Process Monitoring Approach Using Fast Moving Window PCA

Xun Wang, Uwe Kruger,* and George W. Irwin

Intelligent Systems and Control Group, Queen's University Belfast, Belfast BT9 5AH, U.K.

This paper introduces a fast algorithm for moving window principal component analysis (MWPCA) which will adapt a principal component model. This incorporates the concept of recursive adaptation within a moving window to (i) adapt the mean and variance of the process variables, (ii) adapt the correlation matrix, and (iii) adjust the PCA model by recomputing the decomposition. This paper shows that the new algorithm is computationally faster than conventional moving window techniques, if the window size exceeds 3 times the number of variables, and is not affected by the window size. A further contribution is the introduction of an N-step-ahead horizon into the process monitoring. This implies that the PCA model, identified N-steps earlier, is used to analyze the current observation. For monitoring complex chemical systems, this work shows that the use of the horizon improves the ability to detect slowly developing drifts.

1. Introduction

Modern industrial processes often present huge amounts of process data due to the large number of frequently measured variables. This renders the ever-increasing demand for efficient process monitoring a difficult task. Statistically based techniques, collectively referred to as multivariate statistical process control, are known to be effective in detecting and diagnosing abnormal operating conditions in such an environment.[1,2] Two of the most commonly used techniques, principal component analysis (PCA) and partial least squares (PLS), have attracted considerable attention in the literature. PCA is the conceptually simpler technique, as it directly analyzes the recorded variables.

Despite the considerable success of PCA, Gallagher et al.[3] pointed out that most industrial processes are time-varying and that the monitoring of such processes, therefore, requires the adaptation of the PCA model to accommodate this behavior. However, the updated model must still be able to detect abnormal behavior with respect to statistical confidence limits, which themselves may also have to vary with time.[4] This paper studies two techniques that allow such an adaptation of the PCA monitoring model, i.e., moving window PCA (MWPCA) and recursive PCA (RPCA). Two properties are of particular interest. One is the speed of adaptation, describing how fast the process model changes with new events. The other is the speed of computation, the time taken for the algorithm to finish one iteration for model adaptation.

The principle behind the moving window is well-known. As the window slides along the data, a new process model is generated by including the newest sample and excluding the oldest one. Recursive techniques, on the other hand, update the model for an ever-increasing data set that includes new samples without discarding old ones. It offers efficient computation by updating the process model using the previous model rather than completely building it from the original data.[5−8] Although successfully employed for process monitoring,[5] RPCA may be difficult to implement in practice. First, the data set on which the model is updated is ever-growing, leading to a reduction in the speed of adaptation as the data size increases. Second, RPCA includes older data that become increasingly unrepresentative of the time-varying process. If a forgetting factor is introduced to down-weight older samples, the selection of this factor can be difficult without a priori knowledge of likely fault conditions.

MWPCA can overcome some of these deficiencies in RPCA by including a sufficient number of data points in the time-window from which to build the adaptive process model. This allows older samples to be discarded in favor of newer ones that are more representative of the current process operation. Furthermore, the use of a constant number of samples in the window equates to a constant speed of model adaptation. This causes a problem when the window has to cover a large number of data points in order to include sufficient process variation for modeling and monitoring purposes, since the computational speed of MWPCA then drops significantly. If a smaller window size is attempted to improve computational efficiency, data within the window then may not properly represent the underlying relationships between the process variables. An additional danger is that the resulting model may adapt to process changes so quickly that abnormal behavior remains undetected.

To improve computational efficiency without compromising the window size, a fast moving window PCA scheme is proposed in this paper. This method relies on the combined use of RPCA and MWPCA to enhance the application of adaptive condition monitoring. Blending recursive and moving window techniques has proved beneficial in computing the computation of the discrete Fourier transform and in the least-squares approximations.[9,10] Qin[7] discussed the integration of a moving window approach into the recursive PLS algorithm, whereby the process data are grouped into subblocks. Individual PLS models are then built for each of these blocks. When the window moves along the data blocks, a PLS model for the selected window is calculated using

* Corresponding author. Tel.: +44(0)2890 974059. Fax: +44(0)2890 667023. Email: uwe.kruger@ee.qub.ac.uk.

10.1021/ie048873f CCC: $30.25
Published on Web 06/10/2005

the sub-PLS models rather than the original subblock data. The approach proposed in this paper is based on the conventional MWPCA, where the window slides along the original data sample by sample. The general computational benefits of the new algorithm are analyzed in terms of the number of floating point operations required to demonstrate a significantly increased computational efficiency.

A further contribution of the paper lies in the incorporation of an N-step-ahead horizon in the adaptive PCA monitoring procedure. The advantage of using a much older process model for prediction is demonstrated by application to simulated data for fault detection.

The next section gives a brief review of RPCA, providing the necessary background for the new fast MWPCA algorithm, which is derived in Section 3. Section 4 compares the speed of model adaptation of conventional MWPCA with that of the new fast approach. Section 5 then explains that a one-step-ahead prediction may produce adaptation to incipient faults and introduces the concept of an N-step-ahead horizon. Sections 6 and 7 present two application studies, involving a simple example process from the literature and a realistic simulation of a fluid catalytic cracking unit (FCCU), respectively. The conclusions appear in Section 8.

2. Review of Recursive PCA

Since a PCA model is constructed from the correlation matrix of the original process data,[11] its adaptation is achieved efficiently by calculating the current correlation matrix from the previous one rather than by using the old process data.[8] For the original data matrix $\mathbf{X}_k^o \in \mathbb{R}^{k\times m}$, which includes m process variables collected until time instant k, the mean and standard deviation are given by $\mathbf{b}_k$ and $\mathbf{\Sigma}_k = \mathrm{diag}\{\sigma_k(1) \cdots \sigma_k(m)\}$. The matrix $\mathbf{X}_k^o$ is then scaled to produce $\mathbf{X}_k$, such that each variable now has zero mean and unit variance. The correlation matrix, $\mathbf{R}_k$, of the scaled data set is given by

$$\mathbf{R}_k = \frac{1}{k-1}\mathbf{X}_k^T\mathbf{X}_k \tag{1}$$

When a new data point, $\mathbf{x}_{k+1}^o$, becomes available, the mean and standard deviation of the extended data matrix

$$\mathbf{X}_{k+1}^o = \begin{bmatrix} \mathbf{X}_k^o \\ (\mathbf{x}_{k+1}^o)^T \end{bmatrix}$$

are updated as follows

$$\mathbf{b}_{k+1} = \frac{k}{k+1}\mathbf{b}_k + \frac{1}{k+1}\mathbf{x}_{k+1}^o \tag{2}$$

$$(\sigma_{k+1}(i))^2 = \frac{k-1}{k}(\sigma_k(i))^2 + (\Delta b_{k+1}(i))^2 + \frac{(x_{k+1}^o(i) - b_{k+1}(i))^2}{k} \tag{3}$$

with $\Delta\mathbf{b}_{k+1} = \mathbf{b}_{k+1} - \mathbf{b}_k$. Given that $\mathbf{\Sigma}_{k+1} = \mathrm{diag}\{\sigma_{k+1}(1) \cdots \sigma_{k+1}(m)\}$, the new sample, $\mathbf{x}_{k+1}^o$, is scaled as

$$\mathbf{x}_{k+1} = \mathbf{\Sigma}_{k+1}^{-1}(\mathbf{x}_{k+1}^o - \mathbf{b}_{k+1}) \tag{4}$$

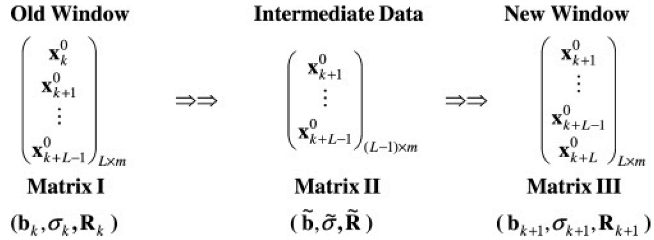

Figure 1. Two-step adaptation to construct new data window.

Utilizing $\Delta\mathbf{b}_{k+1}$, $\mathbf{\Sigma}_{k+1}$, $\mathbf{\Sigma}_k$, $\mathbf{x}_{k+1}$, and the old $\mathbf{R}_k$, the updated correlation matrix $\mathbf{R}_{k+1}$ is given by

$$\mathbf{R}_{k+1} = \frac{k-1}{k}\mathbf{\Sigma}_{k+1}^{-1}\mathbf{\Sigma}_k\mathbf{R}_k\mathbf{\Sigma}_k\mathbf{\Sigma}_{k+1}^{-1} + \mathbf{\Sigma}_{k+1}^{-1}\,\Delta\mathbf{b}_{k+1}\,\Delta\mathbf{b}_{k+1}^T\mathbf{\Sigma}_{k+1}^{-1} + \frac{1}{k}\mathbf{x}_{k+1}\mathbf{x}_{k+1}^T \tag{5}$$

The eigenvalue–eigenvector decomposition of $\mathbf{R}_{k+1}$ then provides the required new PCA model.

3. Fast Moving Window Algorithm

RPCA updates the correlation matrix by adding a new sample to its current value. Conventional MWPCA operates by first discarding the oldest sample from the correlation matrix and then adding a new sample to the matrix. The details of this two-step procedure are shown in Figure 1 for a window size L. The fast MWPCA algorithm is based on this, but it also incorporates the adaptation technique in RPCA. The three matrices in Figure 1 represent the data in the previous window (Matrix I), the result of removing the oldest sample $\mathbf{x}_k^o$ (Matrix II), and the current window of selected data (Matrix III) produced by adding the new sample $\mathbf{x}_{k+L}^o$ to Matrix II. The means the standard deviations and correlation matrices for Matrix II and Matrix III are derived as follows.

3.1. Step 1: Matrix I to Matrix II. The effect of eliminating the oldest sample from Matrix I, on the mean and variance of each process variable and the correlation matrix, can be computed recursively. Equations 6 and 7 describe the variable mean, and eqs 8 and 9 describe the variable variance, while the scaling of the new data point is defined in eq 10

$$\tilde{\mathbf{b}} = \frac{1}{L-1}(L\mathbf{b}_k - \mathbf{x}_k^o) \tag{6}$$

$$\Delta\tilde{\mathbf{b}} = \mathbf{b}_k - \tilde{\mathbf{b}} \tag{7}$$

$$\tilde{\sigma}^2(i) = \frac{L-1}{L-2}(\sigma_k(i))^2 - \frac{L-1}{L-2}(\Delta\tilde{b}(i))^2 - \frac{[x_k^o(i) - b_k(i)]^2}{L-2} \tag{8}$$

$$\tilde{\mathbf{\Sigma}} = \mathrm{diag}\{\tilde{\sigma}(1) \cdots \tilde{\sigma}(m)\} \tag{9}$$

$$\mathbf{x}_k = \mathbf{\Sigma}_k^{-1}(\mathbf{x}_k^o - \mathbf{b}_k) \tag{10}$$

Recursive adaptation of the correlation matrix is based on the above equations. A new matrix $\mathbf{R}^*$ is now introduced to simplify the formation of the equations.

$$\mathbf{R}^* = \frac{L-2}{L-1}\mathbf{\Sigma}_k^{-1}\tilde{\mathbf{\Sigma}}\,\tilde{\mathbf{R}}\tilde{\mathbf{\Sigma}}\mathbf{\Sigma}_k^{-1} \tag{11}$$

Table 1. Procedure to Update Correlation Matrix for the Fast MWPCA Approach

step	equation	description
1	$\tilde{\mathbf{b}} = \frac{1}{L-1}(L\mathbf{b}_k - \mathbf{x}_k^o)$	mean of Matrix II
2	$\Delta\tilde{\mathbf{b}} = \mathbf{b}_k - \tilde{\mathbf{b}}$	difference between means
3	$\mathbf{x}_k = \mathbf{\Sigma}_k^{-1}(\mathbf{x}_k^o - \mathbf{b}_k)$	scale the discarded sample
4	$\mathbf{R}^* = \mathbf{R}_k - \mathbf{\Sigma}_k^{-1}\Delta\tilde{\mathbf{b}}\Delta\tilde{\mathbf{b}}^T\mathbf{\Sigma}_k^{-1} - \frac{1}{L-1}\mathbf{x}_k\mathbf{x}_k^T$	bridge over Matrix I and Matrix III
5	$\mathbf{b}_{k+1} = \frac{1}{L}[(L-1)\tilde{\mathbf{b}} + \mathbf{x}_{k+L}^o]$	mean of Matrix III
6	$\Delta\mathbf{b}_{k+1} = \mathbf{b}_{k+1} - \tilde{\mathbf{b}}$	difference between means
7	$(\sigma_{k+1}(i))^2 = (\sigma_k(i))^2 + (\Delta b_{k+1}(i))^2 - (\Delta\tilde{b}(i))^2 + \frac{[x_{k+L}^o(i) - b_{k+1}(i)]^2 - [x_k^o(i) - b_k(i)]^2}{L-1}$	standard deviation of Matrix III
8	$\mathbf{\Sigma}_{k+1} = \mathrm{diag}\{\sigma_{k+1}(1) \cdots \sigma_{k+1}(m)\}$	
9	$\mathbf{x}_{k+L} = \mathbf{\Sigma}_{k+1}^{-1}(\mathbf{x}_{k+L}^o - \mathbf{b}_{k+1})$	scale the new sample
10	$\mathbf{R}_{k+1} = \mathbf{\Sigma}_{k+1}^{-1}\mathbf{\Sigma}_k\mathbf{R}^*\mathbf{\Sigma}_k\mathbf{\Sigma}_{k+1}^{-1} + \mathbf{\Sigma}_{k+1}^{-1}\Delta\mathbf{b}_{k+1}\Delta\mathbf{b}_{k+1}^T\mathbf{\Sigma}_{k+1}^{-1} + \frac{1}{L-1}\mathbf{x}_{k+L}\mathbf{x}_{k+L}^T$	correlation matrix of Matrix III

$\mathbf{R}^*$ is further derived into

$$\mathbf{R}^* = \mathbf{R}_k - \mathbf{\Sigma}_k^{-1}\Delta\tilde{\mathbf{b}}\Delta\tilde{\mathbf{b}}^T\mathbf{\Sigma}_k^{-1} - \frac{1}{L-1}\mathbf{x}_k\mathbf{x}_k^T \quad (12)$$

The recursion for updating the correlation matrix after elimination of the oldest sample, i.e., Matrix II, is expressed in eq 13.

$$\tilde{\mathbf{R}} = \frac{L-1}{L-2}\tilde{\mathbf{\Sigma}}^{-1}\mathbf{\Sigma}_k\mathbf{R}^*\mathbf{\Sigma}_k\tilde{\mathbf{\Sigma}}^{-1} \quad (13)$$

3.2. Step 2: Matrix II to Matrix III. The effect of including a new data point on the variable means and variances and the correlation matrix is determined using recursive PCA. The updated mean vector and the change in the mean vectors are computed from eqs 14 and 15, while the adaptation of the standard deviations follows from eqs 16 and 17. The scaling of the newest sample and the updating of the correlation matrix are described in eqs 18 and 19.

$$\mathbf{b}_{k+1} = \frac{1}{L}[(L-1)\tilde{\mathbf{b}} + \mathbf{x}_{k+L}^o] \quad (14)$$

$$\Delta\mathbf{b}_{k+1} = \mathbf{b}_{k+1} - \tilde{\mathbf{b}} \quad (15)$$

$$(\sigma_{k+1}(i))^2 = \frac{L-2}{L-1}(\tilde{\sigma}(i))^2 + (\Delta b_{k+1}(i))^2 - \frac{[x_{k+L}^o(i) - b_{k+1}(i)]^2}{L-1} \quad (16)$$

$$\mathbf{\Sigma}_{k+1} = \mathrm{diag}\{\sigma_{k+1}(1) \quad \cdots \quad \sigma_{k+1}(m)\} \quad (17)$$

$$\mathbf{x}_{k+L} = \mathbf{\Sigma}_{k+1}^{-1}(\mathbf{x}_{k+L}^o - \mathbf{b}_{k+1}) \quad (18)$$

$$\mathbf{R}_{k+1} = \frac{L-2}{L-1}\mathbf{\Sigma}_{k+1}^{-1}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{R}}\tilde{\mathbf{\Sigma}}\mathbf{\Sigma}_{k+1}^{-1} + \mathbf{\Sigma}_{k+1}^{-1}\Delta\mathbf{b}_{k+1}\Delta\mathbf{b}_{k+1}^T\mathbf{\Sigma}_{k+1}^{-1} + \frac{1}{L-1}\mathbf{x}_{k+L}\mathbf{x}_{k+L}^T \quad (19)$$

3.3. Step 3: Combination of Step 1 and Step 2. Steps 1 and 2 can be combined to provide a routine that derives Matrix III directly from Matrix I. The adaptation of the standard deviations follows from combining eqs 8 and 16 to give

$$(\sigma_{k+1}(i))^2 = (\sigma_k(i))^2 + (\Delta b_{k+1}(i))^2 - (\Delta\tilde{b}(i))^2 + \frac{[x_{k+L}^o(i) - b_{k+1}(i)]^2 - [x_k^o(i) - b_k(i)]^2}{L-1} \quad (20)$$

Substituting eqs 12 and 13 into eq 19 produces a recursive equation for the updated correlation matrix of Matrix III. Thus,

$$\mathbf{R}_{k+1} = \mathbf{\Sigma}_{k+1}^{-1}\mathbf{\Sigma}_k\mathbf{R}^*\mathbf{\Sigma}_k\mathbf{\Sigma}_{k+1}^{-1} + \mathbf{\Sigma}_{k+1}^{-1}\Delta\mathbf{b}_{k+1}\Delta\mathbf{b}_{k+1}^T\mathbf{\Sigma}_{k+1}^{-1} + \frac{1}{L-1}\mathbf{x}_{k+L}\mathbf{x}_{k+L}^T \quad (21)$$

The combination of steps 1 and 2 constitutes the new fast moving window technique, which is summarized in Table 1 for convenience.

Efficient techniques for updating the eigenvalues and eigenvectors of the correlation matrix have been proposed by Li et al.[7] In addition, the numerically very efficient inverse iteration[14,15] may be employed for this adaptation, since the updated eigenvalues are assumed to be close to the previous set of eigenvalues. This technique is particularly useful, as the number of retained PCs, i.e., the number of eigenvalues and eigenvectors to be updated, is typically small compared to the number of analyzed process variables.

It should be noted, however, that the fast MWPCA technique gains its computational savings by incorporating the efficient update procedure for the correlation matrices from RPCA. Thus, a thorough analysis of the computational speed of both techniques is conducted next. To provide a fair comparison, the eigenvalues and eigenvectors are computed using a singular value decomposition in both cases.

4. MWPCA and Fast MWPCA Computation

This section compares the new fast MWPCA algorithm with conventional MWPCA in terms of computational efficiency. General expressions for the numbers of floating point operations (flops) are determined and compared for varying window sizes and different numbers of recorded data points. This analysis further leads to the development of a straightforward guideline as to when fast MWPCA is to be preferred over conventional MWPCA.

Table 2. Comparison of Flops Consumed for Conventional and Fast MWPCA

MWPCA technique	variable	equation
conventional MWPCA	$\text{flops}_{\text{cov}}$	$\text{flops}_{\text{cov}} = 2Lm^2 + 8Lm + 4m + 1$
fast MWPCA	$\text{flops}_{\text{fast}}$	$\text{flops}_{\text{fast}} = 6m^3 + 20m^2 + 11m + 17$

General algebraic expressions for the number of flops required for updating the correlation matrix in both algorithms have been derived and are given in Table 2. It should be noted that the computation of conventional MWPCA relies on the window length L, as well as on the number of variables m. In contrast, fast MWPCA only depends on m. The two algorithms can usefully be compared by plotting the ratio of conventional MWPCA flops to fast MWPCA flops. Figure 2 shows the results of this comparison for a variety of configurations, i.e., varying window length, L, and number of variables, m.

Figure 2 shows that the computational speed advantage of the fast MWPCA can exceed 100. The larger the window size, the more significant the advantage. However, with an increasing number of variables, the computational advantage is reduced sharply.

For a fixed number of process variables, m, the following criterion can be employed as a straightforward guideline as to when to use which algorithm. Using the equations in Table 2, it can be concluded that, for a given m, the new algorithm is computationally superior if $L > L^*$, where

$$L^* = 3m - 2 + \frac{23m + 16}{2m^2 + 8m} \tag{22}$$

Figure 3 plots the variation in L^*, for m ranging from 1 to 100. The relationship between L and m is almost linear for $m \leq 100$, which leads to the following conclusions. First, for a given m value, if the window size L lies in the area above the dotted line in Figure 3, then the new fast MWPCA algorithm is to be preferred. Second, assuming that the relationship between L and m is indeed linear, the slope of $L(m)$ is $\sim$3. Thus, as a rule of thumb, if the window size L is more than 3 times larger than the number of variables m, fast MWPCA is to be preferred to conventional MWPCA.

5. Process Monitoring Using *N*-Step-Ahead Prediction

Current MWPCA monitoring uses one-step-ahead prediction, which calculates the monitoring statistics based on the previous PCA model. In some situations, for example, where the window size is small or the faults are gradual, the model may then adapt to the faults and, hence, the faults may go unnoticed by the monitoring charts. As a consequence, corrupted models are produced that should not be used for further monitoring. Increasing the window size would seem to be a straightforward solution to this problem. Although the fast MWPCA offers the computational efficiency to support this idea technically, larger window sizes result in a slower adaptation speed. This paper proposes the use of N-step-ahead prediction to improve the application of MWPCA as a condition monitoring tool. This implies that the current PCA model is not applied in analyzing the next recorded sample, as is the case in one-step-ahead prediction. Rather, it is used to evaluate the sample to be recorded N time steps later. Figure 4 shows an incipient fault described by a ramp. When the $(k + 1)$th sample becomes available, the model updated at the $(k - N)$th sample is used to monitor the process rather than that from the previous time instant. The much older model will be more sensitive to the fault since the recent model is likely to have been corrupted by the faulty data from the ramp fault.

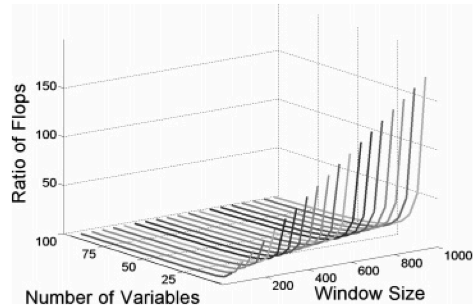

Figure 2. Variation in the ratio of floating point operations for conventional and fast MWPCA for various window sizes L and variable numbers m.

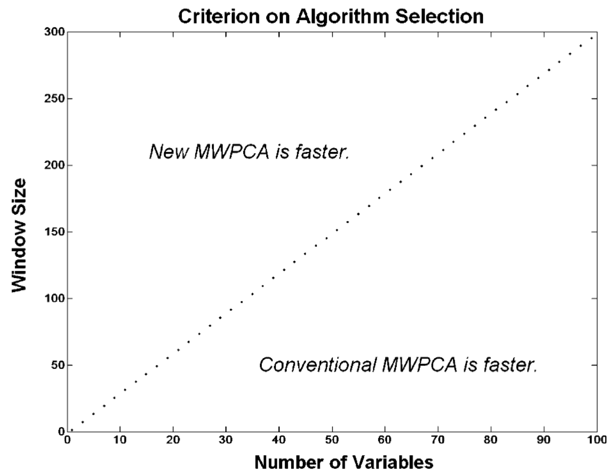

Figure 3. Criterion for selecting the most efficient MWPCA technique.

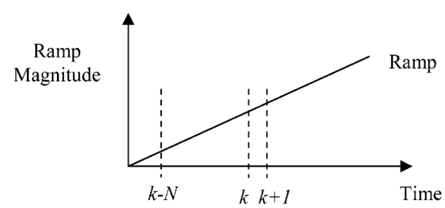

Figure 4. Influence of an N-step-ahead prediction.

The N-step-ahead prediction is now described in more detail. The development of adaptive confidence limits, as discussed by Wang et al.,[4] is first considered. Two univariate monitoring statistics, the T^2 and *SPE* statistics, are usually employed in conventional PCA.[2,12] Here, the T^2 statistic for the kth sample is defined as

$$T_k^{\,2} = \mathbf{x}_k^T\,\mathbf{P}_{k-N}\boldsymbol{\Lambda}_{k-N}^{-1}\mathbf{P}_{k-N}^T\,\mathbf{x}_k \tag{23}$$

Note that both $\boldsymbol{\Lambda}_{k-N}$ and $\mathbf{P}_{k-N}$ are obtained from the $(k - N)$th model, while $\mathbf{x}_k$ is the kth process sample scaled using the mean and variance for that model. The *SPE* statistic for this sample is

$$SPE_k = \mathbf{x}_k^T(\mathbf{I} - \mathbf{P}_{k-N}\mathbf{P}_{k-N}^T)\mathbf{x}_k \tag{24}$$

Table 3. Process Monitoring Using *N*-step-ahead Prediction

Step 1: Build the initial model on the first L data points. This window is denoted as L_1.

Step 2: For a new sample $\mathbf{x}_k$, use the fast MWPCA algorithm to obtain the mean and standard deviation for the latest data set, $\mathbf{x}_{k-L+1}$, ..., $\mathbf{x}_k$, and update the correlation matrix for computing the updated PCA model. This requires discriminating between the following two cases:

- *Case 1*: If $k \leq N + L$, the monitoring statistics for $\mathbf{x}_k$ are calculated using the initial model. The confidence limits remain the same as before.
- *Case 2*: If $k > N + L$, the monitoring statistics for $\mathbf{x}_k$ are calculated using the mean, variance, and PCA model from N steps earlier. The confidence limits for this sample are recalculated based on the statistics for the data used to update the PCA model.

Step 3: When the statistics for $\mathbf{x}_k$ exceed the confident limits, consideration should be given as to whether $\mathbf{x}_k$ should be included for model adaptation.

The conventional one-step-ahead prediction corresponds to $N = 1$. The improved sensitivity in detecting incipient faults in the form of a ramp signal, with N values > 1, is demonstrated later in two simulation studies.

In this paper, the confidence limits are also calculated using a moving window technique. The window size employed is the same as that for updating the PCA model. The monitoring charts presented in the paper employ 95% and 99% confidence limits. The process monitoring procedure using N-step-ahead prediction can be summarized in Table 3.

6. Simple Case Study

In this section, the N-step-ahead prediction procedure is applied to a simplified study process. This simulation is augmented to represent a slowly changing process taking the form of a ramp. Such situations are common in industrial processes, for instance, a leak, pipe blockage, catalyst degradation, or performance deterioration in individual process units. If moving window PCA is applied in such a situation, such slow changes may be accommodated by model adaptation and, hence, remain unnoticed.

6.1. Data Generation. The data used has two predictor variables and two response variables. Both predictor variables were initially equivalent and obtained from a single autoregressive moving average (ARMA) signal

$$v(z^{-1}) = \frac{1 + 0.02z^{-1}}{1 - 0.03z^{-1}}\epsilon(z^{-1}) \tag{25}$$

where $v(z^{-1})$ represents the ARMA signal, $\epsilon(z^{-1})$ represents a normally distributed random signal of zero mean and unit variance, i.e., $\epsilon(z^{-1}) \in N\{0, 1\}$, and z^{-1} is the back-shift operator. Both predictor variables were then augmented by adding uncorrelated, normally distributed random signals of zero mean and a variance of 0.2, $N\{0, 0.2\}$. The response variables are then determined as linear combinations of the predictor variables

$$\begin{pmatrix}\psi_{1k}\\ \psi_{2k}\end{pmatrix}_{\mathrm{t}} = \begin{bmatrix}1.7 & 0.8\\ -0.6 & 0.02\end{bmatrix}\begin{pmatrix}\xi_{1k}\\ \xi_{2k}\end{pmatrix}_{\mathrm{t}};\quad \begin{pmatrix}\xi_{1k}\\ \xi_{2k}\end{pmatrix}_{\mathrm{t}} = \begin{pmatrix}v_k\\ v_k\end{pmatrix} + \begin{pmatrix}\epsilon_{1k}\\ \epsilon_{2k}\end{pmatrix} \tag{26}$$

where ξ_{1k}, ξ_{2k} and ψ_{1k}, ψ_{2k} are the values of the predictor and response variables and v_k, ϵ_{1k}, and ϵ_{2k} are the values of the ARMA signal and the two superimposed random signals for the kth instance in time, respectively. The subscript t refers to the true values of the predictor and the response variables.

After the true predictor and response signals are computed, each process variable is augmented by adding a normally distributed random signal of zero mean and a variance of 0.1, $N\{0, 0.1\}$ to account for measurement noise

$$\begin{pmatrix}\xi_{1k}\\ \xi_{2k}\end{pmatrix}_{\mathrm{m}} = \begin{pmatrix}\xi_{1k}\\ \xi_{2k}\end{pmatrix}_{\mathrm{t}} + \begin{pmatrix}\epsilon_{3k}\\ \epsilon_{4k}\end{pmatrix} \text{ and } \begin{pmatrix}\psi_{1k}\\ \psi_{2k}\end{pmatrix}_{\mathrm{m}} = \begin{pmatrix}\psi_{1k}\\ \psi_{2k}\end{pmatrix}_{\mathrm{t}} + \begin{pmatrix}\epsilon_{5k}\\ \epsilon_{6k}\end{pmatrix} \tag{27}$$

where m denotes the measured values and ϵ_{3k} to ϵ_{6k} are the kth instances of added random signals to the predictor and the response variables.

A ramp signal with an increment of 0.0002 was added to ξ_2 after 1200 data points were simulated. 3000 data points were collected in total. It should be noted that, since the relationships between the variables remain constant, the model structure did not change over time. The simulated process could, therefore, be regarded as changing to a different, and undesired, operating region.

6.2. Process Monitoring Using *N*-Step-Ahead Prediction. The window size to update the PCA model was selected to be 500. The monitoring charts resulting from the one-step-ahead prediction scheme are shown in Figure 5. Both the *SPE* and the T^2 statistics indicated an in-statistical-control situation, implying that the drift remained undetected. This can be explained as follows. When the drift was injected after the 1200th data point, the PCA model applied corresponded to normal process behavior. However, this PCA model was then adapted to the drift. As the window moved forward, the adaptation of the PCA models was done using increasing amount of data that described the influence of the drift. Since an abnormal number of violations of both statistics with respect to their confidence limits did not arise, the model adaptation procedure continued to regard new data as representing normal operation. This led to a corruption of the underlying PCA model, which was then no longer suitable for monitoring this process.

The corresponding results from applying the 500-step-ahead prediction are shown in Figure 6. The T^2 statistic showed a significant number of violations with respect to both 95% and 99% confidence limits between the 1400th and 2700th data point. The confidence limits themselves also increased in value after the 2000th data point as a result of the drift. This result agrees with the underlying nature of the simulated data. As the slow drift was introduced to one of the inputs after 1200 samples, the process drifted away from its current operation region. Since the relationships between the variables remained constant, the *SPE* statistic did not respond to the drift. However, as a measurement of the data variation, the T^2 statistic successfully detected this fault with the 500-step-ahead prediction.

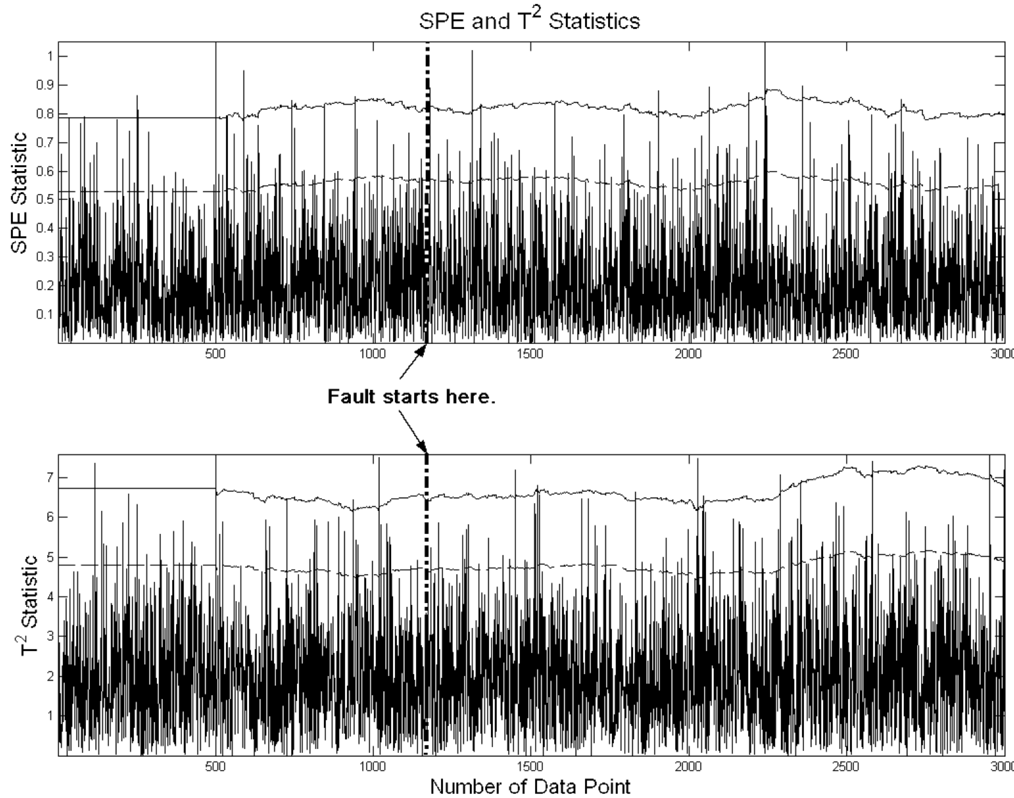

Figure 5. Monitoring a ramp change using one-step-ahead prediction.

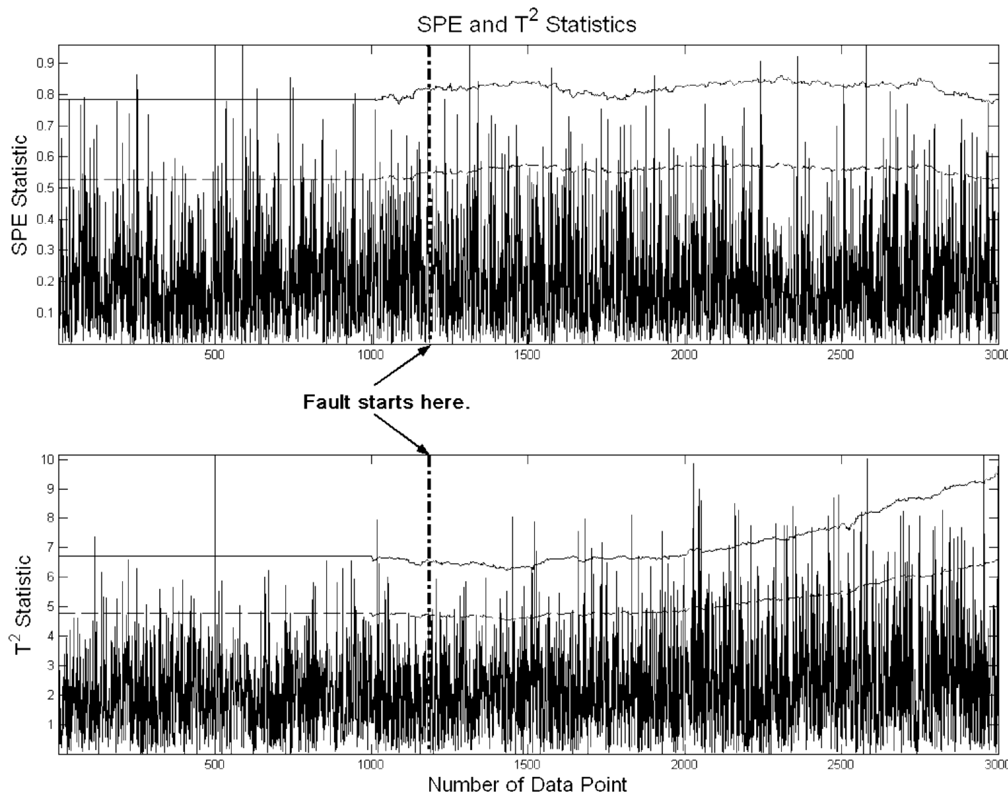

Figure 6. Monitoring a ramp change using 500-step-ahead prediction.

6.3. Sensitivity Analysis Regarding Window Size and Prediction Horizon. The factors affecting the performance of the proposed monitoring scheme are window length L, value of the horizon N, and the slope of the ramp S. The window length affects the efficiency of model adaptation, i.e., the smaller the window, the quicker the model can adapt to process changes. The horizon influences the calculation of both monitoring statistics and control limits. Thus, the larger the horizon, the older the model that is used for analyzing the current data point.

Analysis has been carried out to explore the minimum horizon required to detect a ramp fault of a certain slope using various window lengths. The data sets used were

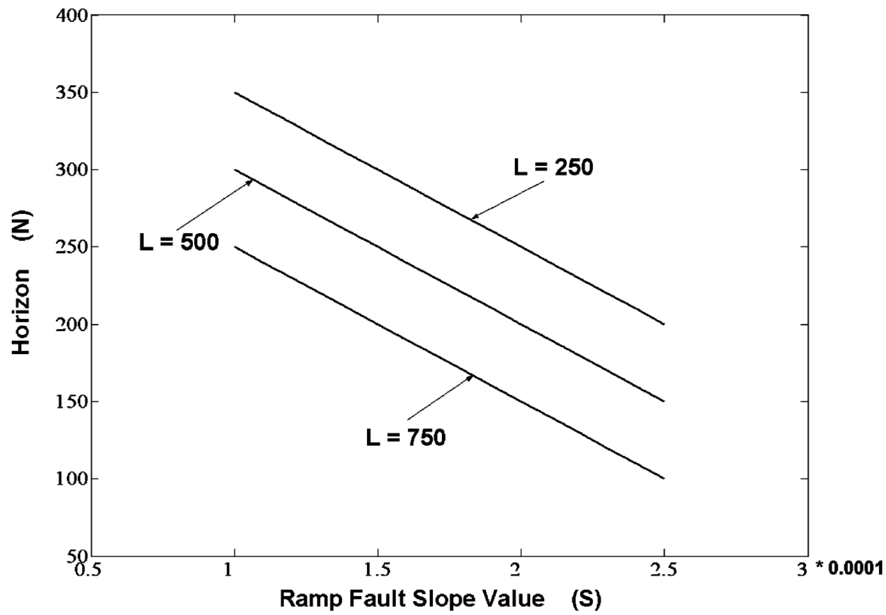

Figure 7. Sensitivity in detecting various slopes.

generated as in section 6.1 but with varying ramp slopes, and window sizes of 250, 500, and 750 were used. With a chosen window size and slope, the minimum horizon that could detect that fault was determined. The results of the complete analysis are displayed in Figure 7, which provides a broad indication of the interrelationship between L, N, and S. It should be noted that the lines in Figure 7 were approximations based on the least squares principle. For each of the ramp sizes analyzed, a number of different data sets were produced on the basis of eqs 25−27, and the smallest horizon N was subsequently obtained.

For a fixed window length L, as the slope S increases in value, the horizon required reduces. Physically, a larger slope implies a more significant change in process performance. In terms of Figure 4, the monitoring procedure then does not require a large value of N to pick up a significant increase in the faulty variable, which aids the detection of the drift. Conversely, if the slope S is very small, the horizon N must be large to notice a significant change in the monitored variable.

With a fixed slope S, a smaller horizon is required if a larger window size is selected. The model adaptation is less effective with a larger window length. When there is a change in the process, the larger the window size the more gradual the changes in the model parameters. Furthermore, monitoring statistics calculated on the basis of a slowly adapting model can detect such drifts more easily than those calculated from a model which adapts more rapidly. Conversely, if the window size L is small, the process models adapts to a ramp fault more effectively, and hence, a large value for N is needed for successful detection.

In summary, using N-step-ahead prediction allows tuning of the monitoring model between adaptability (window length L) and the ability of the model to detect such changes (N value). This important benefit is demonstrated in the next section.

7. Application to a Simulated Industrial Process

In this section, an application study is considered to illustrate the benefits of the proposed N-step-ahead prediction monitoring scheme. The process studied is a simulation of a fluid catalytic cracking unit (FCCU), details of which can be found in ref 13. The particular model IV unit is illustrated in Figure 8.[13]

7.1. Data Generation. The analysis of this unit involved a total of 23 variables, shown in Table 4. ARMA sequences were superimposed on five process input variables to represent measured disturbances. These sequences had the following form

$$u_k = bu_{k-1} + e_k + ae_{k-1} \tag{28}$$

where e follows a normal distribution. The parameters of each ARMA filter, a, b, and var$\{e\}$, are listed in Table 5. Note that each set of parameters was selected such that the variance of the riser temperature = 1. Using the above configuration, a total of 2000 samples were recorded. Two slowly developing drifts, in the form of ramps, were introduced. The first drift, injected at the beginning, related to a gradual decrease in the heat exchanger coefficient. This represented fouling and, hence, a performance deterioration of the exchanger. The second drift injected after the 1500th data sample was more significant and simulated a gradual loss in combustion air blower capacity. In this application, the first fault was assumed to be known and to represent normal operation. The second drift represented an abnormal, and more significant, deterioration in the combustion air blower, and the on-line monitoring model is required to detect this event. The signals of the 23 recorded variables are shown in Figure 9. To justify the use of model adaptation, conventional PCA was compared to its MWPCA counterpart. The use of N-step-ahead prediction was contrasted with the regular one-step-ahead prediction used in the MWPCA model.

7.2. Process Monitoring Results. Three monitoring approaches were investigated using the recorded data set. Conventional PCA was applied first to demonstrate the need for a recursively updated PCA model, followed by a comparison of MWPCA using a one-step-ahead prediction with MWPCA using an N-step-ahead prediction.

7.2.1. Conventional PCA. Figure 10 presents the monitoring charts for PCA applied to the FCCU data.

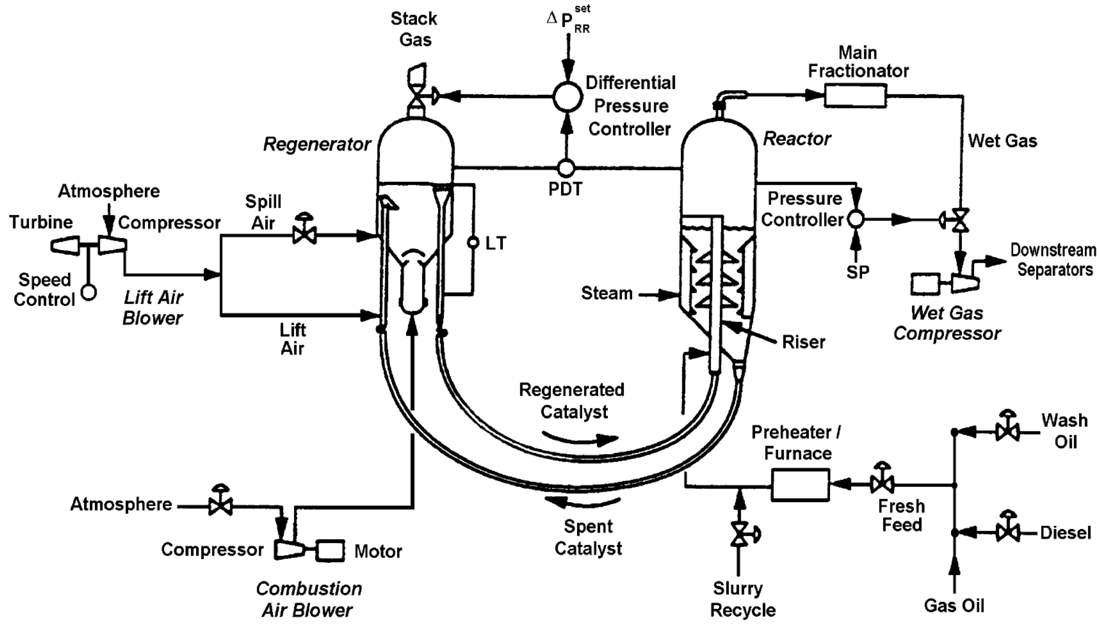

Figure 8. Schematic diagram for FCCU.[13]

Table 4. Considered Process Variables of the FCCU Case Study

variable	description
1	flow of wash oil to reactor riser
2	flow of fresh feed to reactor riser
3	flow of slurry to reactor riser
4	temperature of fresh feed entering furnace
5	fresh feed temperature to riser
6	furnace firebox temperature
7	combustion air blower inlet suction flow
8	combustion air blower throughput
9	combustion air flowrate
10	lift air blower suction flow
11	lift air blower speed
12	lift air blower throughput
13	riser temperature
14	wet gas compressor suction pressure
15	wet gas compressor inlet suction flow
16	wet gas flow to vapor recovery unit
17	regenerator bed temperature
18	stack gas valve position
19	regenerator pressure
20	standpipe catalyst level
21	stack gas O_2 concentration
22	combustion air blower discharge pressure
23	wet gas composition suction valve position

Table 5. Parameters in ARMA Sequences

process variable	var{e}	b	a
slurry flowrate	0.006	0.05	0.5
wash oil feed flowrate	0.006	0.1	0.4
preheater outlet temp	0.115	0.075	0.55
furnace firebox temp	0.014	0.025	0.45
combustion air flowrate	0.014	0.0125	0.55

The first 700 data points were used to identify the PCA model and to determine the control limits for both statistics. The PCA model, along with these fixed control limits, was then applied to the remaining 1300 data points. Figure 10 shows that the *SPE* statistic responded strongly to the second fault, representing a decrease in combustion air blower capacity after 1600 samples. The T^2 statistic, on the other hand, was mainly affected by the first event.

A decrease in the heat exchanger coefficient reduces the amount of heat exchanged to produce a lower temperature in the outlet stream. Therefore, the fresh feed stream entered the riser with a lower temperature than expected. This did not lead to a significant change in the reaction conditions as the hot catalyst provided the necessary heat for the endothermic reactions. Since the PCA monitoring model was established on data that included the influence of the first drift, it was able to describe the relationship between variables correctly, and so, the *SPE* statistic did not respond to the first drift. However, as this drift developed, the process changed to a different operating region, which led to an increase in the T^2 statistic. Since the aim of the study was to mask the impact of the normal performance deterioration of the heat exchanger, the use of a constant mean and variance for each process variable, together with a fixed PCA model, would have given false alarms to the process operator.

The strong response of the *SPE* statistic to the loss of combustion air blower capacity made sense physically. This reduces the airflow into the regenerator, hence reducing the regenerator pressure, which produces a number of consequences. First, the lift air blower increases the level of air pumped into the spent catalyst line. Second, less oxygen is available for recycling the catalyst, which in turn leads to a deterioration in the reaction conditions in the riser. Given the above, most of the variables of the regenerator contributed to this event, leading to a significant change in their interrelationship. The reaction conditions in the riser were also affected, which again contributed to the significant increase in the *SPE* statistic.

This application study revealed that conventional PCA can detect process changes that are benchmarked against historical behavior as described in the reference data, in this case, the first 700 samples. Since conventional PCA is unable to mask slowly developing process changes that are regarded as normal, the application of MWPCA is demonstrated in the next two sections.

7.2.2. MWPCA Based on One-Step-Ahead Prediction. To provide a fair comparison, the window length of the MWPCA technique was set to 700. This implied that its initial PCA model was the same as that utilized

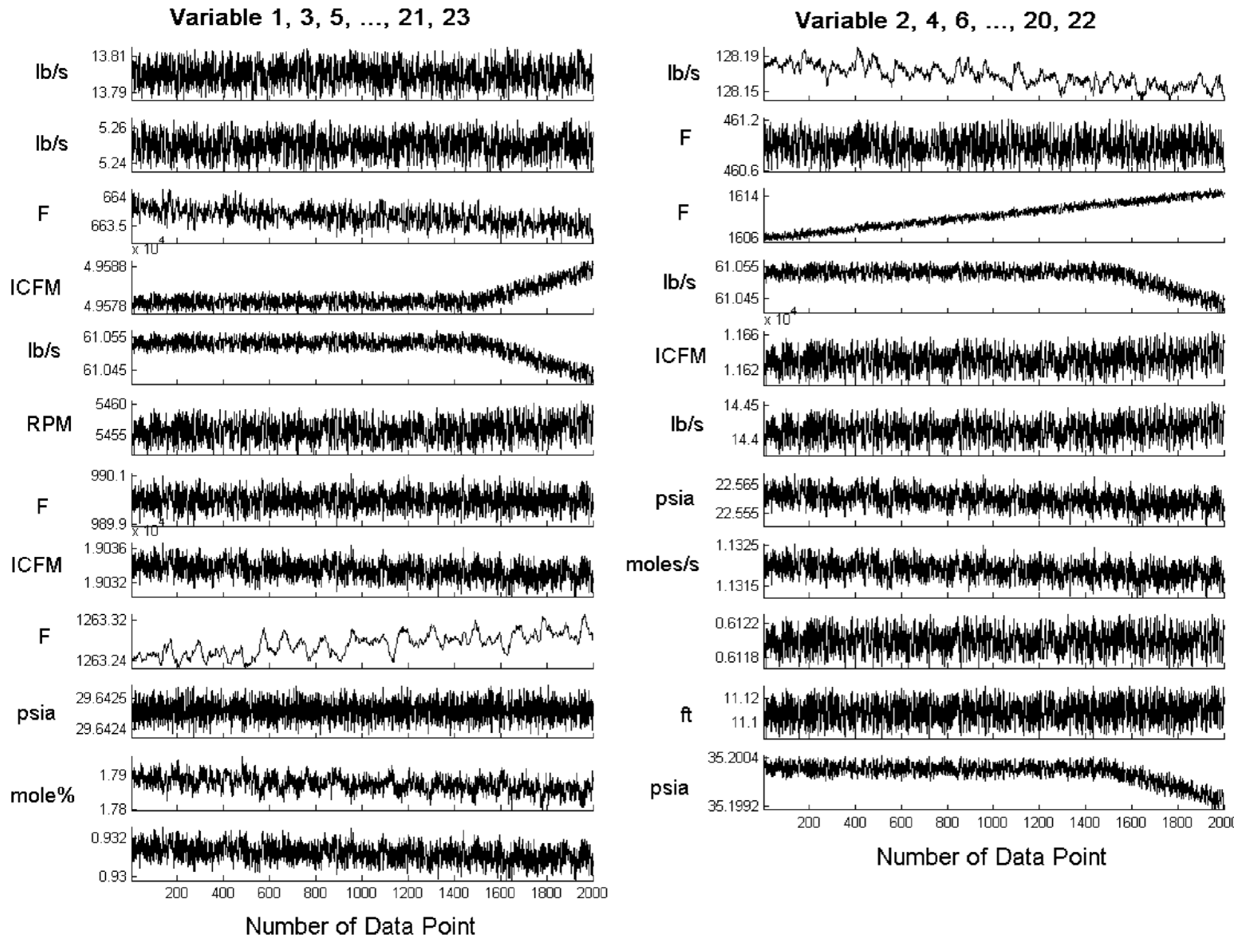

Figure 9. Recorded signals of FCCU simulation.

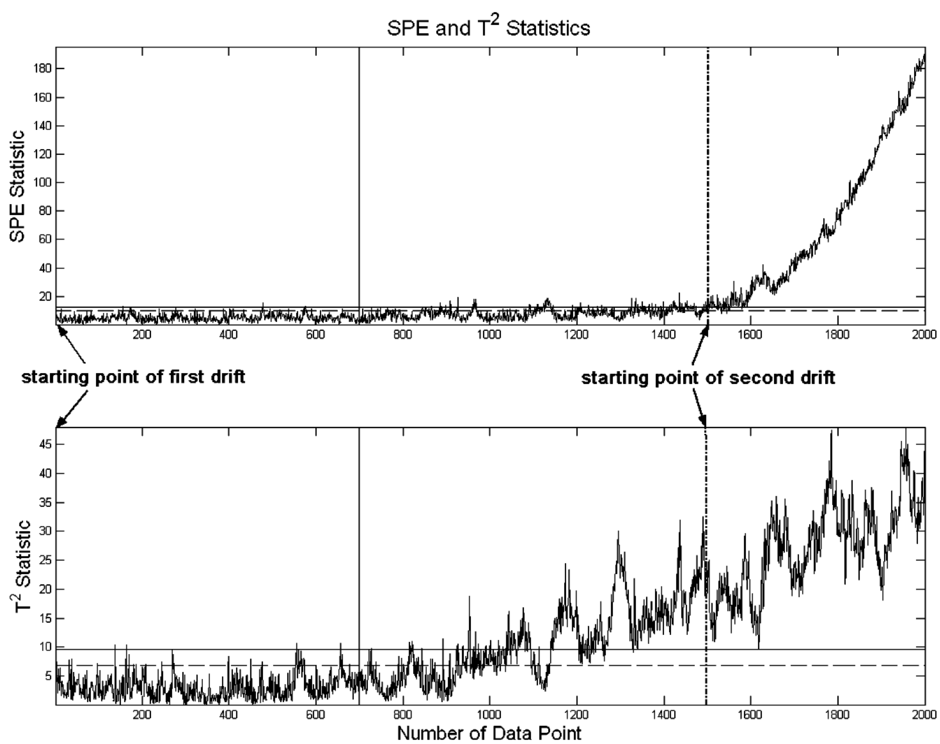

Figure 10. Monitoring charts for conventional PCA on FCCU data.

in the previous study. From the analysis in Section 4, note that the new MWPCA algorithm is >10 times faster than the conventional one in this case. As the window moved along the remaining 1300 data points, the fast MWPCA approach first scaled the data within each window frame to zero mean and unity one. The PCA model was then updated using this scaled data. For one-step-ahead prediction, the previous PCA model

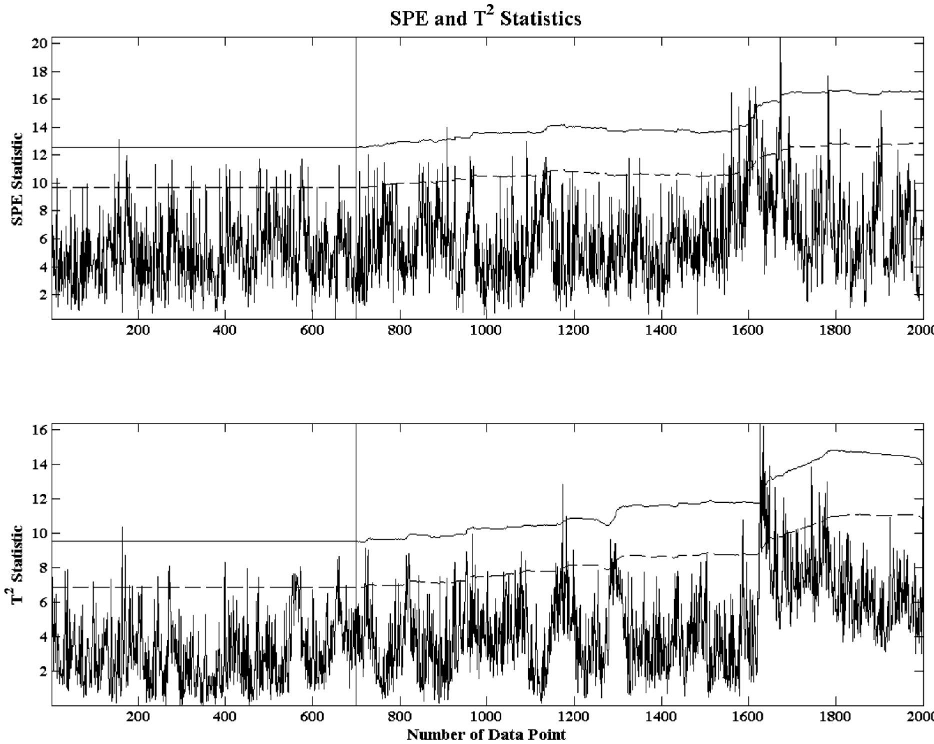

Figure 11. Monitoring charts for MWPCA using one-step-ahead prediction on FCCU data.

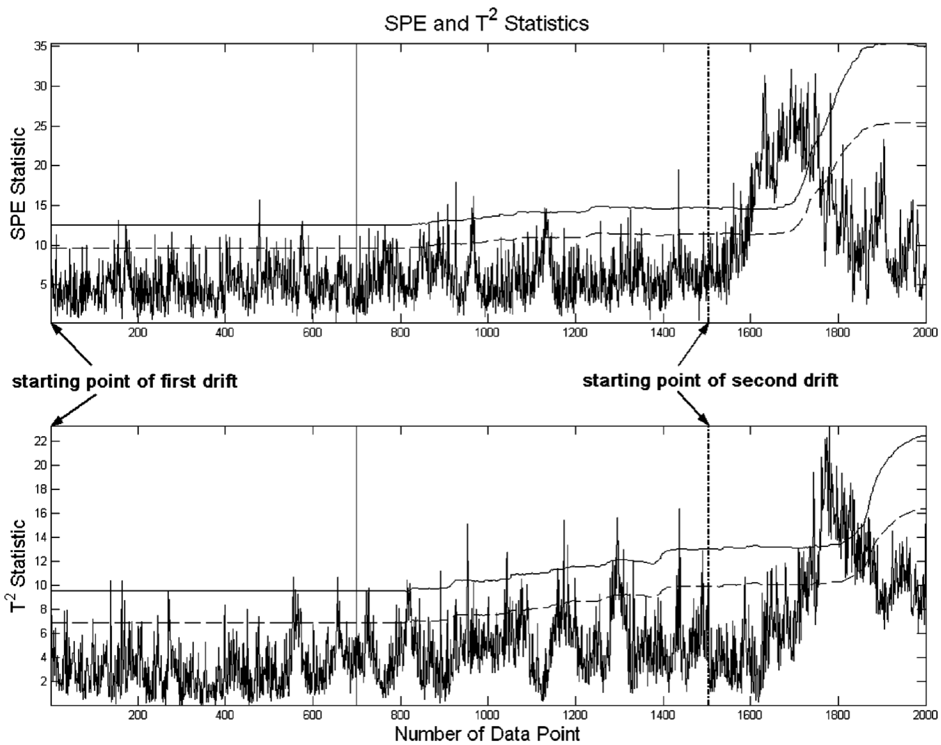

Figure 12. Monitoring charts for MWPCA using *N*-step-ahead prediction on FCCU data.

was used to compute the monitoring statistics and the confidence limits for the current sample.

The results are shown in Figure 11. The impact of the first drift was now completely accommodated, in contrast to conventional PCA (Figure 10). However, the impact of the second drift, which the monitoring approach should identify as abnormal behavior, was not clearly detected. More precisely, although both statistics showed sporadic violations at around the 1600th sample, no significant response to the second ramp was experienced. In fact, this fault could easily have been overlooked, since only a very few observations violated the T^2 or *SPE* control limits over a short period of time. After around 1650 samples, the MWPCA model adapted to the impact of the second drift completely, and hence, this event remained unnoticed thereafter.

7.2.3. MWPCA Based on N-Step-Ahead Prediction. Compared to MWPCA with one-step-ahead prediction, the use of MWPCA with an *N*-step-ahead prediction can slow the model adaptation to gradual drifts, since a much older PCA model is utilized. The window length was again selected to be 700.

In this final study, the horizon N was set at 100 and the monitoring results appear in Figure 12. Both statistics remained normal during the first drift, as required. As the second drift occurred, the *SPE* statistic demonstrated violations of control limits around data point 1600. This abnormal event impacted >150 data points. Given that a total of 100 T^2 and *SPE* values were computed using a PCA model that was not updated using data corresponding to the second ramp, both univariate statistics were significantly more sensitive compared to those for a one-step-ahead prediction. It is important to note, however, that if the adaptation of the PCA model is not suppressed after the second drift fault has been detected, the PCA model will adapt to this drift. This is noticeable after about the 1750th and 1800th samples, where the *SPE* and T^2 statistics, respectively, proved to be insensitive to this event. Finally, a considerable increase in the confidence limits of both statistics was also produced with a lag of ~100 samples, again highlighting the increased sensitivity of the N-step-ahead prediction as compared to the conventional one-step-ahead prediction.

8. Conclusions

This paper introduced a new fast MWPCA technique that enables on-line application of MWPCA with a larger window length. The new MWPCA technique is derived from the principle of a recursively updated PCA model. A detailed analysis of the computational effort of the new MWPCA algorithm compared to that of the conventional MWPCA showed that the new one can be >100 times faster for a window size of 1000. This comparison was based on the adaptation of the mean and variance for each process variable and the recursive determination of the correlation matrix.

The paper has further focused on the issue of model adaptation to anomalous process behavior. To alleviate this unwanted situation, a much older process model was introduced to calculate univariate monitoring statistics and their confidence limits. A simple simulation example showed that such an N-step-ahead prediction could correctly detect behavior considered as abnormal, in contrast to the performance of the conventional one-step-ahead prediction.

To demonstrate that incipiently developing events, which are regarded as normal deterioration in process units, could be accommodated while also being able to detect abnormal butgradually developing faults, the results of a simulated application study were presented. Here, it was shown that a conventional PCA model failed to accommodate the normal performance deterioration, as expected, and can therefore not be relied upon; using MWPCA in association with conventional one-step-ahead prediction gave no persistent indication of the gradually developing fault and eventually accommodated this fault shortly after it was introduced; the use of MWPCA, in conjunction with the N-step-ahead prediction, was able to accommodate the performance deterioration in the heat exchanger, while being able to clearly detect the abnormal event.

Acknowledgment

The authors would like to acknowledge the financial support from Invest Northern Ireland (contract ST237). The authors would also like to thank Dr. R. C. McFarlane of Invensys Process Systems for his kind assistance in dealing with issues regarding the FCCU simulator.

Nomenclature

a, b = parameters of an ARMA filter
$\mathbf{1}_k$ = vector of ones, $\in\mathbb{R}^k$
$\mathbf{b}_k$ = mean of $\mathbf{X}_k^{o}$
$\tilde{\mathbf{b}}$ = mean of matrix II in fast MWPCA approach
$\Delta\mathbf{b}_{k+1} = \mathbf{b}_{k+1} - \mathbf{b}_k$
$\mathbf{I}_k$ = $k \times k$ identity matrix
k = time sample index
L = window length
L^* = window length criterion for using fast MWPCA approach
m = number of variables
N = prediction horizon
$\mathbf{P}$ = loading matrix for PCA model, $\in\mathbb{R}^{m\times m}$
$\mathbf{R}$ = correlation matrix, $\in\mathbb{R}^{m\times m}$
$\tilde{\mathbf{R}}$ = correlation matrix for matrix II in fast MWPCA approach
S = slope value
$\mathbf{X}_k^{o}$ = original data matrix containing data from time instant 1 to k
$\mathbf{X}_k$ = scaled matrix for $\mathbf{X}_k^{o}$
$\mathbf{x}_k^{o}$ = original data collected at time instant k
$\mathbf{x}_k$ = scaled vector for $\mathbf{x}_k^{o}$
$\mathbf{\Sigma}_k$ = diagonal matrix
$\tilde{\mathbf{\Sigma}}$ = diagonal matrix for matrix II in fast MWPCA approach
$\sigma_k(i)$ = the ith element on diagonal position of $\mathbf{\Sigma}_k$
$\tilde{\sigma}(i)$ = the ith element on diagonal position of $\tilde{\mathbf{\Sigma}}$
$\mathbf{\Lambda}$ = diagonal matrix storing the eigenvalues for the PCA model
υ = an ARMA signal
ϵ = noise signal
ξ = predictor variable
ψ = response variable

Literature Cited

(1) MacGregor, J. F.; Marlin, T. E.; Kresta, J. V.; Skagerberg, B. Multivariate statistical methods in process analysis and control. *AIChE Symposium Proceedings of the 4th International Conference on Chemical Process Control*; American Institute of Chemical Engineers: New York, 1991; Publ. No. P-67, p 79.

(2) Wise, B. M.; Gallagher, N. B. The process chemometrics approach to process monitoring and fault detection. *J. Process Control* **1996**, *6* (6), 329.

(3) Gallagher, N. B.; Wise, B. M.; Butler, S. W.; White, D. D.; Barna, G. G. Development and benchmarking of multivariate statistical process control tools for a semiconductor etch process: Improving robustness through model updating. *ADCHEM 1997, Proceedings*, Banff, Canada, June 9–11, 1997; Elsevier: New York, 1997; p 78.

(4) Wang, X.; Kruger, U.; Lennox, B. Recursive partial least squares algorithms for monitoring complex industrial processes. *Control Eng. Pract.* **2003,** *11* (6), 613.

(5) Helland, K.; Berntsen, H. E.; Borgen, O. S.; Martens, H. Recursive Algorithm for Partial Least Squares Regression. *Chemom. Intell. Lab. Syst.* **1991**, *14*, 129.

(6) Dayal, B. S.; MacGregor, J. F. Recursive exponentially weighted PLS and its applications to adaptive control and prediction. *J. Process Control* **1997**, *7* (3), 169.

(7) Qin, S. J. Recursive PLS Algorithms for Adaptive Data Modelling. *Comput. Chem. Eng.* **1998**, *22* (4/5), 503.

(8) Li, W.; Yue, H.; Valle-Cervantes, S.; Qin, S. J. Recursive PCA for adaptive process monitoring. *J. Process Control* **2000**, *10* (5), 471.

(9) Aravena, J. L. Recursive Moving Window DFT Algorithm. *IEEE Trans. Comput.* **1990**, *39* (1), 145.

(10) Fuchs, E.; Donner, K. Fast Least-Squares Polynomial Approximation in Moving Time Windows. *Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal*

Processing, ICASSP, Munich, Germany, April 21−24, 1997; IEEE: Piscataway, NJ, 1997; Vol. 3, p 1965.

(11) Jackson, J. E. Principal Components and Factor Analysis. Part 1: Principal Analysis. *J. Qual. Technol.* **1980**, *12*, 201.

(12) MacGregor, J. F.; Kourti, T. Statistical process control of multivariate processes. *Control Eng. Pract.* **1995**, *3* (3), 403.

(13) McFarlane, R. C.; Reineman, R. C.; Bartee, J. F.; Georgakis, C. Dynamic simulator for a model IV fluid catalytic cracking unit. *Comput. Chem. Eng.* **1993,** *17* (3), 275.

(14) Golub, G. H.; van Loan, C. F. Matrix computation, 3rd ed.; John Hopkins University Press: Baltimore, MD, 1996.

(15) van Huffel, S.; Vandewalle, J. The total least squares problem. Society for Industrial and Applied Mathematics: Philadelphia, PA, 1991.

Received for review November 22, 2004
Revised manuscript received May 6, 2005
Accepted May 6, 2005

IE048873F