What is a Flow Target?

See What is a Flow? in the description of screen Create/Modify Flow. A target flow state is expected normally to be based on actual plant measurements, and so is expressed in terms of flow assays and a measure of overall flow size. The Modeling System accommodates either element or mineral assays (though not a mixture within the same flow target), and can translate from one to the other. A complete flow description requires assays, total solids mass rate, and the flow's weight fraction solids; any of these possible target values may instead be marked “Unspecified.”

Flow targets are used for all process models during model tuning, and for the Pure Empirical model during validation (misnamed “simulation”). The Pure Empirical model reads a file of tuning data and distributes many targets to each flow; if you examine its flows while tuning is in progress, you can view (but not modify) the list of targets. These targets are all deleted when tuning stops. Other process models are tuned using a single target on each flow. These targets must be entered using this screen; and are not deleted after tuning.



Flow name: Rougher feed

Choose target type:
◆ Mineral
◇ Element

	Percent of Solids	
Mineral	Actual	Target
apatite	19.73	19.73

Desired % of solids for apatite: 19.73

Desired % solids overall: 35.00

Desired Solids Mass Rate (LTPH): 18.20

Figure 6-3. Target Levels for Tuning Screen

Prior Target and Next Target Buttons

When a flow has more than one target, one of them is always the current one; this is the target that is initially displayed on this screen. But you can move forward and backward through the entire chain of targets by clicking on these buttons. At present, multiple targets are present only while a Pure Empirical process model is being tuned.

Choose target type: Mineral or Element

You can express a flow target either in terms of mineral assays or in terms of element assays, and the Modeling System can (except in pathological cases) translate from one form to the other.

Target Button

Click to make the flow conform to the current target description. The Modeling System uses a brief genetic algorithm search here, as elsewhere, to solve a problem that may not be entirely well defined. If the target percents sum to more than 100, they will be scaled before the search begins. If the target values are unreachable, the flow is set to values which come as close as possible, in a least-squares sense, to the specified targets.

Current Button

Click to make the target match the current state of the flow.

Mineral List Box (for mineral assay targets), or Element List Box (for element assay targets)

If there is one entry in the list box, its Target Percent of Solids is always available for editing in the first line below the list box. If the list box has more than one entry, click on any line you want to change; that entry will become available for editing in the same line just below the list box.

Desired % of solids for xxxxx
Desired % of solids overall
Desired Solids Mass Rate (LTPH)

If the target is expressed in terms of mineral assays, all these correspond exactly to fields on the prior Create/Modify Flow screen: the first corresponds to values in that screen's Species list box and the others to the first two values controlled by sliders.

OK and CANCEL buttons

Until you click one of these buttons, all target changes have been made to a copy of the flow you are editing in the prior Create/Modify Flow screen (and that flow is itself a copy of the flow model which is attached to a desktop icon, unless you are creating a brand-new model). If you click CANCEL all target and flow changes are discarded; when you click OK they are applied to the flow in the Create/Modify Flow screen.

7. DIALOG SCREEN REFERENCE: SCREENS FOR PROCESSES

7.1 EDIT CONVENTIONAL FLOTATION CELL Screen

Reach this screen from menu choice **Process/Flow / Query/Modify a Process Model** for an existing model, or by double-clicking its flotation cell icon; or double-click an unmodeled flotation cell icon and choose Jordan-Spears Model.

What is the Jordan-Spears First Principles flotation model?

See Reference 3. The model is based on two first-order differential equations, one for the rate of change of unattached particles in the flotation cell and one for the rate of change of attached particles in the system; at equilibrium, both rates are zero. Computations for this model are described in section 3.1 of the Final Report for this project. The diameter of the largest particle that can be floated, d_{maxp}, is not a constant in this implementation of the model, but rather depends on the fluid dissipation energy for each unit volume, which in turn depends on the distribution of fluid velocities.

The discrete time varying rate constant model is given by the following equations:

$$N_{up}(t) = (1 - P_f)\, N_{up}(t - 1)$$

$$N_{ap}(t) = (1 - \frac{Q_a}{V f_a})\, N_{ap}(t-1) + P_f N_{up}(t-1)$$

$$N_{apc}(t) = \frac{Q_a}{V f_a}\, N_{ap}(t)$$

$$P_f = \sum_{db} \sum_{Uf} P_c\, P_a\, P_s\, P_b\, P_u$$

$$P_s = (1 - \frac{d_p}{d_{max_p}})^{\frac{3}{2}}$$

$$P_a = \sec h^2\, (\frac{3\, U_{rpb}\, t_i}{2\, d_b})$$

$N_{ap}(t)$ is the number of attached particles
$N_{up}(t)$ is the number of unattached particles
$N_{apc}(t)$ is the number of attached particles in the concentrate
V is the volume of the cell
Q_a is the volumetric flow rate of air into the cell
f_a is the equilibrium volume fraction of air in the cell
P_f is the flotation probability for a particular size particle

P_c is the probability of collision
P_a is the probability of attachment
P_s is the probability of stable attachment
P_b is the frequency probability of bubble diameter *db*
P_u is the frequency probability of fluid velocity *u*
d_p is the particle diameter
d_{maxp} is the maximum particle size that can be floated
d_b is the bubble diameter
h is the height of the cell
U_{rpb} is the velocity of the average bubble-particle group

$$P_c = C_c\,(\frac{d_p}{d_b})^{C_e}$$

$$d_{max_p} = C_{maxp}\left(\frac{E\,U_f}{\sum(P_{U_f}\,U_f^3)}\right)^{-\frac{1}{3}}$$

$$U_i = \frac{N_b}{V}\frac{N_p}{V}\,[P_c\;5\,(r_b + r_p)^2\;U_{rpb}\frac{1.0}{U_f}]\frac{U_f}{1.0} * \frac{V}{N_p}$$

$$t_i = C_i\,d_p^{\frac{3}{2}}$$

$$U_{rpb} = \frac{0.4\,\varepsilon^{\frac{4}{9}}}{\upsilon^{\frac{1}{3}}}\,[(\frac{\rho_f - \rho_b}{\rho_f})^{\frac{4}{3}}\,(d_b)^{\frac{14}{9}} + (\frac{\rho_p - \rho_f}{\rho_f})^{\frac{4}{3}}\,d_p^{\frac{4}{3}}\,]^{0.5}$$

$$\upsilon = C_v\,\frac{\kappa^2}{\varepsilon}$$

$$\varepsilon = \frac{E}{\sum(P_{U_f}\;U_f^3)}\;U_f^3$$

t_i is induction time
C_c is a collision efficiency constant
C_e is a collision efficiency constant
C_{maxp} is the maximum floatable particle size constant
E is the mean dissipation energy
U_f is the fluid velocity
P_{Uf} is the probability of U_f for a given particle size
N_b/V is the number of bubbles per unit volume of the cell
N_p/V is the number of particles per unit volume
$r_b + r_p$ is the collision radius of the bubble and particle
C_i is the induction time constant
ρ_f is the fluid density
ρ_b is the bubble density
ρ_p is the particle density
υ is kinematic viscosity
ε is fluid dissipation energy
C_v is the kinematic viscosity constant
κ is the kinetic energy of the fluid

Flotation Cell Name

Use the tab key or mouse to place the cursor in this window, then enter any desired text. Names must be unique; menu operations and restoring the desktop from a file expect to identify processes by name. The first five characters of this name will be used to label the process icon on the desktop.

Show RECOVERIES

This button will display assay and recovery values computed for all species for both top and bottom output flows, using the current state of all process flows. An example table is shown in Figure 7-2. For species *water,* the values labeled "grade" are "weight fraction water": that is, 1 minus weight fraction solids (shown as a percentage). The display also shows the current concentration of each species as stored in the process model's internal state. The values are not meaningful if the flows do not represent a consistent process state; a warning is given if the flows are badly out of mass balance.

Time interval between cycles / Total simulated time

Ignore these fields.

Process parameters from the model equations and corresponding screen parameter labels and units are:

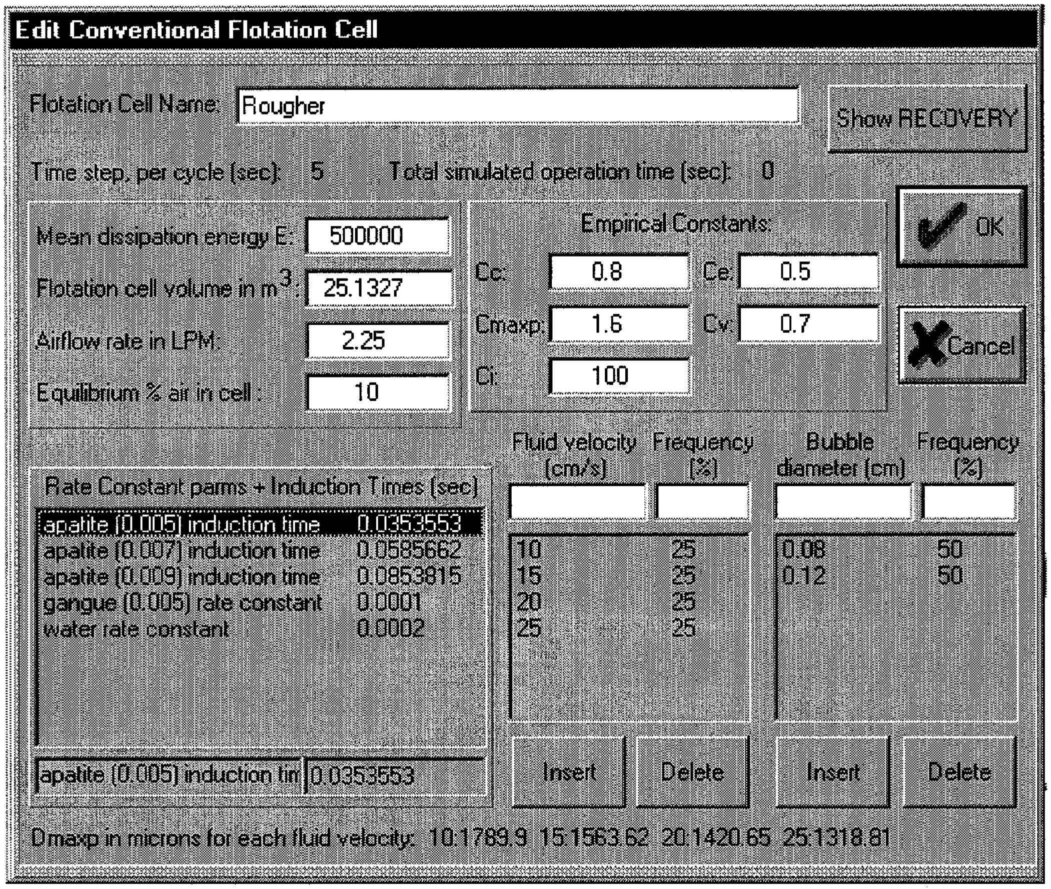
V	Flotation Cell Volume in cubic meters
f_a	Air holdup equilibrium percent volume of air in cell
Q_a	Air rate in liters per minute
C_c	Cc Collision efficiency multiplicative empirical constant
C_e	Ce Collision efficiency exponential empirical constant

C_v Cv Kinematic viscosity empirical constant

C_i Ci Induction time empirical constant

d_{maxp} Cmaxp Maximum attached-particle diameter empirical constant:
d_{maxp} = Cmaxp*(fluid dissipation energy)$^{-1/3}$

E Mean dissipation energy cgs units



Edit Conventional Flotation Cell

Flotation Cell Name: Rougher

Time step, per cycle (sec): 5 Total simulated operation time (sec): 0

Mean dissipation energy E: 500000

Flotation cell volume in m^3: 25.1327

Airflow rate in LPM: 2.25

Equilibrium % air in cell : 10

Empirical Constants:

Cc: 0.8 Ce: 0.5

Cmaxp: 1.6 Cv: 0.7

Ci: 100

Rate Constant parms + Induction Times (sec)

apatite (0.005) induction time	0.0353553
apatite (0.007) induction time	0.0585662
apatite (0.009) induction time	0.0853815
gangue (0.005) rate constant	0.0001
water rate constant	0.0002

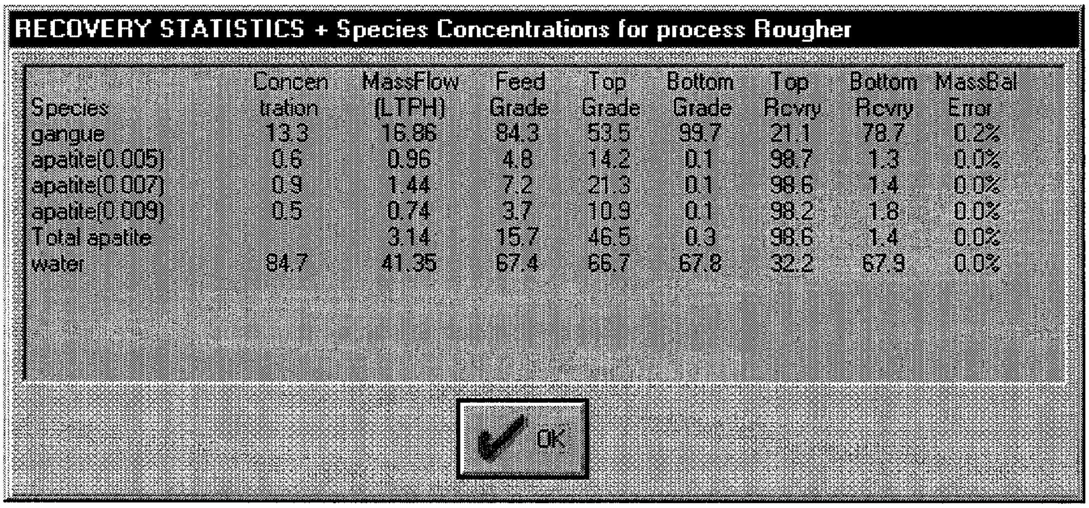
apatite (0.005) induction tim 0.0353553

Fluid velocity (cm/s)	Frequency (%)
10	25
15	25
20	25
25	25

Bubble diameter (cm)	Frequency (%)
0.08	50
0.12	50

Dmaxp in microns for each fluid velocity: 10:1789.9 15:1563.62 20:1420.65 25:1318.81

Figure 7-1. Edit Conventional Flotation Cell Screen, for Jordan-Spears first-principles model



RECOVERY STATISTICS + Species Concentrations for process Rougher

Species	Concentration	MassFlow (LTPH)	Feed Grade	Top Grade	Bottom Grade	Top Rcvry	Bottom Rcvry	MassBal Error
gangue	13.3	16.86	84.3	53.5	99.7	21.1	78.7	0.2%
apatite(0.005)	0.6	0.96	4.8	14.2	0.1	98.7	1.3	0.0%
apatite(0.007)	0.9	1.44	7.2	21.3	0.1	98.6	1.4	0.0%
apatite(0.009)	0.5	0.74	3.7	10.9	0.1	98.2	1.8	0.0%
Total apatite		3.14	15.7	46.5	0.3	98.6	1.4	0.0%
water	84.7	41.35	67.4	66.7	67.8	32.2	67.9	0.0%

Figure 7-2. Table displayed by Show Recoveries button

Rate Constant + Induction Time Parameters

The Jordan-Spears model above is applied to hydrophobic species in the flow; rate constant parameters determine what happens to non-hydrophobic and liquid species. Induction time for hydrophobic species as computed by the Jordan-Spears arithmetic also appears here. Highlight a line to copy its value to the edit field below the list box. Update the value there, then highlight any other line to see the new value copied into the list box. You cannot enter new induction time values directly, but they will be updated if the empirical constant Ci changes (the update occurs when you select a new parameter field).

Distribution of Fluid Velocities
Distribution of Bubble Diameters

Highlight a line (by clicking on it) in either of these distribution list boxes to copy its values to the edit fields above that list box. Click the Delete button under a box to remove its highlighted line. Click the Insert button under a box to enter the edit-field values into the list box. If the edit-field Fluid Velocity value (or Bubble Diameter value) matches an entry already present, that entry's Frequency value will be replaced; otherwise a new line is added to the list. Frequency values will be rescaled to add to 100% if necessary when you leave the screen.

OK and CANCEL buttons

Process-model updates, like Flow-model updates, are performed on a copy of the selected process; no changes are made in the process model actually attached to the desktop icon until you click the OK button. If you click the CANCEL button instead, all changes are abandoned.

7.2 CREATE/EDIT PURE EMPIRICAL FLOTATION CELL MODEL Screen

Reach this screen for an existing conventional flotation cell model from menu choice **Process/Flow / Query/Modify a Process Model,** or by double-clicking its icon; or double-click an unmodeled flotation cell icon, choose Pure Empirical Model, and supply the name of a clustered-data file containing tuning data.

What is the Rate Constant Pure Empirical model?

The mathematical form of the first principles model of Jordan and Spears is a simple first-order differential equation, in which the number of attached particles leaving in the concentrate is directly proportional to the number of attached particles in the flotation cell. The simplest possible model with this essential form is the following differential equation:

$$Qc = kcV$$

where c is the concentration (i.e. volume fraction) of a mineral in the slurry, V is the volume of slurry in the unit (excluding air), *k* is the rate constant associated with the mineral, and Q_C is the volumetric flow rate of the mineral to the concentrate.

If perfect mixing is assumed, the concentration of a mineral in the tails is equal to its concentration in the flotation unit. The volumetric flow rate of the mineral to the tails Q_t is then given by:

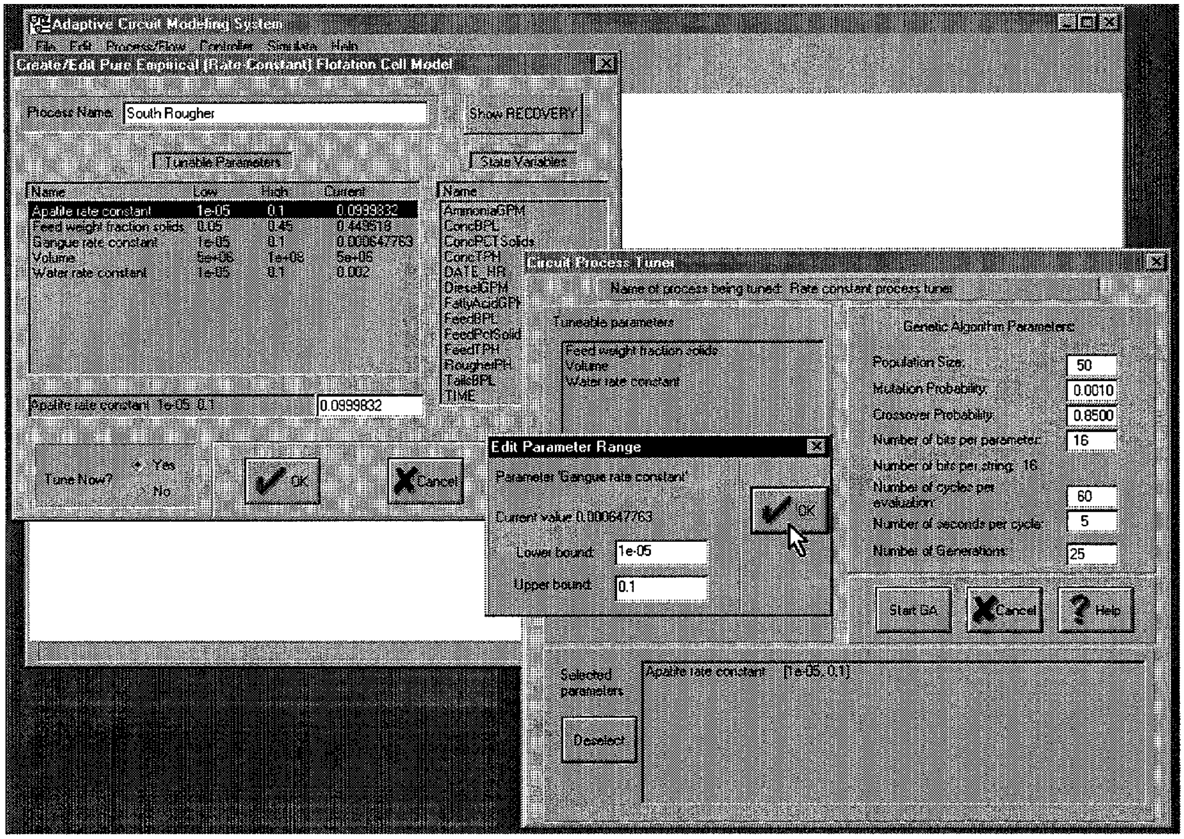
$$Q_t = cQ_{slt}$$

where Q_{slt} is the total volume of the tailing slurry. These equations can be solved for a given feed flow by assuming that the feed is constant long enough for the flotation cell to reach an equilibrium concentration, and then iterating as follows:

$$c_{i+1} = c_i - (Q_c + Q_t - Q_f)\, Dt/V$$

where Q_f and Q_t are the volumetric rates of the floatable species in the feed and tailings, respectively, and *Dt* is a small interval of time. The initial concentration, co, is assumed to equal its concentration in the feed, i.e., $c_O = c_f = Q_f/Q_{slf}$.

The rate constant in this simple model makes sense only for a situation where all of the process parameters such as ore type, conditioning, flotation reagent flow, and bubble size remain constant. This is of course unrealistic. Yet this simplest model can still be effective if it is applied only within a restricted set of operating states. If a number of such states are identified, and a simple model is tuned to perform well for each of those restricted conditions, then the collection of submodels taken together can perform as well as a single complex model tuned over the entire range of process states. The Pure Empirical Rate Constant Model is designed as coordinator for such a collection of separately-tuned submodels. In order to construct such a model, data must be identified corresponding to each of the restricted sets of operating states. The Modeling System uses cluster analysis for this purpose; the Cluster Analyzer is described in Section 8.2.



Process Name: South Rougher

Name	Low	High	Current
Apatite rate constant	1e-05	0.1	0.0999832
Feed weight fraction solids	0.05	0.45	0.449518
Gangue rate constant	1e-05	0.1	0.000647763
Volume	5e+06	1e+08	5e+06
Water rate constant	1e-05	0.1	0.002

Apatite rate constant 1e-05 0.1 0.0999832

Name
AmmoniaGPM
ConcBPL
ConcPCTSolids
ConcTPH
DATE_HR
DieselGPM
FattyAcidGPM
FeedBPL
FeedPctSolid
FeedTPH
RougherPH
TailsBPL
TIME

Tune Now? ● Yes ○ No

Name of process being tuned: Rate constant process tuner

Tuneable parameters:
Feed weight fraction solids
Volume
Water rate constant

Genetic Algorithm Parameters:
Population Size: 50
Mutation Probability: 0.0010
Crossover Probability: 0.8500
Number of bits per parameter: 16
Number of bits per string: 16
Number of cycles per evaluation: 60
Number of seconds per cycle: 5
Number of Generations: 25

Edit Parameter Range
Parameter 'Gangue rate constant'
Current value 0.000647763
Lower bound: 1e-05
Upper bound: 0.1

Selected parameters: Apatite rate constant [1e-05, 0.1]

Figure 7-3. Create/Edit Pure Empirical (Rate Constant) Flotation Cell Model Screen

Process Name

Use the tab key or mouse to place the cursor in this window, then enter any desired text. Names must be unique; menu operations **Delete a Process Model** and **Query/modify a Process Model** identify the targeted process by name. The first five characters of this name will be used to label the process icon on the desktop.

Show RECOVERIES

This button will display assay and recovery values computed for all species for both top and bottom output flows, using the current state of all process flows. For species water, the values labeled "grade" are "weight fraction water": that is, 1 minus weight fraction solids (shown as a percentage). The display also shows the current concentration of each species as stored in the process model's internal state. The values are not meaningful if the flows do not represent a consistent process state; a warning is given if the flows are badly out of mass balance.

State Variables
These are the names of data columns from the clustered-data file.

Tunable Parameters
This list has a rate constant entry for every species in the process's flows, plus an entry for weight fraction solids (same as "Percent Solids" in the Create/Modify Flow screen) and another entry for Volume. You can change the current value for any of these parameters: when you highlight a line in the list, it is copied to the editing line below the list box. Update the value there, then highlight any other line to see the new value copied into the list box. If you enter a value outside the given range, the Low or High limit will stretch as needed. These limits can be modified directly as part of process tuning: when you select a parameter to be automatically tuned, you will be asked to approve or update its upper and lower bounds for the genetic algorithm search.

Tune Now? Buttons
If the Yes button is checked when you click the OK button, you will proceed to the Circuit Process Tuner Screen to select the tunable parameters you actually want the system to set, and to adjust genetic algorithm parameters as desired.

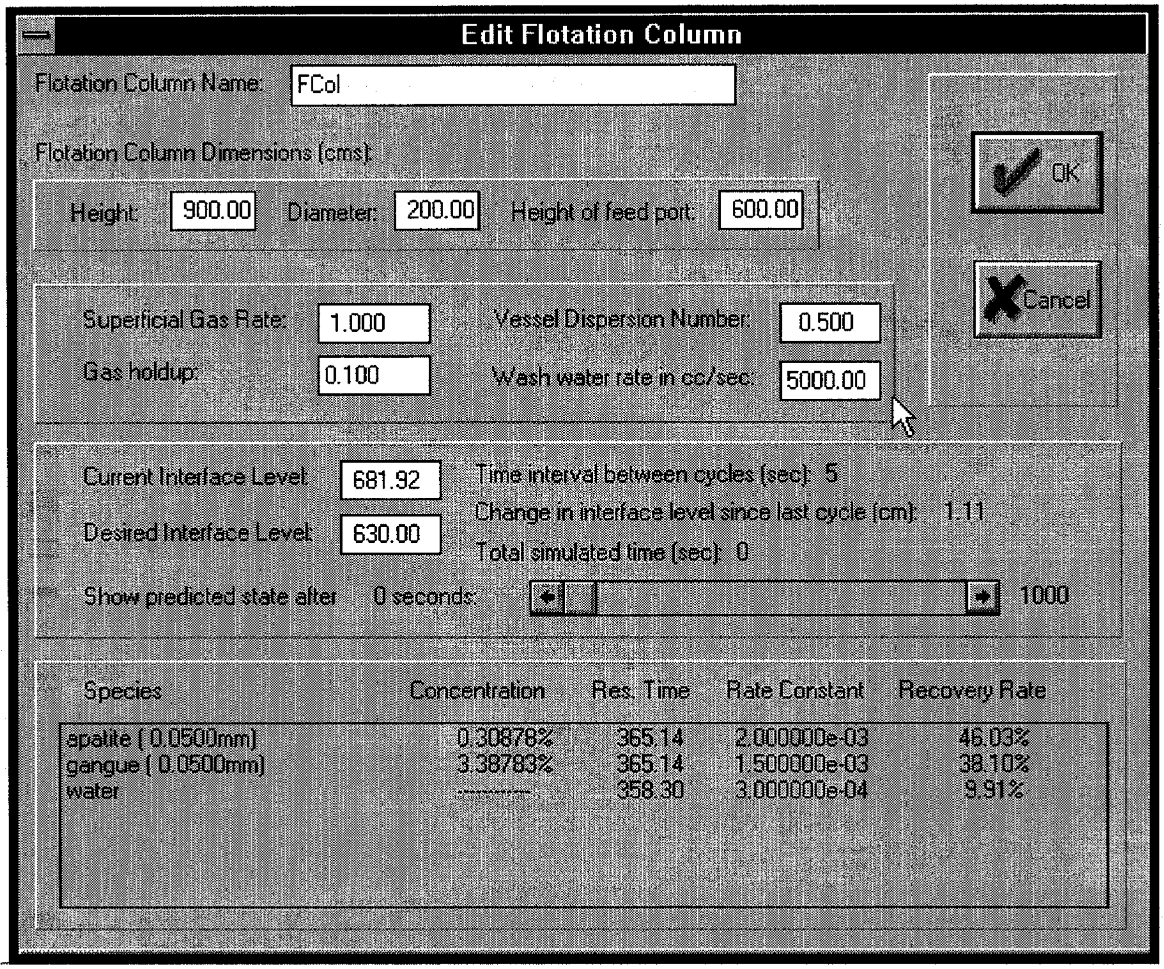
OK and CANCEL buttons
As in all model editing, no changes are actually committed until you click the OK button to exit from this screen.

7.3 EDIT FLOTATION COLUMN Screen

Reach this screen for an existing flotation column model from menu choice **Process/Flow / Query/Modify a Process Model,** or by double-clicking its icon.

What is the flotation column model?

Arithmetic is based on J. A. Finch and G. S. Dobby, *Column Flotation,* Pergamon Press 1990, chapters 2-4. This is the earliest of the models developed for the Modeling System, and is the only model for which fuzzy control was actually implemented. This model, like both flotation-cell models, contains a first-order differential equation for the concentration of hydrophobic particles. The rate constant in the this differential equation depends on such modeled quantities as collection efficiency, slip velocities, bubble and particle Reynolds numbers, and bubble sizes, as well as on model parameters such as column geometry, gas rate, gas holdup, and wash water rate. Most of the listed modeled values are computed by iteration. This model has been little used since the work on fuzzy control, and is now broken: some of the iterative computations do not converge.



Edit Flotation Column

Flotation Column Name: FCol

Flotation Column Dimensions (cms):

Height: 900.00 Diameter: 200.00 Height of feed port: 600.00

Superficial Gas Rate: 1.000 Vessel Dispersion Number: 0.500

Gas holdup: 0.100 Wash water rate in cc/sec: 5000.00

Current Interface Level: 681.92 Time interval between cycles (sec): 5

Desired Interface Level: 630.00 Change in interface level since last cycle (cm): 1.11

Total simulated time (sec): 0

Show predicted state after 0 seconds: 1000

Species	Concentration	Res. Time	Rate Constant	Recovery Rate
apatite (0.0500mm)	0.30878%	365.14	2.000000e-03	46.03%
gangue (0.0500mm)	3.38783%	365.14	1.500000e-03	38.10%
water	----------	358.30	3.000000e-04	9.91%

Figure 7-4. Edit Flotation Column Screen

7.4 CIRCUIT PROCESS TUNER Screen

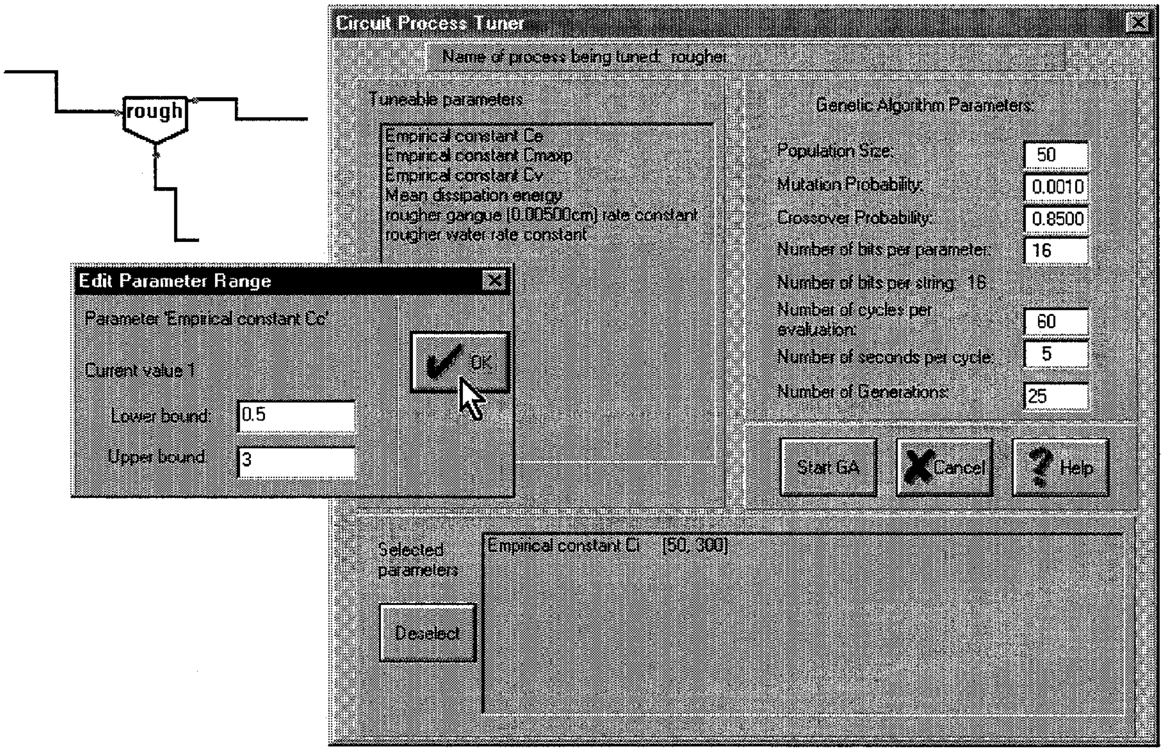
Reach this screen from menu **Process/Flow / Process Tuning / Tune Process** or after setting the Tune Now? Yes button on the Create/Edit Pure Empirical (Rate-Constant) Flotation Cell Model screen.

What is a tuner genetic algorithm?

A genetic algorithm is a very adaptable optimization procedure based on a biological metaphor. There are three key elements in this metaphor. First, view the optimization problem as a search for the best one out of a collection of somethings (best values for a set of model parameters; best order for a set of actions; or whatever); we need a name for these somethings, so call them Ts. Then the first key element is an encoding from the Ts to strings of characters (and back): the metaphor here likens biological organisms to the Ts, and likens chromosomes, genes, and DNA to the character strings that correspond to the Ts. The second key element in the metaphor defines operations on the character strings that correspond to (nonsexual) reproduction in individual biological organisms: exact replication, random mutation, and "crossover": two strands of DNA (or metaphorical DNA) come into contact, and the front part of each string goes off with what used to be the back part of the *other* string. The genetic reassortment provided by crossover is what makes the GA search effective. The third key element in the metaphor defines operations on a *group* of Ts that correspond to reproduction in a population of biological organisms: the better the “fitness” of an organism (or a metaphorical organism), the greater its contribution to the next generation of organisms. This differential reproduction is the engine that drives the GA toward better and better solutions.

Notice that nothing in this scheme guarantees that a GA will find *the best* solution to an optimization problem, only that its solutions should get steadily better as generation follows generation. And nothing tells you when you should stop searching, either. The GA’s population of T-coding character strings is initially filled with random numbers; this population needs to start with enough random variation (that is, needs to be big enough) so that gene shuffling through crossover has a good chance of generating a satisfactory solution.

For some problems, finding a suitable problem encoding and defining the “fitness” function that controls population reproduction is very difficult. But the Process Tuner GA in the Modeling System can use a very simple and straightforward scheme: its “chromosome” is a list of “genes”, one for each model parameter to be tuned; the genes are all the same size; and a gene value is interpreted as a fraction between 0 and 1 telling how far from the corresponding low parameter bound toward the high bound the actual parameter value will be found. The fitness function value for a “chromosome” is a sum (over flows and targets) of sums-of-squared-differences between a flow target and the corresponding flow computed by the model whose parameters come from that chromosome. Note that it is impossible to encode a parameter value smaller than its lower bound or greater than its upper bound, and that the potential precision of a parameter depends on the distance between its bounds.


Circuit Process Tuner

Name of process being tuned: rougher

Tuneable parameters

Empirical constant Ce
Empirical constant Cmaxp
Empirical constant Cv
Mean dissipation energy
rougher gangue [0.00500cm] rate constant
rougher water rate constant

Genetic Algorithm Parameters:

Population Size: 50
Mutation Probability: 0.0010
Crossover Probability: 0.8500
Number of bits per parameter: 16
Number of bits per string: 16
Number of cycles per evaluation: 60
Number of seconds per cycle: 5
Number of Generations: 25

Selected parameters

Empirical constant Ci [50, 300]

Edit Parameter Range

Parameter 'Empirical constant Cc'

Current value 1

Lower bound: 0.5

Upper bound 3

Figure 7-5. **Circuit Process Tuner** Screen for a Conventional Flotation Cell (first-principles) model

Tunable Parameters
Selected Parameters

Between them, these two lists contain all the process’s parameters: see the corresponding process Edit screens. Initially, no parameters are selected. To select a parameter for automatic tuning, double-click on its name in the Tunable Parameters list. First, a small Edit Parameter Range screen gives you the opportunity to adjust the parameter’s upper and lower bounds. Both bounds must be present: the GA uses them in its parameter-encoding scheme. When you click OK to leave the Edit Parameter Range screen, you will see that the parameter name has moved from the Tunable Parameters list to the Selected Parameters list.

Deselect Button

Highlight (by clicking on) a parameter in the Selected Parameters list, then click the Deselect button to move the parameter back to the Tunable Parameters list. You must deselect a parameter first if you want to change its upper and lower bounds again.

Genetic Algorithm Parameters

You are unlikely to want to change the following parameters:

Mutation Probability
Crossover Probability
Number of bits per parameter
Number of cycles per evaluation
Number of seconds per cycle

Number of cycles per evaluation is ignored by the Pure Empirical model but used by the first principles flotation cell model. The number of bits per parameter determines the “graininess” or “mesh” of the parameter search; if you are just experimenting, there is a slight speed advantage when the total number of bits for all parameters is 32 or fewer.

The main choices you must make are

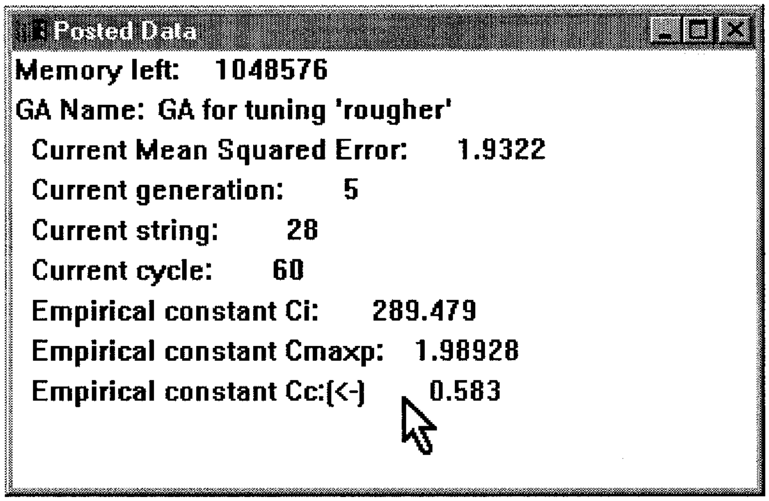
Population size
Number of Generations

The default values (50 strings in the population, reproducing for 25 generations) are as a rule of thumb adequate to tune three parameters (48 bits). For more parameters, increase-the population and generation count proportionally. The time required for each step depends on the number of targets and the type of model as well as on the population size; the tuning display in the Posted Data window (shown in Section 7.5) will show how far the tuning has progressed. Expect that a serious attempt to tune a Pure Empirical model will take hours, not minutes. You can abandon the tuning process with the menu choice **Process/Flow / Process tuning / Quit tuning process model.**

7.5 POSTED DATA Window During Process Tuning

Figure 7-6 shows data displayed while tuning a first principles flotation cell model. As the genetic algorithm finds better and better model parameters, the Current Mean Squared Error decreases toward zero. These error values are not comparable from one model or set of tuning data to another: in one context 0.17 may represent a mediocre model fit, and in another context indicate a very good fit indeed. Instead watch for relative improvement, which should be rapid at first and then slow down.

You can estimate the time tuning will require: “Current string” must step through the numbers 1 to Population Size (as you specified it on the Circuit Process Tuner screen) for each generation; you will also briefly see a higher value, as the GA sets up for the next generation. “Current generation” must step through the numbers from 1 to Number of Generations. Each of these steps should take just about the same length of time, so you can check how many steps get done in fifteen seconds or a minute and then multiply. If you are tuning a Pure Empirical model, these steps must be repeated for each cluster of data points (since each cluster gets its own set of model parameters), and the time required for a step is directly proportional to the number of data points in the cluster. So to estimate the time remaining, you must know something about how many points are in each cluster. You can find this information by looking at the clustered-data output file which was used as input for the model: see the annotated example file in Section 9.2.



```
Memory left:   1048576
GA Name:  GA for tuning 'rougher'
  Current Mean Squared Error:     1.9322
  Current generation:      5
  Current string:       28
  Current cycle:      60
  Empirical constant Ci:      289.479
  Empirical constant Cmaxp:   1.98928
  Empirical constant Cc:[<-]     0.583
```

Figure 7-6. Posted Data Window during process tuning

This window shows values for the best model parameters found so far, at the end of the display. Notice the arrow “(🖫📫)” after rougher empirical constant Cc: this is an indication that this value is very close to its lower limit. A similar right-pointing arrow indicates that a parameter is close to its upper limit. If the best values the GA can find for a parameter are often at one limit, you probably need to adjust the boundaries of the GA’s search.

8. STANDALONE MODELING SYSTEM COMPONENTS

8.1 STANDARD DATA FILE FORMAT

Actual plant operating data is collected from many sources. Data may consist of automatic sensor and control readings, information from on-line chemical analysis, values from off-line laboratory analysis of pooled material samples; derived values, computed from any combination of these; *post* hoc calibration corrections; and so on. Values are taken with different periodicities and represent aggregates over various time intervals and various process and subprocess units of the plant. To organize such data for computer modeling is not a small task.

The Modeling System needs to see data organized into records that say “here is a description of a feed flow and maybe of some current process state; and here is a description of what the process produced under those conditions.” Since ultimately the goal is process control, the interesting outcome values are concentrate grade and recovery (or other measurements from which grade and recovery could be computed). These values are relatively expensive to measure, and tend to be aggregated over relatively long periods. You must look at the data available to you and decide first what the basic modeling time interval will be -- that is, will you try to model average-outcome-over-five-minutes or average-outcome-over-four-hours? Since the Modeling System compares equilibrium model flows to measured flow states, using it to model instantaneous process outcomes would not make sense even if you had the necessary data. Suppose your interval is fifteen minutes. You next begin to assemble data describing fifteen-minute chunks of the process’s feed and corresponding fifteen-minute chunks of the process’s concentrate and tails flows, along with outcome and other measures (conditioner-unit pH, say) that refer, more-or-less, to the same chunk of material. This may require averaging some data series and inserting interpolated values into other series, and is sure to require that you account for expected residence times for various processes and subprocesses and any expected material-flow delays. The way you regularly compute grade and recovery values may mean that your data is automatically consistent and in mass balance; if not, you must decide whether to adjust the data or to live with the fact that the Modeling System could never match your flow targets exactly, so that the minimum possible prediction error is greater than zero.

The previous paragraph said nothing about *what* data you should assemble, except for your outcome measurements. This is because the Modeling System has a data-description language that should be able to identify whatever measurements you bring to the modeling process. The Modeling System does assume that every record has a value for every kind of measurement: that is, that you can arrange your data set as a spreadsheet (columns are measured variables, rows are fifteen-minute material chunks) without any holes. Now to create a file in standard data format you must build a prefix section describing each column of your assembled data, and follow the prefix by your data. The data items do not have to be in fixed locations in each line of the resulting text file, but all numbers must be separated by white space (at least one blank or tab). The data items do not have to fit on a single line of the text file and a line of text may be any length; but items within the line should be limited to 100 characters.

The column descriptors tell (a) the name(s) of the process(es) to which the data refers, (b) the relevant input or output flow(s), (c) the category of data (e.g. percent solids, mineral assay, element assay, flow rate), (d) the units in which the data is expressed, and (e) an indication of whether the data column is to be used for tuning or as information for statistical clustering. In the latter case there is no restriction on the data category. Even before the column descriptors in the prefix section, you must list the processes to which they will refer. The Modeling System expects that one column will contain some kind of record identifier, different for every row of data; this identifier does not have to be numeric, and might well indicate the date and time the records measurements cover..

The data description language is shown below. There are just two kinds of statement: PROCESS and COLUMN. Words in capitals are keywords and must be spelled exactly as shown. Words in lowercase represent information you must supply; see the comments below the line. Quotation marks are required as shown. Items or phrases in square brackets may be omitted. Number format is quite general: if your spreadsheet accepts a number spelling, so should the data description language. Statements do not have to fit on one line of your data file. See annotated files in Section 9 for some examples.

PROCESS number "processname"

number Process numbers need not be consecutive or even in order, but they must all be different.

processname Quotation marks are required. This name matches the name of the Pure Empirical process model that will be tuned using this data.

COLUMN cnumber colname IDENTIFICATION END
COLUMN cnumber colname IGNORE END
COLUMN cnumber colname CLUSTER-ONLY END
COLUMN cnumber colname PROCESS pnum [whichflow] whichinfo [units] [species] [error] END

cnumber The ordinal number for the data column this statement describes: the first column is number 1. Every column in your data must have exactly one descriptor statement. The column descriptors may appear in any order.

colname A name for the data column. Quotation marks are not necessary unless the name contains whitespace or starts like a number (with a digit or one of the three characters +.-) or starts with a keyword.

IDENTIFICATION says this is the required column containing a unique value for each record.

IGNORE says nothing will use the values in this column: the Cluster Analyzer will not copy it.

CLUSTER-ONLY says the values in this column *will* be used for clustering (and hence for identifying which Pure Empirical submodel -- i.e. which of the sets of tuned model parameters -- should be used for a given test-data observation), but the values *will not* be used for constructing any flow targets from the data observations.

PROCESS says this column's values describe a process or flow; and *will* be used in constructing flow targets.

pnum says this column contains a measurement that describes *flow whichflow* of process number *pnum*. The process number must have appeared on a PROCESS statement. The statement must provide a value for *whichflow* unless *whichinfo* is one of the RECOVERY keywords (in that case, the word BOTTOM or TOP serves the same function).

whichflow is one of the three keywords FEED or CONCENTRATE or TAILS.

whichinfo is one of the six keywords or phrases MASS-RATE or VOLUME-RATE or ASSAY or SOLIDS (as in "feed weight fraction solids") or RECOVERY BOTTOM or RECOVERY TOP.

units is one of the seven keyword phrases:

UNITS LPM	liters per minute
UNITS GPM	gallons per minute
UNITS GMPS	grams per second
UNITS TPM	long tons per minute
UNITS TPH	long tons per hour
UNITS PERCENT	dimensionless, 100 means all
UNITS FRACTION	dimensionless, 1 means all

species is keyword MINERAL or ELEMENT followed by a mineral or element name in quotation marks. The name may use upper or lower case letters, but the spelling must match a Modeling System entry as seen in screens Add/Edit Species and Target Levels for Tuning.

error is keyword ERROR or DIGITS followed by a number. This phrase provides an estimate of the accuracy of this column's value, used (when present) in process tuning. ERROR 0.01 is an absolute error bound meaning "value is accurate to plus-or-minus-1/100", DIGITS 3 is a relative error bound meaning "value is accurate to three digits".

The keyword BPL is shorthand for the phrase ASSAY UNITS PERCENT MINERAL "apatite" in a data file, but the Modeling System will always use the long phrase when displaying data descriptions, as in the Cluster Analyzer screen and the Pure Empirical model validation output data file.

8.2 CLUSTER ANALYZER

See Reference 2. The Cluster Analyzer performs principal-components analysis on selected data from a standard data file, then locates clusters within that data, usually but not necessarily based on the extracted principal-components. Its menu offers just three choices: Cluster a data file, Verify an output clustered-data file, and Exit the program; the Verify operation is provided for diagnostic purposes only. The Cluster Analyzer produces two output files: one is the file of tuning data used by the Modeling System, and the other is the "movie" file displayed by the Data Viewer. While the Cluster Analyzer is designed to identify points to be modeled together, sometimes you may need to use it just to format a set of measurements so the Modeling System or Data Viewer can read them.

What are principal-components analysis and cluster analysis?

Both methods are designed to tease useful and concise data characterizations out of masses of highly correlated measurements. Picture a data observation as a point in a space of many dimensions (just as a scatterplot consists of points in a space of two dimensions): each column of measurements in the data file is one dimension in the space. Principal-components analysis tries to find a space of much smaller dimension that contains (or almost contains) nearly all the points, so that each point can be located using just a few values rather than many; cluster analysis tries to classify the points into a small number of groups of similar observations.

If the observations in a data set are measurements made on a physical or chemical process with only a few real degrees of freedom, then (if the measurements are scaled so their dependencies are linear or nearly linear) the measurements certainly ought to lie in a subspace whose dimensions correspond to those real degrees of freedom, cutting slantwise through the measurement space in a way that corresponds to the dependencies among the data measurements. Principal-components analysis can be viewed as a change of coordinates to a system where (as much as possible) the new coordinates for each observed point are almost zero except in the first handful of positions: these first dimensions then identify the lower-dimension space where the points lie. Since the observations ought to lie entirely within that space, their higher-dimension coordinates can be identified with measurement error; removing that error should improve modeling effectiveness.

The cluster analysis method used by the Modeling System is agglomerative: it finds clumps of points very near to each other in the data space, then extends these cluster nuclei by adding points in a stepwise fashion. The Ginsberg-Whiten method is well-adapted for engineering data embodying functional relationships, where in some dimensions corresponding to degrees of freedom the observed values are distributed broadly, while in other dimensions corresponding to functional dependencies the observed values are tightly constrained. This situation leads to long and thin rather than compact clusters. The method compensates for this by rescaling the space so such a cluster *appears* compact: that is, one cluster-distance unit is very large along the long axis of a cluster but is much smaller in other directions. Any clustering method based on sum-of-squares distance computations is sensitive to the scales chosen for its axes, but this method is progressively less sensitive to the initial scaling of the space as it aggregates points into a long thin cluster (when that pattern is in fact found in the data). However when the cluster has very few points its apparent long direction is likely to be very different from the “true” direction, and the adaptive scaling hinders rather than helps cluster recognition. The Cluster Analyzer enforces a minimum value for the number of points in a cluster nucleus to avoid this problem.

When you select the **Cluster** menu item, the Cluster Analyzer first asks you to identify an input file. This must be a standard data file as described in the preceding section. When the prefix section of this file has been read, the Cluster Analyzer asks you to identify output files and set clustering parameters using the busy screen of Figure 8-l.

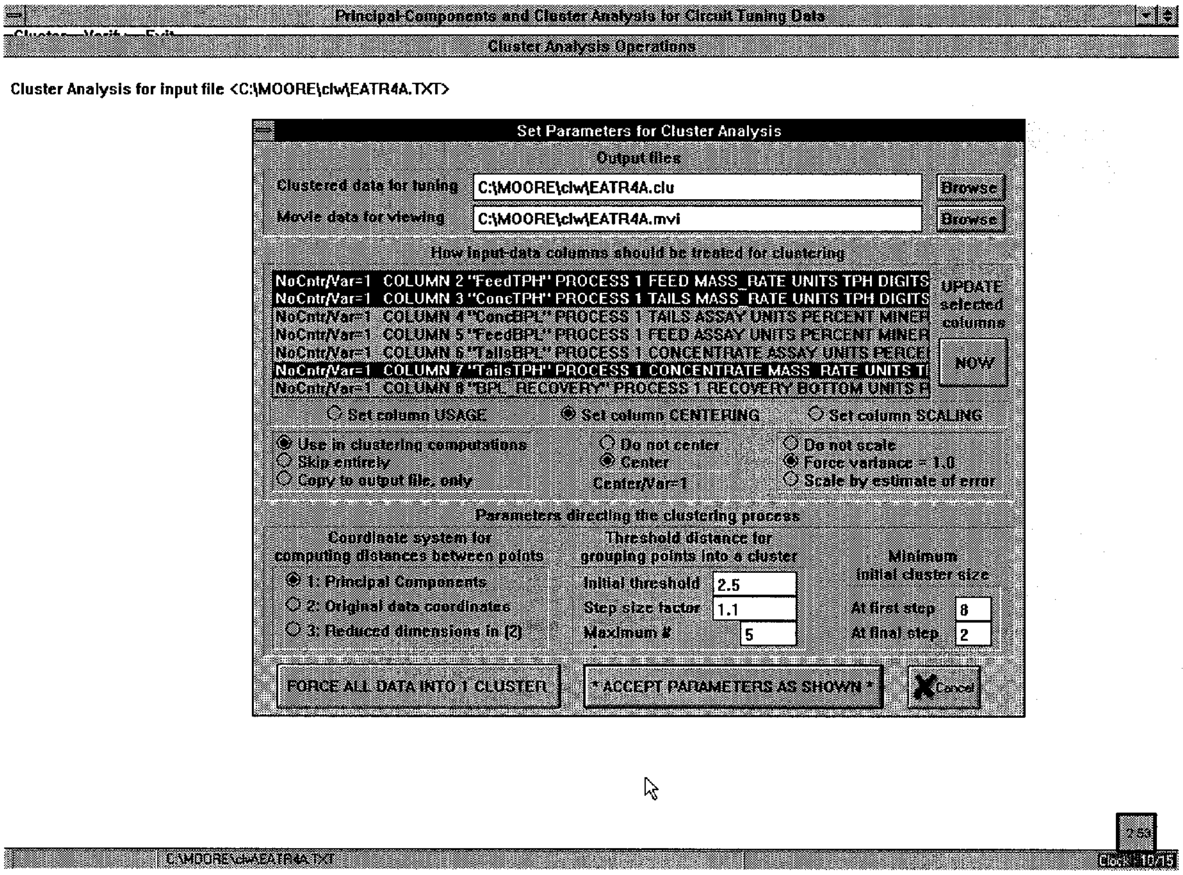
Output Files

The Cluster Analyzer will write two files: one is the tuning-data file that can be used as the basis for a Pure Empirical process model (expected extension *.clu)* and the other contains your data and the results of the principal-components and cluster analyses, all formatted for display by the Data Viewer (expected extension *.mvi*). You can enter the names for these files directly in the edit fields, or you can click the Browse button at the end of an edit field to use the standard Windows file-open dialog to identify the file.

How Input-Data Columns Should Be Treated for Clustering

You must decide which columns in your data file should be used as dimensions of the measurement space for principal-components and cluster analysis; which of the remaining columns you want included in the output files; and which you want to ignore entirely. The IDENTIFICATION column in your data is needed for model tuning but ought not be used for clustering or principal components. You may also want a sequence-number field available for use in scatterplots made by the Data Viewer. Ignored columns must be marked *Skip* in this list, and output-only columns must be marked Copy. For each of the other columns, you must determine whether and how to transform the data: you can center the column (force its mean to be zero: not relevant to clustering but significant to principal-components extraction), and scale it (multiply so its variance or its error is a standard magnitude, significant for both analyses).

Having decided, you must enter your decisions in the list of data columns. To mark one or more columns *Skip,* click on the Set column USAGE button in the row under the list box; all data columns will be displayed in the box. Highlight exactly the lines in the box that need to be changed (click on a line to toggle its highlighting from off to on and back again), and set the Skip entirely button (so it has a dark center). Then click the UPDATE Selected Columns NOW button at the right of the list box: the usage flags at the left of the selected list box entries will be updated to *Skip,* which is the value shown at the bottom center of the button display. (This value always corresponds to the current setting of the usage buttons, even when not all of those settings are relevant to what will happen if you ask to update usage values.) Follow the same procedure to mark columns to be copied, setting the Copy to output file, only button. If some columns are marked *Skip* or *Copy* but you want to include them in the cluster analysis, you must update them in the same way: set the Use in clustering computations button and click the update button. The usage flags for the selected columns will be set to NoCentering/NoScaling. Setting the usage flags for columns included in the cluster analysis requires separate steps for updating the centering choice and the scaling choice. Set one of the buttons Set column CENTERING or Set column SCALING; mark the appropriate usage choice below your selected button; then highlight the listbox lines you want to change (lines whose usage is set to *Skip* or *Copy* will not be displayed), and click the UPDATE Selected Columns NOW button.


Principal-Components and Cluster Analysis for Circuit Tuning Data
Cluster Analysis Operations
Set Parameters for Cluster Analysis
Output files
Browse
Browse
How input-data columns should be treated for clustering
UPDATE selected columns
NOW
Parameters directing the clustering process
FORCE ALL DATA INTO 1 CLUSTER
* ACCEPT PARAMETERS AS SHOWN *
Cancel
Clock 10/15
7:53

Cluster Analysis for input file <C:\MOORE\clw\EATR4A.TXT>

Clustered data for tuning: C:\MOORE\clw\EATR4A.clu

Movie data for viewing: C:\MOORE\clw\EATR4A.mvi

```
NoCntr/Var=1  COLUMN 2 "FeedTPH" PROCESS 1 FEED MASS_RATE UNITS TPH DIGITS
NoCntr/Var=1  COLUMN 3 "ConcTPH" PROCESS 1 TAILS MASS_RATE UNITS TPH DIGITS
NoCntr/Var=1  COLUMN 4 "ConcBPL" PROCESS 1 TAILS ASSAY UNITS PERCENT MINER
NoCntr/Var=1  COLUMN 5 "FeedBPL" PROCESS 1 FEED ASSAY UNITS PERCENT MINER
NoCntr/Var=1  COLUMN 6 "TailsBPL" PROCESS 1 CONCENTRATE ASSAY UNITS PERCE
NoCntr/Var=1  COLUMN 7 "TailsTPH" PROCESS 1 CONCENTRATE MASS_RATE UNITS T
NoCntr/Var=1  COLUMN 8 "BPL_RECOVERY" PROCESS 1 RECOVERY BOTTOM UNITS R
```

○ Set column USAGE ● Set column CENTERING ○ Set column SCALING

● Use in clustering computations
○ Skip entirely
○ Copy to output file, only

○ Do not center
● Center
Center/Var=1

○ Do not scale
● Force variance = 1.0
○ Scale by estimate of error

Coordinate system for computing distances between points
● 1: Principal Components
○ 2: Original data coordinates
○ 3: Reduced dimensions in (2)

Threshold distance for grouping points into a cluster
Initial threshold: 2.5
Step size factor: 1.1
Maximum #: 5

Minimum initial cluster size
At first step: 8
At final step: 2

C:\MOORE\clw\EATR4A.TXT

Figure 8-1. Cluster Analyzer asks for file and parameter information to create a clustered data file.

Parameters Directing the Clustering Process

- Coordinate system for computing distances between points

  Choices *Principal Components* and *Reduced Dimensions* should result in the same clusters; the latter corresponds to error-reduced data. In both cases, after the principal-components analysis you will be shown some results and asked how many dimensions your data should be reduced to.

- Threshold distance for grouping points into a cluster

  The Cluster Analyzer uses a stepwise method for constructing clusters: points whose (scaled) distance to a cluster is less than a threshold will be added to the cluster, and the scaling will be recomputed. When no more points can be assigned to clusters, the threshold is multiplied by the stepsize factor and the search is restarted. Parameters are: an initial threshold value (the greater the dimension of the space, the larger this value should be), the stepsize factor, and the number of steps the Cluster Analyzer will take before it declares remaining unclustered points to be outliers (such points are excluded from the tuning data file).

- Minimum initial cluster size

  The larger the minimum size for a cluster, the harder it is to find a nucleus for a cluster but the less likely it is that changing the order of the data (for example) could change the assignment of points to clusters. Setting the minimum size to a small value at the final step will place isolated small clusters of points into the tuning-data file rather than excluding them as outliers.

FORCE ALL DATA INTO 1 CLUSTER and ACCEPT PARAMETERS AS SHOWN and CANCEL buttons

CANCEL abandons the analysis and returns to the Cluster Analyzer's main menu. *ACCEPT PARAMETERS is a normal OK button: the principal-components and cluster analysis will proceed according to the instructions on this screen. Use FORCE ALL DATA when you want to use the Cluster Analyzer just to format a data set so you can present it to the Modeling System or the Data Viewer: threshold and cluster size parameters on the screen will be ignored, and substitute values will be set to ensure that all points will be included in a single cluster.

8.3 DATA VIEWER

The Data Viewer began as a simple validator for the Cluster Analyzer. While cluster analysis can be very useful, it is not a highly robust technique. With all clustering algorithms, for some data sets the clusters actually constructed can be affected strongly by changing the order of the data points, or by removing one or a handful of the points. So the Viewer's first function was to show an animated clustering "movie" so that a human could verify that the outcome made sense. But the data display itself proved so helpful to the project researchers that those functions are now more prominent than the clustering animation.

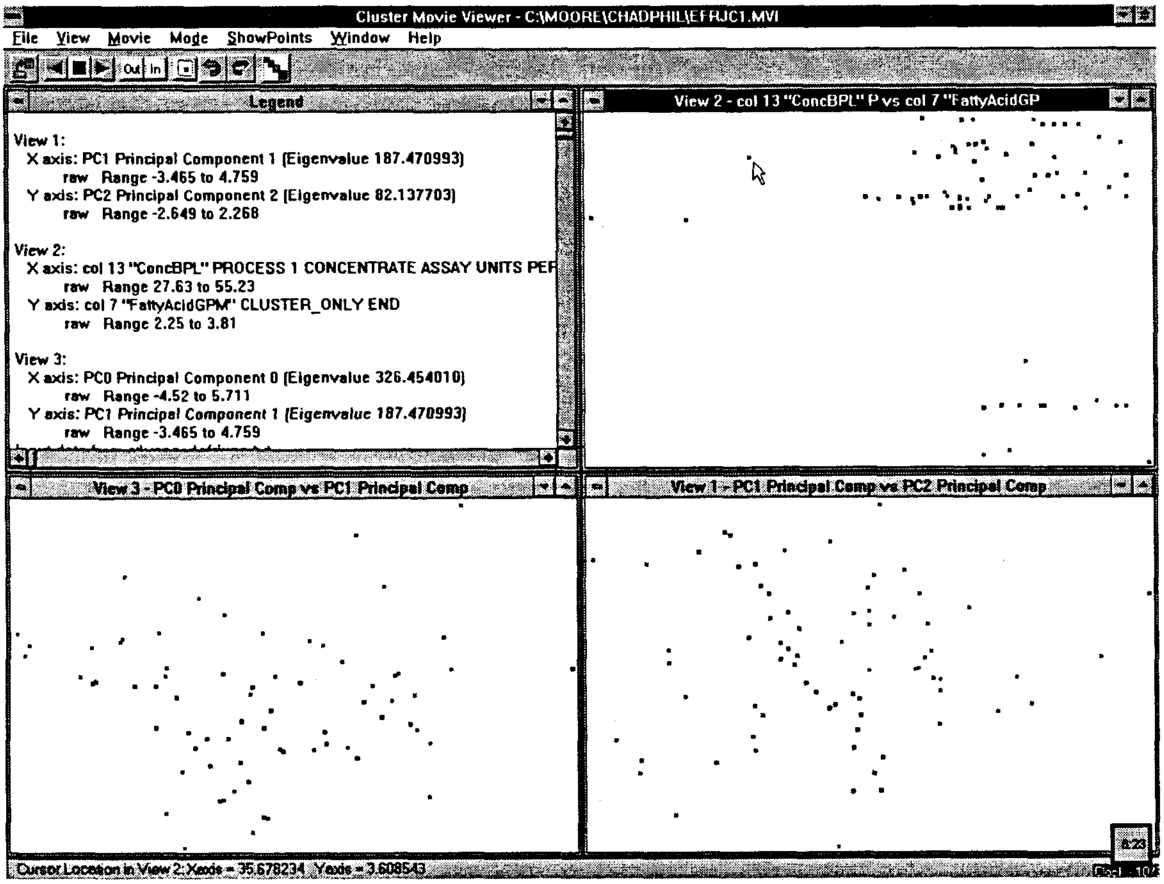
The Viewer offers support for several decisions about the data that need human supervision, since no universal robust algorithm exists for them.

(1) What is the "true" data dimension? The measurements are not independent of one another (though the error in the measurements should be), so one method for removing measurement error from a data set is to reduce its dimension. Usually this is done by projecting the data onto the span of the first several principal components. But how many should be used?

(2) Are the clusters as identified by the clustering algorithm reasonable?

(3) Which data points should be excluded from further analysis because they are "outliers," containing too much measurement error or representing inappropriate plant operating conditions?

The viewer displays scatterplots of the data in coordinate systems derived from the principal-components analysis: each axis may show a measured value, a measured value scaled as requested for the cluster analysis, a corrected measured value after error reduction as described in (1) above (or the corresponding computed error) based on any number of dimensions, or a principal-component loading. Any number of scatterplots can be displayed at the same time. The Viewer uses scatterplot dot color (changing with time) to show the aggregation of points into clusters. Since a point is shown in the same color in every scatterplot, the effect is to highlight data relationships. The Viewer can also step directly through all the points (which will normally be presented in order by time, so this animation traces plant operation through time). The user can run an animation frame-by-frame or continuously (at varying speeds), forwards or backwards. The user can also use the mouse to select one point or a clump of points in one scatterplot; these points are then highlighted in all the views, and can be saved to a separate data file for further analysis. An example showing the results of such a selection operation is shown in Figure 8-2. Since each plot is small when many are displayed at once, it is possible to zoom in and out and pan over each display in order to make it easy to see any area of interest. The user can also take control of plotting-dot symbols and colors, which are defined in an initialization file whose format is shown in Section 9.6..


Cluster Movie Viewer - C:\MOORE\CHADPHIL\EFRJC1.MVI
File View Movie Mode ShowPoints Window Help
Out In
Legend
View 2 - col 13 "ConcBPL" P vs col 7 "FattyAcidGP
View 3 - PC0 Principal Comp vs PC1 Principal Comp
View 1 - PC1 Principal Comp vs PC2 Principal Comp
8:23

View 1:
X axis: PC1 Principal Component 1 (Eigenvalue 187.470993)
raw Range -3.465 to 4.759
Y axis: PC2 Principal Component 2 (Eigenvalue 82.137703)
raw Range -2.649 to 2.268

View 2:
X axis: col 13 "ConcBPL" PROCESS 1 CONCENTRATE ASSAY UNITS PEF
raw Range 27.63 to 55.23
Y axis: col 7 "FattyAcidGPM" CLUSTER_ONLY END
raw Range 2.25 to 3.81

View 3:
X axis: PC0 Principal Component 0 (Eigenvalue 326.454010)
raw Range -4.52 to 5.711
Y axis: PC1 Principal Component 1 (Eigenvalue 187.470993)
raw Range -3.465 to 4.759

Cursor Location in View 2: Xaxis = 35.678234 Yaxis = 3.608543

Figure 8-2. The Data Viewer highlights the same data points in each of three scatterplots based on one data file.

8.3.1 Viewer Menu Operations

File Menu

File | Open	Open a new *.mvi* movie file
File | Close	Close the current movie file
File | Save Selected Points	Write a subset of the data to a file for separate processing (see menu item Mode)
File | ColorDisplay	Show the color patterns that can be drawn with the current colors
File | Exit	Exit the program

View Menu

View | New Movie View	Open a new scatterplot subwindow for the current movie file
View | Legend Window	Open a text subwindow listing axes for scatterplots, the datafile prefix, and information about the clustering process
View | New Axes for View	Show Axis Selection screen to choose new coordinates for active scatterplot

Movie Menu

Movie | Jump to Step	Jump directly to a frame of the animation, selected by sequence number
Movie | Forward	Run animation in forward direction
Movie | Backward	Run animation in backward direction

Movie | Stop	Pause the movie animation
Movie | Zoom in	Decrease coordinate ranges for active scatterplot
Movie | Zoom out	Increase coordinate ranges for active scatterplot

Mode Show Movie Play Mode screen to choose Continuous Play or Single Step or Select Points mode, and set Frames Per Second. Points are selected by surrounding them by a polygon, formed by clicking the left mouse button to add a vertex and clicking the right mouse button to delete the most recent one; the polygon is complete when the last vertex matches the first one. Selected points are drawn using the selection dot pattern that is current when the polygon is completed.

ShowPoints Menu

ShowPoints | Jump to SelectionDot Index	Show Select ColorDot screen to choose dot pattern by number
ShowPoints | Next SelectionDot Pattern	Set selection dot pattern to next one in sequence
ShowPoints | Undo Selection Step	Uncolor the most recently colored set of selected points
ShowPoints | Redo Selection Step	Recolor the next UNDOne set of selected points
ShowPoints | Next PlottingDot Size	Change to next size dot, for all points in all plots

Window Menu

Window | Cascade
Window | Tile
Window | Tile Horizontally

Help Open a text subwindow and fill with lines of help text as found in the *.ini* file

8.3.2 Viewer Toolbar Buttons

All buttons are identical to some menu choice. Buttons are:

📂	File | Open	▣	ShowPoints | Next SelectionDot pattern
◀	Movie | Backward	↶	ShowPoints | Undo Selection Step
■	Movie | Stop	↷	ShowPoints | Redo Selection Step
▶	Movie | Forward	◪	ShowPoints | Next PlottingDot Size
In	Movie | Zoom in		
Out	Movie | Zoom out		

8.3.3 Viewer Mouse Operations

Use the right mouse button to shift the center of a plot display: move the cursor to a selected point, depress the right button, move the cursor (holding the button down), then release the button: the display will shift so that the selected point is drawn under the new cursor location. There is no visual feedback until the display is redrawn. Use the left mouse button to show coordinates in a plot-display: when you move the cursor while the left button is depressed, the cursor's coordinates are displayed in the status line at the bottom of the view window. In point-selection mode, use the left mouse button to draw a selection polygon: click once on each vertex point in order, and finish by clicking on the first point again to close the figure. More exactly, the first depress-left-button action sets the initial polygon node; additional nodes are added by left-button-up actions. If you hold the left button down while moving the mouse, you will see prospective polygon edges as well as completed ones. Nodes can be deleted (last added is first deleted) by clicking the right mouse button in any location; when no nodes are defined, the right mouse button can again be used for panning the display.

9. ANNOTATED EXAMPLE FILES

9.1 Example of STANDARD DATA FILE

The following is the text of the tiny example file SFR.TXT used to tune the Pure Empirical model for the South Rougher process in Section 3: HANDS-ON II: ESTABLISH THEN USE FLOW AND PROCESS MODELS. The measurements are arranged in neat columns so you can read them, but the Modeling System does not require this; it does however require that none be missing. The columns are given somewhat mnemonic names. The reagent measurements are marked CLUSTER_ONLY; measurements that fix the mass balance of apatite, gangue, and water are described using the Modeling System's data language. This data set has no redundant measurements, so there is no direct way to estimate the accuracy of the measurements.

```
PROCESS 1 "South Fine Rougher"
COLUMN 1 DATE_HR IDENTIFICATION END
COLUMN 2 TIME IGNORE END
COLUMN 3 FeedBPL PROCESS 1 FEED ASSAY MINERAL "Apatite" UNITS PERCENT END
COLUMN 4 TailsBPL PROCESS 1 TAILS ASSAY MINERAL "Apatite" UNITS PERCENT END
COLUMN 5 FeedTPH PROCESS 1 FEED MASS_RATE UNITS TPH END
COLUMN 6 FeedPctSolids PROCESS 1 FEED SOLIDS UNITS PERCENT END
COLUMN 7 FattyAcidGPM CLUSTER_ONLY END
COLUMN 8 AmmoniaGPM CLUSTER_ONLY END
COLUMN 9 DieselGPM CLUSTER_ONLY END
COLUMN 10 RougherPH CLUSTER_ONLY END
COLUMN 11 ConcTPH PROCESS 1 CONCENTRATE MASS_RATE UNITS TPH END
COLUMN 12 ConcPCTSolids PROCESS 1 CONCENTRATE SOLIDS UNITS PERCENT END
COLUMN 13 ConcBPL PROCESS 1 CONCENTRATE ASSAY UNITS PERCENT MINERAL "Apatite" END
53100 "00:55:28"  19.78 8.03  669.1357 34.28222 3.45  1.9   0     8.52  236.886  71.1938   41.22042
53101 "01:55:28"  19.95 2.13  650.0562 33.46442 3.46  1.9   0     8.52  233.207  83.53473  51.8026
53102 "02:55:28"  15.34 1.82  546.2584 29.1807  3.44  1.9   0     8.55  176.769  61.85053  43.60003
53104 "04:55:28"  12.37 4.59  649.1897 33.46442 3.35  1.9   0     8.53  182.293  65.09849  32.29647
53105 "05:55:28"   9.34 3.14  568.909  30.06442 3.36  1.92  0.4   8.52  143.999  53.85684  27.63485
53114 "14:16:49"  12.65 3.67  661.1489 34.28222 3.63  1.91  0.38  8.29  187.071  70.54483  35.40724
53115 "15:16:49"  24.07 7.2   681.5004 35.08772 3.55  1.95  0.38  8.51  266.623  82.61603  50.32049
53116 "16:16:49"  20.21 3.32  631.9252 33.46442 3.55  1.98  0.38  8.49  230.315  79.72238  49.66181
53117 "17:16:49"  21.69 2.26  659.66   34.28222 3.55  2     0.38  8.52  246.729  89.21094  54.20846
53118 "18:16:49"  21.37 5.96  694.0212 35.08772 3.64  1.98  0.38  8.55  253.202  82.04914  48.19847
53119 "19:16:49"  28.54 9.56  681.0259 34.28222 3.7   1.98  0.37  8.6   291.279  80.92634  53.93626
53120 "20:16:49"  19.87 4.81  662.3834 34.28222 3.7   1.98  0.38  8.57  235.782  79.04495  47.1181
53121 "21:16:49"  20.31 4.7   583.4647 30.93434 3.69  1.96  0.37  8.62  218.6    66.54737  46.36459
53122 "22:16:49"  19.73 3.58  680.6479 35.08772 3.69  2.04  0.37  8.58  239.428  85.90452  49.49137
53123 "23:16:49"  18.69 4.19  684.6951 35.08772 3.69  2.1   0.37  8.54  233.631  82.43665  46.68469
60100 "00:16:49"  17.99 3     565.3417 30.06442 3.68  1.99  0.37  8.47  198.865  64.82401  45.6142
60101 "01:16:49"  17.71 2.96  585.5978 30.93434 3.67  1.99  0.37  8.5   202.176  67.85629  45.68301
60102 "02:16:49"  17.95 1.47  520.1424 28.28283 3.67  1.97  0.86  8.52  187.111  61.37904  47.28209
60103 "03:16:49"  19.07 4.9   559.0491 30.06442 3.65  1.95  0.86  8.53  204.299  61.76809  43.67516
60104 "04:16:49"  17.21 2.49  577.8169 30.93434 3.65  1.98  0.86  8.56  196.912  68.27013  45.68425
```

The following is another example of column headings used for a test data set. Note that only the columns for Process 1 have descriptors actually entered.

```
PROCESS 1 "EFR"
PROCESS 2 "WFR"
PROCESS 3 "ECR"
PROCESS 4 "WCR"
COLUMN 1 "TIME" IDENTIFICATION END
COLUMN 2 "BATCH#" IGNORE END
COLUMN 3 "TotRecWt" IGNORE END
COLUMN 4 "AmRecWt" IGNORE END
COLUMN 5 "AmRecBPL" IGNORE END
COLUMN 6 "AmCnBPL" IGNORE END
COLUMN 7 "AmCn%Insol" IGNORE END
COLUMN 8 "AmCnTPM" IGNORE END
COLUMN 9 "AmTl%BPL" IGNORE END
COLUMN 10 "AmFd%BPL" PROCESS 1 CONCENTRATE ASSAY UNITS PERCENT MINERAL "Apatite" DIGITS 3 END
COLUMN 11 "AmFdTPM" PROCESS 1 CONCENTRATE MASS_RATE UNITS TPM DIGITS 3 END
COLUMN 12 "Am:amine" IGNORE END
COLUMN 13 "Am:diesel" IGNORE END
COLUMN 14 "Am:pH" IGNORE END
COLUMN 15 "RRecWt" IGNORE END
COLUMN 16 "RRecBPL" PROCESS 1 RECOVERY TOP UNITS PERCENT MINERAL "Apatite" END
COLUMN 17 "FRtl%BPL" PROCESS 1 TAILS ASSAY UNITS PERCENT MINERAL "Apatite" END
COLUMN 18 "Rfd%BPL" PROCESS 1 FEED ASSAY UNITS PERCENT MINERAL "Apatite" END
COLUMN 19 "EF:fAcid" CLUSTER_ONLY END
COLUMN 20 "EF:fOil" CLUSTER_ONLY END
COLUMN 21 "EF:cond" CLUSTER_ONLY END
COLUMN 22 "EFfdTPM" PROCESS 1 FEED MASS_RATE UNITS TPM DIGITS 3 END
COLUMN 23 "WF:fAcid" CLUSTER_ONLY END
COLUMN 24 "WF:fOil" CLUSTER_ONLY END
COLUMN 25 "WF:cond" CLUSTER_ONLY END
COLUMN 26 "WFfdTPM" PROCESS 2 FEED MASS_RATE UNITS TPM DIGITS 3 END
COLUMN 27 "EC:fAcid" CLUSTER_ONLY END
COLUMN 28 "EC:fOil" CLUSTER_ONLY END
COLUMN 29 "ECfdTPM" PROCESS 3 FEED MASS_RATE UNITS TPM DIGITS 3 END
COLUMN 30 "WC:fAcid" CLUSTER_ONLY END
COLUMN 31 "WC:fOil" CLUSTER_ONLY END
COLUMN 32 "WCfdTPM" PROCESS 4 FEED MASS_RATE UNITS TPM DIGITS 3 END
COLUMN 33 "Profit#" IGNORE END
COLUMN 34 "AmSetPt" IGNORE END
```

9.2 Example of CLUSTERED DATA FILE

The following is the text of the Cluster Analyzer output file SFRCL1.CLU derived from standard data file SFR.TXT. The SUMMARY line gives information about the classification: traceW is average distance-to-closest-cluster-center; outliers are unclustered points; points in the wrong cluster are ones that were assigned to a cluster at some early step, but as the clusters grew and the scaling functions changed the assigned cluster's center grew farther away and some other cluster's center grew closer, till finally the point is closer to the other cluster's center than it is to its own cluster's center; points in many clusters are within the threshold distance of more than one cluster center. CLUSTER_COUNT is the number of clusters identified in this cluster analysis. PROCESS and COLUMN lines are copied from the input file's standard-data-file prefix. THRESHOLD, RAWDATA, and TRANSFORM lines tell parameters for this analysis. THRESHOLD shows initial threshold and stepfactor, and comparison with the final threshold gives number of steps actually used. RAWDATA Rows is number of observations; Fields is number of columns in the input file; Select is number of columns used in cluster analysis; Mask shows usage decisions for input columns: 0=Skip, 1=Copy, 8=NoCenter/NoScale, 9=NoCenter/UnitVariance, 10=NoCenter/ScaleByError, 12=Center/NoScale, 13=Center/UnitVariance, 14=Center/ScaleByError. TRANSFORM nVarC is number of variables actually used in clustering; METHOD is clustering coordinate system, PCA (principal-components) or RAW (original) or RED (error-reduced original); MINUS and TIMES show actual centering and scaling values; EIGENSYSTEM list is (eigenvalue)(eigenvector coordinates). Look at a row of eigenvector coordinates to see which data columns contribute most to that eigenvector; look at a column of eigenvector coordinates to see which eigenvectors are affected most by the corresponding column of measurements. Next come cluster distance-scaling matrices: cluster number is first, and NP is number of points in the cluster. Finally comes CLUSTERED_POINTS: for each cluster, cluster number again followed by the *raw* data for the points assigned to the cluster (prefixed by number of columns, length of label, and label or IDENTIFICATION field).

```
CLUSTER_COLLECTION "C:\OOL\project5\temp\SFRCL1.clu"  Mon Dec 02 19:57:04 1996
SUMMARY traceW= 0.890852 #Outliers= 0 #InWrongCluster= 0 #InManyClusters= 0
CLUSTER_COUNT 1
THRESHOLD Final= 130 Init= 130 Factor= 100
RAWDATA Rows= 20 Fields= 13 Select= 11
  Mask= 1 0 13 13 13 13 12 12 12 12 13 13 13
PROCESS 1 "East Fine Rougher On-line" SEPARATOR
COLUMN 1 "DATE_HR" IDENTIFICATION END
COLUMN 2 "TIME" IGNORE END
COLUMN 3 "FeedBPL" PROCESS 1 FEED ASSAY UNITS PERCENT MINERAL "Apatite" END
COLUMN 4 "TailsBPL" PROCESS 1 TAILS ASSAY UNITS PERCENT MINERAL "Apatite" END
COLUMN 5 "FeedTPH" PROCESS 1 FEED MASS_RATE UNITS TPH END
COLUMN 6 "FeedPctSolids" PROCESS 1 FEED SOLIDS UNITS PERCENT END
COLUMN 7 "FattyAcidGPM" CLUSTER_ONLY END
COLUMN 8 "AmmoniaGPM" CLUSTER_ONLY END
COLUMN 9 "DieselGPM" CLUSTER_ONLY END
COLUMN 10 "RougherPH" CLUSTER_ONLY END
COLUMN 11 "ConcTPH" PROCESS 1 CONCENTRATE MASS_RATE UNITS TPH END
COLUMN 12 "ConcPCTSolids" PROCESS 1 CONCENTRATE SOLIDS UNITS PERCENT END
COLUMN 13 "ConcBPL" PROCESS 1 CONCENTRATE ASSAY UNITS PERCENT MINERAL "Apatite" END
 END_DESCRIPTORS
TRANSFORM nVarC= 4 METHOD= PCA
 MINUS= 18.692 4.189 625.59848 32.6307525 3.5885 1.964 0.374 8.5245 218.25885 72.931755 45.294228
 TIMES= 0.2393210021 0.4689733132 0.01800387224 0.4379367502 1 1 1 1 0.02863752865 0.099240758042  0.147080994
 EIGENSYSTEM=
  (91.1576153) (-0.38527621 -0.2803844 -0.3841511 -0.3819640 -0.01733022 -0.00884649  0.02174557 -0.00721773 -0.44709280 -0.41644793 -0.324428230)
  (25.6496864) (-0.39783904  0.2861336  0.4408927  0.4289401 -0.04565622 -0.01168389 -0.09733917 -0.01982691 -0.13439684 -0.02156362 -0.592333540)
  (16.3795015) (-0.29152159 -0.7705770  0.1772120  0.2363525 -0.00820222  0.01436277 -0.03456081 -0.01918362 -0.13632893  0.43766091  0.148293189)
  (1.08555872) (-0.01456946  0.0099959 -0.0456172  0.1539831  0.27395884  0.09963373  0.93751164 -0.04258446 -0.02811044  0.03506512 -0.078450293)
  (0.16572125) (-0.19052131  0.0158871  0.2435399  0.0807723  0.71600626  0.14498576 -0.18982195 -0.12610539 -0.06418896 -0.40244263  0.381984659)
  (0.08798463) (-0.01954517 -0.0528920  0.4691430 -0.5720199  0.12664833  0.25270673  0.06469638  0.56013874  0.11482740  0.12386849 -0.152145079)
  (0.05692502) (-0.15199992  0.1509003 -0.4875219  0.3728490  0.11951777  0.31980775 -0.12867198  0.64727424 -0.06260571  0.12529121  0.075708362)
  (0.03772896) (-0.38649569  0.1877395  0.2230841  0.0063244 -0.47700326 -0.21774638  0.22593604  0.23540487 -0.14408680 -0.22820774  0.557960962)
  (0.02628166) (-0.26261794  0.4126577 -0.0777950 -0.3047783  0.04792674  0.27388147 -0.03883873 -0.35348338 -0.34717279  0.55711235  0.164722116)
  (0.00948233) ( 0.00203755 -0.1174299  0.0624418  0.0808728 -0.38142699  0.81689643  0.02339984 -0.23400467  0.18152981 -0.27132706  0.032910729)
  (0.00602398) (-0.57441587  0.0219065 -0.2185276 -0.1353615  0.06472670 -0.11291214  0.00721833 -0.11673589  0.75312793  0.06524407 -0.037588479)
1 CLUSTER NP= 20 MATRIX=
ROW 0 4.47213595499958 -1.12904206383407e-14 9.52844303025308e-15 4.56149603766248e-15 3.09923256763128e-15
ROW 1 9.54764978398902 1.28593614706282e-13 8.58807315084857e-14 4.61959233344876e-14
ROW 2 5.06455204476367 -5.64908905244244e-14 -3.86531220442424e-14
ROW 3 4.04715981682638 -3.90256098649951e-14
ROW 4 1.04190155497331
CLUSTERED_POINTS
1
  TUPLE 13 5 "53100"
   53100 0 19.78 8.03 669.136 34.2822 3.45 1.9 0 8.52 236.886 71.1938 41.2204
  TUPLE 13 5 "53101"
   53101 0 19.95 2.13 650.056 33.4644 3.46 1.9 0 8.52 233.207 83.5347 51.8026
  TUPLE 13 5 "53102"
   53102 0 15.34 1.82 546.258 29.1807 3.44 1.9 0 8.55 176.769 61.8505 43.6
  TUPLE 13 5 "53104"
   53104 0 12.37 4.59 649.19 33.4644 3.35 1.9 0 8.53 182.293 65.0985 32.2965
  TUPLE 13 5 "53105"
   53105 0 9.34 3.14 568.909 30.0644 3.36 1.92 0.4 8.52 143.999 53.8568 27.6349
  TUPLE 13 5 "53114"
   53114 0 12.65 3.67 661.149 34.2822 3.63 1.91 0.38 8.29 187.071 70.5448 35.4072
  TUPLE 13 5 "53115"
   53115 0 24.07 7.2 681.5 35.0877 3.55 1.95 0.38 8.51 266.623 82.616 50.3205
  TUPLE 13 5 "53116"
   53116 0 20.21 3.32 631.925 33.4644 3.55 1.98 0.38 8.49 230.315 79.7224 49.6618
  TUPLE 13 5 "53117"
   53117 0 21.69 2.26 659.66 34.2822 3.55 2 0.38 8.52 246.729 89.2109 54.2085
  TUPLE 13 5 "53118"
   53118 0 21.37 5.96 694.021 35.0877 3.64 1.98 0.38 8.55 253.202 82.0491 48.1985
  TUPLE 13 5 "53119"
   53119 0 28.54 9.56 681.026 34.2822 3.7 1.98 0.37 8.6 291.279 80.9263 53.9363
  TUPLE 13 5 "53120"
   53120 0 19.87 4.81 662.383 34.2822 3.7 1.98 0.38 8.57 235.782 79.045 47.1181
  TUPLE 13 5 "53121"
   53121 0 20.31 4.7 583.465 30.9343 3.69 1.96 0.37 8.62 218.6 66.5474 46.3646
  TUPLE 13 5 "53122"
   53122 0 19.73 3.58 680.648 35.0877 3.69 2.04 0.37 8.58 239.428 85.9045 49.4914
  TUPLE 13 5 "53123"
   53123 0 18.69 4.19 684.695 35.0877 3.69 2.1 0.37 8.54 233.631 82.4367 46.6847
  TUPLE 13 5 "60100"
   60100 0 17.99 3 565.342 30.0644 3.68 1.99 0.37 8.47 198.865 64.824 45.6142
  TUPLE 13 5 "60101"
   60101 0 17.71 2.96 585.598 30.9343 3.67 1.99 0.37 8.5 202.176 67.8563 45.683
  TUPLE 13 5 "60102"
   60102 0 17.95 1.47 520.142 28.2828 3.67 1.97 0.86 8.52 187.111 61.379 47.2821
  TUPLE 13 5 "60103"
   60103 0 19.07 4.9 559.049 30.0644 3.65 1.95 0.86 8.53 204.299 61.7681 43.6752
  TUPLE 13 5 "60104"
   60104 0 17.21 2.49 577.817 30.9343 3.65 1.98 0.86 8.56 196.912 68.2701 45.6842
```

9.3 Example of Data Viewer MOVIE FILE

The following is the text of the Cluster Analyzer output file SFRCL1.MVI derived from standard data file SFR.TXT. The INFO lines are comments which will appear in the Legend subwindow of the Data Viewer for this file, following information about the axes chosen for each scatterplot subwindow. Other information is the same as in the clustered data file above. NDIMENS shows number of columns used for clustering, number of copied columns, then number of principal-components coordinates; these are the values in the POINT lines that follow (preceded by point sequence number and followed by identification label). Next come lines that define the Data Viewer animations for this file: the first animation runs through all the points, highlighting each in turn, and the second shows the actual search for clusters. There are eight animation statements: RESET p means return point p to its initial (gray) state; CANDIDATE k p means point p is being considered for inclusion in cluster k, so highlight it; STEP separates the frames of the animation; ASSIGN k p means include point p in cluster k; CLUSTERID, THRESHOLD, and COUNT provide information for the Data Viewer status bar while an animation is in progress; PAUSE suspends the animation display.

```
FILE "C:\OOL\project5\temp\SFRCL1.clu"
INFO Input data from C:\OOL\project5\temp\SFR.TXT
INFO Cluster in principal-component coordinates
INFO MinClusterSize 2  InitThreshold 130  Factor 100  #Changes 10
INFO  NoScaling/Center COLUMN 7 "FattyAcidGPM" CLUSTER_ONLY END
INFO  NoScaling/Center COLUMN 8 "AmmoniaGPM" CLUSTER_ONLY END
INFO  NoScaling/Center COLUMN 9 "DieselGPM" CLUSTER_ONLY END
INFO  NoScaling/Center COLUMN 10 "RougherPH" CLUSTER_ONLY END
INFO  ForceUnitVar/Center COLUMN 3 "FeedBPL" PROCESS 1 FEED ASSAY UNITS PERCENT MINERAL "Apatite" END
INFO  ForceUnitVar/Center COLUMN 4 "TailsBPL" PROCESS 1 TAILS ASSAY UNITS PERCENT MINERAL "Apatite" END
INFO  ForceUnitVar/Center COLUMN 5 "FeedTPH" PROCESS 1 FEED MASS_RATE UNITS TPH END
INFO  ForceUnitVar/Center COLUMN 6 "FeedPctSolids" PROCESS 1 FEED SOLIDS UNITS PERCENT END
INFO  ForceUnitVar/Center COLUMN 11 "ConcTPH" PROCESS 1 CONCENTRATE MASS_RATE UNITS TPH END
INFO  ForceUnitVar/Center COLUMN 12 "ConcPCTSolids" PROCESS 1 CONCENTRATE SOLIDS UNITS PERCENT END
INFO  ForceUnitVar/Center COLUMN 13 "ConcBPL" PROCESS 1 CONCENTRATE ASSAY UNITS PERCENT MINERAL "Apatite" END
INFO Statistics for judging true data dimension: (local min in IE&IND, level Scree)
INFO  @n=11 (0.006024)
INFO  @n=10(0.0094823) IND=0.077614 IE=0.074002 Scree=4.4734e-05 ratio=2235.3
INFO  @n=9(0.026282) IND=0.022013 IE=0.079646 Scree=0.00011515 ratio=1929.6
INFO  @n=8(0.037729) IND=0.013114 IE=0.10065 Scree=0.00031032 ratio=1208.1
INFO  @n=7(0.056925) IND=0.0088121 IE=0.11247 Scree=0.00059049 ratio=967.15
INFO  @n=6(0.087985) IND=0.0066077 IE=0.122 Scree=0.0010132 ratio=821.63
INFO  @n=5(0.16572) IND=0.0053723 IE=0.13039 Scree=0.0016666 ratio=718.83
INFO  @n=4(1.0856) IND=0.004818 IE=0.14236 Scree=0.0028972 ratio=602.28
INFO  @n=3(16.38) IND=0.0067108 IE=0.22429 Scree=0.010959 ratio=240.67
INFO  @n=2(25.65) IND=0.017389 IE=0.60059 Scree=0.13259 ratio=29.439
INFO  @n=1(91.158) IND=0.020858 IE=0.62889 Scree=0.32307 ratio=20.953
INFO GUESS true #Principal Components = 11
INFO USE 4 principle components
NDIMENS 11 1 11
COLUMN 3 "FeedBPL" PROCESS 1 FEED ASSAY UNITS PERCENT MINERAL "Apatite" END
COLUMN 4 "TailsBPL" PROCESS 1 TAILS ASSAY UNITS PERCENT MINERAL "Apatite" END
COLUMN 5 "FeedTPH" PROCESS 1 FEED MASS_RATE UNITS TPH END
COLUMN 6 "FeedPctSolids" PROCESS 1 FEED SOLIDS UNITS PERCENT END
COLUMN 7 "FattyAcidGPM" CLUSTER_ONLY END
COLUMN 8 "AmmoniaGPM" CLUSTER_ONLY END
COLUMN 9 "DieselGPM" CLUSTER_ONLY END
COLUMN 10 "RougherPH" CLUSTER_ONLY END
COLUMN 11 "ConcTPH" PROCESS 1 CONCENTRATE MASS_RATE UNITS TPH END
COLUMN 12 "ConcPCTSolids" PROCESS 1 CONCENTRATE SOLIDS UNITS PERCENT END
COLUMN 13 "ConcBPL" PROCESS 1 CONCENTRATE ASSAY UNITS PERCENT MINERAL "Apatite" END
COLUMN 1 "DATE_HR" IDENTIFICATION END
CENTERS 18.692 4.189 625.598 32.6308 3.5885 1.964 0.374 8.5245 218.259 72.9318 45.2942
SCALEFACTORS 4.17849 2.13232 55.5436 2.28343 1 1 1 1 34.9192 10.0765 6.79897
EIGENPAIR 91.1576 -0.385276 -0.280384 -0.384151 -0.381964 -0.0173302 -0.00884649 0.0217456 -0.00721773 -0.447093 -0.416448 -0.324428
EIGENPAIR 25.6497 -0.397839 0.286134 0.440893 0.42894 -0.0456562 -0.0116839 -0.0973392 -0.0198269 -0.134397 -0.0215636 -0.592334
EIGENPAIR 16.3795 -0.291522 -0.770577 0.177212 0.236353 -0.00820223 0.0143628 -0.0345608 -0.0191836 -0.136329 0.437661 0.148293
EIGENPAIR 1.08556 -0.0145695 0.00999598 -0.0456172 0.153983 0.273959 0.0996337 0.937512 -0.0425845 -0.0281104 0.0350651 -0.0784503
EIGENPAIR 0.165721 -0.190521 0.0158871 0.24354 0.0807724 0.716006 0.144986 -0.189822 -0.126105 -0.064189 -0.402443 0.381985
EIGENPAIR 0.0879846 -0.0195452 -0.052892 0.469143 -0.57202 0.126648 0.252707 0.0646964 0.560139 0.114827 0.123868 -0.152145
EIGENPAIR 0.056925 -0.152 0.1509 -0.487522 0.372849 0.119518 0.319808 -0.128672 0.647274 -0.0626057 0.125291 0.0757084
EIGENPAIR 0.037729 -0.386496 0.18774 0.223084 0.00632442 -0.477003 -0.217746 0.225936 0.235405 -0.144087 -0.228208 0.557961
EIGENPAIR 0.0262817 -0.262618 0.412658 -0.077795 -0.304778 0.0479267 0.273881 -0.0388387 -0.353483 -0.347173 0.557112 0.164722
EIGENPAIR 0.00948234 0.00203755 -0.11743 0.0624418 0.0808729 -0.381427 0.816896 0.0233998 -0.234005 0.18153 -0.271327 0.0329107
EIGENPAIR 0.00602399 -0.574416 0.0219066 -0.218528 -0.135362 0.0647267 -0.112912 0.00721834 -0.116736 0.753128 0.0652441 -0.0375885
NPOINTS 20
POINT 1 19.78 8.03 669.136 34.2822 3.45 1.9 0 8.52 236.886 71.1938
        41.2204 53100 -1.16016 1.39815 -1.37796 -0.278972 -0.00226812 -0.0757195 0.0275841 0.0396019
        0.00551363 0.0131957 0.0298066 "53100"
POINT 2 19.95 2.13 650.056 33.4644 3.46 1.9 0 8.52 233.207 83.5347
        51.8026 53101 -1.09932 -0.649458 1.37791 -0.420159 -0.0509863 0.0176164 -0.0831268 0.0248679
        -0.0353488 -0.0162296 0.0120682 "53101"
POINT 3 15.34 1.82 546.258 29.1807 3.44 1.9 0 8.55 176.769 61.8505
        43.6 53102 2.81128 -0.902191 0.136193 -0.551298 0.0410918 -0.0110623 0.0314853 0.0626544
        0.0634847 -0.0184209 -0.0121565 "53102"
POINT 4 12.37 4.59 649.19 33.4644 3.35 1.9 0 8.53 182.293 65.0985
        32.2965 53104 1.62855 2.3351 -0.0117379 -0.210144 -0.0368592 0.0188475 0.0124707 0.0207677
        -0.0761628 0.0205682 -0.0351467 "53104"
POINT 5 9.34 3.14 568.909 30.0644 3.36 1.92 0.4 8.52 143.999 53.8568
        27.6349 53105 4.40834 1.69122 -0.338269 0.0559374 -0.188921 0.109432 -0.00382285 -0.0495426
        0.0401243 -0.00643968 0.0134063 "53105"
POINT 6 12.65 3.67 661.149 34.2822 3.63 1.91 0.38 8.29 187.071 70.5448
        35.4072 53114 1.07455 2.08735 0.69907 0.253286 0.133418 -0.122305 -0.108042 -0.0299895
        0.0175075 -0.0265537 -0.00974971 "53114"
POINT 7 24.07 7.2 681.5 35.0877 3.55 1.95 0.38 8.51 266.623 82.616
        50.3205 53115 -2.94771 0.154136 -0.689035 0.0462079 -0.112957 -0.0937029 -0.0015978 0.0119336
        0.0140294 0.00843602 0.00488131 "53115"
POINT 8 20.21 3.32 631.925 33.4644 3.55 1.98 0.38 8.49 230.315 79.7224
        49.6618 53116 -0.851422 -0.494088 0.658745 0.00332622 -0.0884818 -0.135332 0.052742 -0.0261967
        -0.00782171 0.0221718 -0.0123002 "53116"
POINT 9 21.69 2.26 659.66 34.2822 3.55 2 0.38 8.52 246.729 89.2109
        54.2085 53117 -1.99678 -0.883883 1.55861 -0.00637425 -0.167899 0.00416242 -0.0210173 -0.0495517
        0.012465 0.0025725 0.000310399 "53117"
POINT 10 21.37 5.96 694.021 35.0877 3.64 1.98 0.38 8.55 253.202 82.0491
        48.1985 53118 -2.32789 0.576687 -0.0321875 0.0987845 0.0476489 0.0932093 -0.0614057 0.0568135
        -0.0247132 0.00552499 0.0304187 "53118"
```

```
POINT 11 28.54 9.56 681.026 34.2822 3.7 1.98 0.37 8.6 291.279 80.9263
         53.9363 53119 -3.95429 -0.52415 -2.03161 -0.0488177 -0.00215609 0.0803641 -0.0624384 -0.0237733
         0.0310149 -0.0112407 -0.0391597 "53119"
POINT 12 19.87 4.81 662.383 34.2822 3.7 1.98 0.38 8.57 235.782 79.045
         47.1181 53120 -1.28725 0.327202 0.216821 0.101909 0.0720055 0.0120752 0.0599084 -0.00764945
         -0.0244397 -0.0384312 0.00939284 "53120"
POINT 13 20.31 4.7 583.465 30.9343 3.69 1.96 0.37 8.62 218.6 66.5474
         46.3646 53121 0.56471 -0.825592 -0.865436 -0.0982902 0.0605872 0.0126099 0.0752324 -0.0735372
         -0.0776347 -0.0203597 0.00510191 "53121"
POINT 14 19.73 3.58 680.648 35.0877 3.69 2.04 0.37 8.58 239.428 85.9045
         49.4914 53122 -1.81774 0.236828 1.1493 0.122939 0.0326498 0.0577795 0.0801038 -0.0112402
         0.0265385 -0.0257098 -0.00233815 "53122"
POINT 15 18.69 4.19 684.695 35.0877 3.69 2.1 0.37 8.54 233.631 82.4367
         46.6847 53123 -1.47882 0.724174 0.826749 0.158766 0.107575 0.0755686 0.0548991 0.00456494
         0.0327602 0.0527696 -0.00309627 "53123"
POINT 16 17.99 3 565.342 30.0644 3.68 1.99 0.37 8.47 198.865 64.824
         45.6142 60100 1.63372 -0.991994 -0.247875 -0.116764 0.122521 -0.0165164 -0.0267469 -0.0620118
         0.0241854 0.0245469 0.0102642 "60100"
POINT 17 17.71 2.96 585.598 30.9343 3.67 1.99 0.37 8.5 202.176 67.8563
         45.683 60101 1.20815 -0.671866 0.0605695 -0.0708724 0.120308 0.00111497 -0.00438493 -0.0207256
         0.0148743 0.0128855 0.00359628 "60101"
POINT 18 17.95 1.47 520.142 28.2828 3.67 1.97 0.86 8.52 187.111 61.379
         47.2821 60102 2.67315 -2.02764 -0.106249 0.223932 -0.00473655 0.021219 -0.0692118 0.0155079
         -0.044152 0.0218483 -0.00223635 "60102"
POINT 19 19.07 4.9 559.049 30.0644 3.65 1.95 0.86 8.53 204.299 61.7681
         43.6752 60103 1.48817 -0.782304 -1.24457 0.345532 -0.0648961 -0.0467512 0.0137356 0.0156152
         0.010226 -0.009295 0.0133139 "60103"
POINT 20 17.21 2.49 577.817 30.9343 3.65 1.98 0.86 8.56 196.912 68.2701
         45.6842 60104 1.43076 -0.777682 0.260964 0.391072 -0.0176435 -0.00260942 0.0336335 0.101891
         -0.00245085 -0.0118393 -0.016377 "60104"
RESET 1
RESET 2
RESET 3
RESET 4
RESET 5
RESET 6
RESET 7
RESET 8
RESET 9
RESET 10
RESET 11
RESET 12
RESET 13
RESET 14
RESET 15
RESET 16
RESET 17
RESET 18
RESET 19
RESET 20
STEP
CANDIDATE 0 1
STEP
RESET 1
CANDIDATE 0 2
STEP
RESET 2
CANDIDATE 0 3
STEP
RESET 3
CANDIDATE 0 4
STEP
RESET 4
CANDIDATE 0 5
STEP
RESET 5
CANDIDATE 0 6
STEP
RESET 6
CANDIDATE 0 7
STEP
RESET 7
CANDIDATE 0 8
STEP
RESET 8
CANDIDATE 0 9
STEP
RESET 9
CANDIDATE 0 10
STEP
RESET 10
CANDIDATE 0 11
STEP
RESET 11
CANDIDATE 0 12
STEP
RESET 12
CANDIDATE 0 13
STEP
RESET 13
CANDIDATE 0 14
STEP
RESET 14
CANDIDATE 0 15
STEP
RESET 15
CANDIDATE 0 16
STEP
RESET 16
CANDIDATE 0 17
STEP
RESET 17
CANDIDATE 0 18
STEP
RESET 18
CANDIDATE 0 19
STEP
RESET 19
CANDIDATE 0 20
STEP
RESET 20
PAUSE
THRESHOLD 130
CLUSTERID 0
STEP
CANDIDATE 0 1
CANDIDATE 0 2
CANDIDATE 0 3
CANDIDATE 0 4
CANDIDATE 0 5
CANDIDATE 0 6
CANDIDATE 0 7
CANDIDATE 0 8
CANDIDATE 0 9
CANDIDATE 0 10
CANDIDATE 0 11
CANDIDATE 0 12
CANDIDATE 0 13
CANDIDATE 0 14
CANDIDATE 0 15
CANDIDATE 0 16
CANDIDATE 0 17
CANDIDATE 0 18
CANDIDATE 0 19
CANDIDATE 0 20
STEP
ASSIGN 0 1
ASSIGN 0 2
ASSIGN 0 3
ASSIGN 0 4
ASSIGN 0 5
ASSIGN 0 6
ASSIGN 0 7
ASSIGN 0 8
ASSIGN 0 9
ASSIGN 0 10
ASSIGN 0 11
ASSIGN 0 12
ASSIGN 0 13
ASSIGN 0 14
ASSIGN 0 15
ASSIGN 0 16
ASSIGN 0 17
ASSIGN 0 18
ASSIGN 0 19
ASSIGN 0 20
COUNT 20
CLUSTERID 1
STEP
THRESHOLD 13000
STEP
THRESHOLD 130
INFO 20 points placed in 1 clusters
```

9.4 Example of SAVED DESKTOP or CIRCUIT FILE

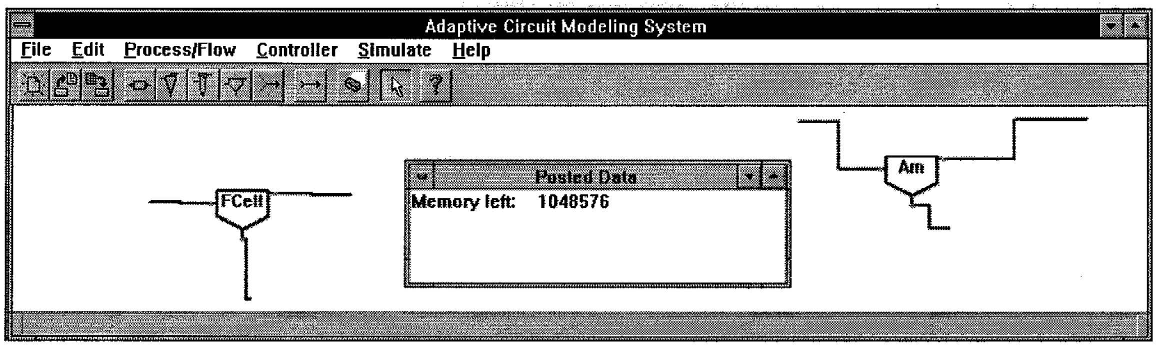
The following is an example of a saved desktop file. The information recorded for flows and processes corresponds to fields in the corresponding edit screens, and for the most part the information labels in this file are the same as the ones used elsewhere. But NOTICE that a tuned Empirical Model still carries a reference to its clustered-data file: eigensystem and cluster-distance information in that file (which can be large) is not copied into the saved desktop file but is needed by the desktop whenever the Empirical Model is used. The final section of the file contains the record of icons on the desktop, which have not been previously described. Most of the numbers in a flow or process GraphicalRepresentation refer to screen coordinates or screen distances, though the unlabeled number on a GRep's second line identifies the icon type. EPSILON indicates how close the cursor must be to a screen object for a mouse click to select the object. Ghost Points are used to display a flow line while you are moving one of its segments. Links between GReps are made using GRep numbers, as in SOURCE_PROCESS or INPUT_FLOWS lines; links to flow and process objects are by name, using MODEL lines. A picture of this desktop follows the listing of the file.

```
FLOW "feed"
SPECIES_COUNT 3
SPECIES "water"
DENSITY 1
VOLUMETRIC_FLOW_RATE 50802.3
LIQUID
VISCOSITY 0.01
END_SPECIES

SPECIES "gangue"
DENSITY 2.6
VOLUMETRIC_FLOW_RATE 1953.94
SOLID
PARTICLE_DIAMETER 0.005
NON_HYDROPHOBIC
END_SPECIES

SPECIES "apatite"
DENSITY 3.15
VOLUMETRIC_FLOW_RATE 179.197
SOLID
PARTICLE_DIAMETER 0.005
HYDROPHOBIC
INDUCTION_TIME 0.04
END_SPECIES

END_FLOW

FLOW "FCell tails"
SPECIES_COUNT 3
SPECIES "water"
DENSITY 1
VOLUMETRIC_FLOW_RATE 31515
LIQUID
VISCOSITY 0.01
END_SPECIES

SPECIES "gangue"
DENSITY 2.6
VOLUMETRIC_FLOW_RATE 1180.67
SOLID
PARTICLE_DIAMETER 0.005
NON_HYDROPHOBIC
END_SPECIES

SPECIES "apatite"
DENSITY 3.15
VOLUMETRIC_FLOW_RATE 23.2415
SOLID
PARTICLE_DIAMETER 0.005
HYDROPHOBIC
INDUCTION_TIME 0.04
END_SPECIES

END_FLOW

FLOW "FCell concentrate"
SPECIES_COUNT 3
SPECIES "water"
DENSITY 1
VOLUMETRIC_FLOW_RATE 19287.4
LIQUID
VISCOSITY 0.01
END_SPECIES

SPECIES "gangue"
DENSITY 2.6
VOLUMETRIC_FLOW_RATE 773.325
SOLID
PARTICLE_DIAMETER 0.005
NON_HYDROPHOBIC
END_SPECIES

SPECIES "apatite"
DENSITY 3.15
VOLUMETRIC_FLOW_RATE 155.956
SOLID
PARTICLE_DIAMETER 0.005
HYDROPHOBIC
INDUCTION_TIME 0.04
END_SPECIES

END_FLOW

FLOW "Am feed"
SPECIES_COUNT 3
SPECIES "water"
DENSITY 1
VOLUMETRIC_FLOW_RATE 565844
LIQUID
VISCOSITY 0.01
END_SPECIES

SPECIES "gangue"
DENSITY 2.6
VOLUMETRIC_FLOW_RATE 12013.9
SOLID
PARTICLE_DIAMETER 0.005
NON_HYDROPHOBIC
```

```
END_SPECIES

SPECIES "apatite"
DENSITY 3.15
VOLUMETRIC_FLOW_RATE 7098.56
SOLID
PARTICLE_DIAMETER 0.005
HYDROPHOBIC
INDUCTION_TIME 0.04
END_SPECIES

END_FLOW

FLOW " tails"
SPECIES_COUNT 3
SPECIES "water"
DENSITY 1
VOLUMETRIC_FLOW_RATE 0
LIQUID
VISCOSITY 0.01
END_SPECIES

SPECIES "gangue"
DENSITY 2.6
VOLUMETRIC_FLOW_RATE 0
SOLID
PARTICLE_DIAMETER 0.005
NON_HYDROPHOBIC
END_SPECIES

SPECIES "apatite"
DENSITY 3.15
VOLUMETRIC_FLOW_RATE 0
SOLID
PARTICLE_DIAMETER 0.005
HYDROPHOBIC
INDUCTION_TIME 0.04
END_SPECIES

END_FLOW

FLOW " concentrate"
SPECIES_COUNT 3
SPECIES "water"
DENSITY 1
VOLUMETRIC_FLOW_RATE 0
LIQUID
VISCOSITY 0.01
END_SPECIES

SPECIES "gangue"
DENSITY 2.6
VOLUMETRIC_FLOW_RATE 0
SOLID
PARTICLE_DIAMETER 0.005
NON_HYDROPHOBIC
END_SPECIES

SPECIES "apatite"
DENSITY 3.15
VOLUMETRIC_FLOW_RATE 0
SOLID
PARTICLE_DIAMETER 0.005
HYDROPHOBIC
INDUCTION_TIME 0.04
END_SPECIES

END_FLOW

FLOW_TARGET "feed"
ID ""
BY_ELEMENT
PCT_SOLIDS_TARGET 10
SOLIDS_MASS_RATE_TARGET 5644.7
BEGIN_MINERAL_TARGETS
MINERAL "apatite" 10
END_MINERAL_TARGETS
BEGIN_ELEMENT_TARGETS
ELEMENT "calcium" 3.97
ELEMENT "phosphorus" 1.84254
ELEMENT "oxygen" 3.80702
ELEMENT "fluorine" 0.376719
END_ELEMENT_TARGETS
END_FLOW_TARGET

PROCESS CONVENTIONAL_FLOTATION_CELL
"FCell"
NUMBER_OF_INPUT_FLOWS 1
NUMBER_OF_OUTPUT_FLOWS 2

INPUT_FLOW 0
"feed"
OUTPUT_FLOW 0
"FCell tails"
OUTPUT_FLOW 1
"FCell concentrate"
LAST_INPUTS "Null_pointer"
Connected 1
numSpecies 3
Flows_Destroyed 0
Flows_Copied 0
K 1e-05 1e-05 1e-05
Tau 427.303 427.303 427.303
T 0
dT 5
Simulating 0
State_Changed 0
NUMBER_OF_ACTIVE_READINGS 0
NUMBER_OF_ACTIVE_ADJUSTMENTS 0
Ce 1
Cc 1
Cmaxp 1
Cv 1
Ci 100
dmaxp 0.005
numVelocities 2
Uf[i] 10 20
pUf[i] 0.5 0.5
numBubbleSizes 2
db[i] 0.08 0.12
nbV[i] 186.51 55.2621
pdb[i] 0.5 0.5
volb[i] 0.000268083 0.000904779
npV[i] -2.01694e-88 551344 17207.4
nupV[i] -1.17023e-170 551344 10853.2
napV[i] -2.85858e-156 0 6354.2
particleVolume -2.99251e+18 6.54498e-08 6.54498e-08
sw 0
Qa 37500
epsa 0.1
V 2.513e+07
E 500000
Feed "feed"
Tails "FCell tails"
Concentrate "FCell concentrate"
numCycles 595
END_PROCESS

PROCESS RATE_CONSTANT_PROCESS
"EAm East Amine"
VOLUME 5e+06
NUMBER_OF_INPUT_FLOWS 1
NUMBER_OF_OUTPUT_FLOWS 2
INPUT_FLOW 1
"Am feed"
OUTPUT_FLOW 1
" tails"
OUTPUT_FLOW 2
" concentrate"
CONCENTRATIONS 0.967327 0.020538 0.0121352
RATE_CONSTANTS 0.002 0.0498349 0.00981752
END_PROCESS

PROCESS MODELLED_PROCESS
"Am"
CLUSTER_FILE_NAME "C:\OOL\data\EATR4z.clu"
REDUCED_DATA_DIMENSION 0
CIRCUIT_PROCESS_NAME "EAm East Amine"
NUMBER_OF_ACTIVE_TUNABLE_PARAMETERS 3
```

```
PARAMETER_INFORMATION
2 "Apatite rate constant" 1e-05 0.1
3 "Feed weight fraction solids" 0.05 0.45
1 "Gangue rate constant" 1e-05 0.1
BAG_OF_MODELS
NUMBER_OF_MODELS 2
DT 0
ALL_MODELS_ACTIVE
PARAMETER_VALUES_FOR_EACH_MODEL
QUALITY_AT_END
1  0.00981294 0.212576 0.0498349 -5.95146
1  0.00981752 0.086524 0.0498349 -1.42906
END_BAG_OF_MODELS
END_PROCESS

FLOW_GRep 2
6
NOT_HIGHLIGHTED
EPSILON 4
SOURCE 107 112
DESTINATION 155 113
NPOINTS 2
131 112
131 113
GHOST_POINTS
107 112
107 111
131 111
131 113
SOURCE_ID -1
DESTINATION_ID 1
MODEL "feed"
SOURCE_PROCESS -1
DESTINATION_PROCESS 1
END_FLOW
FLOW_GRep 3
6
NOT_HIGHLIGHTED
EPSILON 4
SOURCE 205 105
DESTINATION 264 106
NPOINTS 2
234 105
234 106
GHOST_POINTS
-1 -1
-1 -1
-1 -1
-1 -1
SOURCE_ID 1
DESTINATION_ID 0
MODEL "FCell concentrate"
SOURCE_PROCESS 1
DESTINATION_PROCESS 0
END_FLOW
FLOW_GRep 4
6
NOT_HIGHLIGHTED
EPSILON 4
SOURCE 180 141
DESTINATION 186 188
NPOINTS 2
183 141
183 188
GHOST_POINTS
-1 -1
-1 -1
-1 -1
-1 -1
SOURCE_ID 1
DESTINATION_ID -1
MODEL "FCell tails"
SOURCE_PROCESS 1
DESTINATION_PROCESS -1
END_FLOW
FLOW_GRep 11
6
NOT_HIGHLIGHTED
EPSILON 4
SOURCE 611 49
DESTINATION 675 85
NPOINTS 2
643 49
643 85
GHOST_POINTS
643 49
643 84
675 86
675 86
SOURCE_ID -1
DESTINATION_ID -1
MODEL "Am feed"
SOURCE_PROCESS -1
DESTINATION_PROCESS 10
END_FLOW
FLOW_GRep 12
6
NOT_HIGHLIGHTED
EPSILON 4
SOURCE 725 77
DESTINATION 836 47
NPOINTS 2
780 77
780 47
GHOST_POINTS
725 78
780 78
780 77
780 77
SOURCE_ID -1
DESTINATION_ID -1
MODEL " concentrate"
SOURCE_PROCESS 10
DESTINATION_PROCESS -1
END_FLOW
FLOW_GRep 13
6
NOT_HIGHLIGHTED
EPSILON 4
SOURCE 700 113
DESTINATION 729 132
NPOINTS 2
714 113
714 132
GHOST_POINTS
700 114
714 114
714 113
714 113
SOURCE_ID -1
DESTINATION_ID -1
MODEL " tails"
SOURCE_PROCESS 10
DESTINATION_PROCESS -1
END_FLOW
PROCESS_GRep 1
2
NOT_HIGHLIGHTED
EPSILON 4
SOURCE 160 103
WIDTH 4 HEIGHT 3
SCALE 10
MODEL "FCell"
INPUT_FLOWS 2 -1
OUTPUT_FLOWS 4 3
END_PROCESS
PROCESS_GRep 10
2
NOT_HIGHLIGHTED
EPSILON 4
SOURCE 680 76
WIDTH 4 HEIGHT 3
SCALE 10
MODEL "Am"
INPUT_FLOWS 11 -1
OUTPUT_FLOWS 13 12
```

```
END_PROCESS
```


Adaptive Circuit Modeling System
File Edit Process/Flow Controller Simulate Help
Posted Data
FCell
Am

Memory left: 1048576

Figure 9-1. Desktop graphical representation corresponding to example saved-desktop file.

9.5 Example of Model Validation COMPARISON FILE

The following shows the structure but only a few of the data lines of the comparison file produced as validation output when you ask for Simulation of a Pure Empirical model. The file is intended to be read into a spreadsheet program, and the headings at the beginning of the file are labeled in spreadsheet style. The Pure Empirical model that produced this output was based on the standard data file shown at the beginning of this Section. It was tuned for just a few generations, so its fit is very poor.

```
B,C: COLUMN 3 "FeedBPL" PROCESS 1 FEED ASSAY UNITS PERCENT MINERAL "Apatite" END
D,E: COLUMN 4 "TailsBPL" PROCESS 1 TAILS ASSAY UNITS PERCENT MINERAL "Apatite" END
F,G: COLUMN 5 "FeedTPH" PROCESS 1 FEED MASS_RATE UNITS TPH END
H,I: COLUMN 6 "FeedPctSolids" PROCESS 1 FEED SOLIDS UNITS PERCENT END
J,K: COLUMN 11 "ConcTPH" PROCESS 1 CONCENTRATE MASS_RATE UNITS TPH END
L,M: COLUMN 12 "ConcPCTSolids" PROCESS 1 CONCENTRATE SOLIDS UNITS PERCENT END
N,O: COLUMN 13 "ConcBPL" PROCESS 1 CONCENTRATE ASSAY UNITS PERCENT MINERAL "Apatite" END
P,Q: GR COLUMN 1 "feedMassR" PROCESS 1 FEED MASS_RATE UNITS TPH END
R,S: GR COLUMN 2 "feedGrade" PROCESS 1 FEED ASSAY UNITS PERCENT MINERAL "Apatite" END
T,U: GR COLUMN 3 "topMassR" PROCESS 1 CONCENTRATE MASS_RATE UNITS TPH END
V,W: GR COLUMN 4 "topGrade" PROCESS 1 CONCENTRATE ASSAY UNITS PERCENT MINERAL "Apatite" END
X,Y: GR COLUMN 5 "topRecovery" PROCESS 1 RECOVERY TOP UNITS PERCENT MINERAL "Apatite" END
53115  24.07  24.07  7.2   10.686  681.5   681.5   35.088 -1  266.623  145.168  82.616  -1  50.3205  73.5182  681.5   681.5   24.07  24.07  266.623  145.168  50.3205  73.518  81.79   65.0614
53117  21.69  21.69  2.26  10.643  659.66  659.66  34.282 -1  246.729   86.552  89.211  -1  54.2085  94.8409  659.66  659.66  21.69  21.69  246.729   86.552  54.2085  94.841  93.478  57.3713
53118  21.37  21.37  5.96   9.420  694.02  694.02  35.088 -1  253.202  135.34   82.049  -1  48.1985  70.7003  694.02  694.02  21.37  21.37  253.202  135.341  48.1985  70.700  82.286  64.5167
```

9.6 *Example of Data Viewer INITIALIZATION FILE*

The following shows the format of the initialization file used by the Data Viewer. Each section is preceded by a line count; that count may be preceded by any amount of whitespace (including empty lines). The first section defines a table of colors: each line consists of RGB values (red, then green, then blue, in the range 0 to 255) followed by a label. Entries in the table of color patterns defined in the second section consist of three labels from the first table. The first color will be used for the corners of the plotting dot, the second for the middle of its sides, and the third for its center in all dot sizes. The final section consists of lines of help text, which will be displayed on request in the same way the Viewer displays legend text.

```
28
 255 255 255  White
   0   0   0  Black
 192 192 192  LtGrey
 160 160 164  MedGrey
 128 128 128  DkGrey
   0 128 128  GreyGreen
 255   0   0  Red
   0   0 255  Blue
 255 255   0  Yellow
   0 255   0  Green
 255   0 255  Magenta
   0 255 255  Cyan
 128   0   0  Brick
   0 128   0  DkGreen
   0  64 128  DkBlue
 128 128   0  Olive
 128   0 128  DkPurple
 192 220 192  Cream
 192 220 192  Moss
 166 202 240  LtBlue
 128 128 255  MedBlueA
   0 128 255  MedBlueB
   0 128 192  Denim
 255 128  64  Orange
 255   0 128  Pink
 255 128 192  Rose
 255 128 255  Purple
 192  64 192  MedPurple

13
LtGrey   LtGrey   LtGrey
Red      Red      Red
Blue     Blue     Blue
Green    Green    Green
Magenta  Magenta  Magenta
Cyan     Cyan     Cyan
DkGreen  DkGreen  DkGreen
Olive    Moss     Olive
Moss     Olive    Olive
Blue     LtBlue   Blue
DkGrey   DkGrey   Black
 Brick   Brick    Brick
 Yellow  Yellow   Red

50
The viewer displays scatterplots of the data in coordinate systems derived from the principal-
components analysis: each axis may show a measured value,  a measured value scaled as
requested for the cluster analysis, a corrected measured value after error reduction (projection
onto the span of the first few principal components) based on any number of dimensions, the
corresponding computed error, or a principal-component loading.  Any number of scatterplots can
be displayed at the same time.  The Viewer uses scatterplot dot color (changing with time) to show
the aggregation of points into clusters.  Since a point is shown in the same color in every plot, the
effect is to highlight data relationships.  The Viewer can also step directly through all the points
(which will normally be presented in order by time, so this animation traces plant operation
through time).  The user can run an animation frame-by-frame or continuously (at varying
speeds), forwards or backwards.  You can also use the mouse to select one point or a clump of
points in one scatterplot; these points are then highlighted in all the views, and can be saved to a
separate data file for further analysis.  Since each plot is small when many are displayed at once,
you can zoom in and out and pan over each display in order to make it easy to see any area of
interest.
. . .
. . .
```