Qualitative/Quantitative Simulation of Process Temporal Behavior Using Clustered Fuzzy Digraphs

R. F. Li and X. Z. Wang
Dept. of Chemical Engineering, The University of Leeds, Leeds LS2 9JT, U.K.

A methodology is presented for qualitative modeling and simulation of the temporal behavior of individual variables and the composite process. The temporal behavior of a variable in a windowed time scale is captured categorically using principal-component analysis, while the process is modeled as a causal digraph with interacting and recycle nodes. Rigorous, rather than ad hoc, reasoning rules can be devised for the causal digraph using a learning mechanism. Quantitative information can also be incorporated into the method with the introduction of fuzzy c-means clustering without compromising the cognitive level of information embedded in the nodes and the digraph.

Introduction

Process operators are the crucial agents in the overall control system who are responsible not only for many control tasks that are not automated, but more importantly for undertaking supervision of the general strategy and developing an understanding of the process's temporal performance. The understanding is used to identify problems in the current operation, deteriorating performance, and better operational regions that can lead to improved product and operating efficiency. Operators carry out these tasks by assimilating the data collected by the computer-control systems and subsequently developing a mental model of the temporal behavior of individual variables and the composite process. This is a challenging task, because the number of variables that need to be simultaneously monitored is large and they evolve with time and interact with each other. The cognitive level of information presented to operators by modern computer-control systems is low. This deficiency is more obvious when the process is under abnormal operations. This means that techniques and tools need to be developed to help operators in carrying out the role more effectively. Although much work has been done in recent years in developing operational decision support systems using expert systems and neural networks, the focus has been on providing operators with solutions in case of process malfunctions, not specifically on improving the cognitive levels of information on the temporal behavior of processes.

The temporal behavior of a process is defined by the variables and their interacting relationships. Subject to the influence of disturbances, if the degree of freedom of the system is zero and the responses of all variables are bounded, the system will eventually reach a new steady state. Traditionally, the temporal behavior of a process is simulated by simultaneously solving multiple, and generally implicit, differential and algebraic equations. Although numerical simulation is rigorous, it is not able to give a causal picture of the system's temporal behavior, and therefore is not ideal for high-level cognitive decision making, which often requires qualitative and logic reasoning. In addition, rigorous simulation can be hampered when models of systems are incompletely specified, either due to lack of knowledge about the fundamental principles governing the behavior, and so limiting the scope of prediction, or because even if a representative model is available, the parameters cannot be precisely determined.

Efforts were made by Bakshi and Stephanopoulos (1996, 1994a,b) to devise methods to qualitatively describe the temporal behavior of processes. Their procedure was divided into two steps: qualitative representation of the temporal behavior of individual variables, followed by qualitative characterization of the composite process. The temporal behavior of a variable's trajectory was described qualitatively using triangle episodes. The temporal process trends were then obtained using inductive learning that generates a decision tree. In another study, Wang and Li (1999) used principal-component analysis (PCA) to generate qualitative features from trend signals. Inductive learning was also used to extract rules and decision trees from previous operational data. One of the limitations of the preceding methods is that inductive learn-

Correspondence concerning this article should be addressed to X. Z. Wang.

ing tends to give an oversimplified decision tree because the dependent variables are eliminated during the induction. Most importantly, using a decision tree to describe the behavior of a process is not sufficient because in a tree graph, every node except the root node, has only one input link and no recycle links are allowed. This is clearly an oversimplified description of the dynamic behavior of a process, where variables are often dependent in complex ways due to the existence of control loops, recycle streams, and inherent links of variables.

In this article, a digraph methodology is presented for modeling and simulating the temporal behavior of individual variables and the composite process, both qualitatively and quantitatively. The basic idea is described in the following section. There are two critical issues in the proposed method. The first is how to categorically capture the feature of a dynamic transient. The approach employed in this work is described in the fourth section. The fifth section addresses the second issue, which is concerned with devising a rigorous rather than ad hoc reasoning mechanism in such a digraph, particularly when there are interacting and recycle nodes. In both the fourth and fifth sections, the method is presented as a purely qualitative technique. It will be further extended in the sixth section to include quantitative information. This is achieved through the introduction of the method of fuzzy c-means clustering for grouping node values, and a corresponding reasoning approach is subsequently devised that is described in the same section. To ease the introduction of the fourth and sixth sections, the mathematical background to be used in these three sections will be first described, in the third section.

Modeling Process Temporal Behavior Using Digraphs with Interacting and Recycle Nodes

In this study we will use digraphs with interacting and recycle nodes to describe the temporal behavior of a process. A digraph consists of a set of nodes and a set of directed links between the nodes. In this study, the node is used to describe the transient behavior of a variable over a windowed time scale. The directed link between two adjacent nodes represents the cause−effect relationship between them. A complex digraph is always the composition of three basic connections, that is, serial, divergent, and convergent, as shown in Figures 1a, 1b, and 1c.

The basic idea can be illustrated by reference to Figure 2. In the digraph there are sequential relationships, for example, $x_1 \rightarrow x_2 \rightarrow x_3$; interacting relationships, for example, $x_4 \leftrightarrow x_5$, and recycles, for example, $\overrightarrow{x_8 \leftarrow x_7 \leftrightarrow x_6}$. The interacting and recycle relationships can be the result of closed control loops, process recycles, and inherently related interacting causal relationships such as the temperatures of the hot and cold streams of a heat exchanger.

The nodes are shadowed, implying the temporal states of variables. A temporal state of a variable is a trend in a windowed time scale. Figure 3 illustrates the temporal trend of a variable in a windowed time scale. A fixed window width can be given. When the operation moves from the current sampling instant to the next instant, a new data point comes into the window (from the righthand side in Figure 3), and a data point at the left end of Figure 3 will move out of the window.

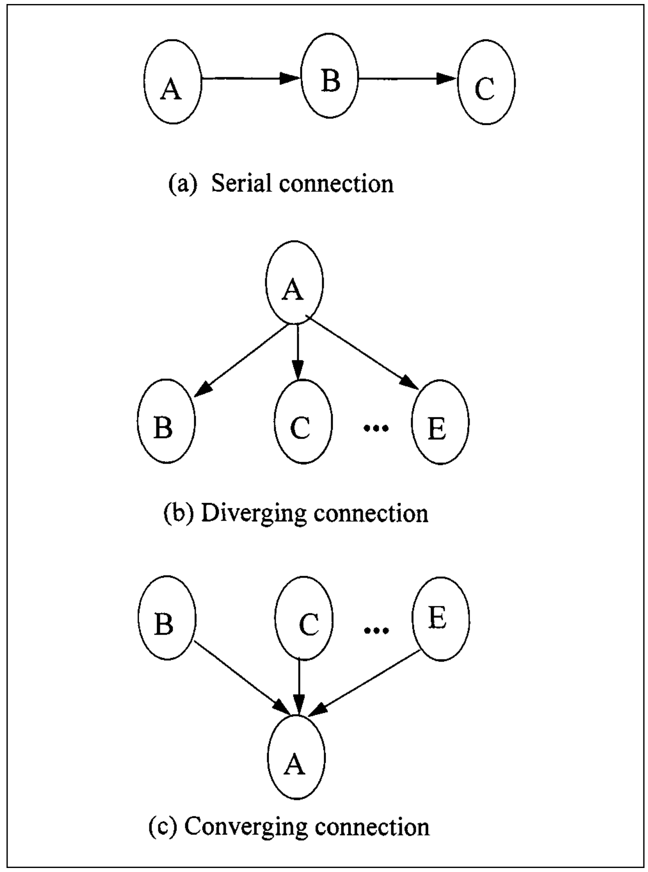

Figure 1. Three basic connections in a digraph.

Suppose at time $t < 0$, the process system is at steady state. At time $t \geq 0$, both nodes x_1 and x_9 are subject to changes due to external factors, the changes will pass the influence onto their adjacent nodes x_2 and x_4, and further propagate to all other nodes of the network in a dynamic and interacting way. The system as a whole will struggle to move to a new steady state. However, it may not always be able to head to a new steady state, indicating a situation that is out of control. Such a diagram containing interacting nodes is clearly a more accurate description of a system's behavior than a decision tree where no recycles are allowed and each node only has one input link.

It is appropriate here to clarify the difference of the current digraph with previous work. Since signed digraph (SDG) was first used by Iri et al. (1979) as a tool for process fault diagnosis, there have been many publications on this topic, which have been reviewed by Huang and Wang (1999) and Wang (1999). The studies have so far focused on fault diagnosis, however, rather than on simulation, which is the task of the current study. In addition, the majority of the studies have considered only steady-state situations. Several efforts in applying SDG to dynamic situations have achieved only very limited success, mainly because of the limitation of the methods used in qualitatively describing the temporal trends of variables. In this study, a new method for representing a variable's temporal behavior, which is able to both qualitatively and quantitatively capture the temporal feature of a variable's transient behavior, will be presented. These differ-

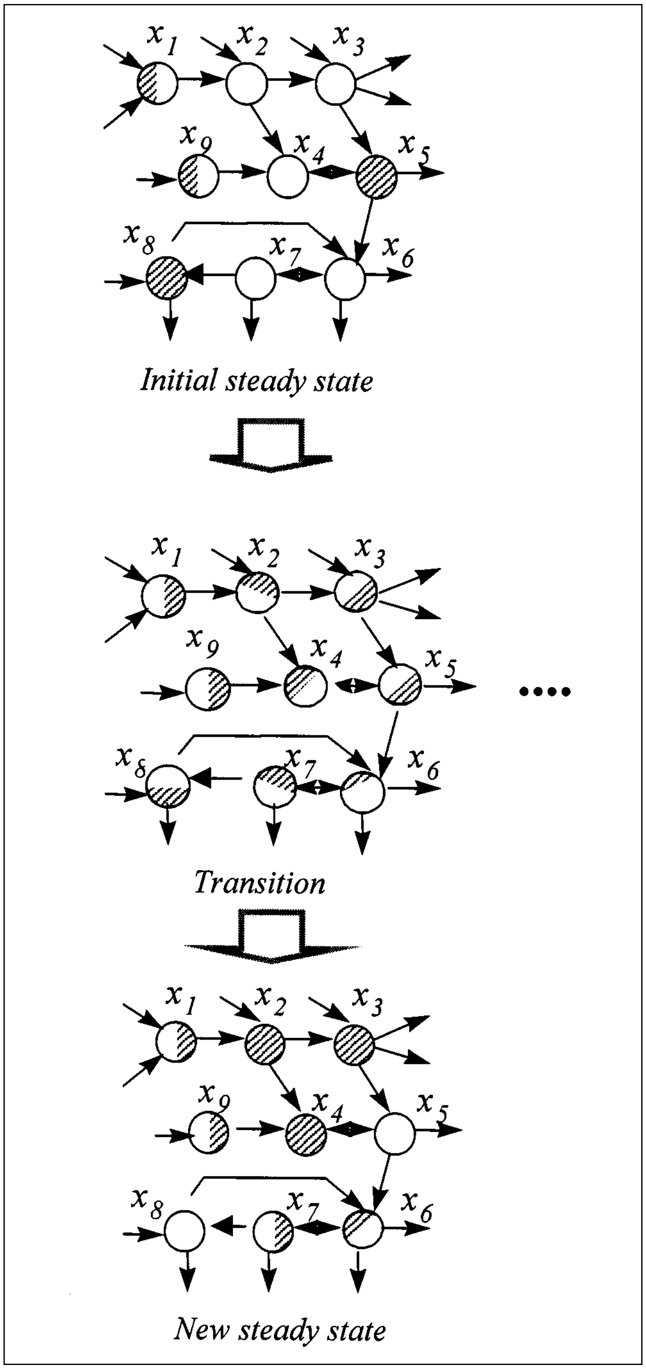

Figure 2. Conceptual illustration of digraph modeling of process temporal behavior.

ences will become clearer when the methods are described in detail in the fourth and sixth sections. It is also important to notice that Figure 2 represents a digraph (DG) but not an SDG, because there is no need to attach a sign to a connection.

It is also necessary to clarify the difference between the current work and that of Vedam and Venkatasubramanian (1999), who proposed an interesting approach of SDG and PCA integration for process monitoring and fault diagnosis. In their system, as in previous works, PCA was used to develop multivariate monitoring systems. SDG was introduced as a tool to automate the interpretation of PCA-based variable contributions. The method proposed in this work is very different because in this study PCA is used as an approach for categorically characterizing the temporal behavior of individual variables, and the new digraph is used to model process temporal behavior.

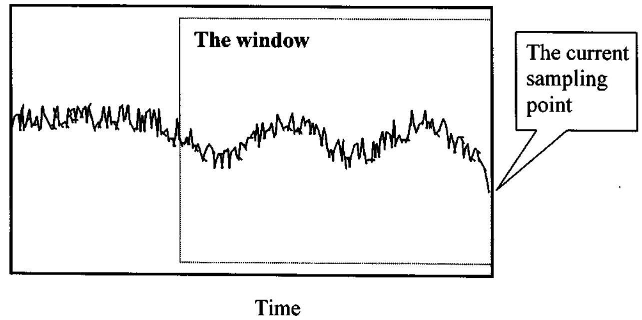

Figure 3. Temporal behavior (trend) of a variable in a windowed time scale.

The procedure of developing a digraph to describe process temporal behavior is as follows.

Step 1. Draw the digraph based on the knowledge of the process.

Step 2. Categorically describe the temporal behavior of all individual variables. There are two options, either using purely qualitative (fourth section) or a combination of qualitative/quantitative descriptions (sixth section).

Step 3. Automatic generation of rules describing the reasoning mechanism in the digraph. There are two methods, depending on whether the categorical description of the variables' temporal behavior obtained in step 2 is purely qualitative or a combination of qualitative and quantitative descriptions.

Before steps 2 and 3 are described in detail in the fourth and sixth sections, it is helpful to first introduce the mathematical techniques that will be used. These include principal-component analysis (PCA) and fuzzy c-means clustering.

Mathematical Techniques Used in the Methodology

Principal-component analysis

PCA was originally developed in the 1900s (Hotelling, 1933; Pearson, 1901), and has now reemerged as an important technique in data analysis. The central idea is to reduce the dimensionality of a data set consisting of a large number of interrelated variables, while retaining as much as possible of the variation present in the data set. Multiple regression and discrimination analysis used variable selection procedures to reduce the dimension, but result in the loss of one or more important dimensions. The PCA approach uses all of the original variables to obtain a smaller set of new variables [principal components (PCs)] that can be used to approximate the original variables. The greater the degree of correlation between the original variables, the fewer the number of new variables required. PCs are uncorrelated and are ordered so that the first few retain most of the variation present in the original set.

Given a data matrix X representing n observations of each of p variables, $x_1, x_2, \ldots, x_p$, the purpose of PCA is to determine a new variable y_1, that can be used to account for the variation in the p variables, $x_1, x_2, \ldots, x_p$. The first principal component is given by a linear combination of the p variables as

$$y_1 = w_{11}x_1 + w_{12}x_2 + \cdots + w_{1p}x_p, \tag{1}$$

where the sample variance is greatest for all of the coefficients (also called weights), $w_{11}, w_{12}, \ldots, w_{1p}$, conveniently written as a vector w_1. The $w_{11}, w_{12}, \ldots, w_{1p}$ have to satisfy the constraint that the sum-of-squares of the coefficients, that is, $\mathbf{w}_1'\mathbf{w}_1$, should be unity.

The jth principal component is a linear combination

$$y_j = w_j' x \tag{2}$$

that has greatest variance subject to $w_j'w_j = 1$ and $w_j'w_i = 0$, for $i < j$.

To find the coefficients defining the first principal component, the elements of $\mathbf{w}_1$ should be chosen so as to maximize the variance of y_1 subject to the constraint, $\mathbf{w}_1'\mathbf{w}_1 = 1$. The variance of y_1 is then given by

$$\mathrm{Var}(y_1) = \mathrm{Var}(\mathbf{w}_1' x) = \mathbf{w}_1' S \mathbf{w}_1, \tag{3}$$

where S is the variance−covariance matrix of the original variables. The solution of $\mathbf{w}_1 = (w_{11}, w_{12}, \ldots, w_{1p})$ to maximize the variance y_1 is the eigenvector of S corresponding to the largest eigenvalue. The eigenvalues of S are roots of the equation,

$$\| S - \lambda I | = 0. \tag{4}$$

If the eigenvalues are $\lambda_1, \lambda_2, \ldots, \lambda_p$, then they can be arranged from the largest to the smallest. The first few eigenvectors are the principal components that can capture most of the variance of the original data, while the remaining PCs mainly represent noise in the data.

In the past few years there has been a great interest in using PCA to develop on-line statistical process control charts for process monitoring (such as MacGregor and Kourti, 1995; Neogi and Schlags, 1998; Dunia et al., 1996; Chen et al., 1996; Negiz and Cinar, 1997; Dong and McAvoy, 1996). In all the applications, the data concern different variables, and the PCA was used to process a two-way data matrix of $X(I \times J)$, where I is the number of variables and J the sampling points. The method also has been extended to batch processes, that is, the multiway PCA, where the data are in three ways, $X(I \times J \times K)$, where I is the number of batch runs, J the number of variables, and K the sampling intervals (Nomikos and MacGregor, 1994). In all these applications, the purpose has been to eliminate the dependencies among the variables. As will be illustrated in the fourth section, the purpose of using PCA in this study is different from the previous works. Our purpose is not to eliminate the dependencies between variables. Instead, PCA is used to categorically characterize the evolving trajectories of individual variables.

Fuzzy c-means approach

The fuzzy c-means clustering is an iterative pattern-recognition approach that is based on fuzzy set theory (Friedman and Kandel, 1999; Bezdek, 1981). A fuzzy set consists of objects and their respective grades of membership in the set. The grade of membership of an object in the fuzzy set ranges from 0 to 1.0 and is given by a subjectively defined fuzzy membership function. Essentially the grade of membership is a quantitative method of expressing the fuzziness of impression. It stems from a grouping of elements into classes that do not have sharply defined boundaries.

Given m data cases $(\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x}_m\}$, each data case x_j is described by n attributes and suppose the m data cases can be partitioned into C clusters. The fuzzy c-means approach is an algorithm that can automatically identify the center of each cluster and calculate the membership values of each data case to each cluster. In order to introduce the approach, it is necessary to define the fuzzy membership function and fuzzy performance index, which can be used to compare different center schemes.

For a cluster C_i that is centered at y_i, the Euclidean distance $\| x_j - y_i \|$ between x_j and y_i is

$$d_{ij} = \| x_j - y_i \|, \qquad 1 \le i \le c, \qquad 1 \le j \le m. \tag{5}$$

The fuzzy membership value of x_j belonging to C_i is defined by

$$\phi_{ij} = \left[\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}} \right)^{2/(\beta-1)} \right]^{-1}, \qquad 1 \le i \le c, \qquad 1 \le j \le m, \tag{6}$$

where β is a tuning parameter that controls the degree of fuzziness in the clustering process, provided that $d_{kj} \neq 0$ for all $1 \le i \le c$. If $d_{kj} = 0$, we define $\phi_{ij} = 1$.

Given the number of clusters, c, the fuzzy clustering process is defined as the process of finding cluster centers $\mathbf{y_1}, \mathbf{y_2}, \ldots, y_c$ that minimize the following fuzzy performance index:

$$\sum_{j=1}^{m} \sum_{i=1}^{c} \phi_{ij} \| z_i - x_j \|^2, \qquad 1 \le i \le c. \tag{7}$$

Clearly it is generally a complex nonlinear problem. The fuzzy c-means algorithm is an interactive procedure that updates z_i using the last iteration's membership values (Friedman and Kandel, 1999; Bezdek, 1981). Updating z_i is done by minimizing c local performance indices, namely,

$$\sum_{j=1}^{m} \phi_{ij} \| z_i - x_j \|^2, \qquad 1 \le i \le c. \tag{8}$$

The updating of the center of an arbitrary fuzzy cluster, C_i, is given by

$$y_i^{(k+1)} = \frac{\sum_{j=1}^{m} \phi_{ij}^{(k)} x_j}{\sum_{j=1}^{m} \phi_{ij}^{(k)}}, \tag{9}$$

Table 1. Example of Fuzzy Clustering

	x_1	x_2	x_3	x_4	x_5
Cluster 1	0.961	0.981	0.946	0.038	0.040
Cluster 2	0.039	0.019	0.054	0.962	0.960

where $\phi_{ij}^{(k)}$ denotes the membership value of x_j in C_i after the kth iteration.

As an example, we consider partitioning five data cases into two clusters, where each data case is of two dimensions, $\mathbf{x_1}=(0,0)^T$, $\mathbf{x_2}=(0,1)^T$, $\mathbf{x_3}=(1,1)^T$, $\mathbf{x_4}=(3,3)^T$, and $\mathbf{x_5}=(4,2)^T$. Starting with an approximation of the two cluster centers, $(0,0)^T$ and $(3,2)^T$, after five steps, the center was found to be $(0.411, 0.715)^T$ and $(3.333, 2.399)^T$, and the membership values of x_i are shown in Table 1.

Categorical Characterization of the Temporal Behavior of Individual Variables

Graphically represented dynamic trend of variables are a convenient representation to operators in evaluating the current operational status of a process and anticipating possible future developments. A survey has revealed that operators spend 71% of their time on monitoring dynamic trends (Yamanaka and Nishiya, 1997). However, in order to make efficient use of trends in a computer-based reasoning system, a proper interpretation method is required. In this section, we first introduce the method used in this study, and then compare it with previously published approaches.

Method

The method mimics the way operators conduct reasoning. Therefore, it is useful to examine how an operator makes use of trend signals. We use Figure 4 to help the discussion. Figure 4 shows a serial connection, $F_i \rightarrow L$, where there is only one incoming connection for L. Figure 4 also shows 75 different temporal trends of F_i in a window of the size of 80 sampling points, as well as the corresponding responses of L. An operator can immediately recognize that the 75 trends of F_i can be clustered to three different types of shapes, denoted as a, b, and c in the figure. Similarly, the trend signals of L can also be grouped into three types of shapes, denoted as d, e, and f. The operator will also find that when F_i has the shape of a, then L will respond, following the shape of d. Similarly, if F_i has the shape of b or c, then L will respond following the trajectories of e or f. Essentially, this is a pattern-recognition process, and each pattern is represented by a single and categorical value.

The approach used in this work mimics this process and was initially developed in our early study (Wang and Li, 1999). Even though readers can refer to Wang and Li (1999) for detailed information on the method, a brief introduction of it is still given here, because it will be extended by introducing fuzzy c-means clustering in the sixth section.

Consider F_i in Figure 4, where the 75 trends in the window equal to the size of 80 sampling points form a data matrix of 75×80. The method involves processing the data matrix using PCA. Plotting the first two principal components gives three clusters, that is, A, B, and C (Figure 4). Similarly, for the variable L, the dynamic trends are also grouped into three

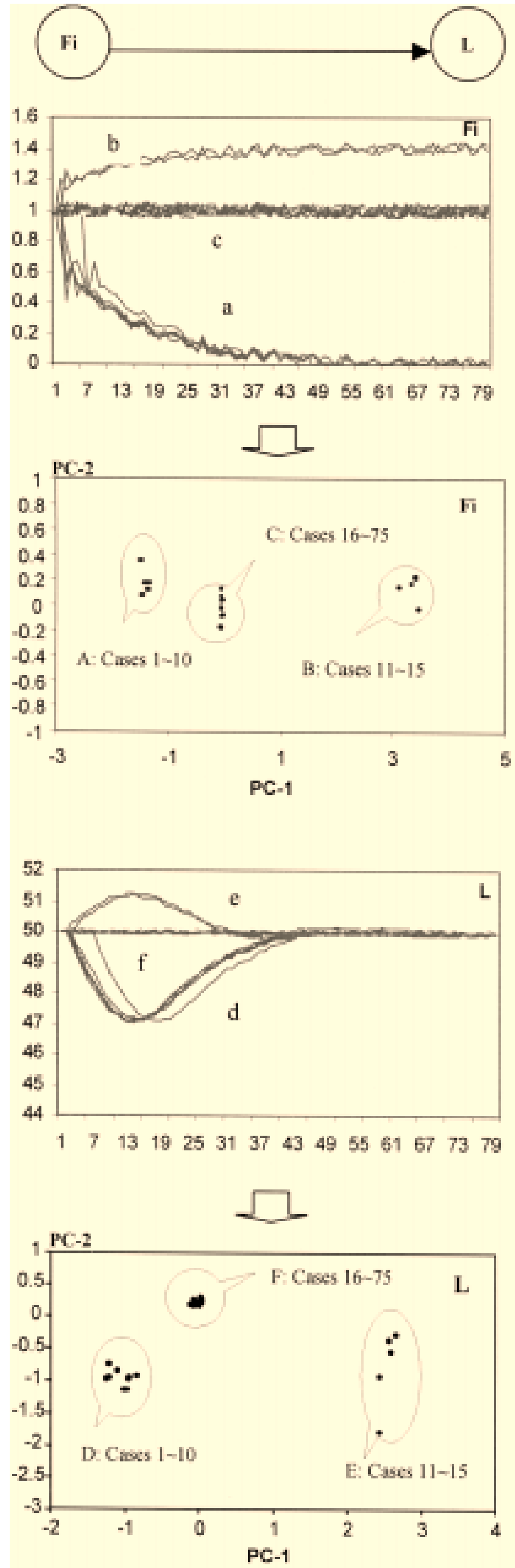

Figure 4. Categorical characterization of dynamic trends using PCA.

clusters, D, E, and F, in the PC1−PC2 plane. Rules can be easily generated for this simple case,

$$\text{IF } F_i = A \quad \text{THEN} \quad L = D$$
$$\text{IF } F_i = B \quad \text{THEN} \quad L = E$$
$$\text{IF } F_i = C \quad \text{THEN} \quad L = F$$

In the case study in this article, it was found that two principal components can effectively capture the features of dynamic trends. Table 2 summarizes the variance that the first two PCs captured for all the variables represented in Figure 5, which relates to a CSTR reactor described in the Appendix. From Table 2 it can be seen that PC1 and PC2 can capture most of the variance for all the variables. Figure 6

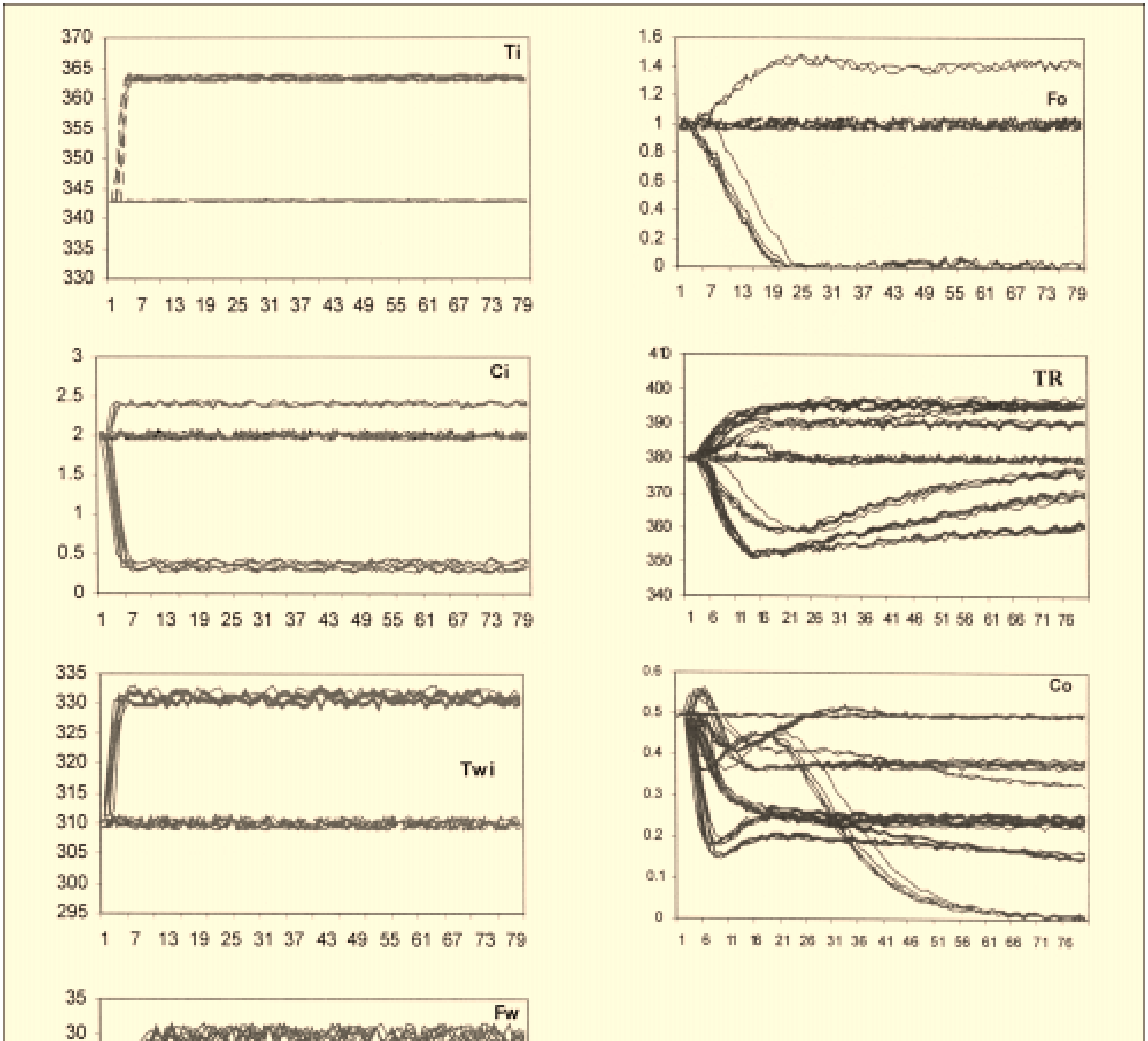

Figure 5. Dynamic trends for the variables of the CSTR reactor.

Table 2. Variance Captured by the First Two Principal Components

Variables	Variance Captured PC1	PC1 + PC2
F_i	95.6	97.0
T_i	97.3	98.7
C_i	96.8	98.6
T_{wi}	97.5	98.9
F_w	92.1	94.2
F_o	92.6	94.4
TR	92.2	95.0
Co	85.6	84.1
L	54.4	75.8

shows the variance captured by PC1 and PC2 for the variables TR, Co, and L. Figure 7 shows the groupings.

In cases where the first two PCs are not able to capture most of the variance (for example, 93%), more PCs need to be included in clustering the trends. For this purpose, fuzzy *c*-means clustering can be used, as will be described in the sixth section. Even in such cases, the two-dimensional PC1−PC2 plot can still be used as a display tool because of its visual effect.

The current use of PCA is clearly different from previous works. In previous works, PCA was almost inevitably used to process data involving a number of variables, and the purpose was to eliminate dependencies between variables. But in this study, it is used as a tool to categorically characterize dynamic trends of individual variables.

Comparison with previous methods

In order to make a comparison with earlier methods, it is necessary to briefly review the literature on qualitative interpretation of dynamic trends. The need to qualitatively describe dynamic trends was first recognized when expert systems were introduced for process monitoring and fault diagnosis. In the early work of expert systems such as G2 (Moore and Kramer, 1986), simple descriptors were employed to describe dynamic trends, such as temperature *increase* or *decrease*. Later various other approaches were proposed, including episodes (Cheung and Stephanopoulos, 1990,a,b; Bakshi and Stephanopoulos, 1994,a,b; Janusz and Venkatasubramanian, 1991), neural networks (Whiteley and Davis, 1992), and wavelets (Bakshi and Stephanopoulos, 1994a,b; Chen et al., 1999). It should be pointed out that the neural-network method of Whiteley and Davis (1992) and the wavelet method of Chen et al. (1999) can hardly be regarded as qualitative approaches. It is probably more appropriate to consider them as dimension-reduction techniques. Among all the approaches just listed, the episode methods (Bakshi and Stephanopoulos, 1994a,b; Janusz and Venkatasubramanian, 1991) have an objective that is most relevant to the current study, and are therefore briefly reviewed and compared with the current method below.

Janusz and Venkatasubramanian (1991) used nine primitives to represent any plots of a function. The method was further evolved to seven episodes, as shown in Figure 8. A dynamic trend of a variable in a windowed time scale is a series of episodes that, when grouped together, can capture the feature of the trend. Cheung and Stephanopolous

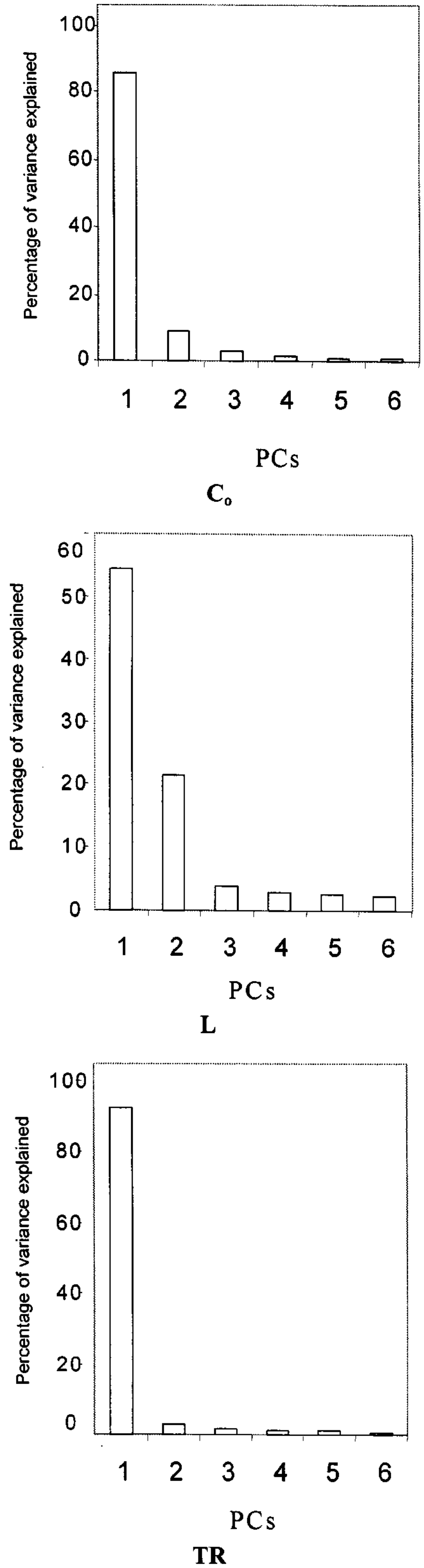

Figure 6. Variance captured by the principal components for three variables.

(1990a,b) developed a similar approach which uses seven primitive triangles to describe a trend. These original episode-based methods suffer from being weak in dealing with

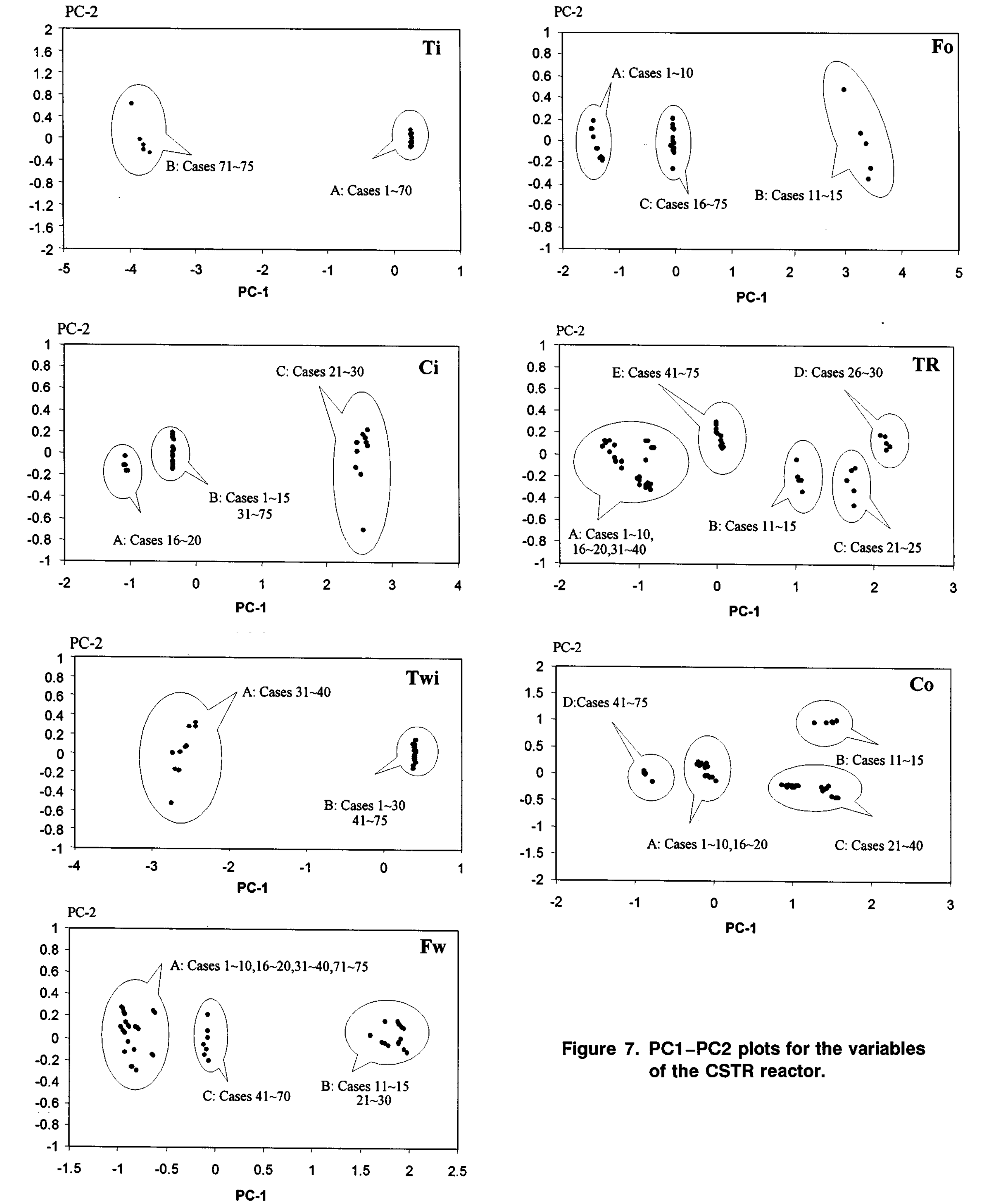

Figure 7. PC1–PC2 plots for the variables of the CSTR reactor.

the adverse effect of noises. As a result, Bakshi and Stephanopolous (1994a,b) further developed the triangular approach by introducing the wavelet-based multiscale signal-analysis technique. Wavelet multiscale analysis is able to eliminate the noise components, and at the same time, automatically identify the inflection points of a trend signal. An inflection point is considered to be the connection of two episodes.

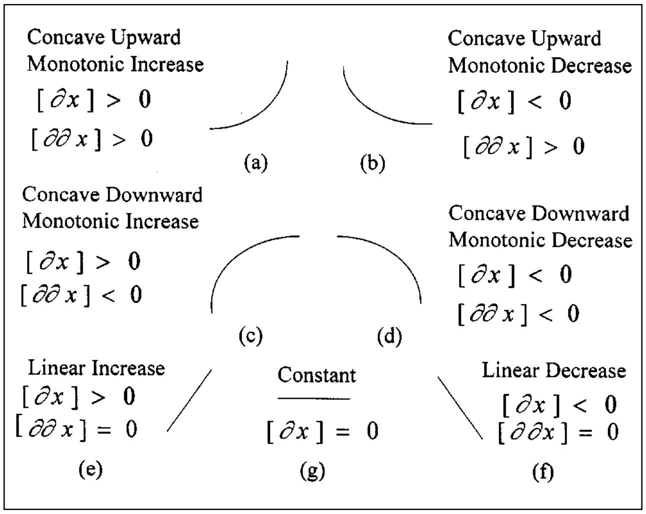

Figure 8. Seven episodes.

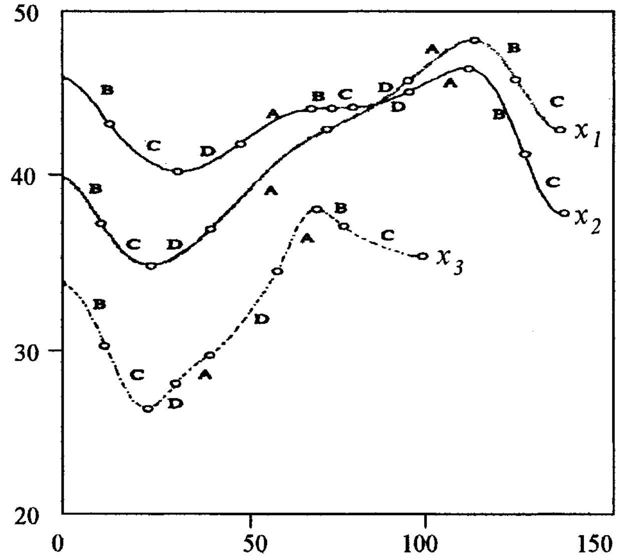

Figure 9. Three trends and their episodes (Bakshi and Stephanopoulos, 1994a,b).

The triangular episode representation was further used to develop inductive rules for product quality monitoring. An example of such rules is "if the trend follows {... *AB CD A D AB* ...}, the batch run is likely to be bad," where *AB*, *CD*, *A*, *D*, and *AB* are different episodes. It is clear that the level of cognition of such a description is still not very high. In addition, the discrimination and clustering of trends is not straightforward. For example, the three trends in Figure 9 can be converted to:

$$x_1 = [B \quad CD \quad A \quad D \quad AB \quad C]$$
$$x_2 = [B \quad CD \quad AB \quad CD \quad AB \quad C]$$
$$x_3 = [B \quad CD \quad A \quad D \quad AB \quad C] \tag{10}$$

Based on visual inspection of the original trends, we can immediately tell that x_1 and x_2 are similar and that they are distinct from x_3. It is not so easy to reach such a conclusion by looking at their qualitative representations using episodes, as shown in Eqs. 10. The episode approaches are not based on pattern recognition. Clearly, the method used in this work using PCA is a more compact and accurate representation of dynamic trends.

Another difference of the current approach with the episode–wavelet approach of Bakshi and Stephanopoulos (1994a,b) is worth noting. The episode–wavelets method analyzes the trends of a variable one by one. For instance, the three trends in Figure 9 were analyzed separately to get the result in Eqs. 10. Comparatively, the PCA-based method deals with the trends of a variable in one step. For instance, to get the PC1–PC2 plane, the 75 trends of F_i in Figure 4 were processed in a single step.

Another point that needs to be addressed is that earlier works on qualitative representation of dynamic trends, and on SDG, were developed separately. The techniques for qualitative representation of dynamic trends have not been effectively incorporated into SDG. For example, even in the most recent publications regarding SDG, for instance, Lee et al. (1999), such descriptors as high, low, and normal are still used in describing trends.

Reasoning Mechanisms

Having described the method for the categorical characterization of the temporal trends of variables in a windowed time scale, it is now appropriate to develop the reasoning mechanism. The reasoning mechanism for a digraph containing no interacting and recycle nodes will be introduced first, and is then extended to deal with interacting and recycle nodes. To aid the illustration of the procedures, a small data set (Table 3), which contains a collection of three data cases representing six variables, each taking three categorical values, is used throughout the discussion. The categorical values are supposed to have been obtained by processing temporal trends of three variables, x_1, x_2, and x_3, using the method described in the previous section.

Basic connections and reasoning

For a digraph containing no interacting nodes, there are three basic connections, that is, serial (Figure 10a), convergent (Figure 10b), and divergent (Figure 10c). For the serial connection of Figure 10a, the reasoning rules that we learned from Table 3 for $x_1 \rightarrow x_2$ and $x_2 \rightarrow x_3$ are

IF $x_1 = A$ THEN $x_2 = B$; IF $x_1 = B$ THEN $x_2 = A$;

IF $x_1 = C$ THEN $x_2 = C$,

IF $x_2 = B$ THEN $x_3 = A$; IF $x_2 = A$ THEN $x_3 = B$;

IF $x_2 = C$ THEN $x_3 = C$.

For the convergent connection of Figure 10b, the rules induced from Table 1 are

IF $x_1 = A$ and $x_2 = B$ THEN $x_3 = A$; IF $x_1 = B$ and $x_2 = A$ THEN $x_3 = B$; IF $x_1 = C$ and $x_2 = C$ THEN $x_3 = C$.

Table 3. Example Data Collection for Illustrating Reasoning Mechanism

Data Case	x_1	x_2	x_3	x_4	x_5	x_6
1	A	B	A	C	C	A
2	B	A	B	B	A	C
3	C	C	C	A	B	B

For the divergent connection of Figure 10c, the rules obtained are

IF $x_1 = A$ THEN $x_2 = B$ and $x_3 = A$; IF $x_1 = B$ THEN $x_2 = A$ and $x_3 = B$; IF $x_1 = C$ THEN $x_2 = C$ and $x_3 = C$.

If the value of a node, such as due to operator's intervention or a fault, is fixed externally, it becomes an independent node. An independent node will not be affected by its precedent nodes, but will still influence its succeeding nodes.

Interacting and recycle nodes

The difficulty with dealing with interacting nodes was first encountered in the work of Iri et al. (1979) when a control loop had to be considered when applying SDG to fault diagnosis. Like Iri et al. (1979), most later developments adopted ad hoc methods for dealing with interacting nodes, which are often based on some assumptions. They are specifically designed for fault diagnosis and adopt a hypothesis−test strategy. The difficulty with the previous works in dealing with interacting nodes was also the result of their inability to describe node values more accurately than simply $+$, $-$, and 0. An interesting and also more reasonable approach was developed by Mo et al. (1997) and Lee et al. (1999), who treat all the nodes related to a single control loop as a cluster. Our method of dealing with interacting as well as recycle nodes is similar to the approach of these authors, though our approach is not restricted to control loops, and node values are not in the form of $+$, $-$, and 0. Various examples of interacting and recycle nodes are depicted in Figure 10. A detailed reasoning procedure is described below based on these situations.

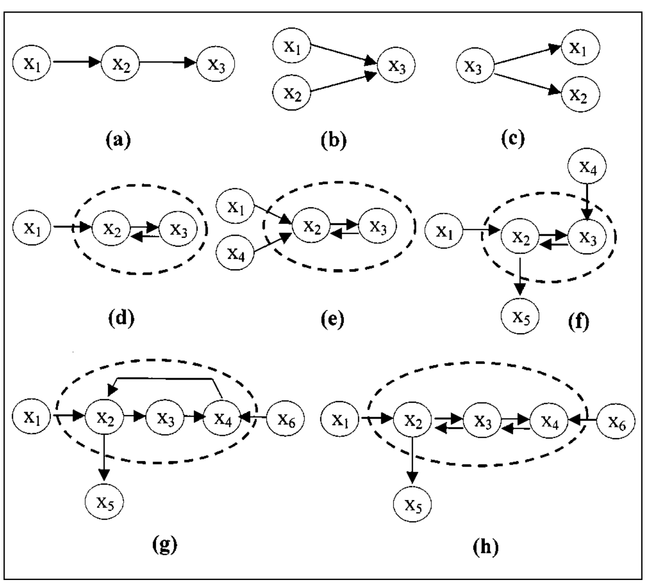

Figure 10. Basic connections in a digraph.

When a node (or nodes) is a dependent node, we consider all its interacting and recycle nodes as a single node. This is a reasonable consideration because, for example, Figure 10d, when there is a change in x_1, the changes in x_2 and x_3 are correlated. If we only consider the first data set in Table 3, then the rules for Figures 10d−10h are:

For Figure 10d IF $x_1 = A$ THEN ($x_2 = B$ and $x_3 = A$)
For Figure 10e IF $x_1 = A$ and $x_4 = C$ THEN ($x_2 = B$ and $x_3 = A$)
For Figure 10f IF $x_1 = A$ and $x_4 = C$ THEN ($x_2 = B$ and $x_3 = A$)
For Figure 10g IF $x_1 = A$ and $x_6 = A$ THEN ($x_2 = B$ and $x_3 = A$ and $x_4 = C$)
For Figure 10h IF $x_1 = A$ and $x_6 = A$ THEN ($x_2 = B$ and $x_3 =$ A and $x_4 = C$).

When considering the effect of a cluster of interacting nodes on other nodes, however, we do not need to treat the interacting nodes as a single node. For example, the rules drawn from the data of Table 3 regarding the connection directly to x_5 in Figure 10f are

IF $x_2 = B$ THEN $x_5 = C$; IF $x_2 = A$ THEN $x_5 = A$;

IF $x_2 = C$ THEN $x_5 = B$.

It is clear that, unlike earlier works on SDG, the causal relationship between two nodes or two node clusters is not simply positive or negative. It is a mapping of two spaces of the categorical values of the two variables. Therefore, the method works independently of the complex relationships between the two nodes or variables (or node clusters) linked by a branch. There is no compromise on the complexity of the relationship in devising the reasoning mechanism.

In contrast, in various versions of SDG, the relationships between two nodes were simply $+$, $-$, or 0, representing positive, negative, or no influential relationship. Consequently, the reasoning rules are derived based on the signs of the connections. This is clearly an oversimplified treatment. Such a simplified treatment of causal variables not only is not accurate but also may lead to difficulties in reasoning. For example, suppose a variable x_3 has two and only two incoming connections from x_1 and x_2, both with $+$ signs. When x_1 is increasing while x_2 is decreasing, it will not be possible to predict whether x_3 will increase, decrease, or remain unchanged. Although there have been several efforts at combating such ambiguity, the methods are far from rigorous.

Fuzzy Quantification of the Qualitative Representation of a Variable's Temporal Behavior

Extraction of qualitative values from the PC1−PC2 plane of a variable's trends based on visual examination is subjective. When the grouping is vague, it is difficult to assign some data cases to clusters. Even in the same cluster of the

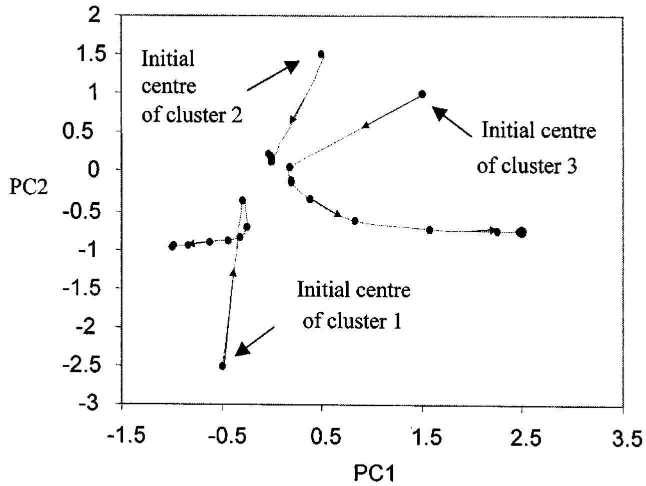

Figure 11. Locus of searching the clusters on a PC1–PC2 plane.

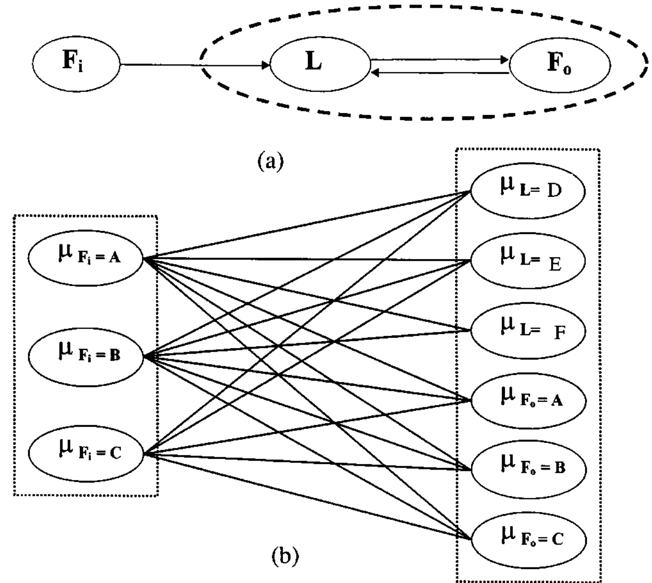

Figure 13. The learning procedure.

PC1−PC2 plane, the data points are not completely overlapped. In other words, the distances of the data points to the cluster center vary. Therefore, considering that all data points in the same cluster have the same value is purely qualitative and not accurate. Furthermore, visual inspection is not an automatic approach, hindering the use of computers. In this section we propose to use the fuzzy *c*-means clustering approach for automatic fuzzy grouping of the data points in the PC1−PC2 plane.

Given the number of clusters and an estimation of the cluster centers, that is, the pairs of values on the PC1 and PC2 axes, the fuzzy *c*-means approach will automatically find the true cluster centers, and for each cluster, calculate the distance of each data case to the center. All the PC1−PC2 clusters in Figure 7 have been processed using the fuzzy *c*-means algorithms. It was found that the method is very tolerant of the initial center approximations and can generally find the real center in 3 to 10 steps. Figure 11 shows an example of the locus of the search for the centers of clusters on a PC1−PC2 plane. It can be seen that in this example, although the starting points are very far from the actual centers of the three clusters, it is able to quickly identify the actual centers in 6 to 7 steps.

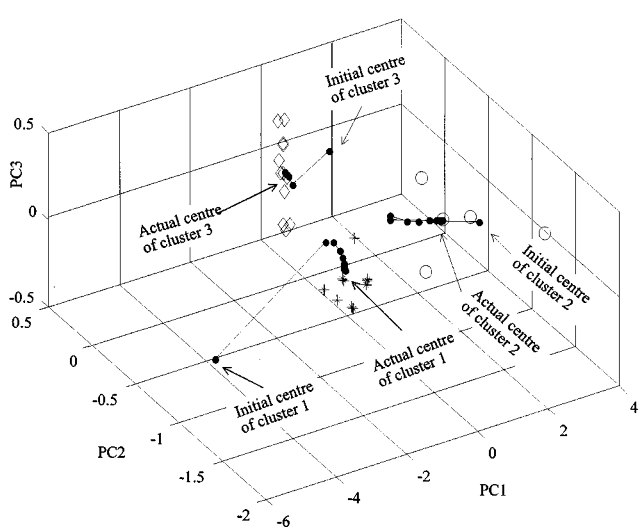

Figure 12. Locus of searching the cluster centers when three PCs are considered.

Another advantage of using the fuzzy *c*-means to help the categorical characterization of a variable's temporal behavior is that fuzzy *c*-mean can handle more than two PCs. If the first two PCs are not able to capture most of the variance, then more PCs should be used. There are various approaches available in the literature for calculating the number of PCs that should be used. Unfortunately these aproaches often give inconsistent results, as discovered by Valle et al. (1999). In this study, we have used a simple rule, which states that the number of PCs chosen should represent more than 93% of the variance (Johnson, 1998). Figure 12 shows an example of how, when three PCs are considered, fuzzy *c*-means can automatically identify the centers of clusters.

Fuzzy qualitative reasoning in the fuzzy digraph

We use Figure 13a, which is part of the digraph of Figure 14 for the CSTR, to illustrate the reasoning procedure in a fuzzy clustered digraph. Nodes L and Fo interact because of the existence of the level controller. The PC1−PC2 planes for F_i, L, and Fo, are shown in Figures 4 and 7, which indicate that each variable takes three categorical values. With the fuzzy quantification of node values, a causal rule between node F_i and the node clusters of L and Fo should be in the form of, as an example, IF $F_i = A(\mu_{F_i-A})$ THEN $L = D(\mu_{L-D})$, and $Fo = A(\mu_{F_o-A})$, where μ_{F_i-A} is the fuzzy membership value of F_i, which belongs to the qualitative value of A in the PC1−PC2 two-dimensional plane of F_i. We use the single-layer percetron shown in Figure 13b to learn the fuzzy membership value. Each variable is split into three

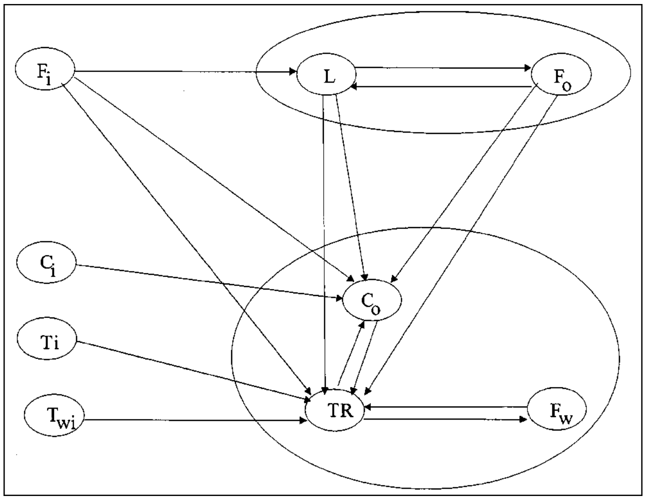

Figure 14. Digraph for the CSTR reactor.

nodes according to the number of qualitative values. The fuzzy membership value, $\mu_{F_i} = A$, of F_i takes the value of A. The error propagation learning algorithm used by three-layer neural networks can be adapted to training such a percetron network.

The first impression of the method seems to be that the way of describing the causal relationships between two nodes is not as intuitive as simply using $+$, $-$, or 0. In essence, this does not necessarily imply a deficiency of the current method, because in the application of either SDG for fault diagnosis or the current fuzzy clustered digraph for temporal behavior modeling, we are only interested in the links and values of the nodes, not in the signs of links. The signs on the links in an SDG are only used to facilitate the reasoning. This is similar to Bayesian networks in which the branches only mean a link between two nodes. The reasoning in a Bayesian network is based on the conditional probability calculation, which also requires a complex conditional probability table.

Application to the CSTR Case Study

The CSTR case study is described in the Appendix and has been referenced in previous sections. In this section, we summarize further the result of applying the method to this CSTR case study.

The digraph for the CSTR reactor of Figure 15 is given in Figure 14. It has four independent nodes—F_i, C_i, T_i, and T_{wi}—and two dependent node clusters—$L \leftrightarrow F_o$, and $C_o \leftrightarrow TR \leftrightarrow F_W$. A dynamic simulator was developed to generate the 75 data cases used in the study. White noise was added to the data before the method was applied.

Nodes F_W, TR, T_{Wi}, C_o, T_i and C_i, and F_i and L were plotted in Figures 4 and 5. Their qualitative characterization results were given in Figure 7. In Figure 7, the data cases that belong to a specific cluster were also indicated.

The digraph (Figure 14) is combined with Figures 4 and 7 to generate the reasoning rules, which are summarized in Table 4. Separate rules are generated for the two dependent node clusters $L \leftrightarrow F_o$, and $C_o \leftrightarrow TR \leftrightarrow F_W$. As an example,

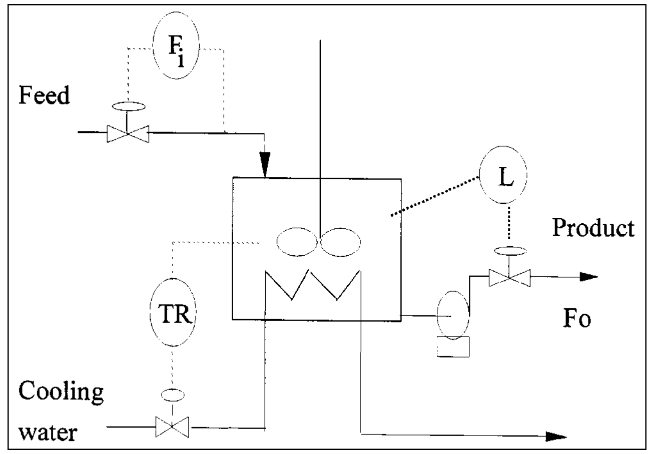

Figure 15. CSTR reactor.

Rule 2 should be interpreted as

$$\text{IF } F_i = B$$
$$\text{THEN } L = E \text{ and } F_o = B.$$

Similarly, Rule 11 should be interpreted as

$$\text{IF } F_i = C \text{ and } T_i = B \text{ and } C_i = B \text{ and } T_{Wi} = B \text{ and } F_o = C \text{ and } L = C$$
$$\text{THEN } C_o = D \text{ and } TR = E \text{ and } F_W = A$$

For the results in Table 4, each node has taken sharp qualitative values. Fuzzy c-means was used only to help assign individual data cases to clusters, but was not used in quantifying the node values.

We use Figure 13, which is part of the CSTR digraph of Figure 14, to illustrate the situation when each node is quantified using a fuzzy membership function. Figure 13a is a digraph linking the node F_i and the node cluster $L \leftrightarrow F_o$. The

Table 4. Rules Generated for the CSTR Reactor

	IF	THEN	
Rules Generated for the Dependent Node Cluster L ↔ Fo			
	F_i	L	F_o
Rule 1	*A*	*D*	*A*
Rule 2	*B*	*E*	*B*
Rule 3	*C*	*F*	*C*

Rules Generated for the Dependent Node Cluster $C_0 \leftrightarrow TR \leftrightarrow F_W$									
	F_i	T_i	C_i	T_{Wi}	F_o	L	C_o	TR	F_W
Rule 4	*A*	*A*	*B*	*B*	*A*	*D*	*A*	*A*	*A*
Rule 5	*B*	*A*	*B*	*B*	*B*	*E*	*B*	*B*	*B*
Rule 6	*C*	*A*	*A*	*B*	*C*	*F*	*A*	*A*	*A*
Rule 7	*C*	*A*	*C*	*B*	*C*	*F*	*C*	*C*	*B*
							or *C*	*D*	*B*
Rule 8	*C*	*A*	*B*	*A*	*C*	*F*	*C*	*A*	*A*
Rule 9	*C*	*A*	*B*	*B*	*C*	*F*	*D*	*E*	*C*
Rule 10	*C*	*B*	*B*	*B*	*C*	*F*	*D*	*E*	*A*
Rule 11	*C*	*B*	*B*	*B*	*C*	*F*	*D*	*E*	*A*

rules generated are in the following general form,

$$\begin{aligned}
&\text{IF} \quad && F_i = (A, \text{ or } B, \text{ or } C, \mu_{F_i}) \\
&\text{THEN} \quad && L = (D, \text{ or } E, \text{ or } F, \mu_L) \\
&\text{and} \quad && F_o = (A, \text{ or } B, \text{ or } C, \mu_{F_o})
\end{aligned}$$

The relationship is represented by the perceptron of Figure 13b, which is trained using the error feedback algorithm adapted from feedforward neural networks.

In practical application of the digraph of Figure 13a, which has been trained using Figure 13b, the trend of F_i is first processed using both PCA and fuzzy *c*-means clustering, and the single-layer perceptron of Figure 13b will predict the values of L and F_o.

As an example, a trend of F_i in a windowed time scale is processed using both PCA and fuzzy *c*-means for qualitative characterization, which give the following result:

$$F_i = (A, 0.99900),\ F_i = (B, 0.00093),\ F_i = (C, 0.00008).$$

Based on this input, the trained single-layer perceptron of Figure 13b predicts the values for F_o and L. The prediction is

$$L = (D, 0.96722),\ L = (E, 0.02591),\ L = (F, 0.00730).$$

$$F_o = (A, 0.98182),\ F_o = (B, 0.00035),\ F_o = (C, 0.02034)$$

It is important to be aware of the differences between the proposed method and traditional dynamic simulation. An obvious difference is that the former is qualitative and the latter is numerical. Another important difference is that the current method is data based and has a statistical basis, while traditional dynamic simulation is model based. As a result, when the current method is applied, it is valid only when the new data fall within the training-data space. In this respect, it is similar to feedforward neural networks that can guarantee correctness and accuracy only when new data fall within the training-data space. This limitation can be overcome through extensive training using data that cover various scenarios, for example, by combining Monte Carlo simulation with dynamic process simulation (Saraiva and Stephanopoulos, 1992). In addition, because the current method is data based, its performance can be continuously improved during use when more and more data are made available.

Conclusions and Future Work

It has been well documented that the information provided to operators by modern computer control systems is hard to understand, hampering correct and rapid development of a mental model of the temporal behavior of the individual variables and the composite process. The majority of effort in qualitative modeling has been focused on steady-state situations. Previous attempts to extend them to dynamic systems have only achieved very limited success. The main difficulties have been due to the time dimension, and the complex dependency and interactions of variables.

The methodology developed in this work allows rigorous qualitative/quantitative representation of the temporal trends of individual variables, as well as the complex dynamics of the composite process. It consists of two major steps: categorical characterization of a variable's temporal behavior within a windowed time scale using PCA, and modeling of the system's dynamics using a digraph containing dependent and interacting nodes. The advantage of PCA qualitative representation of dynamic trends is that it allows one to describe a trend with only one simple categorical value. The digraph method is also more advantageous than other methods, such as decision trees, because the latter does not allow interacting and recycle nodes and is therefore an oversimplified representation. The introduction of fuzzy *c*-means also allows quantitative and more accurate description of the temporal behavior of variables and their dynamic causal relationships.

In the proposed method, it is assumed that a digraph can be drawn before the method is applied. In practice, drawing the digraph can be a very daunting task, particularly when the process is complex and interactions between variables are strong and only partially known. Most earlier studies on SDG have also ignored this issue. There are now data-mining techniques that can automatically discover from data the dependencies of variables (Wang et al., 1997); unfortunately, they are still restricted to very simple data structures. Further study is clearly needed to develop systematic, even automatic methods for drawing digraphs. However, it needs to be pointed out that the methodology developed in this article does not rely on very accurate digraph structures. For instance, if the actual structure is $x_1 \to x_2 \to x_3$, but is mistakenly drawn as $x_1 \to x_3$, it will not have a significant effect on the digraph performance.

In the proposed method, both the categorical characterization of the temporal trends of individual variables as well as the automatic generation of reasoning rules can be regarded as learning procedures. However, the former is not a recursive or incremental learning procedure, though the latter is. This is because PCA is not recursive. For on-line use, it is clearly useful for PCA to be able to learn recursively, that is, learn to improve the performance from each new single example when it is presented, and there is no need to mix the new data with previous data that have been used in training. Biehl and Schlosser (1998) have described recursive PCA methods that are worth investigating in connection with the current study.

Literature Cited

Bakshi, B. R., and G. Stephanopoulos, "Reasoning in Time: Modeling, Analysis and Pattern Recognition of Temporal Process Trends," *Intelligent Systems in Process Engineering: Paradigms from Design and Operations*, G. Stephanopoulos and C. Han, eds., Academic Press, New York (1996).

Bakshi, B. R., and G. Stephanopoulos, "Representation of Process Trends: III. Multiscale Extraction of Trends from Process Data," *Comput. Chem. Eng.*, **18**, 267 (1994a).

Bakshi, B. R., and G. Stephanopoulos, "Representation of Process Trends: IV. Induction of Real-Time Patterns from Operating Data for Diagnosis and Supervisory Control," *Comput. Chem. Eng.*, **18**, 303 (1994b).

Bezdek, J., *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York (1981).

Biehl, M., and E. Schlosser, "The Dynamics of On-Line Principal Component Analysis," *J. Phys. A.-Math. Gen.*, **31**, L97 (1998).

Chen, B. H., X. Z. Wang, S. H. Yang, and C. McGreavy, "Application of Wavelets and Neural Networks to Diagnostic System De-

velopment: 1. Feature Extraction,'' *Comput. Chem. Eng.*, **23**, 899 (1999).
Chen, J. G., J. A. Bandoni, and J. A. Romagnoli, ''Robust PCA and Normal Region in Multivariate Statistical Process Monitoring,'' *AIChE J.*, **42**, 3563 (1996).
Cheung, J. T. Y., and G. Stephanopoulos, ''Representation of Process Trends: 1. A Formal Representation Framework,'' *Comput. Chem. Eng.*, **14**, 495 (1990a).
Cheung, J. T. Y., and G. Stephanopoulos, ''Representation of Process Trends: 2. The Problem of Scale and Qualitative Scaling,'' *Comput. Chem. Eng.*, **14**, 511 (1990b).
Dong, D., and T. J. McAvoy, ''Nonlinear Principal Component Analysis—Based on Principal Curves and Neural Networks,'' *Comput. Chem. Eng.*, **20**, 65 (1996).
Dunia, R., S. J. Qin, T. F. Edgar, and T. J. McAvoy, ''Identification of Faulty Sensors Using Principal Component Analysis,'' *AIChE J.*, **42**, 2797 (1996).
Friedman, M., and A. Kandel, *Introduction to Pattern Recognition: Statistical, Structural, Neural and Fuzzy Logic Approaches*, World Scientific, Singapore (1999).
Hotelling, H., ''Analysis of a Complex of Statistical Variables into Principal Components,'' *J. Educ. Psychol.*, **24**, 417, 498 (1933).
Huang, Y. C., and X. Z. Wang, ''Application of Fuzzy Causal Networks to Waste Water Treatment Plants,'' *Chem. Eng. Sci.*, **54**, 2731 (1999).
Iri, M., K. Aoki, E. O'Shima, and H. Matsuyama, ''An Algorithm for Diagnosis of System Failures in the Chemical Process,'' *Comput. Chem. Eng.*, **3**, 489 (1979).
Janusz, M. E., and V. Venkatasubramanian, ''Automatic Generation of Qualitative Descriptions of Process Trends for Fault Detection and Diagnosis,'' *Eng. App.. Artif. Intell.*, **4**, 329 (1991).
Johnson, D. E., *Applied Multivariate Methods for Data Analysis*, Brooks/Cole Pub., Pacific Grove, CA (1998).
Lee, G. B., B. W. Lee, E. S. Yoon, and C. H. Han, ''Multiple-Fault Diagnosis Under Uncertain Conditions by the Quantification of Qualitative Relations,'' *Ind. Eng. Chem. Res.*, **38**, 988 (1999).
MacGregor, J. F., and T. Kourti, ''Statistical Process Control of Multivariable Processes,'' *Control Eng. Practice*, **3**, 403 (1995).
Mo, K. J., G. Lee, D. S. Nam, Y. H. Yoon, and E. S. Yoon, ''Robust Fault Diagnosis Based on Clustered Symptom Trees,'' *Control Eng. Practice*, **5**, 199 (1997).
Moore, R. L., and M. A. Kramer, ''Expert Systems in On-Line Process Control,'' *Chemical Process Control—CPCIII*, M. Morari and T. J. McAvoy, eds., Elsevier, Amsterdam (1986).
Negiz, A., and A. Cinar, ''Statistical Monitoring of Multivariable Dynamic Processes with State Space Models,'' *AIChE J.*, **43**, 2002 (1997).
Neogi, D., and C. E. Schlags, ''Multivariate Statistical Analysis of an Emulsion Batch Process,'' *Ind. Eng. Chem. Res.*, **37**, 3971 (1998).
Nomikos, P., and J. F. MacGregor, ''Monitoring Batch Processes Using Multiway Principal Component Analysis,'' *AIChE J.*, **40**, 1361 (1994).
Pearson, K., ''On Lines and Planes of Closest Fit to Systems of Points in Space,'' *Phil. Mag.*, **2**, 559 (1901).
Saraiva, P. M., and G. Stephanopoulos, ''Continuous Process Improvement Through Inductive and Analogical Learning,'' *AIChE J.*, **8**, 161 (1992).
Valle, S., W. H. Li, and S. J. Qin, ''Selection of the Number of Principal Components: The Variance of the Reconstruction Error Criterion with a Comparison to Other Methods,'' *Ind. Eng. Chem. Res.*, **38**, 4389 (1999).
Vedam, H., and V. Venkatasubramanian, ''PCA-SDG Based Process Monitoring and Fault Diagnosis,'' *Control Eng. Practice*, **7**, 903 (1999).
Wang, X. Z., and R. F. Li, ''Combining Conceptual Clustering and Principal Component Analysis for State Space Based Process Monitoring,'' *Ind. Eng. Chem. Res.*, **38**, 4345 (1999).
Wang, X. Z., B. H. Chen, and C. McGreavy, ''Data Mining for Failure Diagnosis of Process Units by Learning Probabilistic Networks,'' *Trans. Inst. Chem. Eng.*, **75(Part B)**, 210 (1997).
Wang, X. Z., *Data Mining and Knowledge Discovery for Process Monitoring and Control*, Springer, London (1999).
Whiteley, J. R., and J. F. Davis, ''Knowledge-Based Interpretation of Sensor Patterns,'' *Comput. Chem. Eng.*, **16**, 329 (1992).
Yamanaka, F., and T. Nishiya, ''Application of the Intelligent Alarm System for the Plant Operation,'' *Comput. Chem. Eng.*, **21**, S625 (1997).

Appendix

The single, nonisothermal continuous stirred-tank chemical reactor (CSTR) shown in Figure 15 is used as a case study.

A single reaction, $A \rightarrow B$, takes place in the reactor. It is assumed that the tank is very well mixed, and therefore that the concentration of C_o of component A and the temperature of the product stream leaving the tank are equal to that in the tank. The reaction is exothermic and reaction temperature TR is automatically controlled by a PID controller through manipulation of the cooling-water flow rate. There are also feed flow-rate and liquid-level controllers, as shown in Figure 15. The dynamic behavior of the process is described by the following equations:

Component mass balance

$$V\frac{dC_o}{dt} = F_i C_i - F_o C_o - VK_o e^{-E/RTR} C_o.$$

Energy balance

$$V\rho C_p \frac{dTR}{dt} = \rho C_p F_i T_i - \rho C_p F_o TR - \frac{aF_c^{b+1}}{F_C + \dfrac{aF_c^b}{2\rho_c C_{pc}}}(TR - Twi) + (-\Delta H_{r\times n}) VK_o e^{-E/RTR} C_o.$$

Total mass balance

$$\frac{dV}{dt} = F_i - F_o,$$

where

V = holdup of the reaction mixture, m^3 (tank volume is 1 m^3)
L = liquid level in the tank, $0 \sim 100$
C_o = concentration of component A in the product stream leaving the tank, kmol/m^3, equal to that in the tank
C_i = concentration of component A in the inlet feed stream, kmol/m^3
F_i = feed flow rate, m^3/min
F_c = cooling-water flow rate, m^3/min
F_o = product stream flow rate, m^3/min
$E/R = 8330.1$ K^{-1}
K_o = reaction coefficient, $K_o = 1.0 \times 10^{10}$ min^{-1}
ρ = density of reaction mixture, $\rho = 10^6$ g/m^3
ρ_c = density of cooling water, $\rho_c = 10^6$ g/m^3
C_p = specific heat capacity of the reaction mixture, $C_p = 1$ cal/m^3
C_{pc} = specific heat capacity of cooling water, $C_p = 1$ cal/m^3, $-\Delta H_{r\times n} = 130 \times 10^6$ (cal/kmol)
a = constant, $a = 1.678 \times 10^6$ (cal/min)/K
b = constant, $b = 0.5$
T_i = temperature of the inlet feed, K
TR = temperature of the reaction mixture, K
Twi = inlet temperature of cooling water, K

Manuscript received Apr. 3, 2000, and revision received Sept. 5, 2000.