J. Proc. Cont. Vol. 6, No. 6, pp. 329–348, 1996
Copyright © 1996 Published by Elsevier Science Ltd
Printed in Great Britain. All rights reserved
0959-1524/96 $15.00 + 0.00

PII: S0950-1524(96)00009-1

Full Papers

The process chemometrics approach to process monitoring and fault detection

Barry M. Wise and Neal B. Gallagher

Eigenvector Research, 830 Wapato Lake Road, Manson, WA 98831, USA

Received in revised form 15 March 1996

Chemometrics, the application of mathematical and statistical methods to the analysis of chemical data, is finding ever widening applications in the chemical process environment. This article reviews the chemometrics approach to chemical process monitoring and fault detection. These approaches rely on the formation of a mathematical/statistical model that is based on historical process data. New process data can then be compared with models of normal operation in order to detect a change in the system. Typical modelling approaches rely on principal components analysis, partial least squares and a variety of other chemometric methods. Applications where the ordered nature of the data is taken into account explicitly are also beginning to see use. This article reviews the state-of-the-art of process chemometrics and current trends in research and applications. Copyright © 1996 Published by Elsevier Science Ltd

Keywords: fault detection; multivariable systems; multivariate quality control; statistical analysis

Chemometrics is often thought of as a sub-discipline of chemistry, and in particular, analytical chemistry. Recently, 'chemometric techniques' have been applied to problems that are often thought of as being in the domain of the chemical engineer, i.e. chemical processes. These applications can be roughly divided between those directed at maintenance of process instruments, e.g. calibration, and those that are concerned with maintenance of the process itself, e.g. statistical process control and dynamic modeling. Our focus will be on the latter area. In this article we will review some of the basic chemometric techniques, such as Principal Components Analysis (PCA) and Partial Least Squares (PLS), and some of the newer and more advanced methods, such as Multi-way PCA and PLS. Some example problems utilizing these methods in chemical process monitoring and dynamic process modeling will be considered.

Definition

Before moving on to chemometric techniques, it is important that we define chemometrics. This has been a matter of debate in the technical community for some time;[1,2] however, the authors choose the following definition: Chemometrics is the science of relating measurements made on a chemical system to the state of the system via application of mathematical or statistical methods. It is clear from this definition that chemometrics is data-based. The goal of many chemometrics techniques is the production of an empirical model, derived from data, that allows one to estimate one or more properties of a system from measurements. Chemical systems, of course, include dynamic chemical processes.

PCA: theory and applications

It has been pointed out several times in the recent literature that chemical processes are becoming more heavily instrumented and the data is recorded more frequently.[3,4] This is creating a data overload, and the result is that a good deal of the data is 'wasted', i.e. no useful information is obtained from it. The problem is one of both compression and extraction. Generally, there is a great deal of correlated or redundant information in process measurements. This information must be compressed in a manner that retains the essential information and is more easily displayed than each of the process variables individually. Also, often essential information lies not in any individual process variable but in how the variables change with respect to one another, i.e. how they co-vary. In this case the information must be extracted from the data. Furthermore, in the presence of large amounts of noise, it would be

---

*Paper presented at the IFAC Workshop on On-line Fault Detection and Supervision in the Chemical Process Industries held at Newcastle, UK in June 1995 desirable to take advantage of some sort of signal averaging.

Principal Components Analysis

Principal Components Analysis (PCA) is a favorite tool of chemometricians for data compression and information extraction.[5,6] PCA finds combinations of variables or factors that describe major trends in a data set. Mathematically, PCA relies on an eigenvector decomposition of the covariance or correlation matrix of the process variables. In this work we will use the convention that rows of a data matrix **X** correspond to samples while columns correspond to variables. For a given data matrix **X** with *m* rows and *n* columns the covariance matrix of **X** is defined as

$$\mathrm{cov}(\mathbf{X}) = \frac{\mathbf{X}^{\mathrm{T}}\mathbf{X}}{m-1} \tag{1}$$

This assumes that the columns of **X** have been 'mean centered', i.e. adjusted to have a zero mean by subtracting off the original mean of each column. If the columns of **X** have been 'autoscaled', i.e. adjusted to zero mean and unit variance by dividing each column by its standard deviation, Equation 1 gives the correlation matrix of **X**. (Unless otherwise noted, it is assumed that data is either mean centered or autoscaled prior to analysis.) PCA decomposes the data matrix **X** as the sum of the outer product of vectors $\mathbf{t}_i$ and $\mathbf{p}_i$ plus a residual matrix **E**:

$$\mathbf{X} = \mathbf{t}_1\mathbf{p}^{\mathrm{T}}_1 + \mathbf{t}_2\mathbf{p}^{\mathrm{T}}_2 + \ldots + \mathbf{t}_k\mathbf{p}^{\mathrm{T}}_k + \mathbf{E} \tag{2}$$

Here *k* must be less than or equal to the smaller dimension of **X**, i.e. $k \leq \min\{m, n\}$. The $\mathbf{t}_i$ vectors are known as scores and contain information on how the samples relate to each other. The $\mathbf{p}_i$ vectors are eigenvectors of the covariance matrix, i.e. for each $\mathbf{p}_i$

$$\mathrm{cov}(\mathbf{X})\mathbf{p}_i = \lambda_i\mathbf{p}_i \tag{3}$$

where λ_i is the eigenvalue associated with the eigenvector $\mathbf{p}_i$. In PCA the $\mathbf{p}_i$ are known as loadings and contain information on how variables relate to each other. The $\mathbf{t}_i$ form an orthogonal set ($\mathbf{t}_i^{\mathrm{T}}\mathbf{t}_j = 0$ for $i \neq j$), while the $\mathbf{p}_i$ are orthonormal ($\mathbf{p}_i^{\mathrm{T}}\mathbf{p}_j = 0$ for $i \neq j$, $\mathbf{p}_i^{\mathrm{T}}\mathbf{p}_j = 1$ for $i = j$). Note that for **X** and any $\mathbf{t}_i$, $\mathbf{p}_i$ pair

$$\mathbf{X}\mathbf{p}_i = \mathbf{t}_i \tag{4}$$

This is because the score vector $\mathbf{t}_i$ is the linear combination of the original **X** data defined by $\mathbf{p}_i$. The $\mathbf{t}_i$, $\mathbf{p}_i$ pairs are arranged in descending order according to the associated λ_i. The λ_i are a measure of the amount of variance described by the $\mathbf{t}_i$, $\mathbf{p}_i$ pair. In this context, we can think of variance as information. Because the $\mathbf{t}_i$, $\mathbf{p}_i$ pairs are in descending order of λ_i, the first pair capture the largest amount of information of any pair in the decomposition. In fact, it can be shown that the $\mathbf{t}_1$, $\mathbf{p}_1$ pair capture the greatest amount of variation in the data that it is possible to capture with a linear factor. Each subsequent pair capture the greatest possible amount of variance remaining at that step.

The concept of principal components is shown graphically in *Figure 1*. The figure shows a three-dimensional data set where the data lie primarily in a plane, thus the data are well described by a two principal component (PC) model. The first eigenvector or PC aligns with the greatest variation in the data while the second PC aligns with the greatest amount of variation that is orthogonal to the first PC.

Generally it is found (and it is usually the objective) that the data can be adequately described using far fewer factors than original variables. Thus, the data overload experienced in chemical process monitoring can be solved my monitoring fewer scores (weighted sums of the original variables) than original variables, with no significant loss of information. Also, PCA often produces linear combinations of variables that are useful descriptions, or even predictors, of particular process events. These combinations of variables are often more robust indicators of process conditions than individual variables due to the signal averaging aspects of PCA.

It is also possible to calculate a lack of model fit statistic, *Q*. *Q* is simply the sum of squares of each row (sample) of **E** (from Equation 2), for example, for the i^{th} sample in **X**, $\mathbf{x}_i$:

$$Q_i = \mathbf{e}_i\mathbf{e}_i^{\mathrm{T}} = \mathbf{x}_i(\mathbf{I} - \mathbf{P}_k\mathbf{P}_k^{\mathrm{T}})\mathbf{x}_i^{\mathrm{T}} \tag{5}$$

where $\mathbf{e}_i$ is the i^{th} row of **E**, $\mathbf{P}_k$ is the matrix of the first *k* loadings vectors retained in the PCA model (where each vector is a column of $\mathbf{P}_k$) and **I** is the identity matrix of appropriate size (*n* by *n*). The *Q* statistic indicates how well each sample conforms to the PCA model. It is a measure of the amount of variation in each sample not captured by the *k* principal components retained in the model.

A measure of the variation within the PCA model is given by Hotelling's T^2 statistic. T^2 is the sum of normalized squared scores defined as

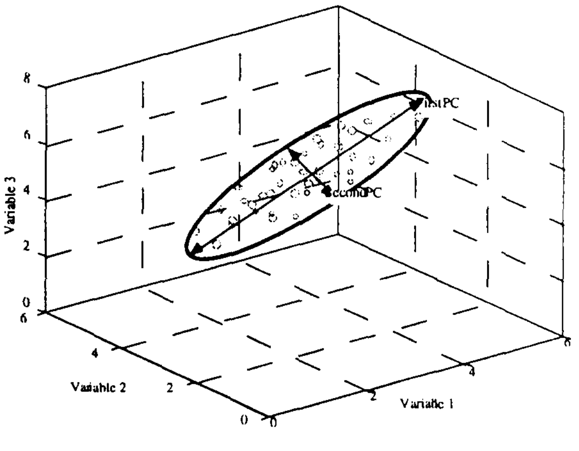

Figure 1 Principal component model of three-dimensional data set lying primarily in a single plane

$$T_i^2 = \mathbf{t}_i \lambda^{-1} \mathbf{t}_i^{\mathrm{T}} = \mathbf{x}_i \mathbf{P} \lambda^{-1} \mathbf{P}^{\mathrm{T}} \mathbf{x}_i^{\mathrm{T}} \qquad (6)$$

where $\mathbf{t}_i$ in this instance refers to the i^{th} row of $\mathbf{T}_k$, the matrix of k scores vectors from the PCA model. The matrix λ^{-1} is a diagonal matrix containing the inverse eigenvalues associated with the k eigenvectors (principal components) retained in the model.

Multivariate statistical process control

Once developed, PCA models can be combined with tools and techniques from univariate statistical process control (SPC) to form multivariate statistical process control (MSPC) tools.[3,4,7–9] Control limits can be placed on the process scores, sum of scores' T^2, residual Q, or residuals of individual variables (single columns of $\mathbf{E}$). There are multivariate analogs to Shewart, range and CUSUM charts. When used for MSPC purposes, PCA models have the additional advantage that the scores variables produced which are linear combinations of the original variables, are more normally distributed than the original variables themselves. This is a consequence of the central limit theorem, which can be stated as follows: if the sample size is large, the theoretical sampling distribution of the mean can be approximated closely with a normal distribution.[10] In our case, we are typically sampling a large number of variables when we form the PCA scores. Thus, we would expect the scores, which are a weighted sum like a mean, to be approximately normally distributed. A word of caution is advised here, however. In some instances, if score values are associated with a controlled property of the system we would not expect the scores to be normally distributed.

Once a PCA model of a data set has been obtained, confidence limits can be established for T^2, the overall residual Q and for the residuals on individual variables. Given the eigenvalues λ_i of the covariance matrix of $\mathbf{X}$ confidence limits can be calculated for Q as follows:[7]

$$Q_\alpha = \Theta_1 \left[\frac{c_\alpha \sqrt{2\Theta_2 h_0^2}}{\Theta_1} + 1 + \frac{\Theta_2 h_0 (h_0 - 1)}{\Theta_1^2} \right]^{\frac{1}{h0}} \qquad (7)$$

where

$$\Theta_i = \sum_{j=k+1}^{n} \lambda_j^i \text{ for } i = 1 - 3 \qquad (8)$$

and

$$h_0 = 1 - \frac{2\Theta_1\Theta_3}{3\Theta_2^2} \qquad (9)$$

In Equation 7 c_α is the standard normal deviate corresponding to the upper $(1 - \alpha)$ percentile. In Equation 8, k is the number of principal components retained in the process model and n is the total number of principal components (equal to the smaller of the number of variables or samples in $\mathbf{X}$).

Statistical confidence limits for T^2 can be calculated by means of the F-distribution as follows:

$$T^2_{k,m,\alpha} = \frac{k(m-1)}{m-k} F_{k,m-1,\alpha} \qquad (10)$$

Here m is the number of samples used to develop the PCA model and k is the number of principal component vectors retained.

Some discussion of the geometric interpretation of Q and T^2 is perhaps in order. As noted above, Q is a measure of the variation of the data outside of the PCA model. Imagine for a moment a process with three variables restricted to lie on a plane that cuts through 3-space such as the one shown in *Figure 1*. Such a system would be well described by a 2 PC model. Q is a measure of the distance off the plane formed by the first 2 PCs. In fact Q is the Euclidean distance of the operating point from the plane formed by the 2 PC model. The Q limit defines a distance off the plane that is considered unusual for normal operating conditions. T^2, on the other hand, is a measure of the distance from the multivariate mean to the projection of the operating point on to the 2 PCs. The T^2 limit defines an ellipse on the plane within which the operating point normally projects.

The residual variance for each variable can be estimated for the PCA model.[3] If it is assumed that the eigenvalues associated with the eigenvectors (PCs) not retained in the model are all the same (a common assumption). then the variance in the residual of the j^{th} variable can be estimated from

$$\hat{s}_j^2 = \left(\sum_{i=1}^{nq} \lambda_i - \sum_{i=1}^{k} \lambda_i \right) \left(1 - \sum_{i=1}^{k} p_{1,j}^2 \right) \qquad (11)$$

which requires only PCs and eigenvalues retained in the model. The first term on the right hand side in Equation 11 can be replaced with the sum of the diagonal elements of the covariance matrix, therefore only the first k eigenvalues of the covariance matrix are required. Given the estimate of the variance of the residuals $\hat{s}_j^2$, the standard F test with the appropriate degrees of freedom can be used to determine if the system or its sensors (variables) have changed. This test is best performed using several samples. The test checks to see if

$$s^2_{j\text{new}} / s^2_{j\text{old}} > F_{\nu\text{new},\, \nu\text{old},\, \alpha} \qquad (12)$$

where

$$\nu_{\text{new}} = m_{\text{new}} \qquad (13)$$

$$\nu_{\text{old}} = m_{\text{old}} - k - 1 \qquad (14)$$

where m_{new} and m_{old} are the number of samples in the test and training sets, respectively, and k is the number of PCs retained in the model. When the inequality of Equation 12 holds, then a change has occurred in the system to a confidence level of $1 - \alpha$.

The mean residual should be zero for all the variables. The t-test can be used to detect a shift in the mean away from zero. In this case the hypothesis that the means are equal is to be tested. Thus, the t-test reduces to

$$t_{\upsilon\text{tot}} = \frac{(\overline{x_{\text{old}}} - \overline{x_{\text{new}}})(\upsilon_{\text{old}} - \upsilon_{\text{new}})^{0.5}}{\left(\frac{1}{\upsilon_{\text{old}}} + \frac{1}{\upsilon_{\text{new}}}\right)^{0.5} (\upsilon_{\text{old}} s_{\text{old}}^2 + \upsilon_{\text{new}} s_{\text{new}}^2)^{0.5}} \tag{15}$$

where the degrees of freedom are both one greater than for the case given above. For the purpose of setting limits, the variances can be assumed to be equal to the variance of the residuals of the calibration set calculated by Equation 11. Once the desired confidence level is chosen, it is possible to solve for the difference between old and new means that is just significant.

Control limits can also be established for the scores, **T**. However, because the scores are related to the states of the process (which will be shown below), they are typically quite correlated in time (autocorrelated) and therefore cannot be assumed to be random and normally distributed. Thus, control limits on the scores must be established based either on judgment concerning desired process operating limits or using more sophisticated time series modeling techniques that are beyond the scope of this manuscript.

Relationship between PCA and state-space models

A theoretical connection exists between PCA and state space models. These models are commonly used by the process control community for describing dynamics of a chemical process. Abundant literature is available concerning the state-space formalism,[11,12] however, it is useful to provide a brief introduction here. Consider a linear, time-invariant discrete, state-space process model:

$$\mathbf{x}(k + 1) = \mathbf{\Phi}\mathbf{x}(k) + \mathbf{\Gamma}\mathbf{u}(k) + \upsilon(k) \tag{16}$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{e}(k) \tag{17}$$

where $x(k)$ is an n by 1 state vector at sampling period k, $\mathbf{u}(k)$ is the r by 1 vector of process inputs and $\mathbf{y}(k)$ is a p by 1 vector of process measurements. The vector $\upsilon(k)$ represents the state noise or disturbance input to the process: $\mathbf{e}(k)$ is measurement noise. The $\mathbf{\Phi}$, $\mathbf{\Gamma}$ and $\mathbf{C}$ matrices are assumed constant. Equation 16 shows that the process state at time $k + 1$ is a function of the state, process inputs and disturbances at time k. Equation 17 relates the process measurements to the process states and measurement noise.

In the typical case of analyzing data from a chemical process, one collects the measurements **y** and forms them into a single matrix **Y**, where each row of **Y** corresponds to a sample. PCA can then be used to decompose **Y**. It has been shown that, for processes where there are more measurements than states variations in the process states appear primarily as variations in the PCA scores, while noise mainly affects the residuals.[3] This is the case where there are more elements in **y** than **x**. This allows one to consider only the noise properties of the system when deriving limits on PCA residuals; the dynamics of the process need not be considered explicitly. The rapid increase in the availability of process analytical instrumentation has drastically increased the instances of processes having more measurements than (significant) states. This connection between PCA and state-space models acts as a guide for the user as to where PCA can be applied directly on process outputs for process monitoring and fault detection purposes.

Application of PCA to a chemical process

As an example we now consider the use of PCA on data obtained from a Slurry-Fed Ceramic Melter (SFCM) test system.[8,13-15] This process converts nuclear fuel reprocessing waste into a stable borosilicate glass. A schematic of the process is shown in *Figure 2*. Feed to the melter, a relatively cool slurry of reprocessing waste and glass forming material is introduced from the top at the center of the melter. The molten glass in the melter is maintained at high temperature by passing electric current through it. Molten glass is drawn off and allowed to solidify for eventual long term storage in a geologic repository.

Data on the SFCM process consists of temperatures measured by thermocouples at 20 locations in the melter. The temperatures are measured at 10 levels in each of two vertical thermowells that are inserted into the molten glass pool. For this example, we will consider the temperature at 16 locations within the molten glass pool as shown in *Figure 3*. (The variables are arranged in the data set so that variables 1–8 are from the bottom to the top of thermowell number 1 and vari-

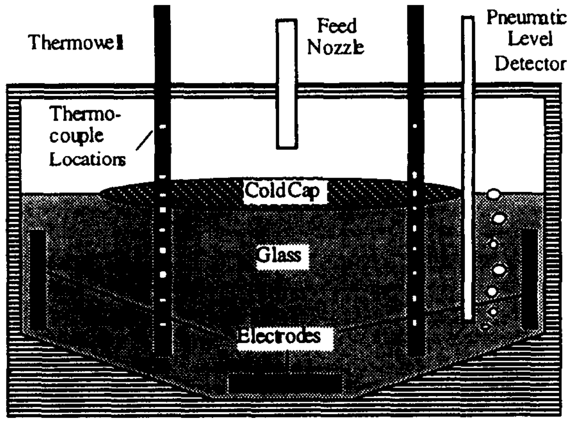

Figure 2 Schematic drawing of slurry-fed ceramic melter

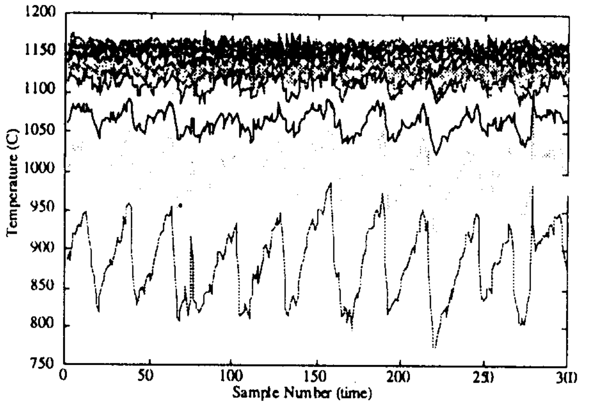

Figure 3 Temperature data from SFCM process

ables 9–16 are from the bottom to the top of thermowell number 2.) It is immediately apparent from *Figure 3* that there is a great deal of correlation in the data. Many of the variables appear to follow a sawtooth pattern.

When PCA is performed on this data (after mean centering), it is found that three factors capture nearly 97% of the variance or information in the data set, as shown in *Table 1*. Thus, the 16 variables can be replaced with three new variables, which are linear combinations of the original variables, with very little loss of information.

The first principal component scores for the SFCM data are shown in *Figure 4*. Recall that these scores are the linear combination of the original temperature variables, defined by the loadings, that describe the greatest variation in the process data. The scores capture the sawtooth nature of the process data. The sawtooth pattern is attributable to changes in the level of the molten glass which is a controlled variable. The loadings plot for the first eigenvector or principal component is shown in *Figure 5*. The variables that contribute most to the first eigenvector are variables 6–8 and 14–16 which correspond to thermocouples near the surface of the molten glass. The second and third PCs (not shown) capture variation that occurs between the two groups of measurement locations (which is not controlled) and variations of the overall process average temperature (which is controlled).

As an example application of MSPC, detection of the failure of a single thermocouple in the SFCM process is shown in *Figures 6* and *7*. In this case, a thermocouple developed a bias, which has resulted in an increase in the model residual Q and in the residuals of the individual variables, $\mathbf{e}_i$. This is because the correlation of the sensor with the remaining sensors has changed due to the bias, i.e. variation outside of the model has occurred. In *Figure 6* the residual Q becomes unusually large starting at sample 51, with the residual increasing to over 100 times the 95% limit. Inspection of the individual residuals[3,16,17] in *Figure 7* identifies the failure as being from thermocouple number 5. In this case there was no need for comparison to the statistical limits on individual variable residuals, however, for less obvious cases a comparison can help correctly identify the faulty sensor.

Note that in this example it is not required that any of the individual variables drift outside their normal range. The MSPC model has detected a change in the system by identifying a change in the relationship between variables. Also, this change was much easier to

Table 1 Variance captured by PCA model of SFCM data

PC number	*Percent variance captured* This PC	 Total
1	88.0711	88.0711
2	6.6974	94.7686
3	2.0442	96.8127
4	0.9122	97.7249
5	0.6693	98.3942
6	0.5503	98.9445
7	0.3614	99.3059
8	0.2268	99.5327

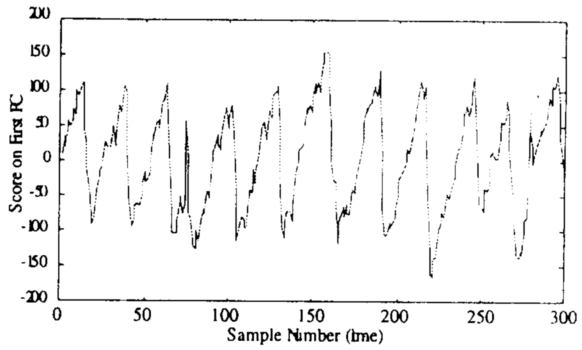

Figure 4 First principal component scores for SFCM data

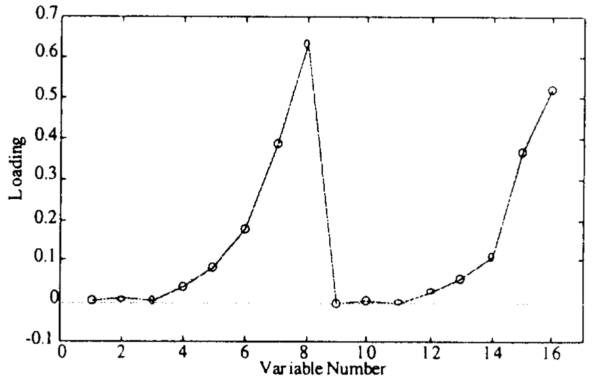

Figure 5 First principal component loadings for SFCM data

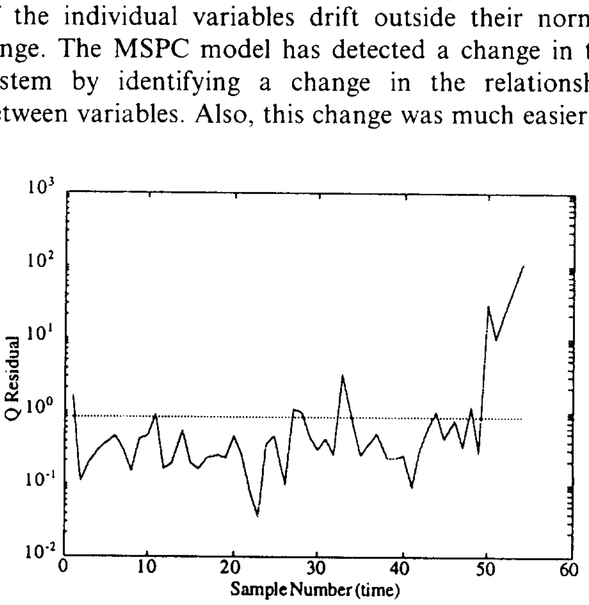

Figure 6 PCA Q residual and 95% limit showing a thermocouple failure at sample 50

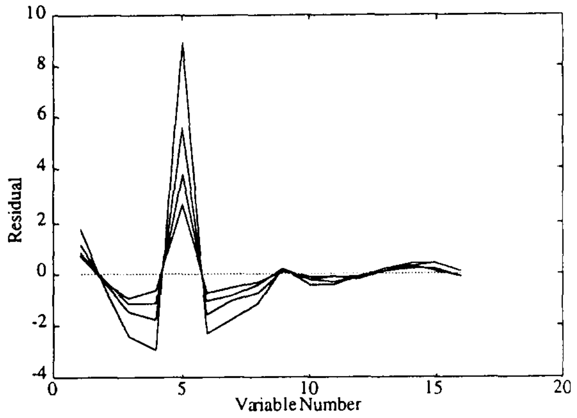

Figure 7 PCA residual on individual vairables identifying failure on thermocouple 5

identify by examining a single plot like *Figure 6* as opposed to several plots as shown in *Figure 3*.

Relationship between PCA and ARX models

PCA can also be used with processes that are not as rich in outputs as the previous example. Consider a linear, time-invariant, discrete, single-input single-output, auto-recursive, extensive variable process model:

$$y(k) = a_1y(k-1) + \ldots + a_ny(k-n) + b_1u(k-1) + \ldots + b_mu(k-m) \qquad (18)$$

Here we see that the process output y at time interval k is linearly related to the n past outputs and the m past inputs u. Imagine now forming the matrix of lagged process outputs and inputs such that $y(k)$ through $y(k-n)$ were used along with the last m values of the process inputs. Thus, form the matrix

$$\mathbf{x} = \begin{bmatrix} y(k) & y(k-1) \ldots & y(k-n) & u(k-1) \ldots & u(k-m) \\ y(k+1 & y(k) \quad \ldots & y(k-n+1) & u(k) \quad \ldots & u(k-m+1) \\ Y(k+2) & y(k+1) \ldots & y(k-n+2) & u(k+1) \ldots & u(k-m+2) \\ \vdots & \vdots \quad \vdots & \vdots & \vdots \quad \vdots & \vdots \\ y(r) & y(r-1) \ldots & y(r-n) & u(r-1) \ldots & u(r-m) \end{bmatrix} \qquad (19)$$

where $r = k + p$, i.e. there are $p + 1$ samples in the matrix. It is clear from the definition of the ARX model in Equation 18 that the first column of the matrix **X** is linearly related to the remaining columns. Thus, the matrix formed above must be rank deficient. If PCA were applied to the matrix, it would be determined that the data could be described using $n + m$ factors rather than $n + m + 1$. This, in fact, is the basis of a test to determine the correct order an ARX model identified from data.[18,19] It is easily seen how PCA could be used with such lagged variable matrices to detect changes in the input/output relationship of dynamic systems. Application of PCA to the lagged matrix has the effect of removing the major dynamics from the system, resulting in residuals that there are much better behaved (uncorrelated) than would otherwise be the case.

As an example of the use of PCA to monitor Single Input/Single Output (SISO) processes using lagged variables, consider the under damped processes defined by the coefficients given in *Table 2*. The step responses of these processes are shown in *Figure 8*. Both of these processes have unit gain and have ~10% noise added to the output.

A pseudo-random binary sequence was used as input to the original model and 300 samples were generated. A matrix of lagged variables was then formed consisting of the last five outputs and last four process inputs. PCA was used to model this matrix. As expected, a 7 PC model captured all of the deterministic variation in the data.

The PCA model was then applied to the modified model. The Q residual for the modified model data on the original is shown in *Figure 9*. Note that now over one-third of the samples have a Q value greater than the previously calculated 95% limit, indicating that a change in the system has occurred. If desired, a statistical test which takes the residuals of multiple samples into account could be used in order to increase the sensitivity of the method.

Multi-way PCA

The PCA method outlined above takes no explicit account of the ordered nature of the data set, i.e. the fact that the data was collected in a sequential manner. Reordering the samples would produce identical results. There are methods that explicitly consider that the data is ordered. These are referred to as multi-way methods because the data is usually organized into time ordered blocks that are each representative of a single sample or process run. The blocks are then arranged into multi-way matrices. Multi-way methods are particularly useful for the analysis of batch process data.

Consider the three-dimensional data array shown in *Figure 10*. A data matrix of this type would be typical of a series of runs of a batch process (or a series of samples from a second order analytical technique such as GC-MS). Here there are $j = 1, 2, \ldots, J$ variables measured at times $k = 1, 2, \ldots, K$ throughout the batch. Similar data will exist on $i = 1, 2, \ldots, I$ runs of the batch process. The data can be summarized in the three-dimensional ($I \times J \times K$) array **X**. Different batch runs (samples) are arranged along the vertical side, different process measurements (variables) along the horizontal

Table 2 ARX models of processes for lagged variable example

Original model
numerator (b) = 0.7796 0.9311 denominator (a) = 1 0.1509 0.5598 gain = 1, noise ~10%
Modified model
numerator (b) = 0.8435 1.0074 denominator (a) = 1 0.1509 0.7000 gain = 1, noise ~10%

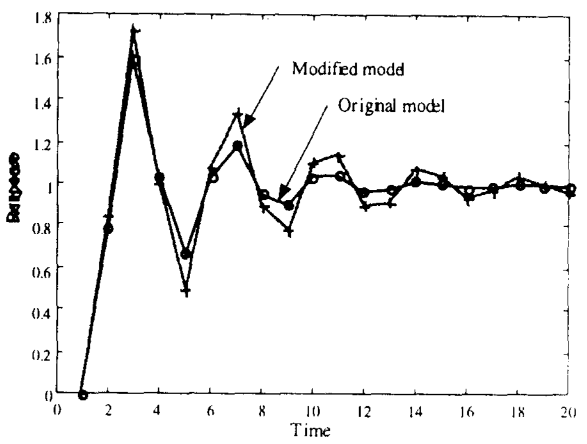

Figure 8 Step response of original and modified model for lagged variable example

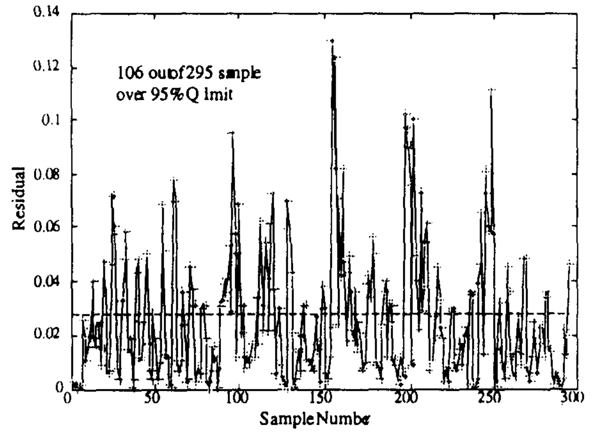

Figure 9 Q residuals for modified process on original PCA model indicating a change has occurred in the system

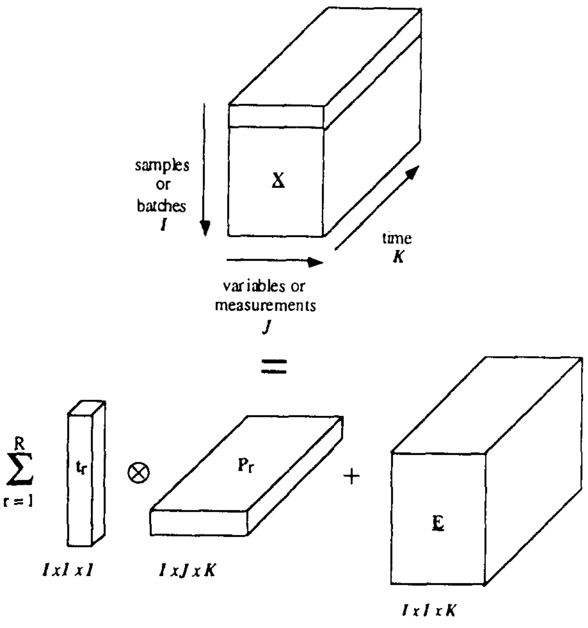

Figure 10 Three-dimensional data array and multi-way PCA decomposition

side, and time recedes into the figure. Each horizontal slice through the array is a $(J \times K)$ matrix representing the time history for all variables of a batch of a particular batch or sample. Each vertical slice made parallel to the front face of the cube is a $(I \times J)$ matrix representing the values of all the variables in all the batches taken at a common time. A vertical slice made parallel to the side of the cube (the time axis) would represent a $(I \times K)$ matrix of all the time histories of a single variable for all the batches.

There are several methods for decomposing the array.[20] These methods include the tri-linear decomposition (TLD),[21,22] parallel factor analysis (PARAFAC)[23,24] and Tucker models.[25] In this work, we will consider one of the more straightforward approaches, that of multi-way PCA (MPCA).[26] Each of the decomposition methods places different constraints on the resulting matrices and vectors.

MPCA is statistically and algorithmically consistent with PCA and has the same goals and benefits.[27] In MPCA the array **X** is decomposed as the summation of the product of score vectors (**t**) and loading matrices (**P**) plus a residual array **E** that is minimized in a least squares sense.

$$\mathbf{X} = \sum_{r=1}^{R} \mathbf{t}_r \otimes \mathbf{P}_r + \mathbf{E} \tag{20}$$

This decomposition is shown graphically in *Figure 10*. This decomposition is done in accordance with the principles of PCA and separates the data in an optimal way into two parts. The noise or residual part **E** is as small as possible and is associated with non-deterministic variation in the data. The systematic part, the sum of the $\mathbf{t}_r \otimes \mathbf{P}_r$, expresses the deterministic variation as one fraction (**t**) related only to batches and a second fraction (**P**) related to variables and their time variation.

MPCA is equivalent to performing PCA on a large two-dimensional matrix formed by unfolding the three-way array **X** in one of six possible ways, only three of which are mathematically unique. For example, one might unfold **X** in such a way as to put each of its vertical slices $(I \times J)$ side by side to the right, starting with the slice corresponding to the first time interval. The resulting two-dimensional matrix has dimensions $(I \times JK)$. This particular unfolding allows one to analyze variability among the batches in **X** by summarizing information in the data with respect to variables and their time variation. A mathematically equivalent unfolding would be to take slices off the side of **X** and place them down the time axis, which also forms a matrix with dimensions $(I \times JK)$. (The latter unfolding orders the matrix with the history of each variable kept together while the former orders the matrix with all the measurements taken at the same time kept together.) One might also be interested in unfolding **X** in other ways, however, the unfolding shown in *Figure 11* (and the equivalent unfolding mentioned above) are the only ways that keep batch (sample) specific information separate from time and variable information.

The MPCA algorithm proceeds as shown in *Figure 11*. First the matrix is unfolded in one of the two equiv-

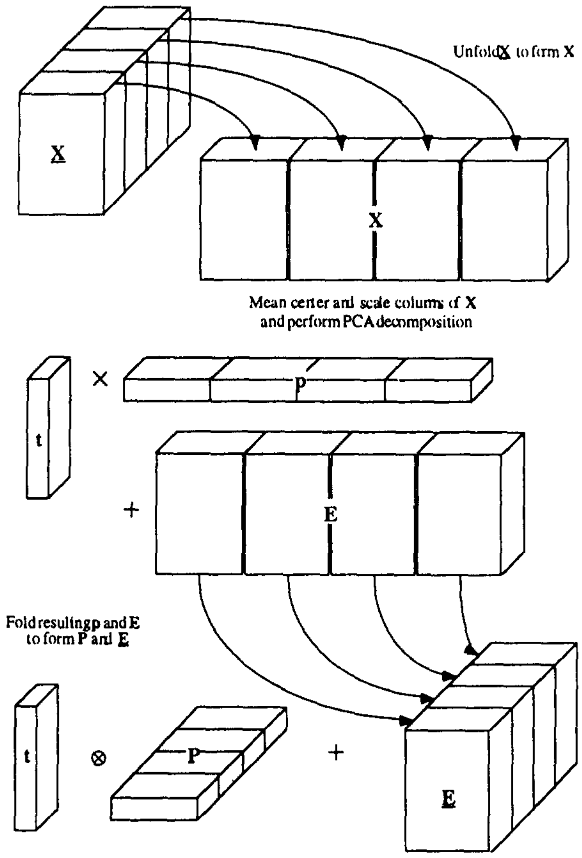

Figure 11 Algorithm for performing MPCA

alent ways described above. Each column of the resulting matrix is then mean centered and, if appropriate, scaled to unit variance (autoscaled). An eigenvector decomposition as described in Equations 1–3 is then applied to the unfolded **X**. Each of the **p**, however, is really an unfolded version of the loadings matrix $\mathbf{P}_r$. After the **p** are obtained, the $\mathbf{P}_r$ can be obtained by reversing the unfolding procedure. In a similar manner, the three array **E** can be formed by folding the PCA residual matrix **E**. The Q and T^2 statistics can be calculated using the unfolded solution as shown in Equations 5 and 6.

This version of MPCA explains variation of measured variables about their average trajectories. Subtracting the average trajectory from each variable (accomplished by mean centering the columns of the unfolded matrix **X**) removes the major nonlinear behaviour of the process. The i^{th} elements of the **t**-score vectors correspond to the i^{th} batch (sample) and summarize the overall variation in this batch with respect to the other batches in the database over the entire history of the batch. The **P** loading matrices summarize the time variation of the measured variables about their average trajectories. The elements of **P** are the weights, which when applied to each variable at each time interval within a batch, give the **t** scores for that batch.

Example of MPCA

As an example of MPCA consider the problem of monitoring a nuclear waste storage tank. A slurry of nuclear waste is stored in a large underground tank. Through processes such as radiolysis constituents in the waste are transformed into hydrogen, ammonia and nitrous oxide. To avoid build up of large volumes of gas bubbles in the waste a pump periodically agitates the slurry causing releases of small volumes of gas. This is referred to as the 'mitigation' process. Release gas, and a large volume of inlet air, exit through a filter system in an exit vent that contains a gas chromatograph used for measuring hydrogen concentration, and an FTIR used for measuring concentrations of ammonia and nitrous oxide. These gases are monitored as a function of time and, when referenced to pump initiation, yield a signature of the pump induced off-gassing. The pump operation time can be considered short compared to off-gas times. The process can be considered batch and is repeated several times per week.

Figure 12 shows mean signatures for the three measured gases where pump initiation occurs at time 10. These three concentration trajectories define the signature of the mitigation process. A three-way data matrix of signatures can be formed as in *Figure 8* by arranging each operation of the pump (a sample or batch) along the vertical axis, and each variable (gas concentration) along the horizontal azis with the time history receding into the figure. In our example we have 57 samples, three variables and consider 280 time increments. Thus, for this analysis our array is 57 × 3 × 280.

Results of the MPCA decomposition of the mean centered data are given in *Table 3*. The first eight PCs capture over 98% of the variation in the data. The remaining PCs (not shown) captured less than 0.5% of the variation each.

Loadings for the first principal component, which captures nearly 58% of the variance, are shown in *Figure 13*. Here the loadings matrix has been unfolded in order clarify the presentation. This PC corresponds to gross changes in signature magnitude relative to the mean signature. It is apparent that most of the change is at times immediately after pump operation, which would be expected. A large fraction of the variance not captured by this linear combination of signatures (over 40%) is related to variation in signature shapes. This

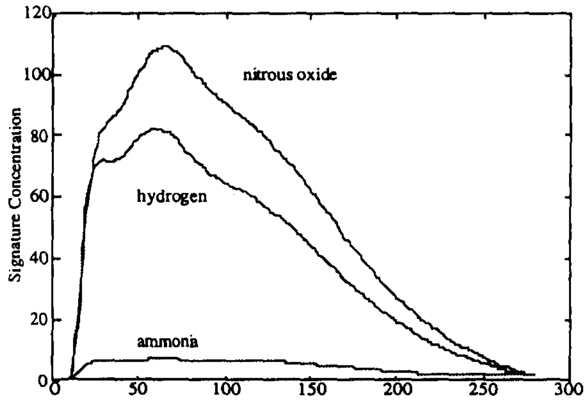

Figure 12 Mean signature for measured gases from mitigation process

Table 3 Variance captured by MPCA model of nuclear waste mitigation process data

PC number	*Percent variance captured* This PC	Total
1	57.67	57.67
2	21.69	79.36
3	7.91	87.27
4	5.06	92.33
5	2.80	95.13
6	1.90	97.03
7	0.87	97.90
8	0.54	98.44
9	0.46	98.90
10	0.20	99.10

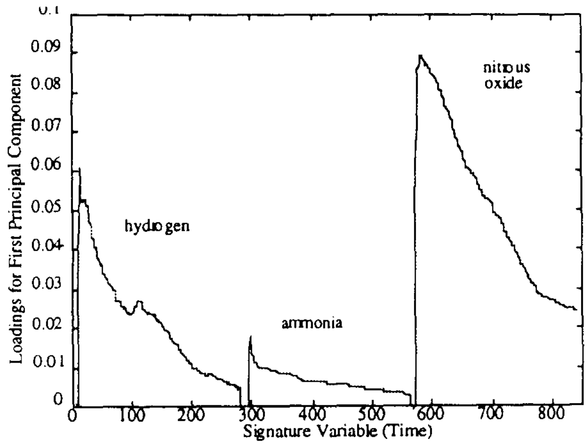

Figure 13 Loadings for first principal component from MPCA analysis of mitigation process data

wide variety of signature shapes makes identification of subtle trends in signature magnitudes difficult to identify when examining individual signatures from each pump operation. However, examination of the scores for the first principal component, which are shown in *Figure 14* as a function of pump operation date, indicates that signature magnitude decreases slightly with date. MPCA in this example improves the monitoring capability of the nuclear waste mitigation process.

Other factor analysis techniques

While PCA is perhaps the most commonly used factor analysis technique, it is not the only one. In the sections that follow we consider two other techniques that are useful for extracting information like data produced by dynamic systems. Like MPCA, both of these techniques consider the ordered nature of the data set explicitly.

Evolving factor analysis

Evolving factor analysis (EFA) is a general technique for analysis of multivariate data that has an intrinsic order.[28] Like PCA, EFA can be used to determine how many 'factors' are present in a data set, i.e. how many independent sources of variation exist. In addition, EFA can be used to determine where in a data set the factors first appear and where they disappear.

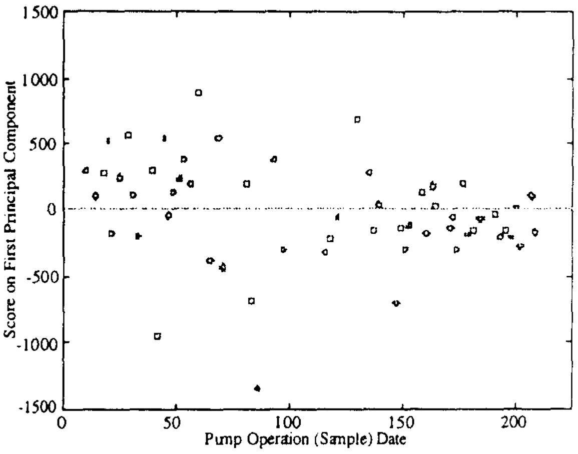

Figure 14 Scores on first principal component from MPCA analysis of mitigation process data

The fundamental idea of EFA is to follow the singular values of a data matrix as rows (samples) are added. Given a data matrix **X** (usually not mean-centered when performing EFA) with m samples and n variables one starts by determining the singular value of the first row (sample), then the first two rows, then the first three and so on until all m samples have been considered. Since the number of samples considered, and thus number of factors, evolves as the method proceeds, the technique is known as EFA. Of course, the number of non-zero singular values can never be greater than the lesser of the number of rows and columns in the sub-matrix considered. It is also customary to do EFA in the reverse direction, i.e. determine the singular values starting from the last row in the matrix and working upwards until all rows have been considered.

Once the singular values have been determined, they (or the logarithms, since they usually span many orders of magnitude) can be plotted versus the ordered variable, e.g. time. As new factors enter the data new significant singular values will rise from the baseline of small singular values associated with noise in the data. Working in the reverse direction, significant singular values will rise from the baseline when factors disappear from the data matrix.

As an example of EFA, consider a batch chemical process that is being monitored by an NIR spectrometer. Suppose that there are three analytes of interest in the system, and that their concentration profiles (as a function of time) are as shown in the upper half of *Figure 15*. For this example, the first analyte to appear can be considered a reactant, the second a desired product, and the third an unwanted byproduct of a side reaction. Furthermore, suppose that the pure component spectra for each analyte is as shown in the lower half of *Figure 15*. Now imagine that neither the concentration profiles or the pure component spectra are known *a priori*. Instead, the available data appear as in *Figure 16*, which shows the measured spectra as a function of time.

Figure 17 shows the results of performing EFA on this data. From the forward moving singular values (the solid lines in the figure), it is apparent that no (spectrally active) analytes are present initially. The first one

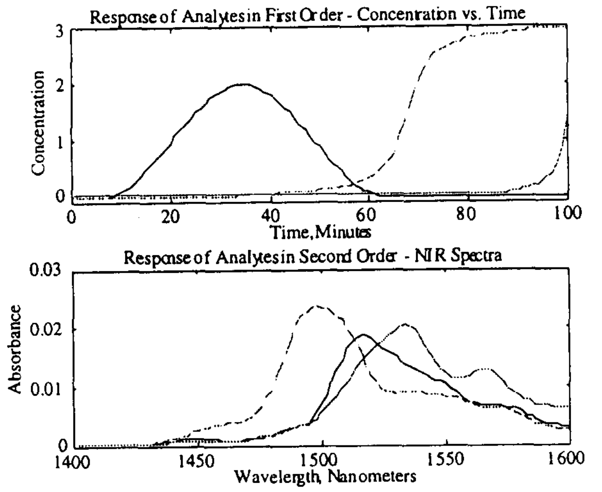

Figure 15 Concentration profiles and pure component spectra for example batch process.

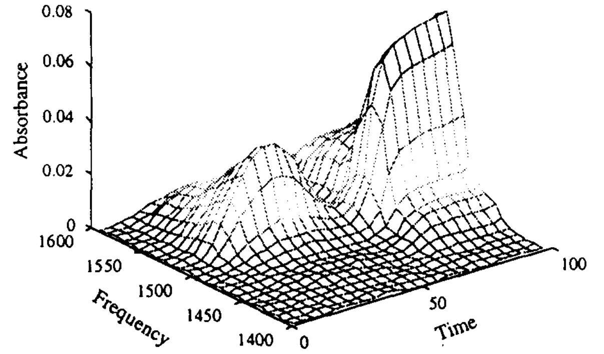

Figure 16 Measured response of spectra versus time for example batch process

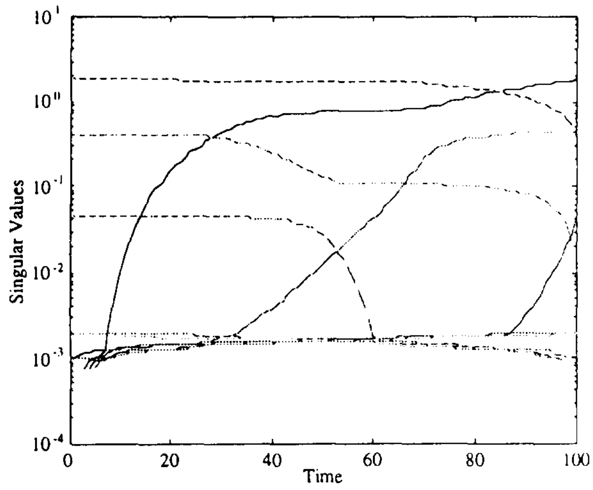

Figure 17 Evolving factor analysis of example batch process data showing forwards (solid) and backwards singular values (dashed)

appears at sample 5. A new analyte appears at sample 30, where the singular value goes from the background level to a significant value. No other analytes appear until after sample 80. Working from the backwards singular values (the dashed lines), it is apparent that there were two analytes present at the end of the data set since there are two significant singular values. Working from right to left, it is apparent that another singular value becomes non-zero at sample 60. This indicates the addition of analyte when moving backwards in time, which corresponds to the disappearance of an analyte when moving forward in time.

Working from the EFA curves with the assumption that the first analyte to appear was also the first to disappear, we would conclude that the analyte which appeared at sample 5 disappeared at about sample 60. Additional analytes appeared at sample 30 and sample 80 and remained until the end of the run. Thus, it would be determined from the analysis that some unwanted analyte appeared near the end of the batch.

EFA provides important information concerning the times of existence of particular factors in the data set. This may be used to identify the onset of unusual conditions such as process disturbances, the failure of process instruments, or, as in our example, the production of an unwanted byproduct.

Multivariate curve resolution

EFA gives us important information regarding the system: the total number of factors present and their times of appearance and disappearance. However, it would be impossible to identify the compounds based on this information alone (unless one was very familiar with the system at hand). However, the results of EFA do provide a starting point for the determination of the concentration profiles and pure component spectra. Techniques for obtaining this additional information are known as self modeling curve resolution (SMCR)[29] or multivariate curve resolution (MCR)[30]

Simply put, the goal of MCR is to extract the number of analytes, concentration profiles and pure component spectra with as few assumptions about the data as possible. Restated mathematically, given a data matrix **X** (not mean centered) that is the product of concentration profiles **C** and pure component spectra **A**.

$$\mathbf{X} = \mathbf{CA} \tag{21}$$

we wish to obtain physically meaningful **C** and **A**. Obviously, this cannot be obtained directly from a principal components decomposition of **X** without additional information because the equation:

$$\hat{\mathbf{X}} = \mathbf{TP}^{\mathrm{T}} = \mathbf{TRR}^{-1}\mathbf{P}^{\mathrm{T}} = \mathbf{CA} \tag{22}$$

where

$$\mathbf{X} = \hat{\mathbf{X}} + \mathbf{E} = \mathbf{TP}^{\mathrm{T}} + \mathbf{E} \tag{23}$$

has an infinite number of solutions for any arbitrary transformation matrix **R**. This results in a rotational and intensity ambiguity for **C** and **A** if no other information is known.

Fortunately, more information is typically available. In particular, both concentrations and spectra are necessarily non-negative. Using this constraint, it is possible to extract the pure component spectra through a procedure of alternating and constrained least squares optimization. Starting with the results of the EFA, it is possible to obtain an initial estimate of the concentration profiles, or at least, the range of existence in the data set of each component. These concentration profiles are used as the initial estimates in the constrained and alternating least squares optimization. At each iteration of the optimization, a new estimate of the spectra matrix **A** and of the concentration profiles **C** is obtained. Iteration between the following two equations is performed:

$$\mathbf{A} = \mathbf{C}^{+}\hat{\mathbf{X}} \tag{24}$$

$$\mathbf{C} = \hat{\mathbf{X}}\mathbf{A}^{+} \tag{25}$$

where $\mathbf{C}^{+}$ and $\mathbf{A}^{+}$ are the pseudo inverses (see Equation 30 below) of the matrices **C** and **A** respectively. At each iteration, any negative elements of **C** and **A** are reset to zero, as are any elements known to be zero from inspection of the EFA results and application of any known properties of the system. Obviously, selection of the correct number of components in the calculation of $\hat{\mathbf{X}}$ is important. Use of this matrix instead of the original data **X** improves the stability of the calculations $\hat{\mathbf{X}}$ is a noise filtered estimate of **X**.

It should be noted that there will be an intensity ambiguity in the results of the MCR, i.e. an ambiguity in the absolute magnitude of the concentration profiles and pure component spectra obtained since, for any single pure component matrix $\mathbf{X}_{p}$

$$\mathbf{X}_{p} = \mathbf{c}\,\mathbf{a} = \alpha\mathbf{c}\frac{1}{\alpha}\mathbf{a} \tag{26}$$

for any arbitrary constant α. If, however, the concentration of a component is known at any point within the data, it is possible to resolve the ambiguity for that particular component.

If additional information is known about the problem, it is possible to incorporate it into the iteration procedure as a constraint. For instance, it is often assumed that the concentration profiles are unimodal. In other cases, closure or some known stoichiometry may be used. When additional known (correct) constraints are used, the quality of the solution obtained generally improves. It is also possible to combine data sets from different runs of the process under consideration simply by forming the augmented matrices as follows:

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_{1} \\ \vdots \\ \mathbf{X}_{\mathrm{NP}} \end{bmatrix} = \begin{bmatrix} \mathbf{C}_{1} \\ \vdots \\ \mathbf{C}_{\mathrm{NP}} \end{bmatrix} \mathbf{A} = \mathbf{C}\mathbf{A} \tag{27}$$

where $\mathbf{X}_{1}$ to $\mathbf{X}_{\mathrm{NP}}$ are the ordered data from NP individual process runs and $\mathbf{C}_{1}$ to $\mathbf{C}_{\mathrm{NP}}$ are the associated concentration profiles from the runs. Combining data from different runs may also contribute to the stability of the solutions obtained.

As an example of MCR, we now return to the data used in the EFA example. Recall that we had determined the range of existence of each of the three components present in the data. Thus, as the initial concentration profiles, we will use the forward singular values from the EFA with the first singular value re-set to zero after sample 60, where we assume (from consideration of the backward EFA) the associated component disappeared.

The resulting spectral estimates and concentration profiles are compared with the known values in *Figure 18*. In this problem the intensity ambiguity exists so the concentration profiles and spectra have been scaled for comparison to the actual values. Note that the spectra and concentration profiles have been recovered almost exactly from the original data, with the exception of a small error on the estimate of the spectra of the 'unwanted byproduct' analyte present in the lowest concentration. Using MCR, we have now detected the presence of the byproduct and obtained an estimate of its pure component spectra. This gives us the opportunity to search the spectra library for the true identity of the compound.

To the author's knowledge, no applications of MCR have been published where fault detection and real-time process monitoring was the goal. The existing literature consists of studies that were done 'after the fact' when it was known that contaminant species had developed in process runs and their identity was not known. However, it is easy to see how MCR and EFA might be implemented on-line in a batch process. Such a technique would allow one to detect and fingerprint the unwanted analytes in a reacting system.

Multivariate regression modeling

In this section we discuss several regression methods that have found utility for process monitoring and fault detection. As we shall see, these methods can be used in a number of different ways depending upon the applications.

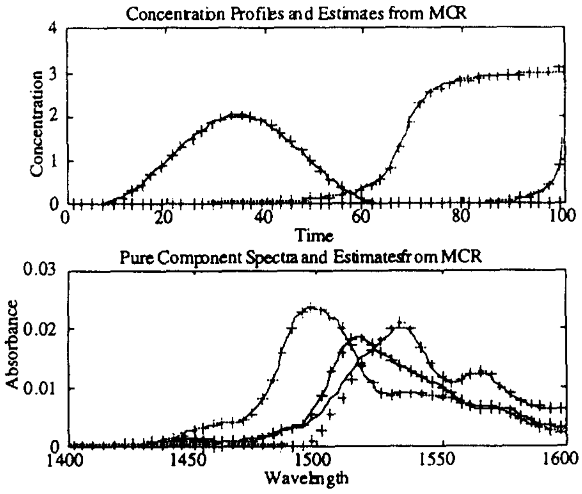

Figure 18 Comparison of actual concentration profiles and pure components spectra and estimates (+s) from MCR

Inverse Least Squares

Inverse Least Squares (ILS), also known as Multiple Linear Regression (MLR) or Ordinary Least Squares (OLS) can be useful for predicting properties of a system based on variables which are only indirectly related to the property.[31] ILS assumes that a regression vector **b** can be used to determine a property of the system y from the measured variables **x** (a row vector such as a spectra or collection of process variables). Thus, the ILS model is

$$\mathbf{x}\,\mathbf{b} = y \tag{28}$$

The regression vector **b** must be determined using a collection of measurements **x** and the known values of the property of interest, y. Thus, **b** is estimated from

$$\mathbf{b} = \mathbf{X}^{+}y \tag{29}$$

where $\mathbf{X}^{+}$ is the pseudo inverse of **X**. There are many ways to determine a pseudo inverse, but perhaps the most obvious is Multiple Linear Regression (MLR, also known as ordinary least squares). In this case, $\mathbf{X}^{+}$ is defined by

$$\mathbf{X}^{+} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}} \tag{30}$$

Unfortunately, this approach often fails in practice because of the collinearity of **X**, e.g. some columns of **X** (variables) are linear combinations of other columns, or because **X** contains fewer samples than variables (few rows than columns). For example, the spectroscopy calibration problem is extremely ill-conditioned due to a high degree of correlation between absorbances at nearby wavelengths. Similarly, in applications where one would like to predict product properties based on process variables, the process variables are often highly correlated. It is also typical that there are fewer samples available than the number of wavelengths considered.

When Equation 30 is used with systems that produce nearly collinear data, the solution for **b** is unstable, i.e. small perturbations in the original data, possibly due to noise or experimental error, cause the method to produce wildly different results. While the calibrations may fit the data, they are typically not useful for predicting properties of new samples.

Principal Components Regression

Principal Components Regression (PCR) is one way to deal with the problem of ill-conditioned matrices.[32] Instead of regressing the system properties (e.g. concentrations) on the original measured variables (e.g. spectra), the properties are regressed on the principal component scores of the measured variables (which are orthogonal and, therefore, well conditioned). Thus, $\mathbf{X}^{+}$ is estimated as

$$\mathbf{X}^{+} = \mathbf{P}\,(\mathbf{T}^{\mathrm{T}}\mathbf{T})^{-1}\,\mathbf{T}^{\mathrm{T}} \tag{31}$$

As in PCA, the number of principal components to retain in the model must be determined. In our case, the purpose of the regression model is to predict properties of interest for new samples. Thus, we would like to determine the number of PCs that optimizes the predictive ability of the PCR model. This is typically done by cross-validation, a procedure where the available data is split between training and test sets. The prediction residual error on the test samples is determined as a function of the number of PCs retained in the regression model formed with the test data. The procedure is usually repeated several times, with each sample in the original data set being part of the test set at least once. The total prediction error over all test sets as a function of the number of PCs is then used to determine the optimum number of PCs, i.e. the number of PCs which produces minimum prediction error. If all of the PCs are retained in the model, the result is identical to that for MLR (at least in the case of more samples than variables). In some sense, it can be seen that the PCR model 'converges' to the MLR model as PCs are added.

Partial Least Squares

Partial Least Squares (PLS) regression[33–36] is related to both PCR and MLR and can be thought of occupying a middle ground between them. PCR finds factors that capture the greatest amount of variance in the predictor variables, e.g. spectra or the temperatures in the SFCM example. MLR seeks to find a single factor that best correlates predictor variables with predicted variables, e.g. concentrations or level. PLS attempts to find factors which do both, i.e. capture variance and achieve correlation. We commonly say that PLS attempts to maximize covariance.

In PLS, the scaled and mean-centered **X** and **Y** matrices are decomposed as:

$$\mathbf{X} = \mathbf{T}\mathbf{P}^{\mathrm{T}} + \mathbf{E} = \sum_{a=1}^{A} \mathbf{t}_a\mathbf{p}_a^{\mathrm{T}} + \mathbf{E} \tag{32}$$

$$\mathbf{Y} = \mathbf{U}\mathbf{Q}^{\mathrm{T}} + \mathbf{F} = \sum_{a=1}^{A} \mathbf{u}_a\mathbf{q}_a^{\mathrm{T}} + \mathbf{F} \tag{33}$$

Note that here we are allowing for multivariate **Y**. One of the unique features of PLS is that it can be used to form models relating more than one predicted variable to many predictor variables. The latent vectors $\mathbf{t}_a$ are computed from the data for each PLS dimension ($a = 1, 2, \ldots, A$) such that the linear combination of the **X** variables defined by the latent variable $\mathbf{t}_a = \mathbf{w}_a^{\mathrm{T}}\mathbf{x}$ and the linear combination of the **Y** variables defined by the latent variable $\mathbf{u}_a = \mathbf{q}_a^{\mathrm{T}}\mathbf{y}$ maximizes the covariance between **X** and **Y** that is explained at each dimension, i.e. as latent variables are added.

There are several ways to calculate PLS model parameters (see for example the work of de Jong 1993[37]), however, perhaps the most instructive method is known as NIPALS for Non-iterative Partial Least Squares.

NIPALS calculates scores and loadings (similar to those used in PCR) and an additional set of vectors known as weights, **W**. The addition of weights is required to maintain orthogonal scores. Unlike PCR and MLR, the NIPALS algorithm for PLS also works when there is more than one predicted variable, **Y**, and therefore scores **U** and loadings **Q** are also calculated for the **Y**-block. A vector of 'inner-relationship' coefficients, **b**, which relates the **X**- and **Y**-block scores, must also be calculated. Using NIPALS the scores, weights, loadings and inner-coefficients are calculated sequentially as shown below.

The PLS decomposition is started by selecting one column of **Y**, $\mathbf{y}_j$, as the starting estimate for $\mathbf{u}_1$ (usually the column of **Y** with greatest variance is chosen). Of course in the case of univariate y, $\mathbf{u}_1 = \mathbf{y}$. Starting in the **X** data block:

$$\mathbf{w}_1 = \frac{\mathbf{X}^T\mathbf{u}_1}{\|\mathbf{X}^T\mathbf{u}_1\|} \tag{34}$$

$$\mathbf{t}_1 = \mathbf{X}\mathbf{w}_1 \tag{35}$$

In the y data:

$$\mathbf{q}_1 = \frac{\mathbf{u}_1^T\mathbf{t}_1}{\|\mathbf{u}_1^T\mathbf{t}_1\|} \tag{36}$$

$$\mathbf{u}_1 = \mathbf{Y}\mathbf{q}_1 \tag{37}$$

Check for convergence by comparing $\mathbf{t}_1$ in Equation 35 with the one from the previous iteration. If they are equal within rounding error, proceed to Equation 38. If they are not return to Equation 34 and use the $\mathbf{u}_1$ from Equation 37. If the **Y**-block is univariate, Equations 36 and 37 can be omitted, set $\mathbf{q}_1 = 1$, and no iteration is required.

Calculate the **X** data block loadings and rescale the scores and weight accordingly:

$$\mathbf{p}_1 = \frac{\mathbf{X}^T\mathbf{t}_1}{\|\mathbf{t}_1^T\mathbf{t}_1\|} \tag{38}$$

$$\mathbf{p}_{1\mathrm{new}} = \frac{\mathbf{p}_{1\mathrm{old}}}{\|\mathbf{p}_{1\mathrm{old}}\|} \tag{39}$$

$$\mathbf{t}_{1\mathrm{new}} = \mathbf{t}_{1\mathrm{old}}\|\mathbf{p}_{1\mathrm{old}}\| \tag{40}$$

$$\mathbf{w}_{1\mathrm{new}} = \mathbf{w}_{1\mathrm{old}}\|\mathbf{p}_{1\mathrm{old}}\| \tag{41}$$

Find the regression coefficient *b* for the inner relation:

$$b_1 = \frac{\mathbf{u}_1^T\mathbf{t}_1}{\mathbf{t}_1^T\mathbf{t}_1} \tag{42}$$

After the scores and loadings have been calculated for the first factor (commonly called a latent variable [LV] in PLS), the **X**- and **Y**-block residuals are calculated as follows:

$$\mathbf{E}_1 = \mathbf{X} - \mathbf{t}_1\mathbf{p}_1^T \tag{43}$$

$$\mathbf{F}_1 = \mathbf{Y} - b_1\mathbf{u}_1\mathbf{q}_1^T \tag{44}$$

The entire procedure is now repeated for the next latent variable starting from Equation 34. **X** and **Y** are replaced with there residuals $\mathbf{E}_1$ and $\mathbf{F}_1$, respectively, and all subscripts are incremented by 1.

It can be shown that PLS forms the matrix inverse defined by:

$$\mathbf{X}^+ = \mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}(\mathbf{T}^T\mathbf{T})^{-1}\,\mathbf{T}^T \tag{45}$$

where the **W**, **P** and **T** are as calculated above. Note that the scores and loadings calculated in PLS are not the same as those calculated in PCA and PCR. They can be thought of, however, as PCA scores and loadings that have been rotated to be more relevant for predicting y. Also, as in PCR, the PLS model converges to the MLR solution if all latent variables are included.

If the PLS algorithm does not appear transparent, do not despair. The important thing to remember is that PLS attempts to find factors or LVs which are correlated with **Y** while describing a large amount of the variation in **X**. This is in contrast to PCR, where the factors (in this case PCs) are selected solely on the amount of variation they explain in **X**. It is for this reason that PLS has found great utility in process monitoring applications.

An application of PCR and PLS

As noted in the introduction to this section, the possible applications of 'calibration' methods go beyond the spectral calibration problem. Therefore, as an example of the application of PCR and PLS, we will now develop a regression model that relates the temperatures measured in the SFCM (which take the place of spectra in the calibration problem) to the level of the molten glass (which takes the place of concentration). The glass level for the period of temperatures shown in *Figure 2* is shown in *Figure 19*. As noted previously, the measured temperatures rise and fall with the glass level, which is a controlled variable. A model relating temperatures to level would serve as a backup and/or fault detection system for the pneumatic level measuring instrument.

In our example, there are 300 samples available for the calibration. (We will also use all 20 of the available temperature measurements, rather than just the subset of 16 used in the PCA example.) We must first decide how to split the available data for cross-validation to determine the optimum number of PCs to retain in the PCR, or similarly, LVs in the PLS model. When many training samples are available, a good rule of thumb is to use the square root of the number of samples for each test set and the number of test sets. Given 300

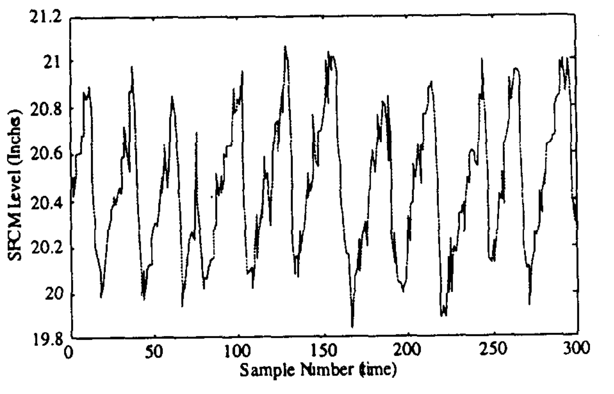

Figure 19 SFCM molten glass level

samples this would lead to 17.32 samples per test set and 17.32 test sets. In our case, it is more convenient to use 10 training sets of 30 samples each.

We must also decide how to select the test sets from the data. One possibility would be to form each test set using every 10th point of the data, each subset starting from sample 1, 2, 3, etc. When dealing with data from a dynamic system, however, this is usually not a good choice because noise in the system is usually not completely random. Instead, noise is generally serially correlated, which means that the amount of variation due to noise on the n^{th} point is correlated to the amount of noise on the $n+1^{th}$ point. Thus, if the test sets were chosen using every 10th point, noise on the test data would be highly correlated with noise in the training data, and the result would be a model that fit much of the system noise. Instead, when dealing with time series data produced by a dynamic system, it is better to choose contiguous blocks of data for the test sets. Thus, the first test set will be samples 1–30, the second will be samples 31–60 and so on.

Figures 20 and *21* show results of the cross-validation procedure on the temperature/level data for the PCR and PLS models, respectively. Note how the prediction error or PRESS (prediction residual error sum of squares) is a minimum at 11 PCs for the PCR model. We might choose 11 PCs for construction of the final model using all 300 training samples, however, experience has shown that in cases like this a better choice would be to select the first minimum with PRESS close to the global minimum. Thus, we would choose six PCs for the model. Note, however, that some PCs do not contribute to the prediction error in a positive way. When PCs 2, 4 and 5 are added, the PRESS actually gets worse. This suggests that these factors are not relevant for prediction of the level.

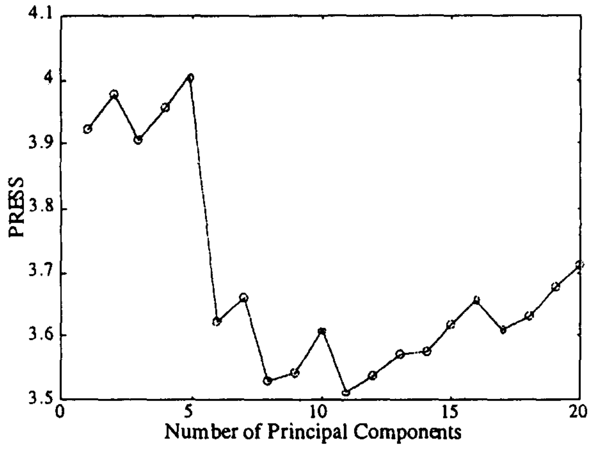

Figure 20 Prediction error (PRESS) as a function of number of PCs in PCR model for SFCM data

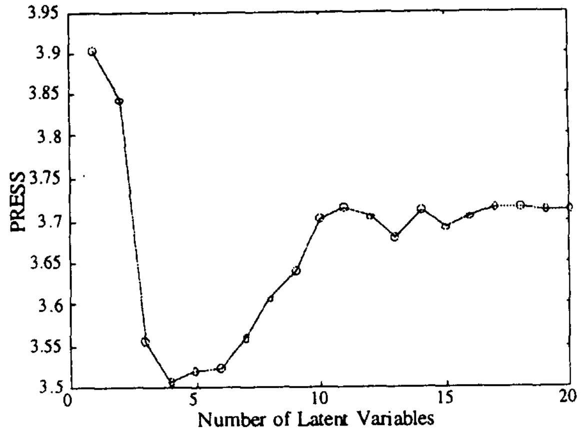

Figure 21 Prediction error (PRESS) as a function of number of LVs in PLS model for SFCM data

Figure 21 shows the PRESS versus number of LVs for the PLS model. Note how the PRESS goes through a minimum at five LVs. as opposed to the 11 PCs used in the PCR model, and does not have many local minima. This is typical of the behavior of PLS relative to PCR. Because PLS attempts to find factors that describe variance (like PCR) and achieve correlation (like MLR), PLS models tend to 'converge' to the MLR solution faster than PCR models. For the same number of factors, they also tend to capture more of the relevant information in **X**, which results in minimum PRESS at a smaller number of factors and with fewer irrelevant factors. In practice, we would choose four LVs for the model because the change in the PRESS from four to five LVs does not appear significant (it is less than 2%).

As mentioned above, there are often factors in PCR models which do not contribute positively to the predictive ability of the model, e.g. factors 2, 4, 5, 7 and 10 in our PCR example, where the PRESS goes up (see *Figure 20*). This happens in PCR because the factors (the PCs) are chosen without consideration of how they relate to the predicted variable. This happens much less frequently in PLS, because the factors (the latent variables) are chosen with regard to how correlated the factor scores are to the predicted variable. It is possible to form the PCR model leaving out these factors, simply by deleting the appropriate columns from the **P** and **T** matrices in Equation 31.

We can compare the regression vectors calculated by PLS and PCR with the regression vector calculated by MLR, as shown in *Figure 22*. It can be seen that the MLR regression coefficients vary considerably switching from positive to negative and back several times. This 'ringing' in the coefficients is typical of models produced by MLR when the problem is ill-conditioned.

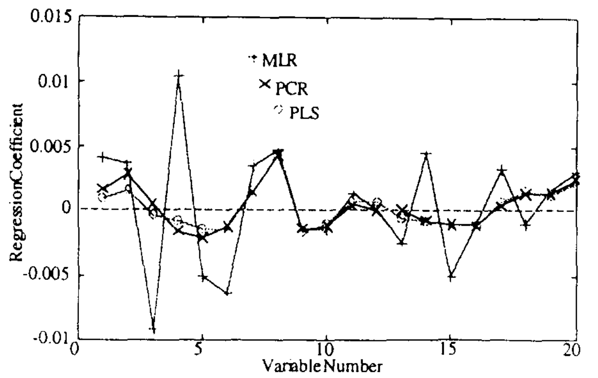

Figure 22 Comparison of regression coefficients from MLR, PCR and PLS models for SFCM level

Note how the PLS and PCR models are much smoother. In fact, the PLS and PCR models make more 'sense' from a physical standpoint. Because of the close proximity of the temperature sensors in this system, we would not expect the relationship between adjacent sensors and the level to be of opposite signs, as they are in the MLR model.

Ridge regression

Ridge regression is another technique for dealing with ill-conditioned data. Ridge regression gets its name because a constant is added to the 'ridge' of the covariance matrix in the process of forming the pseudo inverse:

$$X^{-} = (X^{T}X + I\Theta)^{-1}X^{T} \quad (46)$$

The addition of the constant Θ to the ridge has the effect of stabilizing the values of the coefficients determined from the regression. It also has the effect of shrinking the coefficients. It has been shown[38] that the regression vector calculated from ridge regression is the least squares solution subject to the constraint that the vector is confined to a sphere centered around the origin. Thus, in some sense, RR assumes that the regression coefficients are more likely to be small (near zero) than large. Note that it is also possible to impose a non-spherical constraint on the regression vector by using a specific diagonal matrix instead of the identity matrix in Equation 46. This is often useful when one knows that certain coefficients are more likely to be close to zero than others. This is the case with certain types of dynamic models, as will be shown below.

Ridge regression is also used quite often when one is concerned with the values of the regression coefficients themselves, rather than in prediction. If one can derive needed information from the coefficients themselves, RR may be an appropriate choice. The trick in ridge regression is to determine the optimum value of Θ for developing a predictive model. A common statistical method was outlined by Hoerl *et al.*[39] It is also possible to determine the optimum value through cross-validation.

As an example of the use of ridge regression, we will now revisit the SFCM data used in the examples of PCR and PLS. The regression coefficients are shown as a function of the ridge parameter in *Figure 23*. Note how the values shrink as the value of Θ is increased. The value of the ridge parameter using the method of Hoerl and Kennard is 0.0074, while the value determined using the cross-validation procedure outlined above is 0.0076. The regression coefficients are compared with th ose determined from MLR in *Figure 24*. Note how the coefficients have been shrunk in RR as compared to MLR.

Comparison of linear models on SFCM example data

Of course, the real reason for forming the regression models is to make predictions of the SFCM level given the temperatures in the event of a failure of the level instrument or as a check on the performance of the instrument. Thus, the best test of the models is their ability to predict the level given new temperature measurements.

Before comparing the predictive ability of the models, it is useful to introduce several measures of a model's fit to the data and predictive power. In all of the measures considered, we are attempting to estimate the 'average' deviation of the model from the data. The root-mean-square error of calibration (RMSEC) tells us about the fit of the model to the calibration data. It is defined as:

$$\mathrm{RMSEC} = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}} \quad (47)$$

where the $\hat{y}_i$ are the values of the predicted variable when all samples are included in the model formation and n is the number of calibration samples. RMSEC is a measure of how well the model fits the data.

This is in contrast to the root-mean-square error of cross-validation (RMSECV) which is a measure of a model's ability to predict new samples. The RMSECV is defined as in Equation 47, except the $\hat{y}_i$ are prediction for samples not included in the model formulation. RMSECV is related to the PRESS value for the number

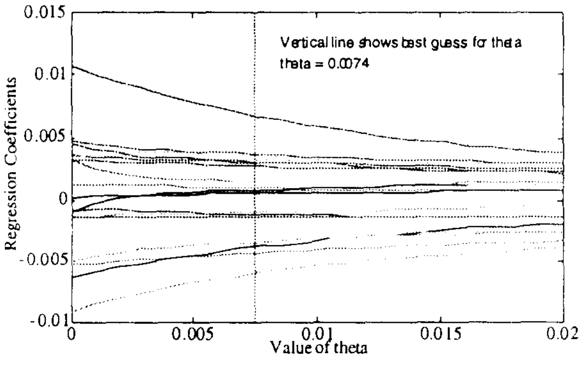

Figure 23 Value of regression coefficients as a function of the ridge parameter Θ

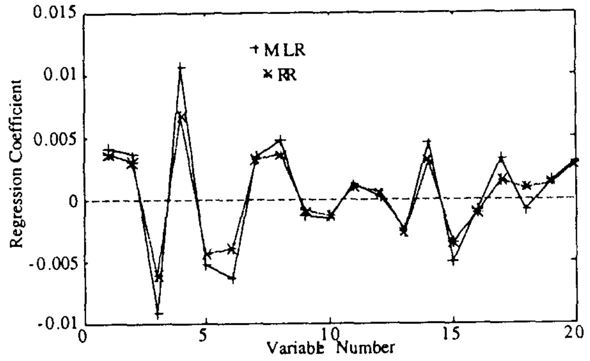

Figure 24 Regression coefficients from MLR and RR models on SFCM example

of PCs or LVs included in the model, i.e.

$$\text{RMSECV}_k = \sqrt{\frac{\text{PRESS}_k}{n}} \tag{48}$$

where PRESS_k is the sum of squares prediction error for the model which includes k factors. Of course, the exact value of RMSECV depends not only on k, but on how the test sets were formed. It is also common to calculate PRESS and thus RMSECV, for a leave one out cross-validation. i.e. where each sample is left out of the model formulation and predicted once.

It is also possible to calculate a root-mean-square error of prediction (RMSEP) when the model is applied to new data provided that the reference values for the new data are known. RMSEP is calculated exactly as in Equation 38 except that the estimates $\hat{y}_i$ are based on a previously developed model, not one in which the samples to be 'predicted' are included in the model building.

The MLR, PCR, PLS and RR models developed for the SFCM example can now be compared using the measures just introduced. This information is summarized in *Table 4*. A new data set of 200 samples was used for the calculation of RMSEP. The actual level is compared with the prediction form PLS in *Figure 25*.

Note how the MLR model shows the smallest value for RMSEC, the error of calibration. This indicates that, as expected, the MLR model 'fits' the data best.

Note, however, from the RMSECV that the MLR does not predict the best, even for samples within the original calibration data. Here the 'best' model is PLS. This is also true for the RMSECV for leave-one-out cross-validation (RMSECV-LOO). Note that the numbers for RMSECV-LOO are all lower than those for RMSECV. This is expected, because the ability to predict improves as the number of samples included in the model formulation is increased.

Table 4 Comparison of MLR, PCr, PLS and RR models used for SFCM example

	MLR	PCR	PLS	RR
RMSEC	.0991	.1059	.1034	.0996
RMSECV	.1122	.1108	.1098	.1046
RMSECV -LOO	.1062	.1079	.1045	.1067
RMSEP	.1496	.1366	.1396	.1471

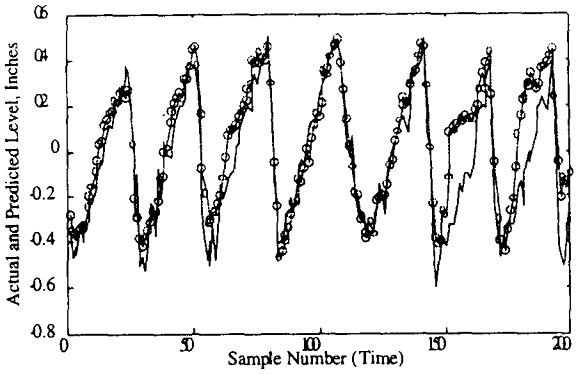

Figure 25 Comparison of actual SFCM level and predictions from PLS model

For the prediction on new samples the PCR model outperforms all other models. The RMSEP for the MLR model is ~10% higher than the prediction error for the PCR model. The PLS model prediction error is ~2% higher than the PCR model, while the RR model RMSEP is ~8% higher. While these differences are small, they illustrate the point that when the data has a great deal of redundancy between variables, methods that take advantage of the structure generally outperform those that don't.

It should be evident from the comparison of models that fit and prediction are entirely different aspects of a model's performance. If prediction is the goal, a model should be built with this criteria in mind. The calculation of PRESS from a cross-validation procedure is one way to measure a collection of model's predictive abilities. Also, the robustness of the models in the face of minor changes to the system must also be considered. PLS and PCR models are known to be relatively robust compared to MLR.

Using the PLS predictions, scores and residuals for process monitoring

Up to this point the predictive aspects of PLS models have been emphasized. However, this is not the only way in which PLS models may be applied. By virtue of its construction, the PLS model describes the variation in **X** that is predictive for **Y**. Thus, it is logical to use the PLS weights in a manner analogous to use of the loadings in PCA, i.e. to determine if the structure of the **X** data has changed in a significant way. All of the statistical tests used for PCA may, in fact, be used with little modification for PLS. For example, one many monitor the PLS **X** block scores. This scheme is particularly useful as it monitors the variations in **X** which are known to be related to the properties to be predicted, **Y**. Residual Q and T^2 can also be developed, as will be shown.

Going back to our SFCM example, we can now use the PLS model for predicting level to detect a failure of the level sensor. Using the difference between the actual and predicted levels from the calibration data it is possible to set a limit on the expected size of this residual in practice. Review of the residuals showed them to be approximately normally distributed (not shown) and so a limit of 2.33 standard deviations of the prediction error was used to establish a 99% limit. *Figure 26* shows the detection of a simulated failure. Here the residuals from the level prediction are monitored. A ramped bias is added starting at sample 20 which goes from 0. to 0.6 inches over the range. Note how the residuals go outside the 99% limits detecting the failure of the level sensor.

The same model can also be used to detect the thermocouple failure shown previously using the Q statistic and/or the T^2 statistic. *Figure 27* shows a plot of T^2 versus Q for the same samples shown in *Figure 6*. As in the previous example, the individual residuals could be used to assign the failure to thermocouple number 5 (not shown).

The PLS model can also be used to detect a process upset as shown in *Figure 28*. Here the T^2 indicates a disturbance at samples 50–70. T^2 alone, however, does not provide much information regarding the root cause of the disturbance. The contribution of the individual variables to T^2 is shown in *Figure 29*. This is calculated by multiplying the normalized scores used in the calculation of T^2 by the loadings matrix. *Figure 29* shows the contributions to T^2 for samples 51–60 as a bar chart. The pattern of variables contributing to T^2 suggests a cold-cap upset, a condition where the cold-cap atop the molten glass breaks and lets liquid slurry come into direct contact with the glass. The pattern of widely different contributions on variables 15–20 suggests that the upset was primarily confined to the right side of the melter.

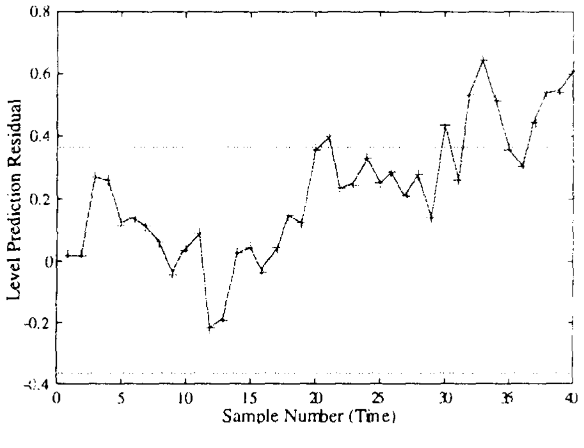

Figure 26 Level sensor prediction residuals from PLS model showing detection of sensor failure

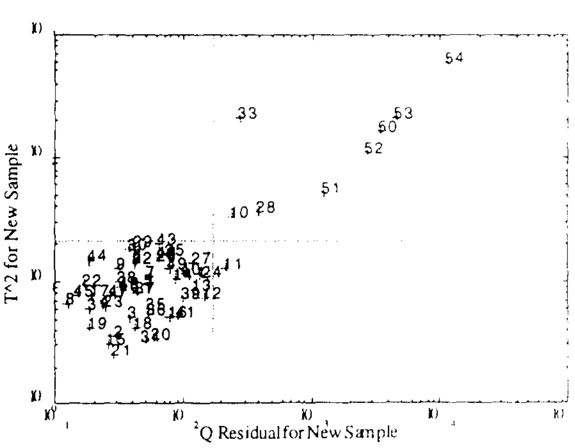

Figure 27 T^2 versus Q plot for PLS model showing detection of failed thermocouple

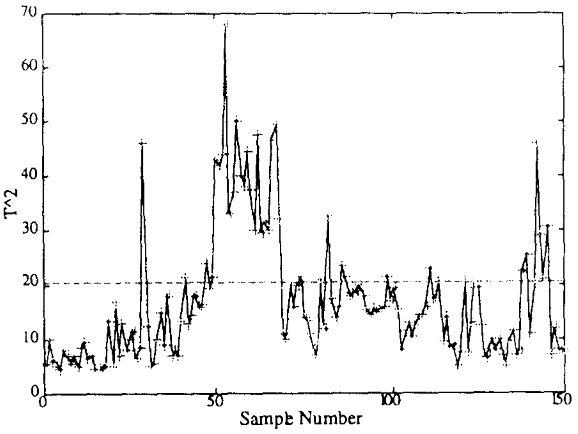

Figure 28 T^2 versus time for SFCM process showing process upset at samples 50–70

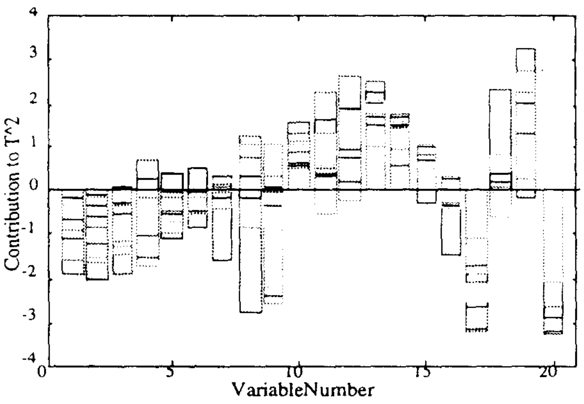

Figure 29 Contribution of individual variables to T^2 for upset period of *Figure 28*

PLS for regression adjusted variables

It has been demonstrated[40,41] that PCA-like residuals can be generated with PLS, and that the detection limits for process faults and sensor failures are often much improved over PCA monitoring. This requires that PLS models be obtained that relate each process output to the remaining outputs in the system. Thus, for a system with n outputs, n PLS models would be required. The final regression vectors from the n PLS models can be formed into a single matrix, with each model being a column vector. Because each of the variables does not contribute to its own prediction, the resulting prediction matrix, $\mathbf{M}_p$, has zeros on the diagonal. Thus, the PLS 'filtered' estimate of a data matrix $\mathbf{X}$ can be obtained by simple matrix multiplication

$$\hat{\mathbf{X}} = \mathbf{X}\mathbf{M}_p \qquad (49)$$

The $\hat{\mathbf{X}}$ in Equation 49 have also been referred to as 'regression adjusted' variables. A residuals matrix, $\mathbf{D}_{pls}$, can be calculated from

$$\mathbf{D}_{pls} = \mathbf{X} - \hat{\mathbf{X}} = \mathbf{X} - \mathbf{X}\mathbf{M}_p = \mathbf{X}(\mathbf{I} - \mathbf{M}_p) = \mathbf{X}\mathbf{R}_{pls} \quad (50)$$

where $\mathbf{R}_{pls}$ is the PLS equivalent of the $\mathbf{I}$-$\mathbf{P}\mathbf{P}^T$ matrix in PCA. Once the $\mathbf{R}_{pls}$ matrix has been calculated it can be used with new data to produce PCA-like residuals. These residuals can be used with statistical tests to detect any changes to the process or its sensors. It has been shown that the use of predictive ability as the basis for model selection makes PLS more sensitive to process upsets than PCA. The major drawback of PLS in this application is that it produces residuals that are autocorrelated, due to mapping of state information into the residuals. The autocorrelation in the residuals can change when the autocorrelation in the states changes, invalidating the calculated control limits.

Multi-block PLS

Multi-block data anlaysis methods have their origins in path analysis and path modeling in the fields of sociology and econometrics. Multivariate projection methods for analyzing such block data are largely due to Herman Wold,[42] Bruce Kowalski[33] and Svante Wold.[43] In this article we will use a variation of the multi-block or hierarchical PLS algorithms of Wold *et al.*[44] and Wangen and Kowalski[45] The basic concepts will be illustrated here using the case shown in *Figure 30* where there is a single $\mathbf{Y}$ block and two $\mathbf{X}$ blocks. The multi-block PLS decomposition proceeds as follows:

(1) Start by selecting one column of $\mathbf{Y}$, $\mathbf{y}_j$, as the starting estimate for $\mathbf{u}$.

(2) Perform part of a PLS round on each of the blocks $\mathbf{X}_1$ and $\mathbf{X}_2$ to get ($\mathbf{w}_1$, $\mathbf{t}_1$) and ($\mathbf{w}_2$, $\mathbf{t}_2$) as in Equations 34–41 above.

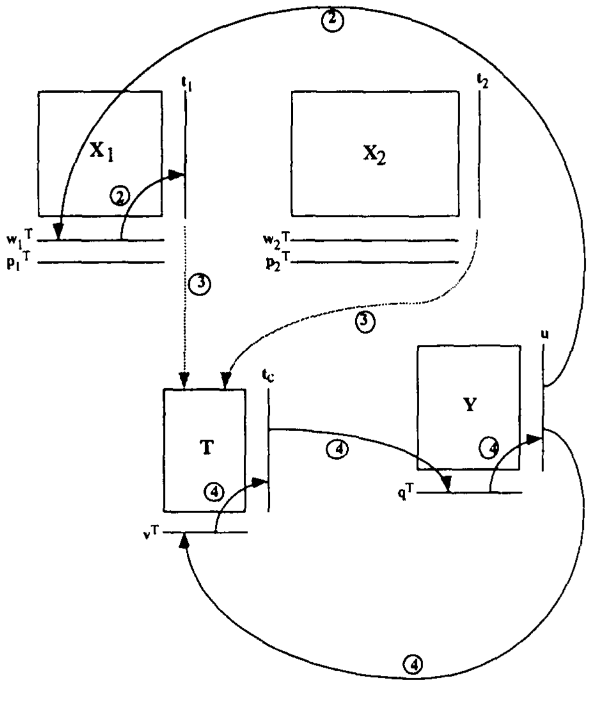

Figure 30 Multi-block PLS algorithm

(3) Collect all the score vector $\mathbf{t}_1$, $\mathbf{t}_2$ in the consensus matrix $\mathbf{T}$.

(4) Make one round of PLS with $\mathbf{T}$ as $\mathbf{X}$ (Equations 34–41 above) to get a loading vector $\mathbf{v}$ and a score vector $\mathbf{t}_c$ for the $\mathbf{T}$ matrix, as well as a loading vector $\mathbf{q}$ and a new score vector $\mathbf{u}$ for the $\mathbf{Y}$ matrix.

(5) Return to step 2 and iterate until convergence of $\mathbf{u}$.

(6) Compute the loadings $\mathbf{p}_1 = \mathbf{X}_1^T\mathbf{t}_1/\mathbf{t}_1^T\mathbf{t}_1$ and $\mathbf{p}_2 = \mathbf{X}_2\mathbf{t}_2/\mathbf{t}_2^T\mathbf{t}_2$ for the $\mathbf{X}_1$ and $\mathbf{X}_2$ matrices.

(7) Compute the residual matrices $\mathbf{E}_1 = \mathbf{X}_1 - \mathbf{t}_1\mathbf{p}_1^T$; $\mathbf{E}_2 = \mathbf{X}_2 - \mathbf{t}_2\mathbf{p}_2^T$, $\mathbf{F} = \mathbf{Y} - \mathbf{t}_c\mathbf{q}^T$.

(8) Calculate the next set of latent vectors by replacing $\mathbf{X}_1$, $\mathbf{X}_2$ and $\mathbf{Y}$ by their residual matrices $\mathbf{E}_1$, $\mathbf{E}_2$, and $\mathbf{F}$, and repeating from step 1.

Although there are many possible variations of multi-block PLS algorithms, the one illustrated here leads to loading vectors $\mathbf{w}$ and score vectors $\mathbf{t}$ for each block that are orthogonal to one another, but the overall score vectors $\mathbf{t}_c$ are not orthogonal to one another. This follows from the computation of the residual matrix $\mathbf{E}_1$ and $\mathbf{E}_2$ in step 7, the loading vectors and orthogonal latent vectors for each block $\mathbf{X}$.

Multi-block PLS is appropriate when the process of interest can be broken down into logical blocks of variables.[46] An example of this might be different zones in aa continuous reactor or measurements made on several different feed stocks that are blended together to form a final product.

Multi-way PLS

Multi-way PLS[47–49] is useful when one wants to relate a three-way array of predictor variables, such as that shown in *Figure 8*, to a matrix of predicted variables. The development of a multi-way PLS model is very similar to the development of a multiway PCA model. The array of predictor variables $\mathbf{X}$ is unfolded as in *Figure 9* to form a matrix $\mathbf{X}$. PLS for multivariate $\mathbf{Y}$ is then used to form a model relating $\mathbf{X}$ to $\mathbf{Y}$.

Multi-way PLS is particularly useful for modeling batch systems. Generally, the end point quality parameters compose $\mathbf{Y}$ and are predicted using the history of the process variables. Monitoring can be done during the batch by using the missing data versions of PLS.

Applications of multi-block and multi-way PLS

Multi-way and multi-block PLS have been demonstrated several times using both simulated and actual process data. In a very early application of multi-block PLS, Frank *et al.* considered the manufacture of 3M adhesive tape samples. Improvements to the quality control of the adhesives was reported. Nomikos and MacGregor have reported using multi-way PLS in simulated monitoring of a styrene butadiene batch reactor. MacGregor *et al.*[46] reported the use of multiblock PLS on a simulated low density polyethylene process. Additional applications of MPCA include that of Kosanovich *et al.*[50] a proprietary process.

Conclusions

This paper has provided a summary of many of the concepts and methods of the chemometrics approach to process monitoring and fault detection. These methods take full advantage of the multivariate nature of the data. Because of this, the methods are often much more powerful techniques for detecting changes in a system than methods that look at variables one at a time.

Chemometric methods provide a rich tool set for process monitoring and fault detection applications because the models:

- handle large numbers of correlated variables very well
- take advantage of structure in data
- provide data compression
- provide a great deal of diagnostic information

Furthermore, the development of models is typically straightforward because the models are deterministic (same model each time for a given data set) and the computational requirements relatively low. This generally allows the model developer to participate directly and in real time with the model buillding process, which often leads to deeper process insight. Finally, model development requires only examples of desired operation, as opposed to a full theoretical model of the process.

The methods considered here can be applied in a number of ways depending upon the particular objective at hand. The general procedure for their application is always the same, however. A model which describes the desired behavior of the system is identified using data from the actual process. Future process data is compared with the models to determine if a change has occurred in the system.

If recent trends continue, we expect to see more applications of PCA and PLS in the process environment. Much work remains to be done on the multi-way and multi-block methods, and this is a very active research area. Other methods that take explicit account of the ordered nature of the data are also being investigated. Applications of multi-way methods to real process are already beginning, however, and we expect this trend to continue.

Acknowledgement

The authors wish to thank John F. MacGregor for assistance in acquiring the references upon which this manuscript is based. Additional discussions with Bruce R. Kowalski were also helpful.

References

1 Brown, S.D. *NAmICS Newsletter* 1991 (1)
2 Wise, B.M. *NAmICS Newsletter* 1991 (2)
3 Wise, B.M., Ricker, N.L., Veltkamp, D.J. and Kowalski, B.R. *Process Control and Quality* 1990, (1) 41
4 Kresta, J.V., MacGregor, J.F. and Marlin, T.E. *Can. J. of Chem. Engng.* 1991, **69**, 35
5 Jackson, J.E. *J. Qual. Tech.* 1981, **13**(2)
6 Jackson, J.E. 'A User's Guide to Principal Components' Wiley, New York, 1991
7 Jackson, J.E. and Mudholkar G.S. *Techometrics* 1979, **21**(3)
8 Wise, B.M., Veltkamp, D.J., Ricker, N.L., Kowalski, B.R., Barnes, S.M. and Arakali, V. 'Waste Management '91 Proceedings', Tucson, AZ, 1991b
9 Nomikos, P. and MacGregor, J.F. *Chemo. and Intell. Lab. Sys.* 1995 in press
10 Freund, J.E. 'Statistics A First Course' 3rd Edition, Prentice Hall, NJ, 1981, 236
11 Åstrom, K.J. and Wittenmark, B. 'Computer Controlled Systems' Prentice-Hall, Englewood Cliffs, NJ, 1984
12 Kwakernaak, H. and Sivan, R. 'Linear Optimal Control Systems' Wiley, New York, NY, 1972
13 Wise, B.M. and McMakin, A.H. 'A statistical technique for analyzing data from liquid-fed ceramic melters' PNL-SA-15267, Pacific Northwest Laboratory, Richland, WA, 1987
14 Wise, B.M. and Ricker, N.L. *AIChE Symposium Series on Process Sensing* 1989, **85** (267)
15 Wise, B.M., Veltkamp, D.J., Davis, B., Ricker, N.L. and Kowalski, B.R. 'Waste Management '88 Proceedings' Tucson, AZ, 1988
16 Wise, B.M. *Process Control and Quality* 1993, **5**, 73
17 Miller, P., Swanson, R.E. and Heckler, C.F. 37th Annual Conference ASCQ Rochester, NY, 1993
18 Ljung, L. 'System Identification: Theory for the User' Prentice-Hall, New Jersey, 1987
19 Ljung, L. 'System Identification Toolbox User's Guide' The MathWorks, Natick, MA, 1991
20 Geladi, P. *Chemo. and Intell. Lab. Sys.* 1989, **7**, 11
21 Sanchez, E. and Kowalski, B.R. *J. Chemometrics*, 1990, **4**, 29
22 Booksh, K.S., Lin, Z., Wang, Z. and Kowalski, B.R. *Anlytical Chemistry*, 1994, **66**, 2561
23 Smilde, A.K. and Doornbos, D.A. *J. Chemometrics* 1991, **5**, 345
24 Smilde, A.K., Wang, Y. and Kowalski, B.R. *J. Chemometrics* 1994, **8**, 21
25 Kroonenberg, P.M. and de Leeuw, J. *Psychometrika*, 1980, **45**, 69
26 Wold, S., Geladi, P., Esbensen, K. and Ohman, J. *J. Chemometrics*, 1987, **1**, 41
27 Nomikos, P. and MacGregor, J.F. *Technometrics*, 1995, **37** (1)
28 Keller, H.R. and Massart, D.L. *Chemometrics and Intelligent Laboratory Systems* 1992, B12, 209
29 Tauler, R., Izquierdo-Ridorsa, A. and Casassas, E. *Chemometrics and Intelligent Laboratory Systems* 1993, **18**, 293
30 Tauler, R., Kowalski, B.R. and Fleming, S. *Analytical Chemistry*, 1993, 65, 2040
31 Wise, B.M. and Kowalski, B.R. 'Process Analytical Chemistry' Blackie Acadamic, 1995
32 Næs, T. and Martens, H. *J. Chemometrics* 1988, **2**
33 Frank, I.E., Feikema, J., Constantine, N. and Kowalski, B.R. *Journal of Chemical Information and Computer Sciences* 1983, **24**, 20
34 Geladi, P. and Kowalski, B.R. *Anal. Chim. Acta* 1986, **185** (1)
35 Lorber, L. 'System Identification: Theory for the User' Prentice-Hall, New Jersey, 1987
36 Höskuldsson, A. *J. Chemometrics* 1988, **2**, 211
37 de Jong, S. *Chemometrics and Intelligent Laboratory Systems* 1993, **18**, 251
38 Draper, H. R. and Smith, H. 'Applied Regression Analysis' Second Edition, John Wiley, new York, 1981
39 Hoerl, A.E., Kennard, R.W. and Baldwin, K.F. *Communications in Statistics*, 1975, **4**, 105
40 Wise, B.M. and Ricker, N.L. in 'ADCHEM '91' Toulouse, France, pp. 125, 1991c
41 Wise, B.M. and Gallagher, N.B. 'PLS Toolbox for Use with MATLAB' Version 1.5 Software, 1995b
42 Wold, H. 'Systems Under Indirect Observations' (eds K.G. Goreskog and H. Wold) North Holland, Amsterdam, 1982, Chap. 1, Vol. 2
43 Wold, S., Ruhe, A., Wold, H. and Dunn, W.J. *SIAM J. Sci. Stat. Comput.* 1984, **5**, 735
44 Wold, S., Esbensen, K. and Geladi, P. *Chemo. and Intell. Lab. Sys.* 1987a, **2**, 37
45 Wangen, L.E. and Kowalski, B.R. *J. Chemometrics* 1988, **3**, 3
46 MacGregor, J.F., Jaeckle, C., Kiparissides, C. and Koutoudi, M.

AIChE Journal, 1994, **40** (5), 826

47 Nomikos, P. and MacGregor, J.F. *AIChE Journal*, 1994, **40** (8) 1361

48 Kourti, T., Nomikos, P. and MacGregor, J.F. *J. Process Control*, 1995, 277

49 Kourti, T., Nomikos, P. and MacGregor, J.F. *Chemo. and Intell. Lab. Sys.* 1995, **28**, 3

50 Kosanovich, K.A., Piovoso, M.J., Dahl, K.S., MacGregor, J.F. and Nomikos, P. 'Multi-way PCA applied to an industrial batch process' 1994 American Control Conference