Ind. Eng. Chem. Res. **2008,** *47,* 419−427

Variable MWPCA for Adaptive Process Monitoring

Xiao Bin He and Yu Pu Yang*

Department of Automation, Shanghai Jiao Tong University, 800 Dongchuan Road, Min Hang District, Shanghai, China

An adaptive process monitoring approach with variable moving window principal component analysis (variable MWPCA) is proposed. On the basis of recursively updating the correlation matrix in both samplewise and blockwise manners, the approach combines the moving window technique with the classical rank-*r* singular value decomposition (R-SVD) algorithm to construct a new PCA model. Compared with previous MWPCA algorithms, the method not only improves the computation efficiency but also reduces the storage requirement. Furthermore, instead of a fixed window size, a variable moving window strategy is described in detail for accommodating normal process changes with different changing rates. The proposed method is applied to an illustrative case and a continuous stirred tank reactor process, and the monitoring results show better adaptability to both a slow drift and a set-point change than the results of using the conventional MWPCA with a fixed window size.

1. Introduction

A multivariate statistical process control based on principal component analysis (PCA) has been successfully applied to on-line continuous and batch process monitoring.[1−3] As a multivariate statistical projection technique, PCA extracts the first few principal components from highly correlated process data to capture the key process information. Typically, two major monitoring indices: Hotelling T^2 and the squared prediction error (SPE), are used to detect abnormal situations.[4,5]

Under the assumption that the industrial process is stationary or time-invariant, the static PCA used for process monitoring is reasonable. However, the time-varying character of most industrial processes always violates the assumption. When the PCA model for some particular process conditions is used to monitor these processes with normal changes, a number of false alarms often arise.[6]

To address the challenge, recently, several adaptive PCA schemes have been proposed. Wold[7] proposed EWM-PCA with the use of a exponentially weighted moving average (EWMA) filter to re-estimate the mean and variance for updating PCA models. On the basis of the same EWMA technique, recursive PCA[8] (RPCA) efficiently updates the model by recursively calculating the correlation matrix. Although its recursive characteristic is suitable for on-line process monitoring, one limitation is that the data set for updating the model is ever increasing, leading to a reduction in the speed of adaptation. In addition, RPCA includes ever-growing old data that are increasingly unrepresentative of the time-varying process, resulting in a decrease in model precision. In contrast, for the MWPCA based on the simpler concept of a moving window filter, the moving window adds the newest data that are more representative of the current process operation and discards those oldest data along the sample data.[9,10] By including sufficient new samples, MWPCA keeps good efficiency and precision of model adaptation. Recently, Wang[11] proposed a fast MWPCA that significantly increases computational efficiency by employing the recursive technique in RPCA to the conventional MWPCA. However, we still lack an efficient algorithm suitable for MWPCA to update the eigenvalues and eigenvectors of the correlation matrix in both samplewise and blockwise manners. Moreover, similar to the forgetting factor in RPCA, an adaptive mechanism to select the optimal moving window size is indispensable.

The paper focuses on developing a variable MWPCA algorithm and makes the following contributions: first, since a PCA model is constructed from the correlation matrix of the sample data, recursive schemes are introduced to update the correlation matrix in samplewise and blockwise approaches. Second, by blending the classical R-SVD algorithm and the moving window technique, a new MWPCA algorithm that increases the computation efficiency and decreases the storage requirement is discussed. Finally, an effective formula is derived to select the optimal length of the moving window. Instead of a fixed length, the window size should vary according to how fast the normal process can change.

The rest of the paper is structured as follows. We offer an efficient recursive calculation for the sample mean, variance, and correlation matrix in section 2. Section 3 presents the new algorithm to update a PCA model. A variable moving window strategy is proposed in section 4. Then, section 5 gives an introduction to an overall adaptive monitoring procedure using the variable MWPCA. In section 6, a practical guidance to select the parameters affecting the window size is described with a simple example, where the monitoring results of the proposed method are also evaluated. Section 7 illustrates an application to a continuous stirred tank reactor (CSTR) process, and section 8 concludes the paper.

2. Recursive Calculation for the Correlation Matrix

The recursive calculation for the current correlation matrix is achieved efficiently using the previous one rather than using the old process data.[8,11] Similar to the two-step procedure of the conventional MWPCA: discarding the oldest sample from the correlation matrix and adding the newest sample into the matrix, the recursive calculation procedure includes two main parts: downdating and updating. The details are expatiated in a blockwise approach. The samplewise approach is a special case of the blockwise approach in that the block size is set to be one.

* To whom correspondence should be addressed. Tel: +86-21-34204261. Fax: +86-21-34204427. Email: ypyang@sjtu.edu.cn (Y. P. Yang).

10.1021/ie070712z CCC: $40.75
Published on Web 12/13/2007

At time instant k, let $\mathbf{X}_k^0 \in \mathscr{R}^{L\times m}$ be the data matrix I of L samples (the window size or rows) and m variables (columns). Assume that $\mathbf{b}_k \in \mathscr{R}^{1\times m}$,$\sigma_k \in \mathscr{R}^{1\times m}$, $\mathbf{X}_k \in \mathscr{R}^{L\times m}$, and $\mathbf{R}_k \in \mathscr{R}^{m\times m}$ are the mean vector, the standard deviation vector, the scaled data matrix and the correlation matrix of matrix I, respectively. The task for the recursive calculation is to calculate $\mathbf{b}_{k+1}$, σ_{k+1}, $\mathbf{X}_{k+1}$, and $\mathbf{R}_{k+1}$ when the next block of data $\mathbf{x}_{\mathrm{new}}^0 \in \mathscr{R}^{n_{k+1}\times m}$ is available at time instant $k+1$.

2.1. Downdating. Downdating is the procedure where matrix I becomes the matrix II $\tilde{\mathrm{X}}^0 \in \mathscr{R}^{(L-n_{k+1})\times m}$ by discarding the oldest data block $\mathbf{x}_{\mathrm{old}}^0 \in \mathscr{R}^{n_{k+1}\times m}$ from the moving window. Then, the mean vector $\tilde{\mathrm{b}}$ and variance of each process variable $\tilde{\sigma}_i^2$ ($i = 1,\cdots m$) can be calculated recursively as.[8,11]

$$\tilde{\mathbf{b}} = \frac{1}{L-n_{k+1}-1}(L\mathbf{b}_k - \mathbf{I}_{k+1}^T\mathbf{x}_{\mathrm{old}}^0) \quad (1)$$

$$\tilde{\sigma}_i^{\,2} = \frac{L-1}{L-n_{k+1}-1}\sigma_i^{\,2} - \frac{L-n_{k+1}}{L-n_{k+1}-1}\Delta\tilde{b}_i^{\,2} - \frac{1}{L-n_{k+1}-1}\|\mathbf{x}_{\mathrm{old}}^{\;0}(:,i) - \mathbf{I}_{k+1}b_k(i)\|^2 \quad (2)$$

where $\Delta\tilde{\mathbf{b}} = \mathbf{b}_k - \tilde{\mathrm{b}}$, $\mathbf{I}_{k+1} = [1,1,\cdots,1,1]^T \in \mathscr{R}^{n_{k+1}}$, $\mathbf{x}_{\mathrm{old}}^0(:,i)$ is the ith column of $\mathbf{x}_{\mathrm{old}}^0$, $\Delta\tilde{\mathrm{b}}_i$ and $b_k(i)$ are the ith column of $\Delta\tilde{\mathrm{b}}$ and $\mathbf{b}_k$, respectively.

The matrix II $\tilde{\mathrm{X}}^0$ scaled to zero mean and unit variance is given by

$$\tilde{\mathbf{X}} = (\tilde{\mathbf{X}}^0 - \mathbf{I}_{k+1}\tilde{\mathbf{b}})\tilde{\Sigma}^{-1} \quad (3)$$

where $\tilde{\Sigma} = \mathrm{diag}\,(\tilde{\sigma}_1,\cdots\tilde{\sigma}_m)$.

The recursive calculation of $\mathbf{R}_k$ is given by[8,11]

$$\mathbf{R}_k = \frac{L-n_{k+1}-1}{L-1}\Sigma_k^{-1}\tilde{\Sigma}\,\tilde{\mathbf{R}}\tilde{\Sigma}\,\Sigma_k^{-1} + \frac{L-n_{k+1}}{L-1}\Sigma_k^{-1}\Delta\tilde{\mathbf{b}}^T\Delta\tilde{\mathbf{b}}\Sigma_k^{-1} + \frac{1}{L-1}\mathbf{x}_{\mathrm{old}}^T\mathbf{x}_{\mathrm{old}} \quad (4)$$

For the covariance-based PCA, the variance scaling is dispensable since it affects only the relative weighting of each variable. As a result, we can use the initial variance to scale the data and do not update the variance.[8] Then, the correlation matrix of matrix II is calculated as

$$\tilde{\mathbf{R}} = \frac{L-1}{L-n_{k+1}-1}\mathbf{R_k} - \frac{L-n_{k+1}}{L-n_{k+1}-1}\Sigma_k^{-1}\Delta\tilde{\mathbf{b}}^T\Delta\tilde{\mathbf{b}}\Sigma_k^{-1} - \frac{1}{L-n_{k+1}-1}\mathbf{x}_{\mathrm{old}}^T\mathbf{x}_{\mathrm{old}} \quad (5)$$

2.2. Updating. When the new block of data $\mathbf{x}_{\mathrm{new}}^0 \in \mathscr{R}^{n_{k+1}\times m}$ is available at time instant $k+1$, we add the new block into matrix II. Then, matrix II becomes matrix III $\mathbf{X}_{k+1}^0 \in \mathscr{R}^{L\times m}$. We call this procedure "updating". Similar to the downdating procedure, mean vector $\mathbf{b}_{k+1}$ and correlation matrix $\mathbf{R}_{k+1}$ can be computed recursively.

$$\mathbf{b}_{k+1} = \frac{1}{L-1}(L\mathbf{b}_k + \mathbf{I}_{n_{k+1}}^T\mathbf{x}_{\mathrm{new}}^0) \quad (6)$$

$$\mathbf{R}_{k+1} = \frac{L-n_{k+1}-1}{L-1}\tilde{\mathbf{R}} + \frac{L-n_{k+1}}{L-1}\Sigma_k^{-1}\Delta\mathbf{b}_{k+1}^T\Delta\mathbf{b}_{k+1}\Sigma_k^{-1} + \frac{1}{L-1}\mathbf{x}_{\mathrm{new}}^T\mathbf{x}_{\mathrm{new}} \quad (7)$$

where $\Delta\mathbf{b}_{k+1} = \mathbf{b}_{k+1} - \tilde{\mathbf{b}}$.

Substituting eq 5 into eq 7, then

$$\mathbf{R}_{k+1} = \mathbf{R}_k - \frac{L-n_{k+1}}{L-1}\Sigma_k^{-1}\Delta\tilde{\mathbf{b}}^T\Delta\tilde{\mathbf{b}}\Sigma_k^{-1} - \frac{1}{L-1}\mathbf{x}_{\mathrm{old}}^T\mathbf{x}_{\mathrm{old}} + \frac{L-n_{k+1}}{L-1}\Sigma_k^{-1}\Delta\mathbf{b}_{k+1}^T\Delta\mathbf{b}_{k+1}\Sigma_k^{-1} + \frac{1}{L-1}\mathbf{x}_{\mathrm{new}}^T\mathbf{x}_{\mathrm{new}} \quad (8)$$

3. Updating the PCA Model

After the new correlation matrix is calculated recursively, a number of approaches can be used to calculate its eigenvalues and eigenvectors. Batch SVD is one of the simplest approaches but has low computation efficiency. More recently, Li et al.[8] introduced the rank-one modification algorithm and the Lanczos tridiagonalization algorithm in the samplewise and blockwise RPCA, respectively. Similar methods were also introduced into MWPCA.[11] However, all of the eigenpairs have to be calculated in the rank-one modification algorithm, although only a few principal eigenpairs of the collation matrix are needed, which is an excessive computational burden.[8] The number of principal components should be known in advance because the Lanczos algorithm is known to be inaccurate for the smaller singular values.[12] Moreover, these methods become memory consuming making them impractical when the number of measured variables is fairly large because they both require storing the full correlation matrix. Consequently, a new PCA model-updating algorithm is proposed. Because of the combination of the moving window technique and the classical R-SVD method, the algorithm improves the computational efficiency and reduces the storage requirement. Incremental PCA based on the R-SVD has been applied widely in the machine-learning community.[13−15] Exploiting the properties of orthonormal bases and block structure, the R-SVD algorithm computes the new eigenpairs efficiently.

Assuming that the kth PCA model can be constructed with the decomposition of matrix $\mathbf{R}_k$,

$$\mathbf{R}_k \approx \frac{1}{L}\mathbf{X}_k^T\mathbf{X}_k \approx \mathbf{U}_k\Lambda_k\mathbf{U}_k^T = \mathbf{U}_k\,diag\{\lambda_1,\lambda_2,\cdots\lambda_r\}\,\mathbf{U}_k^T \quad (9)$$

where $\mathbf{U}_k$ is the loading matrix at the k time point, λ_i are the eigenvalues of the covariance matrix in descending order, and r is the number of principal components of the kth PCA model.

Substituting eq 9 into eq 8,

$$\mathbf{R}_{k+1} = \mathbf{R}_k - \frac{L-n_{k+1}}{L-1}\Sigma_k^{-1}\Delta\tilde{\mathbf{b}}^T\Delta\tilde{\mathbf{b}}\Sigma_k^{-1} - \frac{1}{L-1}\mathbf{x}_{\mathrm{old}}^T\mathbf{x}_{\mathrm{old}} + \frac{L-n_{k+1}}{L-1}\Sigma_k^{-1}\Delta\mathbf{b}_{k+1}^T\Delta\mathbf{b}_{k+1}\Sigma_k^{-1} + \frac{1}{L-1}\mathbf{x}_{\mathrm{new}}^T\mathbf{x}_{\mathrm{new}} \approx [\mathbf{U}_k\ \Delta\hat{\mathrm{b}}\ \hat{\mathrm{x}}_{\mathrm{old}}\ \Delta\hat{\mathrm{b}}_{k+1}\ \hat{\mathrm{x}}_{\mathrm{new}}] \begin{pmatrix} \Lambda_k & & & & \mathbf{0} \\ & -\mathbf{1} & & & \\ & & -\mathbf{E}_{k+1} & & \\ & & & \mathbf{1} & \\ \mathbf{0} & & & & \mathbf{E}_{k+1} \end{pmatrix} [\mathbf{U}_k\ \Delta\hat{\mathbf{b}}\ \hat{\mathbf{x}}_{\mathrm{old}}\ \Delta\hat{\mathbf{b}}_{k+1}\ \hat{\mathbf{x}}_{\mathrm{new}}]^T = [\mathbf{U}_k\ \mathbf{B}]\mathbf{C}[\mathbf{U}_k\ \mathbf{B}]^T \quad (10)$$

where $\Delta\hat{\mathrm{b}} = \sqrt{(L-n_{k+1})/(L-1)}\ \Sigma_k^{-1}\Delta\tilde{\mathrm{b}}^T$, $\hat{\mathrm{x}}_{\mathrm{old}} = \sqrt{1/(L-1)}\ \mathbf{x}_{\mathrm{old}}{}^T$, $\Delta\hat{\mathrm{b}}_{k+1} = \sqrt{(L-n_{k+1})/(L-1)}\ \Sigma_k^{-1}\Delta\mathbf{b}_{k+1}^T$, $\hat{\mathrm{x}}_{\mathrm{new}} = \sqrt{1/(L-1)}\ \mathbf{x}_{\mathrm{new}}{}^T$, $\mathbf{B} = [\Delta\hat{\mathrm{b}}\ \hat{\mathrm{x}}_{\mathrm{old}}\ \Delta\hat{\mathrm{b}}_{k+1}\ \hat{\mathrm{x}}_{\mathrm{new}}]$, $\mathbf{C} = diag(\Lambda_k, -\mathbf{1},$

Table 1. Procedure to Update the PCA Model for the Proposed MWPCA[a]

step	equation	step	equation
1	$\tilde{\mathbf{b}} = (L\mathbf{b}_k - \mathbf{I}_{\text{old}}^T\mathbf{x}_{\text{old}}^0)/(L - n_{k+1} - 1)$	8	$\tilde{\mathbf{x}}_{\text{new}} = \sqrt{1/(L-1)}\mathbf{x}_{\text{new}}^T$
2	$\mathbf{b}_{k+1} = (L\tilde{\mathbf{b}} + \mathbf{I}_{\text{new}}^T\mathbf{x}_{\text{new}}^0)/(L - \mathbf{1})$	9	$\mathbf{B} = [\Delta\hat{\mathbf{b}}\hat{\mathbf{x}}_{\text{old}}\ \Delta\hat{\mathbf{b}}_{k+1}\ \hat{\mathbf{x}}_{\text{new}}]$
3	$\tilde{\mathbf{X}} = (\tilde{\mathbf{X}}^0 - \mathbf{I}_{k+1}\tilde{\mathbf{b}})\tilde{\Sigma}^{-1}$	10	$\mathbf{C} = diag(\Lambda_k, -\mathbf{1}, -\mathbf{E}_{k+1}, \mathbf{1}, \mathbf{E}_{k+1})$
4	$\mathbf{X}_{k+1} = (\mathbf{X}_{k+1}^0 - \mathbf{I}_{k+1}\mathbf{b}_{k+1})\Sigma_{k+1}^{-1}$	11	$\mathbf{Q}_{\mathbf{B}}\mathbf{G}_{\mathbf{B}} \xleftarrow{QR} (\mathbf{E}_{n_0} - \mathbf{U}_k\mathbf{U}_k^T)\mathbf{B}$
5	$\Delta\tilde{\mathbf{b}} = \sqrt{(L-n_{k+1})/(L-1)}\Sigma_k^{-1}(\mathbf{b}_k - \tilde{\mathbf{b}})^T$	12	$\mathbf{G}_{k+1} = \begin{pmatrix} \mathbf{E} & \mathbf{U}_k^T\mathbf{B} \\ \mathbf{0} & \mathbf{G}_{\mathbf{B}} \end{pmatrix}$
6	$\Delta\tilde{\mathbf{b}}_{k+1} = \sqrt{(L-n_{k+1})/(L-1)}\Sigma_k^{-1}(\mathbf{b}_{k+1} - \tilde{\mathbf{b}})^T$	13	$\tilde{\mathbf{U}}\tilde{\mathbf{\Lambda}}\tilde{\mathbf{U}}^T \xleftarrow{SVD} \mathbf{G}_{k+1}\mathbf{C}\mathbf{G}_{k+1}^T$
7	$\hat{\mathbf{x}}_{old} = \sqrt{1/(L-1)}\mathbf{x}_{old}^T$	14	$\mathbf{U}_{k+1} = [\mathbf{U}_k\mathbf{Q}_{\mathbf{B}}]\tilde{\mathbf{U}}, \mathbf{\Lambda}_{k+1} = \tilde{\Lambda}$

$-\mathbf{E}_{k+1}, \mathbf{1}, \mathbf{E}_{k+1})$, $\mathbf{E}_{k+1} \in \mathscr{R}^{n_{k+1}\times n_{k+1}}$is an unit matrices, and $\mathbf{x}_{\text{new}}$ is the scaled block data of $\mathbf{x}_{\text{new}}^0$.

These columns of $\mathbf{B}$ are added into $\mathbf{U}_k$ and they disturb the orthogonality of $\mathbf{U}_k$. To restore the orthogonality, the decomposition of the vectors of $\mathbf{B}$ should be made into two components: $\mathbf{B}_{\mathbf{U}_k^\perp}$ and $\mathbf{B}_{\mathbf{U}_k}$.[16] $\mathbf{B}_{\mathbf{U}_k^\perp}$ and $\mathbf{B}_{\mathbf{U}k}$ are orthogonal and parallel to the $\mathbf{U}_k$ in $\mathbf{U}_k^\perp$ (the orthogonality complement space of $\mathbf{U}_k$), respectively. So

$$\mathbf{B} = \mathbf{B}_{\mathbf{U}_k^\perp} + \mathbf{B}_{\mathbf{U}_k} = (\mathbf{E}_{n_0} - \mathbf{U}_k\mathbf{U}_k^T)\mathbf{B} + \mathbf{U}_k\mathbf{U}_k^T\mathbf{B} \qquad (11)$$

where $\mathbf{E}_{n_0} \in \mathscr{R}^{n_0\times n_0}$ is a unit matrix and $n_0 = 2(1 + n_{k+1})$.

$\mathbf{B}_{\mathbf{U}_k}$ makes the eigenvectors rotated, but $\mathbf{B}_{\mathbf{U}_k^\perp}$ increases the rank of the eigenspace $\mathbf{U}_k$. To make the column vectors of $\mathbf{B}_{\mathbf{U}_k^\perp}$ mutually orthogonal, we computed the QR decomposition $\mathbf{Q}_{\mathbf{B}}\mathbf{G}_{\mathbf{B}} \xleftarrow{QR} \mathbf{B}_{\mathbf{U}_k^\perp}$.[16] Then

$$[\mathbf{U}_k\ \mathbf{B}] = [\mathbf{U}_k\ \mathbf{Q}_B]\begin{pmatrix} \mathbf{E} & \mathbf{U}_k^T\mathbf{B} \\ \mathbf{0} & \mathbf{G}_{\mathbf{B}} \end{pmatrix} = :\mathbf{U}'_{k+1}\ \mathbf{G}_{k+1} \qquad (12)$$

and

$$\mathbf{R}_{k+1} \approx \mathbf{U}'_{k+1}\,(\mathbf{G}_{k+1}\mathbf{C}\mathbf{G}_{k+1}^T)\mathbf{U}'^{\,T}_{k+1} \qquad (13)$$

To obtain the optimal eigenvectors, the singular value decomposition of the small matrix $\mathbf{G}_{k+1}\mathbf{C}\mathbf{G}_{k+1}^T$ is necessary.

$$\mathbf{R}_{k+1} \approx \mathbf{U}'_{k+1}(\tilde{\mathbf{U}}\tilde{\Lambda}\tilde{\mathbf{U}}^T)\mathbf{U}'^{\,T}_{k+1} = \mathbf{U}_{k+1}\Lambda_{k+1}\mathbf{U}_{k+1}^{\,T} \qquad (14)$$

where $\mathbf{U}_{k+1} = \mathbf{U}'_{k+1}\tilde{\mathbf{U}}$ and $\Lambda_{k+1} = \tilde{\Lambda}$.

To sum up, the PCA model at time instant $k+1$ can be efficiently constructed by using the smaller matrices and the SVD of $\mathbf{G}_{k+1}\mathbf{C}\mathbf{G}_{k+1}^T$. The computation complexity is $O(m(r + 2n_{k+1} + 2)^2)$ and the storage requirement is $O(m(r + 2n_{k+1} + 2))$.[17] The procedure to update the PCA model is summarized in Table 1 for convenience.

In practice, a numerical error that makes $\mathbf{U}_k$ and $\mathbf{Q}_{\mathbf{B}}$ not exactly orthogonal should be taken into consideration. Applying modified Gram−Schmidt orthogonalization to $\mathbf{Q}_{\mathbf{B}}$ may make the algorithm numerically robust when the inner product of its first and last columns is slightly away from zero.[15]

4. Variable Moving Window

In MWPCA, how to choose an optimal size of the moving window is another issue. Generally, the window size should be large enough to cover sufficient sample data for modeling and monitoring. However, the large window size leads to a significant reduction in the computational efficiency. Furthermore, when the process changes rapidly, the window covers too much outdated sample data so that MWPCA fails to exactly track the process change. Although a smaller window size can enhance computational efficiency, data within the window may not properly represent the underlying relationships between the process variables. Besides, the too small window may result in a potential danger in that the generated model may adapt so quickly to process changes that abnormal behavior is undetected.[11] Up to now, most model-updating approaches have empirically used a fixed window size. Similar to the forgetting factor in RPCA or RPLS, the window size is a tunable parameter that varies depending on how fast the normal process can change. When the process changes rapidly, the window size should be small; and vice versa, when the change becomes slow, the window size should be large. To the best of our knowledge, there is little concern for how to determine the optimal size of a moving window. But many literatures[18,19] have reported algorithms on the variable forgetting factor in RPCA and RPLS. In particular, Choi et al.[20] proposed an algorithm for the variable forgetting factors for updating the mean and covariance. One merit is that the forgetting factor is contingent upon the changes of the mean and covariance structures that directly represent process change, unlike those previous approaches in which the forgetting factor depends on Hotelling's T^2 and the SPE statistic. This idea is extended to the variable moving window in the article.

Typically, Hotelling's T^2 and the SPE statistic are used as two monitoring indices. In this sense, their variation shows all of the information about the process change. Furthermore, the variation on the monitoring plots of SPE and Hotelling's T^2 is owed to the abnormal fluctuation of their expectation values: E(SPE) and $E(T^2)$.[21]

Let $\mathbf{U}_e = [\mathbf{U}_{r+1}, \mathbf{U}_{r+2}, \cdots, \mathbf{U}_m]$ and $\mathbf{U}_t = [\mathbf{U}_1, \mathbf{U}_2, \cdots, \mathbf{U}_r]$, then

$$E(\text{SPE}) = tr(\mathbf{U}_e^T\mathbf{R}\mathbf{U}_e) + E(\mathbf{X}^0)\mathbf{U}_e\mathbf{U}_e^T\mathbf{E}^T(\mathbf{X}^0) \qquad (15)$$

and

$$E(T^2) = tr(\Lambda^{-1}\mathbf{U}_t^T\mathbf{R}\mathbf{U}_t) + E(\mathbf{X}^0)\mathbf{U}_t\Lambda^{-1}\mathbf{U}_t^T\mathbf{E}^T(\mathbf{X}^0) \qquad (16)$$

Because E(SPE) and $E(T^2)$ are the functions of the mean vector $E(\mathbf{X}^0)$ and the correlation matrix $\mathbf{R}$, for a given statistic model, both the mean vector and the correlation matrix changes are the essential factors representing the process change.[21] The window size for MWPCA at time point k is calculated as

$$L_k = L_{\min} + (L_{\max} - L_{\min})\exp\left\{-\left(\alpha\frac{||\Delta\mathbf{b}_k||}{||\Delta\mathbf{b}_0||} + \beta\frac{||\Delta\mathbf{R}_k||}{||\Delta\mathbf{R}_0||}\right)^{\gamma}\right\} \qquad (17)$$

where $L_{\max}$ and $L_{\min}$ are the maximum and minimum values of the moving window, respectively. α, β, and γ are three function

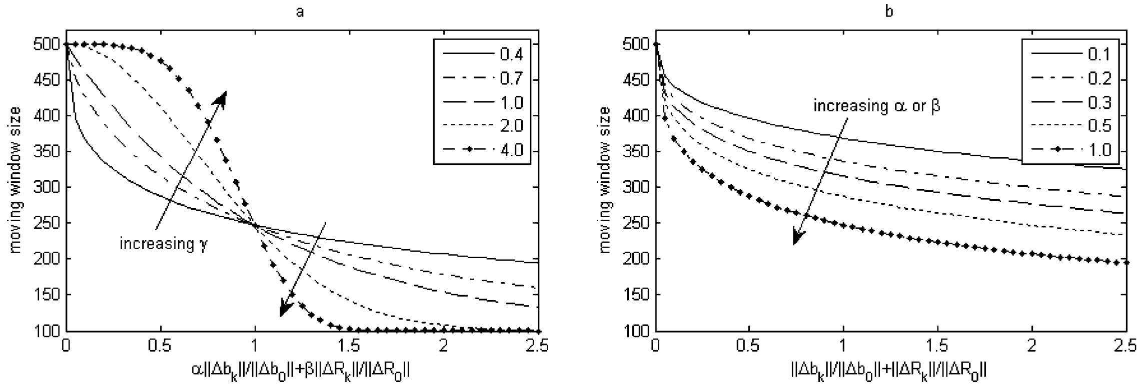

Figure 1. Effect of function parameters (α, β, and γ) on the variable window size: (a) effect of γ, (b) effect of α or β. Here, assume that $L_{max} = 500$, $L_{min} = 100$, $\alpha = \beta$, and $\gamma = 0.4$.

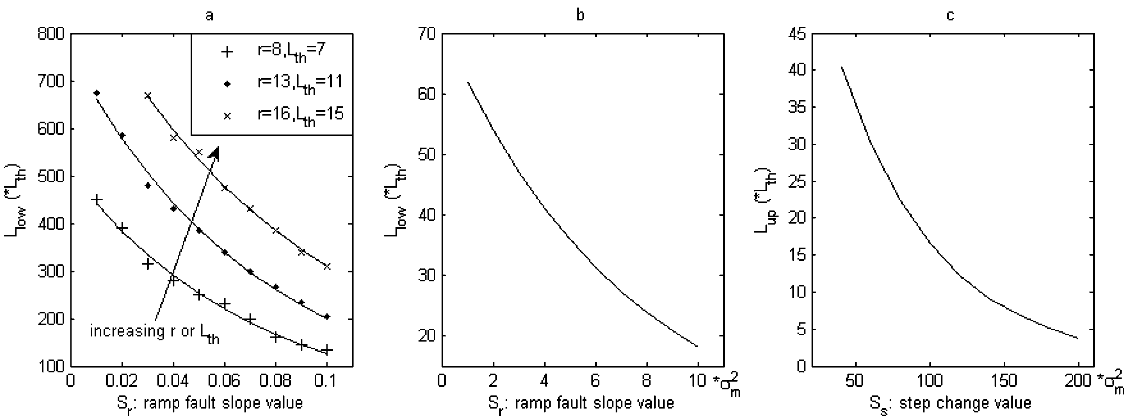

Figure 2. Guidance to select L_{max} and L_{min}: (a) The lower limit value L_{low} of the window size to detect the corresponding ramp fault, (b) the lower limit value of the window size (multiples of L_{th}) to detect the corresponding ramp fault (multiples of σ_m^2), and (c) the upper limit value of the window size (multiples of L_{th}) to accommodate the corresponding step change (multiples of σ_m^2).

parameters, and $\|\Delta \mathbf{b}_k\| = \|\mathbf{b}_k - \mathbf{b}_{k-1}\|$ is the Euclidean vector norm of difference between two consecutive mean vectors: $\mathbf{b}_k$ and $\mathbf{b}_{k-1}$. Similarly, $\|\Delta \mathbf{R}_k\| = \|\mathbf{R}_k - \mathbf{R}_{k-1}\|$ is the Euclidean matrix norm of the difference between two consecutive correlation matrices: $\mathbf{R}_k$ and $\mathbf{R}_{k-1}$. $\|\Delta \mathbf{b}_0\|$ means the basic mean change in steady state, where there is not any operating mode change or abnormity. $\|\Delta \mathbf{b}_0\|$ is the averaged $\|\Delta \mathbf{b}\|$ obtained using a training data set. Similarly, $\|\Delta \mathbf{R}_0\|$ is the basic correlation matrix change. $\|\Delta \mathbf{b}_k\|$ and $\|\Delta \mathbf{R}_k\|$ are normalized by dividing $\|\Delta \mathbf{b}_0\|$ and $\|\Delta \mathbf{R}_0\|$ respectively so that they have the same dimension. $\|\Delta \mathbf{b}_k\|/\|\Delta \mathbf{b}_0\|$ is the ratio of two slope values: $\|\Delta \mathbf{b}_k\|/\Delta t$ and $\|\Delta \mathbf{b}_0\|/\Delta t$. Here, Δt is the sampling interval. So, $\|\Delta \mathbf{b}_k\|/\|\Delta \mathbf{b}_0\|$ means the relative mean change rate. Similarly, $\|\Delta \mathbf{R}_k\|/\|\Delta \mathbf{R}_0\|$ means the relative correlation matrix change rate. Parts a and b of Figure 1 show the effects of three function parameters (α, β, and γ) on the window size. The sensitivity of the window size to the process change is determined by γ. The larger γ is, the more sensitive to the process change L_k is. As γ increases, L_k changes rapidly about the point (1, 247.16). When γ tends to be infinite, L_k becomes a binary variable. That is, L_k is equal to L_{max} when $\alpha\|\Delta \mathbf{b}_k\|/\|\Delta \mathbf{b}_0\| + \beta\|\Delta \mathbf{R}_k\|/\|\Delta \mathbf{R}_0\| > 1$; otherwise, L_k is equal to L_{min}. Parameters α and β are the impact factors of $\|\Delta \mathbf{b}_k\|/\|\Delta \mathbf{b}_0\|$ and $\|\Delta \mathbf{R}_k\|/\|\Delta \mathbf{R}_0\|$ on the process change, respectively. The effect of $\|\Delta \mathbf{b}_k\|/\|\Delta \mathbf{b}_0\|$ (or $\|\Delta \mathbf{R}_k\|/\|\Delta \mathbf{R}_0\|$) on the window size has a gradual increase with α (or β).

For the window size, L_{max} and L_{min} determine its adjustive range. When the process has an extreme slow change, that is, the change rates of the mean and correlation matrix tend to be zero ($\|\Delta \mathbf{b}_k\| \rightarrow 0$ and $\|\Delta \mathbf{R}_k\| \rightarrow 0$), the window size tends to the maximum value ($L \rightarrow L_{max}$); when the process experiences an extreme quick change, that is, the change rates of the mean and correlation matrix tend to be infinite ($\|\Delta \mathbf{b}_k\| \rightarrow \infty$ and $\|\Delta \mathbf{R}_k\| \rightarrow \infty$), and the window size tends to be the minimum value ($L \rightarrow L_{min}$). The larger L_{max} is, the more sample data the moving window can include and the better ability to detect those faults with less ramp slope the PCA model has; however, it results in the reduction of computation efficiency and the deterioration of model adaptability. The smaller L_{min} is, the better adaptability to more quick process changes the PCA model can show. However, the value of L_{min} should be large enough to prevent the covariance matrix from being inaccurate or unreliable. Without regard to serial correlation, the number of data points Sp for building a stable PCA model should be larger than or at least equal to the number of the independent parameters M_r in the model,[22] that is

$$Sp \geq Sp_{min} = M_r \tag{18}$$

where $Sp_{min} = L_{th} \times m$. L_{th} is the threshold value of the window size for building a stable PCA model. To a PCA model, we also have[22]

$$M_r = \frac{r + 2rm - r^2}{2} \tag{19}$$

So, L_{th} can be calculated as follows:

$$L_{th} = \frac{r + 2rm - r^2}{2m} \tag{20}$$

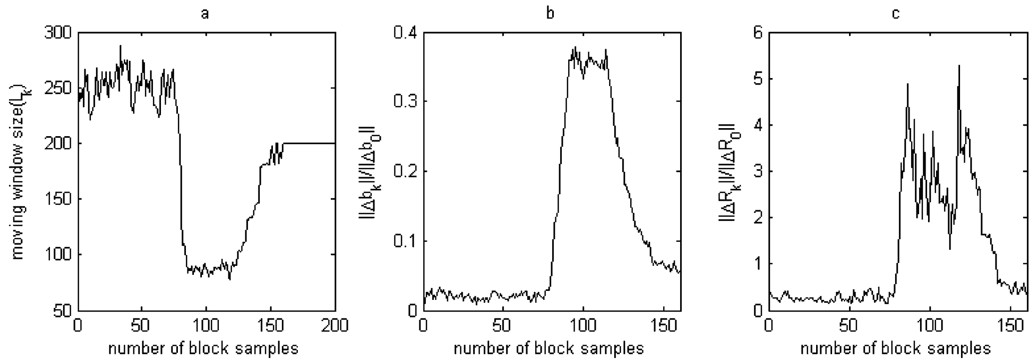

Figure 3. The moving window size and the relative changing rates of the mean and correlation matrix vary with time: (a) The moving window size, (b) the relative changing rate of the mean, and (c) the relative changing rate of the correlation matrix.

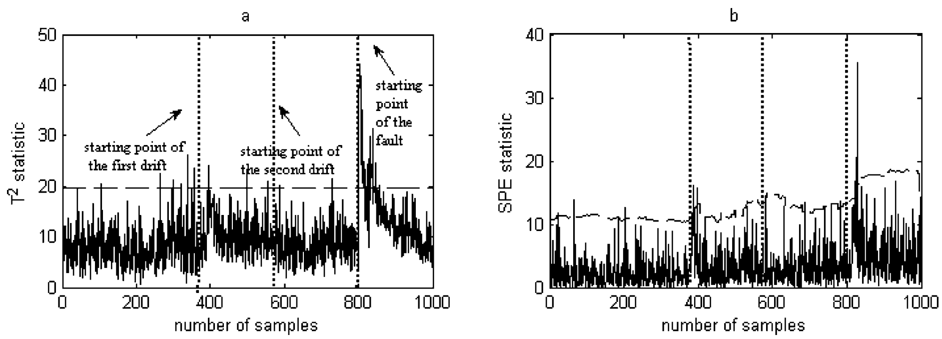

Figure 4. Process monitoring charts based on the variable MWPCA: (a) Hotelling's T^2 statistic, (b) SPE statistic.

Further practical guidance to select these parameters is discussed in detail in section 6 via a simulated example.

5. A Complete Adaptive Monitoring Procedure Using the Variable MWPCA

The overall procedure for adaptive process monitoring based on the variable MWPCA is as follows.

Off-line Learning:

(1) Calculate the initial value of the sample mean vector $\mathbf{b}_0$, the standard variance vector σ_0, and the PCA model (the loading matrix $\mathbf{U}_0$, the eigenvalue matrix Λ_0, and the number of principal components r) using the training data set.

(2) Calculate the two monitoring limits: $T^2_{\lim,0}$ and $Q_{\lim,0}$.

(3) Calculate $L_{\rm th}$ in terms of the values of r and m.

(4) Determinate $L_{\max}$ and $L_{\min}$, discussed in section 6

(5) Select α, β, and γ, discussed in section 6.

(6) Calculate $||\Delta\mathbf{b}_0||$ and $||\Delta\mathbf{R}_0||$. At first, calculate the initial value of $\mathbf{b}$ and $\mathbf{R}$ using the beginning part of the training data set with the window size $L = 0.5(L_{\max} + L_{\min})$. Then, as the window slides along the training data set in either the sample-wise or the blockwise approach, repeatedly calculate the values of $\mathbf{b}$, $\mathbf{R}$, $||\Delta\mathbf{b}||$, and $||\Delta\mathbf{R}||$. Finally, average all of the values of $||\Delta\mathbf{b}||$ and $||\Delta\mathbf{R}||$ to obtain $||\Delta\mathbf{b}_0||$ and $||\Delta\mathbf{R}_0||$, respectively.

On-line Monitoring:

Assume that these values: $\mathbf{b}_k$, σ_k, $\mathbf{R}_k$, $\mathbf{U}_k$, Λ_k, $T^2_{\lim,k}$, $Q_{\lim,k}$, and L_k are calculated at time instant k. The new block of data $\mathbf{x}^0_{new} \in \mathscr{R}^{n_{k+1}\times m}$ is available at time point $k + 1$.

(1) Calculate T^2_k and Q_k for new block $(\mathbf{x}^0_{k+1})_{n_{k+1}\times m}$ after normalization.

(2) If $T^2_k > T^2_{\lim,k}$ or $Q_k > Q_{\lim,k}$, judge whether the out-of-limit is due to outliers or faults. If it is due to outliers, calculate the robust estimated values of these outliers using the M-estimator[23] or other algorithms.[24,25] Otherwise, stop updating the model and trigger a process alarm as a result of an abnormal process situation.

(3) Determine L_{k+1} by calculating $||\Delta\mathbf{b}_{k+1}||/||\Delta\mathbf{b}_0||$ and $||\Delta\mathbf{R}_{k+1}||/||\Delta\mathbf{R}_0||$.

(4) If $L_{k+1} \le L_k$, discard the oldest $L_k - L_{k+1}$ sample data, then carry out the downdating and updating procedure using the new data block. Otherwise, only perform the updating procedure according to the new data block, and do not carry out the downdating procedure. Then set $L_{k+1} = L_k + n_{k+1}$.

(5) Update the PCA model based on the procedure in Table 1.

(6) Determine the number of principal components, and calculate two monitoring limits: $T^2_{\lim,k+1}$ and $Q_{\lim,k+1}$.

(7) Set $k = k + 1$ and go to step 2.

6. Illustrative Case study

The following multivariate static process[20] is considered: $\mathbf{x} = \Phi\omega$, where $\Phi \in \mathscr{R}^{30\times 10}$ is a normally distributed random transformation matrix with zero mean and unit variance, that is, $\Phi_{i,j} \in N\{0, 1\}$, and $\omega \in \mathscr{R}^{10}$ is a normally distributed random vector with zero mean and a variance of 0.01, that is, $\omega_i \in N\{0, 0.01\}$. Hence, the measured vector $\mathbf{x} \in \mathscr{R}^{30}$ also follows a normal distribution with zero mean and a variance of 0.01. In this study, 500 samples are generated to form the training data set in the normal process operation. In the test data set, 1000 samples are collected in total. These data are generated using *MATLAB 6.0* where different seeds are used for random number generation, and a blockwise updating approach is adopted and the block size is 5.

6.1. Practical Guidance to Select the Variable Window Parameters. For the window size, five parameters ($L_{\max}$, $L_{\min}$ α, β, and γ) need to be determined in eq 17. An analysis has been carried out to effectively select them.

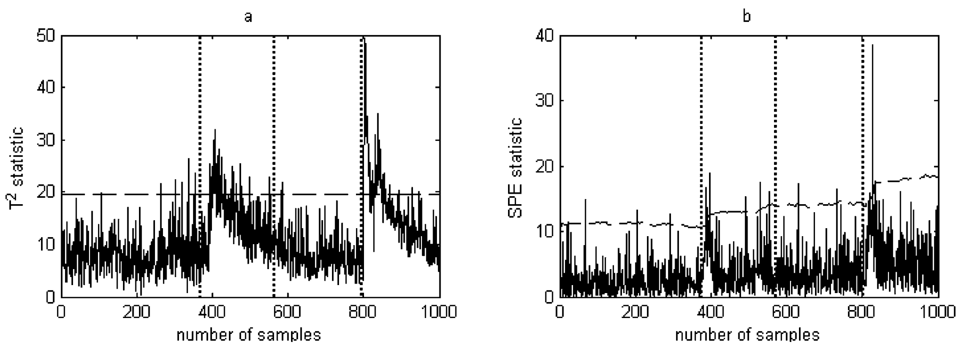

Figure 5. Process monitoring charts based on the MWPCA with a fixed moving window size: (a) Hotelling's T^2 statistic, (b) SPE statistic.

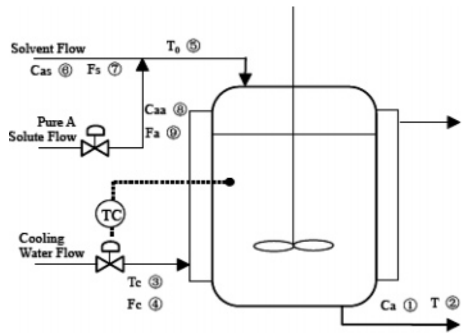

Figure 6. Nonisothermal CSTR process and measured variables: ①outlet concentration, ②outlet temperature, ③cooling water temperature, ④cooling water flow rate, ⑤inlet temperature, ⑥solvent concentration, ⑦solvent flow rate, ⑧reactant A concentration, and ⑨reactant A flow rate.

$L_{\max}$ and $L_{\min}$ affect the changing range of the window size. In a computational resource constraint, $L_{\max}$ should be large enough to make the window size sufficiently large for detecting slow ramp faults. Part a of Figure 2 shows that, as the ramp fault slope value S_r increases, the lower limit value of the window size L_{low} required also increases to detect the fault. It includes three curves with different experimental data. The data set in the curve 1 is generated in the simulated process described in the beginning part of the section. As S_r increases from 0.01 to 1 by the step of 0.01, the lower limit value L_{low} increases to 450 from 140 to detect the corresponding ramp fault. During the simulation, because the rank of the covariance matrix does not change over time, the number of principal components r is fixed at eight. Hence, the threshold value L_{th} is 7 in terms of eq 20. It should be noted that the curve is a fitting result based on the least-squares law. To analyze the influence of the number of measured variables or principal components on the lower limit value, two different data sets are generated via repairing the illustrative case. Through doubling the column of the random vector ω, that is, $\omega \in \mathscr{R}^{20}$, $r = 13$ and $L_{\text{th}} = 11$ are derived in the curve 2. In addition, $r = 16$ and $L_{\text{th}} = 15$ can be obtained in curve 3 by increasing the dimensions of the translation matrix to $\Phi \in \mathscr{R}^{40 \times 10}$. These curves show explicitly that, with the number of principal components increasing, the lower limit value L_{low} should increase to detect the same ramp fault. Nevertheless, the shapes of these curves stay mostly uniform. That is to say, to some fault slope value, the ratio of the lower limit value L_{low} to the threshold value L_{th} is a constant value despite the change in the number of principal components or the measured variables. Hence, three curves in part a of Figure 2 can be unified to a curve in part b of Figure 2, where the ramp fault slope value is described to be a multiple of the minimum variance of measured variables σ_m^2. Then, part b of Figure 2 presents the more general relation. To some slow ramp fault that we want to detect during process monitoring, the required maximum window size $L_{\max}$ should be larger than the lower limit value L_{low}. Considering the computation efficiency, generally, define $L_{\max} = \sim 60-80 L_{\text{th}}$.

At least, the value of $L_{\min}$ should be equal to the threshold value L_{th} to prevent the covariance matrix from being inaccurate or unreliable. In the same way, part c of Figue2 shows the upper limit value L_{up} required to accommodate the corresponding step change. The upper limit value L_{up} is represented by a multiple of L_{th} to eliminate the influence of the variation of the number of measured variables or principal components. Therefore, $L_{\min}$ should be less than the upper limit value L_{up} for adapting to the step change in the monitored process. In practice, $L_{\min} = \sim 4-6 L_{\text{th}}$.

As mentioned before, α and β are the weighted coefficients of $||\Delta \mathbf{b}_k||/||\Delta \mathbf{b}_0||$ and $||\Delta \mathbf{R}_k||/||\Delta \mathbf{R}_0||$, and γ affects the sensitivity to the process change. When $||\Delta \mathbf{b}_k||/||\Delta \mathbf{b}_0|| = 1$ and $||\Delta \mathbf{R}_k||/||\Delta \mathbf{R}_0|| = 1$, which indicates that the current mean and correlation matrix changes are the same as the basic mean and correlation matrix change, L is set to be a medial value between $L_{\max}$ and $L_{\min}$, that is, $L = L_{\min} + 0.5(L_{\max} - L_{\min})$. The value of γ should be less than unity to match the slow exponential reduction of the window size with increasing $||\Delta \mathbf{b}_k||/||\Delta \mathbf{b}_0||$ and $||\Delta \mathbf{R}_k||/||\Delta \mathbf{R}_0||$. Empirically, the default value of γ is taken as $\sim$0.4−0.6. To simplify these parameters' selection, assume that α equals to β. Then

$$\begin{Bmatrix} \alpha = \beta = 0.20, & \mathit{if}\ \gamma = 0.4 \\ \alpha = \beta = 0.24, & \mathit{if}\ \gamma = 0.5 \\ \alpha = \beta = 0.27, & \mathit{if}\ \gamma = 0.6 \end{Bmatrix} \tag{21}$$

It should be noted that the selecting procedure, that is, fixing one of two residual function parameters (α and γ) and varying the other, might be repeated iteratively.

6.2. Process Monitoring Using Variable MWPCA. In the unit, two slow drifts and a ramp fault are simulated. Two drifts are assumed to be known and to represent normal operation. One drift with a slow increment of 0.07 is added to x_1 after 370 data points, that is, $x_1(t + 1) = x_1(t) + 0.07(t - 370)$; another drift with a decrement of 0.06 is added to x_1 after 570 data points to counteract the first signal and form a slower increment of 0.01, that is, $x_1(t + 1) = x_1(t) + 0.07(t - 370) - 0.06(t - 570)$. The ramp fault, whose slope value is 0.06, is added to x_{11} after 800 data points, that is, $x_{11}(t + 1) = x_{11}(t) + 0.06(t - 800)$. After off-line learning, the parameters: $L_{\max}$, $L_{\min}$ α, β, and γ, are selected to be 500, 35, 0.2, 0.2, and 0.4, respectively.

Parts a−c of Figure 3 show the changes of the window size L_k, $||\Delta \mathbf{b}_k||/||\Delta \mathbf{b}_0||$ and $||\Delta \mathbf{R}_k||/||\Delta \mathbf{R}_0||$, respectively. As shown in part a of Figure 3, the window size becomes small to adapt to the slow drift and reverts to be larger when a slower drift happens at data point 570. When no normal

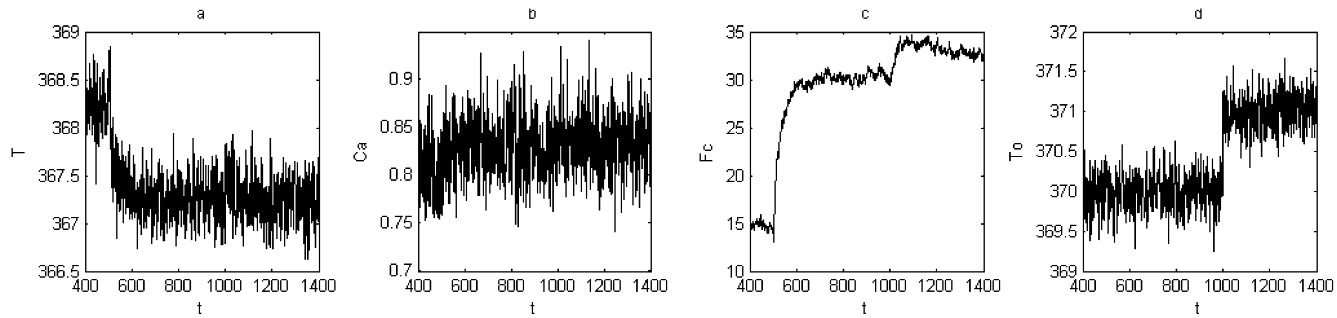

Figure 7. Time-series plots of measured variables: (a) the outlet temperature T, (b) the outlet concentration C_a, (c) the cooling water flow rate F_c, and (d) the inlet temperature T_0.

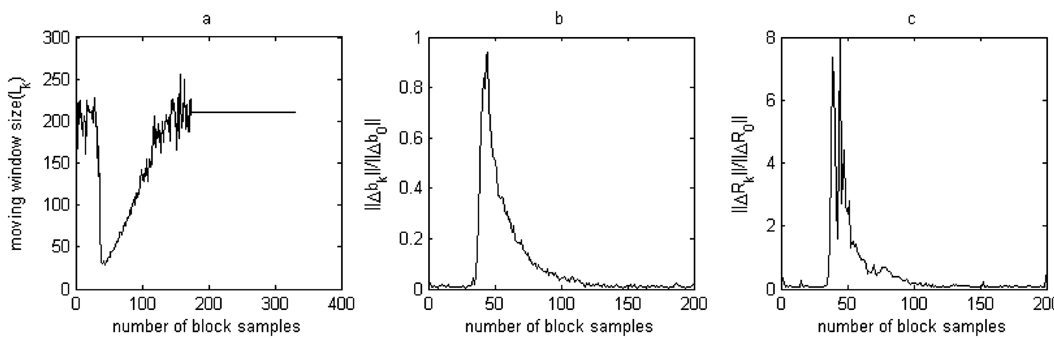

Figure 8. The moving window size and the relative changing rates of the mean and correlation matrix over time: (a) the moving window size, (b) the relative changing rate of the mean, and (c) the relative changing rate of the correlation matrix.

process changes are anticipated and the window size keeps relative stable, the update of the window size is automatically stopped in case the model accommodates faults. With the measured variable x_1 going though two normal drifts, $||\Delta \mathbf{b}_k||/||\Delta \mathbf{b}_0||$ and $||\Delta \mathbf{R}_k||/||\Delta \mathbf{R}_0||$ promptly track this normal process changes and increase rapidly to a larger value and become stable later, respectively. However, $||\Delta \mathbf{R}_k||/||\Delta \mathbf{R}_0||$ experiences a big fluctuation when the window size becomes small for adapting to process change. Therefore, it is advisable that the impact factor β should be less than the impact factor α to reduce the effect of the fluctuation on the window size. Note that parts b and c of Figure 3 only show the relative changing rates of the mean and correlation matrix in a normal situation because they become nonsensical in an abnormal situation.

The monitoring charts using the variable moving window strategy are shown in Figure 4. After two different drifts are introduced at the 370th and 570th sample points, the PCA model perfectly accommodates the drifts by adjusting the window size. Subsequently, the fault is clearly detected. The corresponding results with the window size fixed to be 300 are shown in Figure 5. The T^2 statistic shows a significant number of violations because the window size is too large to adapt to the first drift. In contrast to conventional PCA with a fixed window size, the variable MWPCA not only completely accommodates different normal drifts by adjusting the window size but also keeps enough capability to clearly detect faults by stopping the updating of the window size in case the model accommodates faults and fails to function.

There is a risk that the current model adapts to process faults or disturbances. It is important to discriminate between normal process changes and process faults. This, however, cannot be achieved with the use of data-driven methods alone. Priori process knowledge must be incorporated to identify normal process change from disturbance. The procedure to identify them is out of the scope of this paper. Further discussion can been found in refs 26 and 27.

7. Application to a Simulated Industrial Process

A nonisothermal continuous stirred tank chemical reactor model[28,29] is used to illustrate the proposed method for the adaptive process monitoring. The reacting system includes heat transfer and heat of reactor. The reaction is the first order ($A \rightarrow B$). Suppose that the volume and the physical properties are constant. The mathematic model consists of the component material balance on reactant A and the total energy balance as follows:

$$\frac{dC_{\mathrm{A}}}{dt} = \frac{F}{V}C_{\mathrm{A0}} - \frac{F}{V}C_{\mathrm{A}} - k_0 e^{-E/RT\,C_{\mathrm{A}}} \tag{22}$$

$$V\rho C_{\mathrm{p}}\frac{dT}{dt} = \rho C_{\mathrm{p}}F(T_0 - T) - \frac{\beta_{\mathrm{UA}} a F_C^{b+1}}{F_{\mathrm{C}} + \beta_{\mathrm{UA}} a F_{\mathrm{C}}^{b}/2\rho_{\mathrm{C}}C_{\mathrm{PC}}} \times (T - T_{\mathrm{C}}) + (-\Delta H_{r\times n})V\beta_{\mathrm{r}}k_0 e^{-E/RT\,C_A} \tag{23}$$

The schematic of the process and nine measured variables are given in Figure 6. The reactant and the cooling water flows control the outlet concentration and temperature, respectively. However, only the temperature controller is active in this application. All of the stochastic disturbances are constructed as first order autoregressive models. In addition, Gaussian white noises with different variances are added to all of the measured variables. More detailed model information and simulation conditions are provided in ref 28.

Different from the above example, a known set-point change in the reactor temperature and a bias fault in the inlet temperature are taken into account. The duration of the simulation is 1400 min. The set-point change and the bias fault are introduced at

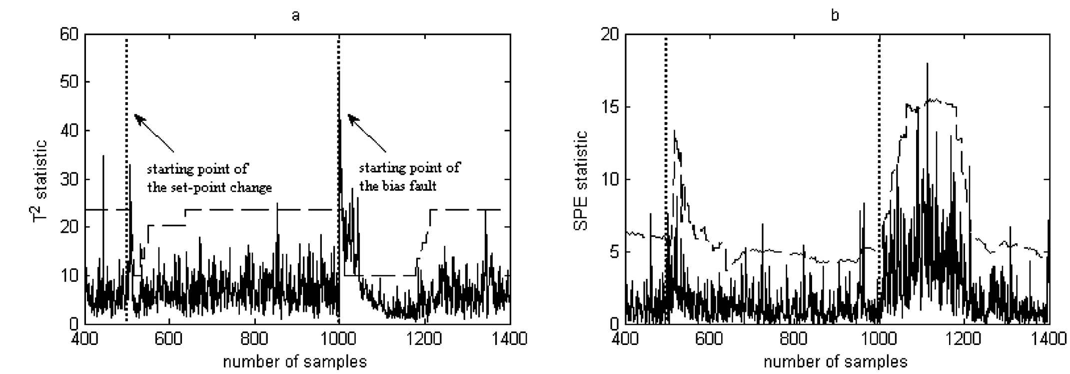

Figure 9. Process monitoring charts based on the variable MWPCA: (a) Hotelling's T^2 statistic, (b) SPE statistic.

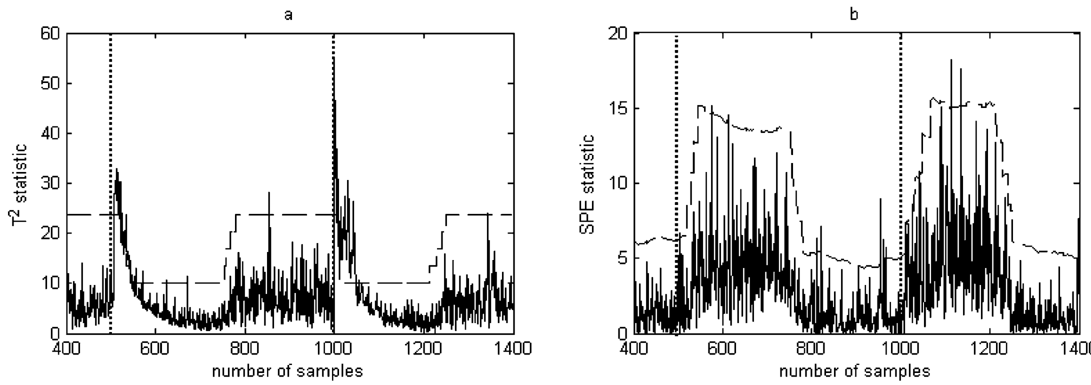

Figure 10. Process monitoring charts using the MWPCA with a fixed moving window size: (a) Hotelling's T^2 statistic, (b) SPE statistic.

t = 500 min and t = 1000 min, respectively. Nine process variables (C_a, T, T_c, F_c, T_0, C_{as}, F_s, C_{aa}, and F_a) are measured every minute. A data set of 1400 samples is generated. The training data set includes the first 400 samples (t = ~1−400 min). The other 1000 samples (t = ~ 401−1400 min) are used to update the model for on-line adaptive process monitoring. The blockwise updating approach is adopted, and the block size is 3.

Using off-line learning based on the training data set, the initial PCA model is obtained with the number of the principal components $r = 6$. Then, $L_{th} = 5$. According to the proposed guidance, the parameters L_{max}, L_{min} α, β, and γ are selected to be 400 (80 × L_{th}), 25 (5 × L_{th}), 0.3, 0.1, 0.4, respectively.

The reactor temperature decreases to 367.25 from 368.25 at t = 500 min. The cooling water flow rate is regulated to meet the requirement because of the close-loop control of the temperature. The set-point change also influences the outlet concentration. The bias of 1.0 °C is added to the inlet temperature starting at t = 1000 min. Apart from the cooling water flow rate, which is adjusted to meet the close-loop control requirement, other variables have no obvious change in the bias fault. Figure 7 illustrates the changes of four measured variables: T, C_a, F_c, and T_0.

The window size L_k and the changes of $||\Delta \mathbf{b}_k||/||\Delta \mathbf{b}_0||$ and $||\Delta \mathbf{R}_k||/||\Delta \mathbf{R}_0||$ are illustrated in parts a−c of Figure 8. As a result of the set-point change, $||\Delta \mathbf{b}_k||/||\Delta \mathbf{b}_0||$ and $||\Delta \mathbf{R}_k||/||\Delta \mathbf{R}_0||$ have a significant step change. After sharply reducing, the window size resumes a stable value. When no normal process changes are anticipated, stop updating the window size automatically and keep the window size at a constant value.

In Figure 9, the monitoring charts using the variable MWPCA show that the process remains in control after the step change has been introduced. The number of the principal components decreases to be 3 because the associated data (from the sample about 550 to 750) change abruptly. The effect of the step change is completely accommodated due to the sharp reduction of the window size. At t = 500 min, when the bias fault is introduced, the update of the window size is stopped. The monitoring results show that the variable MWPCA not only perfectly accommodates the set-point change but also clearly detects the bias fault. In comparison, the conventional MWPCA with the fixed window size set to be 220 causes a number of false alarms before the process comes back to a relative stable state (Figure 10) because the window size is selected to be too large to adapt to the rapid change.

8. Conclusions

This paper proposes a variable MWPCA for time-varying process monitoring. The paper starts with a recursive strategy for updating the correlation matrix in samplewise and blockwise approaches, and then combines the moving window technique with the classical R-SVD algorithm to efficiently construct a new PCA model. As a result of storing only principal eigenpairs rather than the whole correlation matrix, the MWPCA algorithm requires less storage space than other approaches. Furthermore, instead of the fixed moving window, a variable moving window strategy is discussed for MWPCA adapting to normal process changes with a different changing rate. The variety of the window size depends on how fast the normal process can change. In addition, the changes of the sample mean vector and correlation matrix mainly reflect the normal process change. According to the current change rates of the mean and correlation matrix, the effective formula is derived to select the optimal window size. Further consideration is taken into the method of selecting the parameters in the formula. The application of the proposed adaptive process monitoring method to a simple illustrative case and a nonisothermal CSTR process demonstrates that, as expected, the variable MWPCA could accommodate both slow drift and rapid step change on the premise of clearly detecting the abnormal event. In comparison, a number of false alarms arise when the conventional MWPCA with a fixed window size is used.

Literature Cited

(1) Wise, B. M.; Gallagher, N. B. The Process Chemometrics Approach to Process Monitoring and Fault Detection. *J. Process Control* **1996**, *6*, 329.

(2) Lane, S.; Martin, E. B.; Kooijmans, R.; Morris, A. J. Performance Monitoring of a Multi-Product Semi-Batch Process. *J. Process Control* **2001**, *11*, 1.

(3) Rannar, S.; MacGregor, J. F.; Wold, S. *Adaptive Batch Monitoring Using Hierarchical PCA.* AICHE Annual Meeting, Los Angeles, November, 1997; 16.

(4) Jackson, J. E. *A User's Guide to Principal Components;* John Wiley & Sons: New York, 1991.

(5) Qin, S. J. Statistical Process Monitoring: Basics and Beyond. *J. Chemom.* **2003**, *17*, 480.

(6) Alwan, L. C.; Roberts, H. V. The Problem of Misplaced Control Limits. *Appl. Statistics* **1995**, *44*, 269.

(7) Wold, S. Exponentially Weighted Moving Principal Components Analysis and Projections to Latent Structures. *Chemom. Intell. Lab. Syst.* **1994**, *23*, 149.

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

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

(10) Lee, D. S.; Vanrolleghem, P. A. Adaptive Consensus Principal Component Analysis for On-Line Batch Process Monitoring. *Environ. Monit. Assess.* **2004**, *92*, 119.

(11) Wang, X.; Kruger, U.; Irwin, G. W. Process Monitoring Approach Using Fast Moving Window PCA. *Ind. Eng. Chem. Res.* **2005**, *44*, 5691.

(12) Berry, M. W. Large Scale Singular Value Computation. *International Journal of Supercomputer Applications* **1992**, *6*, 13.

(13) Li, Y. On Incremental and Robust Subspace Learning. *Pattern Recognit.* **2004**, *37* (7), 1509.

(14) Lim, J.; Ross, D.; Lin, R.; Yang, M. Incremental Learing for Visual Tracking. *Advances in NIPS*; **2004**, 793.

(15) Brand, M. Incremental Singular Value Decomposition of Uncertain Data with Missing Values. *ECCV*; **2002**, 707.

(16) Hoegaerts, L.; De Lathauwer, L.; Goethals, I.; Suykens, J. A. K.; Vandewalle, J.; De Moor, B. Efficiently Updating and Tracking the Dominant Kernel Principal Components. *Neural Networks* **2007**, *20*, 220.

(17) Golub, G. H.; Van Loan, C. F. *Matrix Computations*; The Johns Hopkins University Press: Baltimore, Maryland, 1996.

(18) Fortescue, T. R.; Kershenbaum, L. S.; Ydstie, B. E. Implementation of Self-Tuning Regulators with Variable Forgetting Factors. *Automatica* **1981**, *17*, 831.

(19) Lane, S.; Martin, E. B.; Morris, A. J.; Gower, P. Application of Exponentially Weighted Principal Component Analysis for the Monitoring of a Polymer Film Manufacturing Process. *Trans. Inst. Meas. Control* **2003**, *25* (1), 17.

(20) Choi, S. W.; Martin, E. B.; Morris, A. J.; Lee, I. B. Adaptive Multivariate Statistical Process Control for Monitoring Time-varying Process. *Ind. Eng. Chem. Res.* **2006**, *45*, 3108.

(21) Feng, X.; Yang, X.; Xu, Y. Squared Prediction Error Analysis of Multivariate Statistical Process Control. *J. Tsinghua Univ. Sci. Technol.* **1999**, *39* (7), 41.

(22) Chiang, L. H.; Russell, E. L.; Braatz, R. D. *Fault Detection and Diagnosis in Industrial Systems*; Springer: London, 2001.

(23) Huber, P. J. A Robust Estimation of a Location Parameter. *Ann. Math. Stat.* **1964**, *35*, 135.

(24) Dunia, R.; Qin, J.; Edgar, T. F.; McAvoy, T. J. *Sensor Fault Identification and Reconstruction Using Principal Component Analysis*, 13th IFAC World Congress, San Francisco, 1996; N2959, 264.

(25) Arteaga, F.; Ferrer, A. Dealing with Missing Data in MSPC: Several Methods, Different Interpretations, Some Examples. *J. Chemom.* **2002**, *16*, 408.

(26) Jin, H. D.; Lee, Y.; Lee, G.; Han, C. Robust Recursive Principal Component Analysis Modeling for Adaptive Monitoring. *Ind. Eng. Chem. Res.* **2006**, *45*, 696.

(27) Hwang, D. H.; Han, C. Real-Time Monitoring for a Process with Multiple Operating Models. *Control Eng. Pract.* **1999**, *7*, 891.

(28) Yoon, S.; MacGregor, J. F. Fault Diagnosis with Multivariate Statistical Models. Part I: Using Steady-State Fault Signatures. *J. Process Control* **2001**, *11*, 387.

(29) Choi, S. W.; Lee, C.; Lee, J.; Park, J. H.; Lee, I. B. Fault Detection and Identification of Nonlinear Processes Based on Kernel PCA. *Chemom. Intell. Lab. Syst.* **2005**, *75*, 55.

Received for review May 18, 2007
Revised manuscript received October 10, 2007
Accepted October 15, 2007

IE070712Z