ELSEVIER

Journal of Process Control 10 (2000) 471–486

JOURNAL OF PROCESS CONTROL

www.elsevier.com/locate/jprocont

Recursive PCA for adaptive process monitoring

Weihua Li, H. Henry Yue, Sergio Valle-Cervantes, S. Joe Qin *

Department of Chemical Engineering, The University of Texas at Austin, Austin, TX 78712, USA

Abstract

While principal component analysis (PCA) has found wide application in process monitoring, slow and normal process changes often occur in real processes, which lead to false alarms for a fixed-model monitoring approach. In this paper, we propose two recursive PCA algorithms for adaptive process monitoring. The paper starts with an efficient approach to updating the correlation matrix recursively. The algorithms, using rank-one modification and Lanczos tridiagonalization, are then proposed and their computational complexity is compared. The number of principal components and the confidence limits for process monitoring are also determined recursively. A complete adaptive monitoring algorithm that addresses the issues of missing values and outlines is presented. Finally, the proposed algorithms are applied to a rapid thermal annealing process in semiconductor processing for adaptive monitoring.

Keywords: Recursive principal component analysis; Adaptive process monitoring; Rank-one modification; Lanczos tridiagonalization

1. Introduction

Principal component analysis (PCA) has been successfully applied to the monitoring of industrial processes, including chemical and microelectronics manufacturing processes [1,2]. Formulated as a multivariate statistical process control (MSPC) task, these applications use PCA to extract a few independent components from highly correlated process data and use the components to monitor the process operation. Typically, two major monitoring indices are calculated, the squared prediction error (SPE) and the Hotelling T^2 index. An abnormal situation will cause at least one of the two indices to exceed the control limits. Multi-way PCA, as a variation of PCA, was successfully applied to batch monitoring, in which the data have three dimensions: batches, variables, and samples in a batch [3].

A major limitation of PCA-based monitoring is that the PCA model, once built from the data, is time-invariant, while most real industrial processes are time-varying [4]. The time-varying characteristics of industrial processes include: (i) changes in the mean; (ii) changes in the variance; and (iii) changes in the correlation structure among variables, including changes in the number of significant principal components (PCs). When a time-in variant PCA model is used to monitor processes with the aforementioned normal changes, false alarms often result, which significantly compromise the reliability of the monitoring system.

Industrial processes commonly demonstrate slow time-varying behaviors, such as catalyst deactivation, equipment aging, sensor and process drifting, and preventive maintenance and cleaning. So far, little has been reported on the development of adaptive process monitoring. Gallagher et al. [4] pointed out the need to monitor microelectronics manufacturing processes. Wold [5] discussed the use of exponentially weighted moving average (EWMA) filters in conjunction with PCA and PLS. Rigopoulos et al. [6] use a similar moving window scheme to identify significant modes in a simulated paper machine profile. Rannar et al., [7] use a hierarchical PCA for adaptive batch monitoring in a way that is similar to EWMA based PCA. However, we still lack a recursive PCA that resembles the adaptation mechanism of recursive PLS [8].

A complete recursive PCA scheme should consider the following issues:

1. Recursive update of the mean for both covariance-based PCA and correlation-based PCA.
2. Efficient algorithms for the PCA calculation, including sample-wise update and block-wise update.

* Corresponding author. Tel.: +1-512-471-4417; fax: +1-512-471-7060.

E-mail address: qin@che.utexas.edu (S.J. Qin).

0959-1524/00/$ - see front matter
PII: S0959-1524(00)00022-6

3. Recursive determination of the number of principal components, as the significant number of components changes; and
4. Recursive determination of the confidence limits for SPE and T^2 in real-time to facilitate adaptive monitoring.

Here we propose two recursive PCA (RPCA) algorithms to adapt for normal process changes and thus reduce false alarms. The challenge is to efficiently make use of the old model to calculate the new model. Since the variables in PCA are usually scaled to zero mean and unit variance, changes in the variable mean and variance complicate the recursive computation. While in recursive regression, such a recursive PLS in Qin [8], one can simply use a bias term to adapt for the mean changes, the same technique does not work for recursive PCA. Let $x^o \in \mathcal{R}^m$ be a vector of sensor measurements with possibley changing means. A recursive PLS scheme can use the augmented vector $[x^{oT}\ 1]$ as a regressor to adapt for the effect of the time-varying mean [8]. However, this augmentation in recursive PCA does not yield a model that is consistent with one from mean-centered data. For example, consider as a limiting case a process with only one variable whose mean is shifted from 0 to 0.5. As shown in Fig. 1(a), a proper adaptation of the mean will result in a PCA model with one component that coincides with the x_1^0 axis. With the augmented data as shown in Fig. 1(b), two PCs are required. The first PC goes through the origin; the second PC must be orthogonal to the first. However, neither captures the data variation correctly.

We offer an efficient recursive calculation for the correlation matrix with time varying mean and variance (Section 2) and propose two numerically efficient algorithms (Section 3) to update the PCA model: (i) use of rank-one modification for sample-wise recursion; and (ii) use of Lanczos tridiagonalization for block-wise recursion (Section 4). Section 5 compares the computational costs of the two PCA algorithms. The recursive determination of the number of PCs is presented in Section 6, where the variance of reconstruction error (VRE) method, Akaike information criterion (AIC), minimum description length (MDL) and other methods are compared. Section 7 investigates the recursive determination of confidence limits for SPE and T^2. Section 8 illustrates an application to a rapid thermal annealing process in semiconductor processing, and we draw our conclusions in Section 9.

2. Recursive update for the correlation matrix

2.1. Conventional PCA

In the conventional batch-wise PCA, the raw data matrix $\mathbf{X}^0 \in \mathcal{R}^{n\times m}$ of n samples (rows) and m variables (columns) is first normalized to a matrix $\mathbf{X}$ with zero

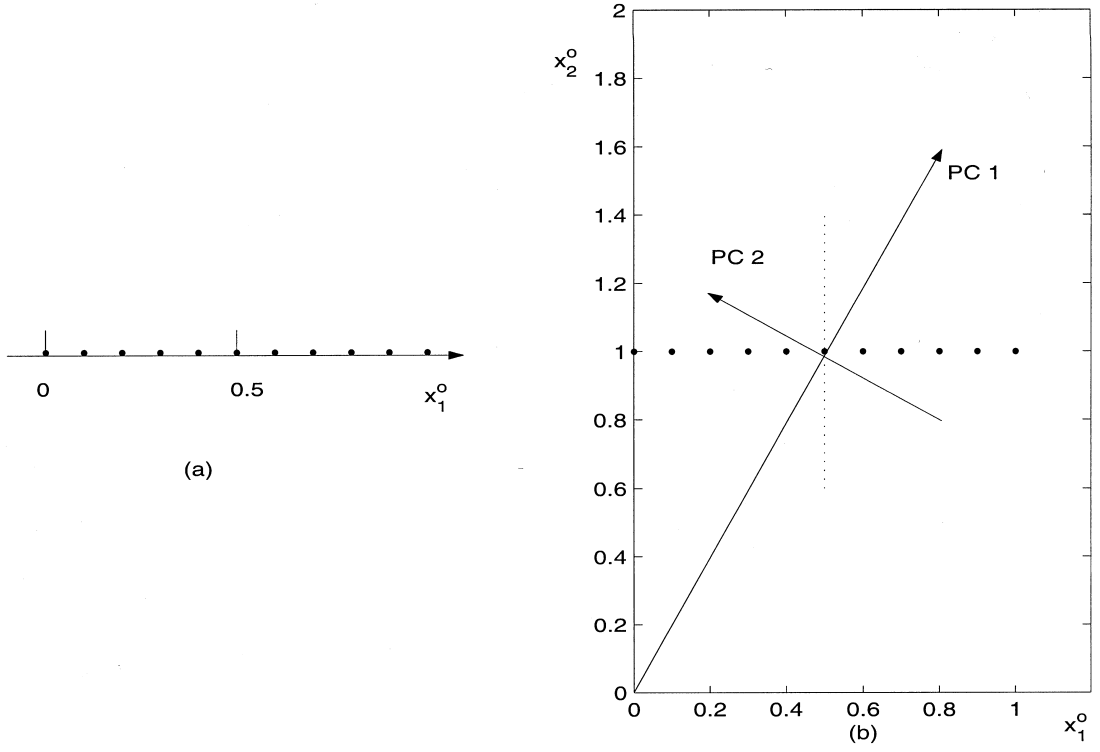

Fig. 1. The significance of mean adaptation in RPCA.

mean and unit variance. Then the normalized matrix $\mathbf{X}$ is decomposed as follows:

$$\begin{aligned}\mathbf{X} &= \mathbf{T}\mathbf{P}^T + \tilde{\mathbf{X}} \\ &= \mathbf{T}\mathbf{P}^T + \tilde{\mathbf{T}}\tilde{\mathbf{P}}^T = \begin{bmatrix}\mathbf{T} & \tilde{\mathbf{T}}\end{bmatrix}\begin{bmatrix}\mathbf{P} & \tilde{\mathbf{P}}\end{bmatrix}^T\end{aligned} \tag{1}$$

where $\mathbf{T} \in \mathcal{R}^{n\times l}$ and $\mathbf{P} \in \mathcal{R}^{m\times l}$ are the PC scores and loadings, respectively. The PCA projection reduces the original set of variables to l principal components. The residual matrix $\tilde{\mathbf{X}}$ can be further decomposed into $\tilde{\mathbf{T}}\tilde{\mathbf{P}}^T$ if desired. The decomposition is made such that $[\mathbf{T}\,\tilde{\mathbf{T}}]$ is orthogonal and $[\mathbf{P}\,\tilde{\mathbf{P}}]$ is orthonormal.

The correlation matrix of the variables can be approximated as:

$$R \approx \frac{1}{n-1}\mathbf{X}^T\mathbf{X} \tag{2}$$

The columns of $\mathbf{P}$ are actually eigenvectors of $\mathbf{R}$ associated with the l largest eigenvalues, and the columns of $\tilde{\mathbf{P}}$ are the remaining eigenvectors, converting the calculation of the PCA model ($\mathbf{P}$) to an eigenvector problem. If only the $\mathbf{P}$ matrix is needed, very efficient numerical algorithms based on the Lanczos method are available [9]. Occasionally, the covariance matrix is used to derive a PCA model. In this case, the data is scaled to zero-mean, but the variance is unscaled. The variance scaling affects the relative weighting of all variables [10].

2.2. Recursive correlation matrix calculation

If a block of process data has been used to build an initial PCA model, we need to update the PCA model when a new block of data becomes available. Let $\mathbf{X}_1^0 \in \mathcal{R}^{n_1\times m}$ be the raw initial data block. Then the mean of each column is given in the vector

$$\mathbf{b}_1 = \frac{1}{n_1}\left(\mathbf{X}_1^0\right)^T 1_{n_1} \tag{3}$$

where $1_{n_1} = [1, 1, \cdots, 1]^T \in \mathcal{R}^{n_1}$. The data scaled to zero mean and unit variance is given by

$$\mathbf{X}_1 = \left(\mathbf{X}_1^0 - 1_{n_1}\mathbf{b}_1^T\right)\Sigma_1^{-1} \tag{4}$$

where

$$\Sigma_1 = \mathrm{diag}(\sigma_{1\cdot 1}, \ldots, \sigma_{1\cdot m}) \tag{5}$$

whose ith element is the standard deviation of the ith sensor ($i = 1, \ldots, m$). The correlation matrix is

$$\mathbf{R}_1 = \frac{1}{n_1 - 1}\mathbf{X}_1^T\mathbf{X}_1 \tag{6}$$

The new block of data is expected to augment the data matrix and calculate the correlation matrix recursively. Assume that $\mathbf{b}_k$ $\mathbf{X}_k$ and $\mathbf{R}_k$ have been calculated when the kth block of data is collected. The task for recursive calculation is to calculate $\mathbf{b}_{k+1}$, $\mathbf{X}_{k+1}$ and $\mathbf{R}_{k+1}$ when the next block of data $\mathbf{X}_{n_{k+1}}^n \in \mathcal{R}^{n_{k+1}\times m}$ is available. Denoting

$$\mathbf{X}_{k+1}^0 = \begin{bmatrix}\mathbf{X}_k^0 \\ \mathbf{X}_{n_{k+1}}^0\end{bmatrix} \tag{7}$$

for all the $k+1$ blocks of data, the mean vector $\mathbf{b}_{k+1}$ is related to $\mathbf{b}_k$ by the following relation:

$$\left(\sum_{i=1}^{k+1} n_i\right)\mathbf{b}_{k+1} = \left(\sum_{i=1}^{k} n_i\right)\mathbf{b}_k + \left(\mathbf{X}_{n_{k+1}}^0\right)^T 1_{n_{k+1}} \tag{8}$$

Denoting $N_k = \sum_{i=1}^{k} n_i$, Eq. (8) yields the following recursive calculation:

$$\mathbf{b}_{k+1} = \frac{N_k}{N_{k+1}}\mathbf{b}_k + \frac{1}{N_{k+1}}\left(\mathbf{X}_{n_{k+1}}^0\right)^T 1_{n_{k+1}} \tag{9}$$

The recursive calculation of $\mathbf{X}_{k+1}$ is given by

$$\begin{aligned}\mathbf{X}_{k+1} &= \left[\mathbf{X}_{k+1}^0 - 1_{k+1}\mathbf{b}_{k+1}^T\right]\Sigma_{k+1}^{-1} \\ &= \left[\begin{bmatrix}\mathbf{X}_k^0 \\ \mathbf{X}_{n_{k+1}}^0\end{bmatrix} - 1_{k+1}\mathbf{b}_{k+1}^T\right]\Sigma_{k+1}^{-1} \\ &= \begin{bmatrix}\mathbf{X}_k^0 - 1_k\Delta\mathbf{b}_{k+1}^T - 1_k\mathbf{b}_k^T \\ \mathbf{X}_{n_{k+1}}^0 - 1_{n_{k+1}}\mathbf{b}_{k+1}^T\end{bmatrix}\Sigma_{k+1}^{-1} \\ &= \begin{bmatrix}\mathbf{X}_k\Sigma_k\Sigma_{k+1}^{-1} - 1_k\Delta\mathbf{b}_{k+1}^T\Sigma_{k+1}^{-1} \\ \mathrm{X}_{n_{k+1}}\end{bmatrix}\end{aligned} \tag{10}$$

where

$$\begin{aligned}\mathbf{X}_k &= \left(\mathbf{X}_k^0 - 1_k\mathbf{b}_k^T\right)\Sigma_k^{-1} \\ \mathbf{X}_{n_k+1} &= \left(\mathbf{X}_{n_{k+1}} - 1_{n_{k+1}}\mathbf{b}_{k+1}^T\right)\Sigma_{k+1}^{-1} \\ \Sigma_j &= \mathrm{diag}\left(\sigma_{j\cdot 1}, \cdots, \sigma_{j\cdot m}\right),\ j = k, k+1 \\ \Delta\mathbf{b}_{k+1} &= \mathbf{b}_{k+1} - \mathbf{b}_k\end{aligned} \tag{11}$$

Note that $1_k = [1, \cdots, 1]^T \in \mathcal{R}^{N_k}$.

The recursive computation of the standard deviation, derived in Appendix A, has the following relationship,

$$\begin{aligned}(N_{k+1} - 1)\sigma_{k+1\cdot i}^2 = (N_k - 1)\sigma_{\mathrm{ki}}^2 + N_k\Delta b_{k+1}^2(i) \\ + \left\|\mathbf{X}_{n_{k+1}}^0(:, i) - 1_{n_{k+1}}b_{k+1}(i)\right\|^2\end{aligned} \tag{12}$$

where $\mathbf{X}_{n_{k+1}}^0(:, i)$ is the ith column of the associated matrix; $b_{k+1}(i)$ and $\Delta b_{k+1}(i)$ are the ith elements of the associated vectors.

Similarly, the recursive calculation of the correlation matrix, derived in Appendix B, has the following form,

$$
\begin{aligned}
R_{k+1} = & \frac{1}{N_{k+1}-1}\mathbf{X}_{k+1}^T\mathbf{X}_{k+1} \\
& - \frac{N_k-1}{N_{k+1}-1}\Sigma_{k+1}^{-1}\Sigma_k\mathbf{R}_k\Sigma_k\Sigma_{k+1}^{-1} \\
& + \frac{N_k}{N_{k+1}-1}\Sigma_{k+1}^{-1}\Delta\mathbf{b}_{k+1}\Delta\mathbf{b}_{k+1}^T\Sigma_{k+1}^{-1} \\
& + \frac{1}{N_{k+1}-1}\mathbf{X}_{n_{k+1}}^T\mathbf{X}_{n_{k+1}}
\end{aligned} \tag{13}
$$

Regarding the above recursive relation, not that:

1. The effect of mean changes $\Delta\mathbf{b}_{k+1}$ on the correlation matrix is only a rank-one modification.
2. As a special case, if the model needs to be updated after every new sample is taken, the recursive relation reduces to:

$$
\begin{aligned}
\mathbf{R}_{k+1} = & \frac{k-1}{k}\Sigma_{k+1}^{-1}\Sigma_k\mathbf{R}_k\Sigma_k\Sigma_{k+1}^{-1} \\
& + \Sigma_{k+1}^{-1}\Delta\mathbf{b}_{k+1}\Delta\mathbf{b}_{k+1}^T\Sigma_{k+1}^{-1} + \frac{1}{k}\mathbf{x}_{k+1}\mathbf{x}_{k+1}^T
\end{aligned} \tag{14}
$$

which is two rank-one modifications. Algorithms are available for the eigenvector calculation of rank-one modifications [11], which will be elaborated later in this paper (Section 3).

3. Often old data are exponentially ignored as they do not represent the current process. The recursive calculations for Eqs. (9), (12) and (13) with a forgetting factor are:

$$
\mathbf{b}_{k+1} = \mu\mathbf{b}_k + (1-\mu)\frac{1}{n_{k+1}}\left(\mathbf{X}_{n_{k+1}}^0\right)^T 1_{n_{k+1}} \tag{15}
$$

$$
\begin{aligned}
\sigma_{k+1\cdot i}^2 = & \mu\left(\sigma_{\mathrm{k\cdot i}}^2 + \Delta\mathbf{b}_{k+1}^2(i)\right) + (1-\mu)\frac{1}{n_{k+1}} \\
& \times \left\|\mathbf{X}_{n_{k+1}}^0(:,i) - 1_{n_k+1}\mathbf{b}_{k+1}(i)\right\|^2
\end{aligned} \tag{16}
$$

and

$$
\begin{aligned}
\mathbf{R}_{k+1} = & \mu\Sigma_{k+1}^{-1}\left(\Sigma_k\mathbf{R}_k\Sigma_k + \Delta\mathbf{b}_{k+1}\Delta\mathbf{b}_{k+1}^T\right)\Sigma_{k+1}^{-1} \\
& + (1-\mu)\frac{1}{n_{k+1}}\mathbf{X}_{n_{k+1}}^T\mathbf{X}_{n_{k+1}}
\end{aligned} \tag{17}
$$

for $N_k >> 1$. In the above relations, $0 < \mu \leqslant \frac{N_k}{N_{k+1}} < 1$ is the forgetting factor. Smaller μ tends to forget old data more quickly, whereas

$$
\mu = \frac{N_k}{N_{k+1}} \tag{18}
$$

recovers the case of no forgetting. Similar to the window size in a moving window approach, the forgetting factor is a tuning parameter that varies depending on how fast the normal process can change.

3. RPCA by rank-one modification

If one chooses to update the RPCA model after each sample, the recursive correlation matrix becomes

$$
\begin{aligned}
\mathbf{R}_{k+1} = & \mu\Sigma_{k+1}^{-1}\left(\Sigma_k\mathbf{R}_k\Sigma_k + \Delta\mathbf{b}_{k+1}\Delta\mathbf{b}_{k+1}^T\right)\Sigma_{k+1}^{-1} \\
& + (1-\mu)\mathbf{x}_{k+1}\mathbf{x}_{k+1}^T
\end{aligned} \tag{19}
$$

while the change in mean is only a rank-one modification to the correlation matrix, the change in variance (Σ_{k+1}) completely changes the eigenstructure. Note that the role of variance update is different from that of the mean update. If the mean changes but is not updated, as shown in Fig. 1, the calculated principal component direction $(\mathbf{p}_1)$ can be very different from the actual principal component direction $(\mathbf{p}_1')$ after the mean is updated. On the other hand, the variance-update affects only the relative weighting of each variable. If the variance does not change dramatically, we can use the initial variance to scale the data and do not update variance. For the case of covariance-based PCA, there is no need to scale the variance at all. Therefore, Eq. (19) without variance updating reduces to,

$$
\mathbf{R}_{k+1} = \mu\left(\mathbf{R}_k + \Sigma_1^{-1}\Delta\mathbf{b}_{k+1}\Delta\mathbf{b}_{k+1}^T\Sigma_1^{-1}\right) + (1-\mu)\mathbf{x}_{k+1}\mathbf{x}_{k+1}^T \tag{20}
$$

In other words, $\mathbf{R}_{k+1}$ is modified with two rank-one matrices, $\Sigma_1^{-1}\Delta\mathbf{b}_{k+1}\Delta\mathbf{b}_{k+1}^T\Sigma_1^{-1}$ and $\mathbf{x}_{k+1}\mathbf{x}_{k+1}^T$. Others have studied the rank-one updating of the eigenproblem and have proposed related special algorithms in the symmetric case [11,12].

The problem of rank-one modification can be summarized as follows: given a diagonal matrix $\mathbf{\Lambda}^0 = \mathrm{diag}\left(\lambda_1^0,\ldots,\lambda_m^0\right)$, we seek the eigenpairs $\{\mathbf{p},\lambda\}$ of $\mathbf{\Lambda}^\circ + \rho^2\mathbf{z}\mathbf{z}^T$ such that $\left(\mathbf{\Lambda}^\circ + \rho^2\mathbf{z}\mathbf{z}^T\right)\mathrm{p}-\lambda\mathrm{p}$, where $\mathbf{z}$ is of unit norm.

This problem leads to the system

$$
\left(\mathbf{\Lambda}^\circ - \lambda\mathbf{I} + \rho^2\mathbf{z}\mathbf{z}^T\right)\mathbf{p} = \mathbf{0} \tag{21}
$$

The eigenvalues λ_i for $i-1,\cdots,m$ can be obtained by finding the roots of the following secular equation [12]:

$$
f(\lambda) - 1 + \rho^2\sum_{j=1}^{m}\frac{\mathrm{z}_j^2}{\left(\lambda_j^0 - \lambda\right)} = 0 \tag{22}
$$

where z_j denotes the jth entry of $\mathbf{z}$. Once the eigenvalues λ_i are known, the corresponding eigenvectors can be calculated from Eq. (21) as:

$$\mathbf{p}_i = \frac{(\mathbf{\Lambda}^{\circ} - \mathbf{\Lambda}_i\mathbf{I})^{-1}\mathbf{z}}{\left\|(\mathbf{\Lambda}^{\circ} - \mathbf{\Lambda}_i\mathbf{I})^{-1}\mathbf{z}\right\|}, \quad 1 \leqslant i \leqslant m \tag{23}$$

where the vector $(\mathbf{\Lambda}^{\circ} - \lambda_i\mathbf{I})^{-1}\mathbf{z}$ is previously calculated in finding roots of Eq. (22).

To compute each $\mathbf{p}_i$, we need only additional $2m$ multiplications; computing $\mathbf{p}_1, \ldots, \mathbf{p}_m$ thus needs additional $2m^2$ multiplications. Bunch et al. [11] reported that the average number of iterations required to find a single root is about 4.4, making the computational cost of the eigenpair of $\mathbf{\Lambda}^{\circ} + \rho^2\mathbf{z}\mathbf{z}^T$ very low.

Using the above algorithms, we may compute the eigenpairs of $\mathbf{R}_{k+1}$ by applying the rank-one modification algorithm twice. Since $\mathbf{R}_k = \mathbf{P}_k\mathbf{\Lambda}_k\mathbf{P}_k^T$ is known in the existing model, where $\mathbf{\Lambda}_k$ is a diagonal matrix containing all the eigenvalues, Eq. (20) can be rewritten as follows,

$$\begin{aligned}\mathbf{R}_{k+1} &= \mu\mathbf{P}_k\left(\mathbf{\Lambda}_k + \delta\mathbf{b}_{k+1}\delta\mathbf{b}_{k+1}^T\right)\mathbf{P}_k^T \\ &\quad + (1-\mu)\mathbf{x}_{k+1}\mathbf{x}_{k+1}^T\end{aligned} \tag{24}$$

where $\delta\mathbf{b}_{k+1} = \mathbf{P}_k^T\Sigma_1^{-1}\Delta\mathbf{b}_{k+1}$. After the first rank-one modification,

$$\mu\left(\mathbf{\Lambda}_k + \delta\mathbf{b}_{k+1}\delta\mathbf{b}_{k+1}^T\right) = \mathbf{P}_{k+}\mathbf{\Lambda}_{k+}\mathbf{P}_{k+}^T \tag{25}$$

we obtain,

$$\begin{aligned}\mathbf{R}_{k+1} &= \mathbf{P}_k\mathbf{P}_{k+}\mathbf{\Lambda}_{k+}\mathbf{P}_{k+}^T\mathbf{P}_k^T + (1-\mu)\mathbf{x}_{k+1}\mathbf{x}_{k+1}^T \\ &= \bar{\mathbf{P}}_{k+}\mathbf{\Lambda}_{k+}\bar{\mathbf{P}}_{k+}^T + (1-\mu)\mathbf{x}_{k+1}\mathbf{x}_{k+1}^T \\ &= \bar{\mathbf{P}}_{k+}\left(\mathbf{\Lambda}_{k+} + (1-\mu)\mathbf{z}_{k+1}\mathbf{z}_{k+1}^T\right)\bar{\mathbf{P}}_{k+}^T\end{aligned} \tag{26}$$

where $\mathbf{z}_{k+1} = \bar{\mathbf{P}}_{k+}^T\mathbf{x}_{k+1}$ and $\bar{\mathbf{P}}_{k+} = \mathbf{P}_k\mathbf{P}_{k+}$ is an orthogonal matrix because $\mathbf{P}_k$ and $\mathbf{P}_{k+}$ are orthogonal.

Applying the rank-one modification algorithm again,

$$\mathbf{\Lambda}_{k+} + (1-\mu)\mathbf{z}_{k+1}\mathbf{z}_{k+1}^T = \mathbf{V}_{k+1}\mathbf{\Lambda}_{k+1}\mathbf{V}_{k+1}^T \tag{27}$$

the eigendecomposition of the correlation matrix is

$$\mathbf{R}_{k+1} = \bar{\mathbf{P}}_{k+}\mathbf{V}_{k+1}\mathbf{\Lambda}_{k+1}\mathbf{V}_{k+1}^T\bar{\mathbf{P}}_{k+}^T = \mathbf{P}_{k+1}\mathbf{\Lambda}_{k+1}\mathbf{P}_{k+1}^T \tag{28}$$

where $\mathbf{P}_{k+1} = \bar{\mathbf{P}}_{k+}\mathbf{V}_{k+1}$.

As shown above, computing the eigenpair of $\mathbf{R}_{k+1}$ requires two rank-one modifications and two matrix–matrix products, i.e. $\bar{\mathbf{P}}_{k+} = \mathbf{P}_k\mathbf{P}_{k+}$ and $\mathbf{P}_{k+1} - \bar{\mathbf{P}}_{k+}\mathbf{V}_{k+1}$. The computational cost of two rak-one modifications is negligible as compared with the cost of computing the two matrix–matrix products, which requires $O(2m^3)$ multiplications. Therefore, based on the known eigenstructure of $\mathbf{R}_k$, we require $O(2m^3)$ multiplications to compute the eigenstructure of $\mathbf{R}_{k+1}$.

A major drawback of the rank-one modification algorithm is that all the eigenpairs have to be calculated, although only a few principal eigenpairs of the correlation matrix are needed. This is an excessive computational burden. Moreover, if one needs to update the model based on a block of data instead of a sample, a costly sequence of rank-one modifications (depending on the rank of the matrix $\mathbf{X}_{n_{k+1}}^T\mathbf{X}_{n_{k+1}}$) has to be applied. The following alternative method avoids these drawbacks.

4. RPCA by the Lanczos tridiagonalization

The Lanczos algorithms for symmetric matrices were introduced in the 1950s and are described, for instance, in Golub and Van Loan [13]. The algorithms received more attention following Paige's results [14] and were studied in detail by Parlett [15]. The main advantages of the Lanczos tridiagonalization methods come from the following:

1. The original matrix is not overwritten.
2. Little storage is required since only matrix–vector products are computed; and
3. Only a few (l_k) of the largest eigenvalues are calculated. Usually l_k is much smaller than the dimension of the matrix $\mathbf{R}_k$.

These features make the Lanczos procedure useful for large matrices, especially if they are sparse.

4.1. Lanczos tridiagonalization

Applying the Lanczos procedure to the symmetric correlation matrix $\mathbf{R}_k \in \mathcal{R}^{m\times m}$ yields a symmetric tridiagonal matrix $\mathbf{\Gamma}$, i.e.

$$\begin{aligned}\Gamma &= \Upsilon^T\mathbf{R}_k\Upsilon \\ &= \begin{bmatrix} \alpha_1 & \beta_1 & \ldots & 0 \\ \beta_1 & \alpha_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \beta_{m_1-1} \\ 0 & \ldots & \beta_{m_1-1} & \alpha_{m_1} \end{bmatrix}\end{aligned} \tag{29}$$

where $\Upsilon \in \mathcal{R}^{m\times m_1}$ is a matrix containing the orthogonal Lanczos vectors and $m_1 < m$. Although Γ has a smaller dimension than $\mathbf{R}_k$, their extreme eigenvalues are approximately the same [8,9].

Note that because of roundoff errors, a loss of orthogonality among the computed Lanczos vectors occurs after some eigenpairs have already been calculated, leading to several undesirable phenomena [64,15]. To cope with this problem, a re-orthogonalization procedure is usually introduced. Therefore, we compute the principal eigenpairs of the correlation matrix using the tridiagonalization

algorithm with complete re-orthogonalization. The whole procedure includes two stages:

1. The tridiagonalization of the correlation matrix $\mathbf{R}_k$ using the Lanczos algorithm with a re-orthogonalization procedure.
2. The calculation of the principal eigenpairs of the resulting tridiagonal matrix.

A typical algorithm to compute the diagonal elements α_i and the sub-diagonal elements β_i for $i = 1,\cdots, m_1$ is given in Golub and Van Loan [8]. Approximations to the eigenvectors of $\mathbf{R}_k$ require the storage of the generated Lanzcos vectors and the computation of the eigenvectors of the matrix Γ. The algorithm terminates as soon as m (or fewer) orthogonal Lanczos vectors have been generated. In each step, the complexity is dominated by a matrix–vector product.

4.2. Termination of the Lanczos algorithm

The dimension of the tridiagonal matrix Γ, m_1, which is usually smaller than the dimension of the correlation matrix $\mathbf{R}_k$, must be determined in the Lanczos tridiagonalization algorithm. A properly selected m_1 should guarantee that the eigenvalues of Γ are approximately the same as the m_1 largest eigenvalues of $\mathbf{R}_k$. In this paper, we propose a practical criterion associated with the eigenvalues of $\mathbf{R}_k$ for the termination of the Lanczos algorithm. It is known that trace ($\mathbf{R}_k$) is equal to the sum of all the eigenvalues of $\mathbf{R}_k$. If Γ includes all significant eigenvalues of $\mathbf{R}_k$, trace (Γ) will be very close to trace ($\mathbf{R}_k$). Therefore, we terminate the Lanczos algorithm and determine the value of m_1 if

$$\frac{\text{trace}(\Gamma)}{\text{trace}(\mathbf{R}_k)} \geqslant \epsilon$$

where ϵ is a selected threshold. For example, $\epsilon = 0.995$ means that 99.5% of the variance in $\mathbf{R}_k$ is represented by the m_1 eigenvalues.

With the computed Γ, we can calculate its first few largest eigenvalues $\lambda_1,\cdots,\lambda_{l_k}$, which are approximately equal to the l_k principal eigenvalues of $\mathbf{R}_k$, and the associated eigenvectors $\boldsymbol{C}_{l_k} = [c_1,\cdots,c_{l_k}]$ with $i \in \mathcal{R}^{m_1}$ for $i = 1,\cdots, l_k$. Consequently, the PCA loading matrix for $\mathbf{R}_k$ is

$$\mathbf{P}l_k = \Upsilon\mathbf{C}_{l_k} \tag{30}$$

4.3. Computation of the principal eigenvalues

In this subsection, we present an efficient approach to computing the principal eigenvalues $\{\lambda_1 \geqslant \lambda_2 \geqslant,\cdots, \geqslant \lambda_{l_k}\}$ of Γ in descending order with $l_k \leqslant m_1$. Let Γ_κ denote the leading $\kappa \times \kappa$ principal sub-matrix of the tridiagonal matrix Γ and $p_\kappa(\lambda) = \det(\Gamma_\kappa - \lambda\mathbf{I})$ denote the associated characteristic polynomial for $\kappa = 1,\cdots, m_1$. A simple determinantal expansion [13] shows that

$$p_\kappa(\lambda) = (\alpha_\kappa - \lambda)p_{\kappa-1}(\lambda) - \beta^2_{\kappa-1}p_{\kappa-2}(\lambda), \quad \kappa = 2,\ldots, l_k \tag{31}$$

where $p_0(\lambda) = 1$.

Denote the upper and lower limits of λ by $\lambda_{\max}$ and $\lambda_{\min}$, respectively. For any $\lambda_{\min} \leqslant \lambda \leqslant \lambda_{\max}$ with $p_0(\lambda) = 1$, using Eq. (31), we can generate a sequence.

$$\{p_0(\lambda), p_1(\lambda), \ldots, p_{m_1}(\lambda)\} \tag{32}$$

Let $\alpha(\lambda)$ denote the number of sign changes in this sequence. Then, based on Gershgorin Theorem and Sturm sequence property [13], we develop an algorithm to calculate the principal eigenvalues of Γ, given in Appendix C.

We evaluate the overall complexity to compute $\mathbf{P}_{l_k}$ as follows:

1. The updating of the correlation matrix $\mathbf{R}_k$ in Eq. (19) requires $n_k m^2 + 4m^2 + 4m$ multiplications.
2. The Lanczos tridiagonalization to compute Γ and Υ requires $4m^2m_1 - 1.5mm_1^2 + 6.5mm_1 - m^2 - 2m_1^2$ multiplications [13].
3. The computation of l_k eigenvalues requires $3m_1l_ks$ multiplications, where s is the number of iterations required to find an eigenvalue and is evaluated in Appendix C.
4. The computation of $\mathbf{P}_{l_k}$ from Υ and $\mathbf{C}_{l_k}$ requires mm_1l_k multiplications.

Therefore, the overall complexity is approximately equal to $n_km^2 + 4m^2m_1 - 1.5mm_1^2 + mm_1l_k + 3m_1l_ks$ multiplications, where $l_k \leqslant m_1 \leqslant m$. Obviously, the complexity reaches its maximum value $n_km^2 + 2.5m^3 + m^2l_k + 3ml_ks$ when $m_1 = m$.

5. Comparison of computational complexity

In this section, we compare the computational costs for rank-one modification, Lanczos tridiagonalization, and the standard singular value decomposition (SVD) algorithm for RPCA calculation. For the block-wise case, since $\mathbf{X}^T_{n_k}\mathbf{X}_{n_k}$ can be decomposed into a sum of n_k rank-one matrices, using rank-one modification requires $(n_k + 1)m^3$ multiplications to compute the PCA loading matrix. In this case, the RPCA by Lanczos tridiagonalization is much more efficient. Because it is recognized as one of the most efficient algorithms among the SVD algorithms [13], we choose for comparison the symmetric QR, which also includes two steps: (1) the tridiagonalization of the correlation matrix $\mathbf{R}_k$; (2) calculation for the eigenpairs of the computed tridiagonal matrix. Based on

Golub and Van Loan [13] (Algorithm 8.3.1), the tridiagonalization of $\mathbf{R}_k$ sing the symmetric QR algorithm needs $\frac{8}{3}m^3$ multiplications. As pointed out previously, even in the worst case ($m = m_1$), tridiagonalizing $\mathbf{R}_k$ using the Lanczos approach needs $2.5m^3$ multiplications. The computational costs can be further reduced if the complete re-orthogonalization of the Lanczos vectors is replaced by partial re-orthogonalization [14]. In addition, the second step of the symmetric QR algorithm requires $m^2 l_k + 3ml_k s$. Therefore, the Lanczos tridiagonalization approach for RPCA is more efficient than SVD based on the symmetric QR algorithm. The overall complexity using these three approaches is listed in Table 1.

In the special case of sample-wise updating, i.e. $n_k = 1$, computing the PCA loading matrix $\mathbf{P}l_k$ using the RPCA by Lanczos tridiagonalization needs approximately $(2.5m^2 m_1)$ multiplications, while the rank-one modification requires $O(2m^3)$ multiplications. We have compared the computational costs among the aforementioned three RPCA approaches when they are applied to correlation matrices with different dimensions. As depicted in Fig. 2, RPCA by rank-one modification is the most efficient, the RPCA by Lanczos tridiagonalization is the second most efficient, and the standard SVD is most costly. The Lanczos algorithm can be more efficient when there are many variables that are highly correlated.

6. Recursive determination of the number of PCs

Since the number of significant principal components can change over time, it is necessary to determine this number recursively in RPCA modeling. There are many ways of determining the number of PCs in batch-wise PCA, including:

- Autocorrelation [17]
- Cross-validation [18,19]
- Cumulative percent variance [20]
- Scree test [21]
- Average eigenvalues
- Imbedded error function [20]
- Xu and Kailath's approach [9]
- Akaike information criterion [22,23]
- Minimum description length criterion [23,24]

Table 1
Complexity of the RPCA computations

Approach used	Overall complexity
Rank-one modification	$(n_k + 1)m^3$
Lanczos tridiagonalization	$n_k m^2 + 4m^2 m_1 - 1.5mm_1^2 + mm_1 l_k + 3m_1 l_k s$
The symmetric QR algorithm	$n_k m^2 + \frac{8}{3}m^3 + m^2 l_k + 3ml_k s$

- Variance of reconstruction error [25]

However, not all the approaches are suitable for recursive PCA. For example, the cross-validation approach is not suitable because old data are not representative for the current process. Therefore, we consider only the following methods to determine the number of PCs recursively.

1. Cumulative percent variance (CPV): The CPV is a measure of the percent variance captured by the first l_k PCs:

$$\mathrm{CPV}(l_k) = \frac{\Sigma_{j=1}^{l_k}\lambda_j}{\Sigma_{j=1}^{m}\lambda_j}100\%$$

Since only the first l_k eigenvalues are calculated in the block-wise RPCA, we use the following formula:

$$\mathrm{CPV}(l_k) = \frac{\Sigma_{j=1}^{l_k}\lambda_j}{\mathrm{trace}(\mathbf{R}_k)}100\% \tag{33}$$

The number of PCs is chosen when CPV reaches a predetermined limit, say 95%.

2. Average eigenvalue (AE): The AE approach selects the eigenvalues greater than the mean of all eigenvalues as the principal eigenvalues and discards those smaller than the mean. Therefore, for $\mathbf{R}_k$, only the eigenvalues greater than

$$\frac{\mathrm{trace}(\mathbf{R}_k)}{m}$$

are used as the principal eigenvalues.

3. Imbedded error function (IEF). The IEF is given by:

$$\mathrm{IEF}(l_k) = \sqrt{\frac{l_k\theta_1}{N_k m(m-l_k)}} \tag{34}$$

where

$$\theta_1 = \sum_{j=l_{k+1}}^{m}\lambda_j = \mathrm{trace}(\mathbf{R}_k) - \sum_{j=1}^{l_k}\lambda_j$$

$$\approx \mathrm{trace}(\Gamma) - \sum_{j=1}^{l_k}\lambda_j \tag{35}$$

and N_k is the length of the data set used to compute $\mathbf{R}_k$. If forgetting factor μ is employed, N_k is estimated from Eq. (18) as

$$N_k = \frac{n_k}{1-\mu}$$

If the noise in each variable has the same variance, $\mathrm{IEF}(l_k)$ will have a minimum corresponding to the number of PCs. Note that IEF works for covariance-based PCA only.

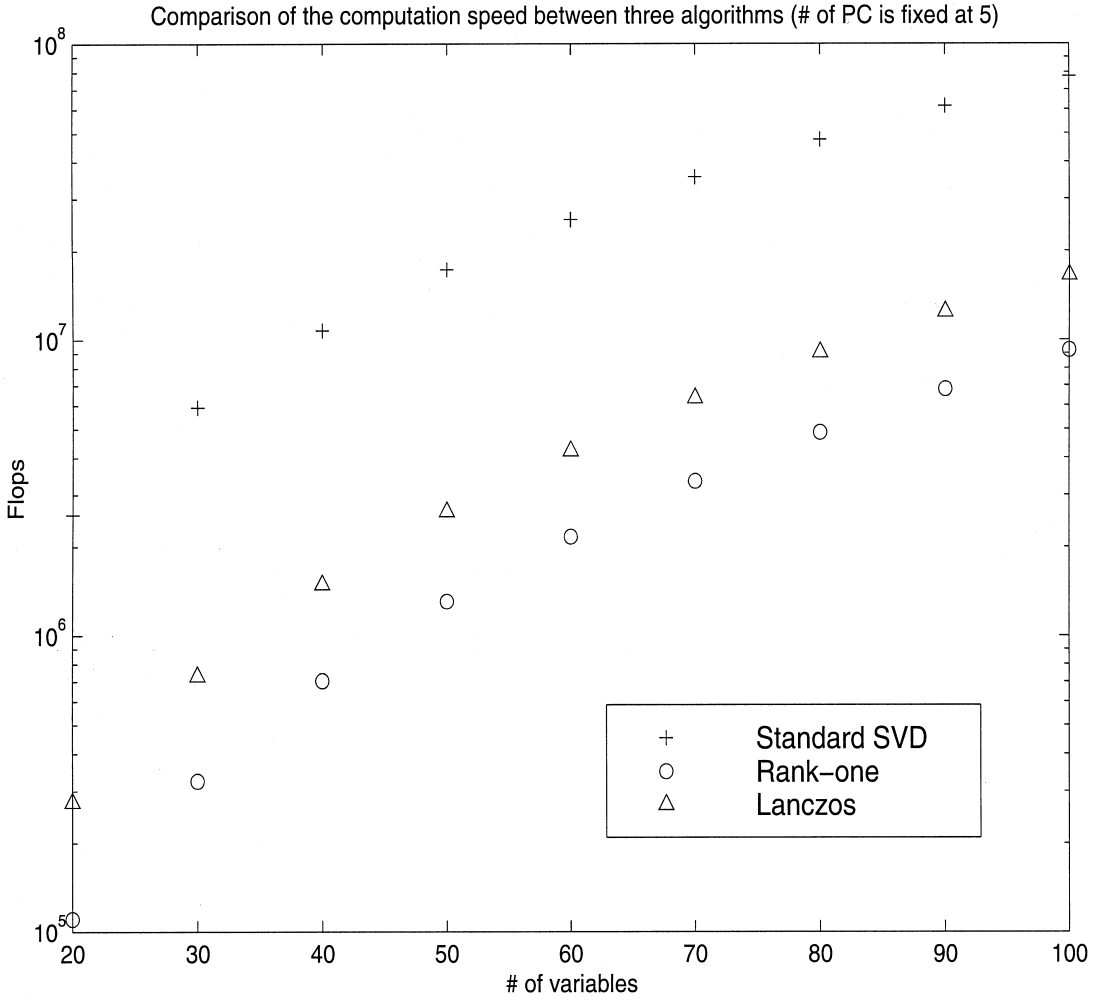

Fig. 2. Comparison of computational costs in the sample-wise case.

4. Xu and Kailath's approach: Xu and Kailath [9] determine the number of PCs for a covariance matrix based on a few of the largest eigenvalues. This approach also assumes that the smallest eigenvalues of the covariance matrix $\mathbf{R}_k$ are equal. If the remaining eigenvalues after l_k are identical,

$$\phi_{l_k} = \log\left(\frac{\sqrt{\frac{\theta_2}{m-l_k}}}{\frac{\theta_1}{m-l_k}}\right)$$

should be approximately zero, where

$$\theta_2 = \sum_{j=l_{k+1}}^{m} \lambda_j^2 = \operatorname{trace}\left(\mathbf{R}_k^2\right) - \sum_{j=1}^{l_k} \lambda_j^2 \approx \operatorname{trace}\left(\Gamma^2\right) - \sum_{j=1}^{l_k} \lambda_j^2 \tag{36}$$

Xu and Kailath [9] showed that if the observation vector x_k is sampled from a zero-mean multivariate normal process, then $N_k(m-l_k)\phi_{l_k}$ will be asymptotically chi-square distributed with $\frac{1}{2}(m-l_k)(m-l_k+1)-1$ degrees of freedom. Therefore, with a chosen level of significance β, we can select the threshold χ_β^2 for ϕ_l. If $N_k(m-l_k)\phi_l \leqslant \chi_\vartheta^2$, we accept the null hypothesis that $l = l_k$. This approach is valid only for covariance-based PCA.

5. Information theoretic criteria: Wax and Kailath [23] propose to determine the number of PCs for a covariance matrix based on AIC [22] and MDL [24] criteria. This approach assumes that the observation vector x_k is sampled from a zero-mean multivariate normal process and the covariance matrix has repeated smallest eigenvalues. Wax and Kailath [23] propose a practical criterion and show that the following two functions will be minimized when l_k is equal to the number of PCs.

$$\mathrm{AIC}(l_k) = -2\log\left(\frac{\Pi_{j=l_{k+1}}^{m} \lambda_j^{\frac{1}{m-l_k}}}{\frac{\theta_1}{m-l_k}}\right)^{(m-l_k)N_k} + 2M$$

$$\mathrm{MDL}(l_k) = -2\log\left(\frac{\Pi_{j=l_{k+1}}^{m}\lambda_j^{\frac{1}{m-l_k}}}{\frac{\theta_1}{m-l_k}}\right)^{(m-l_k)N_k} + M\log(N_k)$$

where

$$\Pi_{j=l_{k+1}}^{m}\lambda_j = \frac{\det(\mathbf{R}_k)}{\Pi_{j=1}^{l_k}\lambda_j} \approx \frac{\det(\Gamma)}{\Pi_{j=1}^{l_k}\lambda_j}$$

For real valued signals,

$$M = \frac{l_k(2m - l_k + 1)}{2}$$

and for complex valued signals,

$$M = l_k(2m - l_k)$$

Wax and Kailath [23] point out that MDL gives a consistent estimate but AIC usually overestimates the number of PCs when N_k is very large. In addition, AIC and MDL criteria are valid only for covariance-based PCA and fail for rank deficient covariance matrices, e.g. more variables than samples.

6. Variance of reconstruction error. Qin and Dunia [25] propose selecting the number of PCs based on the best reconstruction of the process variables. An important feature of this approach is that the index has a minimum corresponding to the best reconstruction and is valid for both covariance and correlation-based PCA, no matter how the observation vector x_k is distributed. The VRE method assumes that each sensor (or group of sensors) is faulty and reconstructed from other sensors based on the PCA model, $\mathbf{P}_{l_k}$. The variance of the reconstruction error for the ith sensor is [26]:

$$u_i = \frac{r_k(i,i) - 2c_k^T(:,i)\mathbf{r}_k(:,i) + c_k^T(:,i)\mathbf{R}_k c_k(:,i)}{(1 - c_k(i,i))^2}, \quad i = 1,\cdots,m \tag{37}$$

where $c_k(:,i)$ and $c_k(i,i)$ are the ith column and $iith$ element of matrix $\mathbf{P}_{l_k}\mathbf{P}_{l_k}^T$, respectively. Similarly, $\mathbf{r}_k(:,i)$

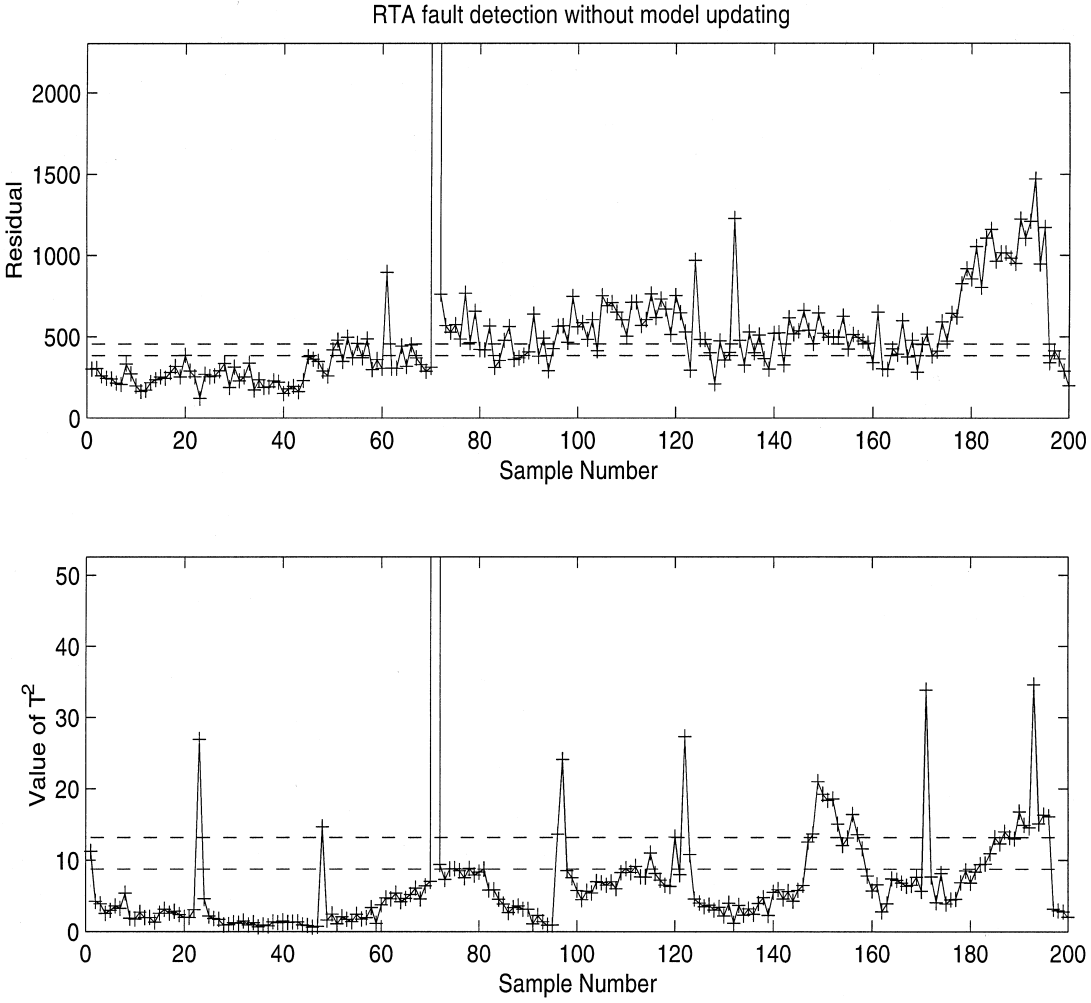

Fig. 3. SPE, and T^2 calculated from a fixed model.

and $r_k(i, i)$ are the ith column and iith element of matrix $\mathbf{R}_k$, respectively. The VRE is defined to account for all sensors,

$$\mathrm{VRE}(l_k) = \sum_{i=1}^{m} \frac{u_i}{\mathrm{var}(x_i)} \tag{38}$$

where $\mathrm{var}(x_i)$ stands for the variance of the ith element of the observation vector. Therefore, $\mathrm{VRE}(l_k)$ can be calculated recursively using only the eigenvalues and eigenvectors of $\mathbf{R}_k$ up to l_g. The VRE method selects the number of PCs that gives a minimum VRE.

7. Recursive process monitoring

When PCA is used to monitor an industrial process, the squared prediction error Q and the Hotelling T^2 are usually used. For slowly time-varying processes, the confidence limits for those detection indices will change with time, making adaptation of these limits necessary for on-line monitoring. In this section, we discuss the recursive calculation of the confidence limits for these two statistics. Q and T^2 using the RPCA models. Since our objective is to use recursive PCA for adaptive process monitoring in real time, we must consider the presence of outliers and missing values. A complete monitoring algorithm that handles these practical issues will be discussed in this section.

7.1. Adaptation of the alarm thresholds

The Q statistic is given by:

$$Q_k = \mathbf{x}^T\left(\mathbf{I} - \mathbf{P}_{l_k}\mathbf{P}_{l_k}^T\right)\mathbf{x} \tag{39}$$

where $\mathbf{x}$ is a new observation vector to be monitored. The Q statistic indicates the extent to which each sample conforms to the PCA model. It is a measure of the amount of variation not captured by the principal component model.

Based on the work by Jackson and Mudholkar [27], the upper limit for the Q statistic is given by,

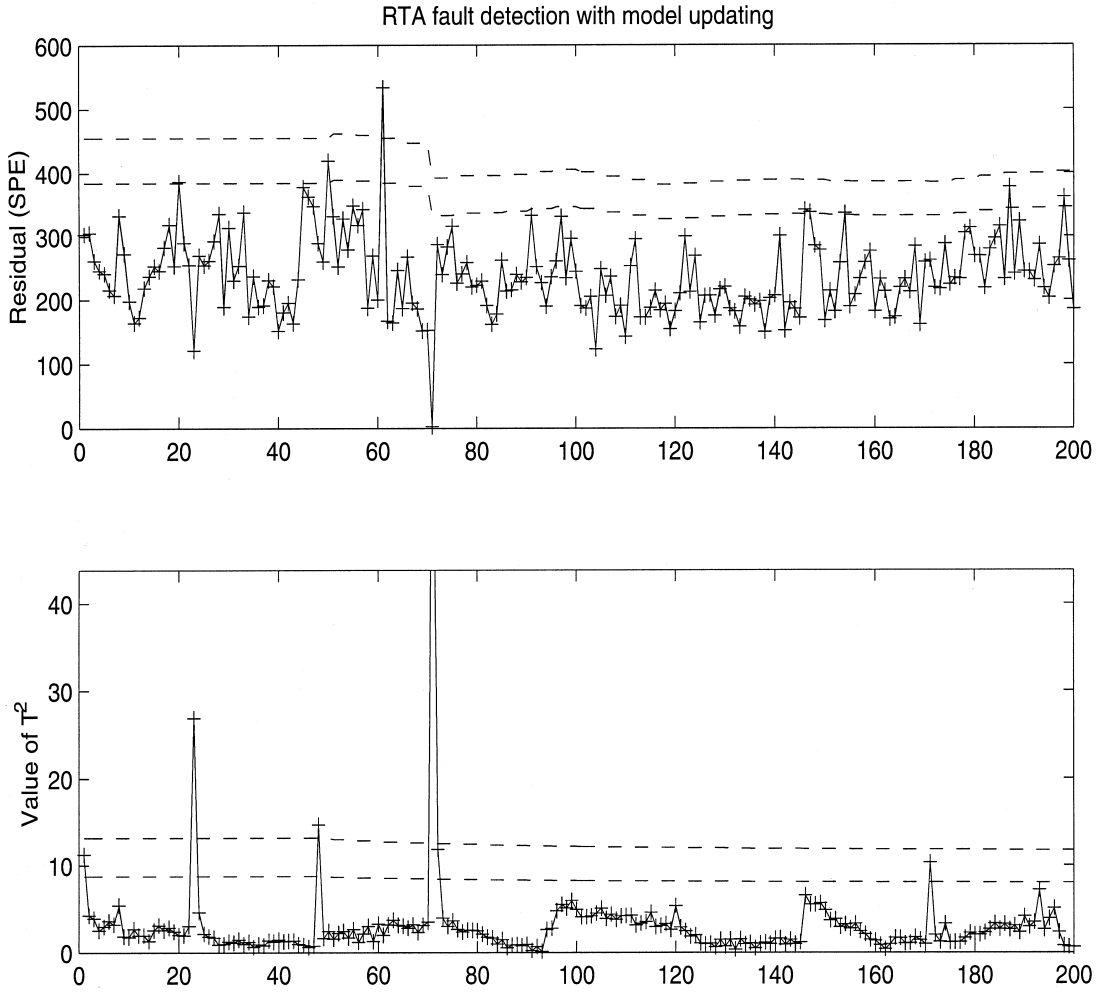

Fig. 4. The recursively calculated SPE and T^2.

$$Q_{k\cdot\alpha} = \theta_1 \left[\frac{\eta_\alpha \sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right]^{\frac{1}{h_0}} \tag{40}$$

where

$$h_0 = 1 - \frac{2\theta_1 \theta_3}{3\theta_2^2} \tag{41}$$

and η_α is the normal deviate corresponding to the upper $(1-\alpha)$ percentile. The calculation of θ_1 and θ_2 has been shown in Eqs. (35) and (36), respectively. Further,

$$\theta_3 = \sum_{i=l_{k+1}}^{m} \lambda_l^3 = \operatorname{trace}\left(\mathbf{R}_k^3\right) - \sum_{j=1}^{l_k} \lambda_j^3 \approx \operatorname{trace}\left(\Gamma^3\right) - \sum_{j=1}^{l_k} \lambda_j^3 \tag{42}$$

In the recursive implementation, θ_1, θ_2, θ_3 and h_0 are updated using the l_k largest eigenvalues after each new data block, making the limit $Q_{k\cdot\alpha}$, time-varying.

The T^2 statistic is defined by

$$T_k^2 = \mathbf{x}^T \mathbf{P}_{l_k} \Lambda_{l_k}^{-1} \mathbf{P}_{l_k}^T \mathbf{x} \tag{43}$$

where

$$\Lambda_{l_k} = \operatorname{diag}\left(\lambda_1, \cdots, \lambda_{l_k}\right)$$

follows a chi-square distribution with l_k degrees of freedom [10]. Given a level of significance β, the upper limit for the T^2 statistic is $\chi_\beta^2(l_k)$, which also needs to be updated for each new data block because l_k can be time-varying.

7.2. A complete adaptive monitoring scheme

To make the proposed recursive PCA algorithms useful in real time, we propose an adaptive monitoring scheme that addresses the issues of missing values and outliers, frequently seen in industrial processes with a large number of sensors.

Outliers are invalid sensor readings that violate the normal PCA model. Therefore, we can detect outliers using Q and T^2 indices before the PCA model is updated.

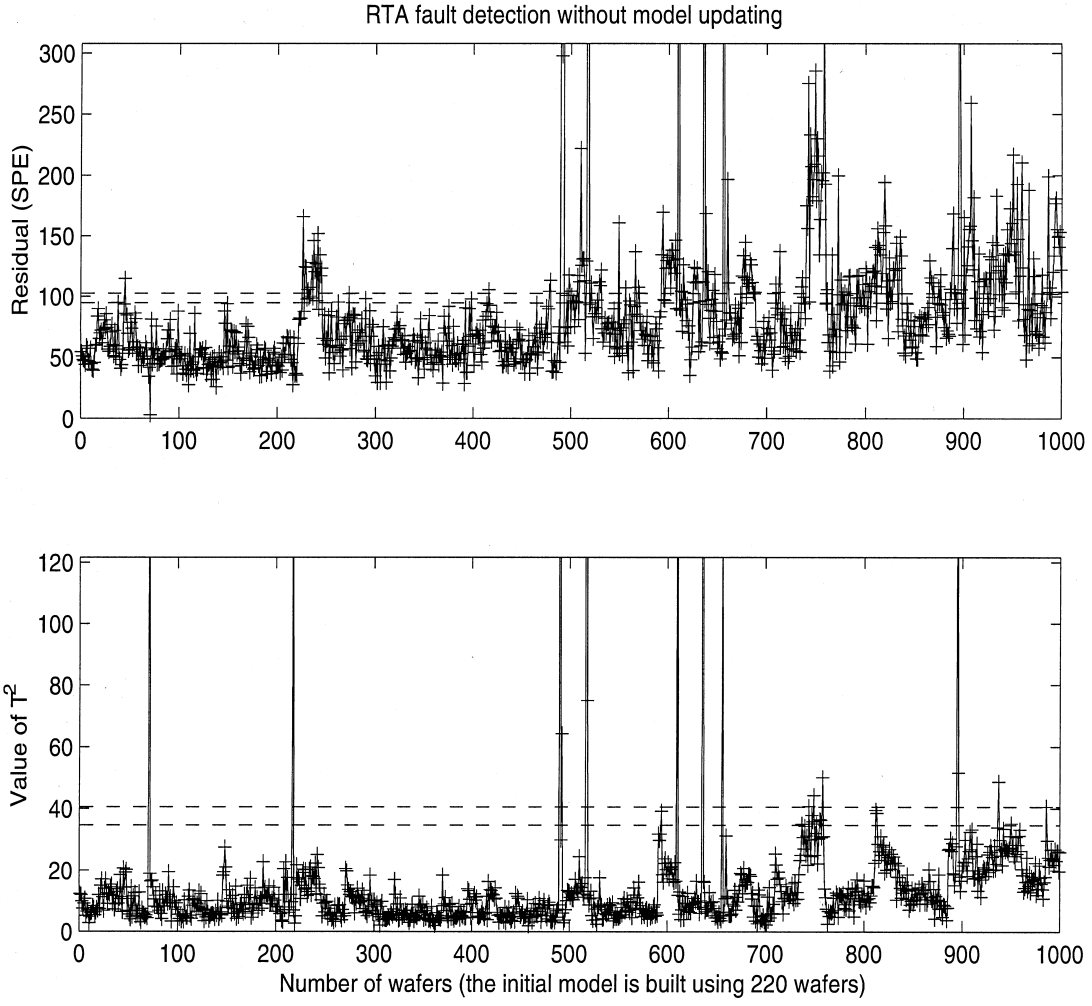

Fig. 5. SPE and ^{2} calculated from a fixed model for 1000 wafers.

The sensor reconstruction and identification methods in Dunia et al. [26] are easily used to differentiate an occasional outlier (invalid sensor reading) from an abnormal process fault. If there are missing values in the current measurement, the same reconstruction algorithm in Dunia et al. [26] can reconstruct the missing values, then use the reconstructed measurement for process monitoring.

With the tolerance limits $Q_{k\cdot\alpha}$ and $\chi^2_\beta(l_k)$ and the computed PCA loading matrix $\mathbf{P}_{l_k}$ a complete procedure for adaptive process monitoring, including normal data adaptation and fault detection, can be implemented in real-time, as follows:

1. For an initial data block $(k-1)$, calculate $\mathbf{P}_{l_k}$, l_k, $Q_{k\cdot\alpha}$ and $\chi^2_\beta(l_k)$.
2. Collect new data samples. If there are missing values in the samples, reconstruct them using the algorithms in Dunia et al. [26].
3. Calculate Q_k from Eqs. (39) and (43). If $Q_k > Q_{k\cdot\alpha}$ or $T^2_k > \chi^2_\beta(l_k)$, conduct sensor reconstruction and identification to see if the out-of-limit is due to outliers or sensor faults. If they are outliers, reconstruct them using Dunia et al. [26] and accumulate them until a new block size is reached (n_{k+1}). Otherwise, the out-of-limit is due to an abnormal process situation. The model updating is terminated and a process alarm is triggered.
4. Update the PCA model based on the new data block if no process alarms have occurred. Calculate $\mathbf{P}_{l_{k+1}}, Q_{k+1\cdot\alpha}$ and $\chi^2_\beta(l_{k+1})$. Set $k = k+1$and go to step 2.

A potential adaptation problem is that the principal component direction could change directions with minor perturbations in the data [5]. This can happen because of the bilinear decomposition, $\mathbf{X} = \mathbf{TP}^T + \mathbf{E} = (-\mathbf{T})(-\mathbf{P})^T + \mathbf{E}$, which can switch the signs of the loadings and scores simultaneously. While this switching could introduce instability if one monitors the scores in an SPC chart, it does not affect the results if one uses fault directions and a subspace approach [25,26].

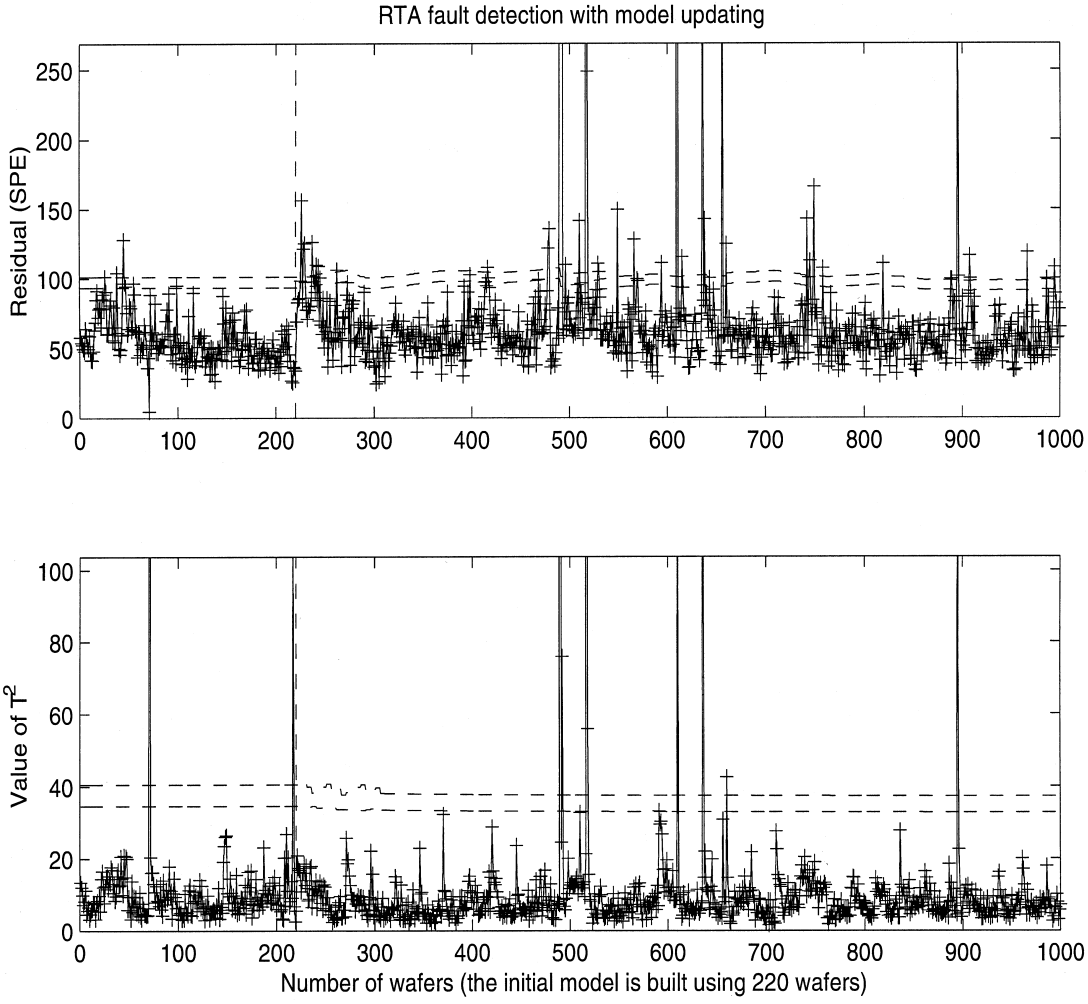

Fig. 6. The recursively calculated SPe and T^2 for 1000 wafers using a forgetting factor μ−0.995.

8. Application to a rapid thermal annealing process

To illustrate the recursive PCA and adaptive monitoring method, data from a rapid thermal annealing (RTA) process in semiconductor processing are used. The objective of this application is to build a RPCA model to adapt for slow, normal process changes and detect abnormal operations. The RTA processes experience normal drifts from wafer to wafer due to machine cleaning and contamination build-up. RTA is also subject to frequent operational errors, which usually result in off-specification products. The RTA operation is a batch process that lasts about 2 min. Six process measurements including temperature, flow and power are relevant to the monitoring task. Due to the batch nature of this process, the RPCA model is a three-way PCA that augments the batch time dimension as variables. Overall 1000 wafers (batches) of data are used to demonstrate how to update the RPCA models, determine the number of PCs, and calculate the confidence limits for SPE and T^2 recursively.

The first experiment emulates the case of limited samples to initialize the monitoring procedure. An initial PCA model is built using the first 50 wafers. When this initial PCA model is used to monitor the process and not updated with new data blocks (Fig. 3), the calculated SPE and T^2 frequently exceed their respective confidence limits even though the process is operating normally. Fig. 4 shows the SPE, and T^2 calculated from the recursively updated PCA model as well as the 95 and 99% confidence. Wafers 23 and 48, outside the T^2 limit, are deleted from the initial modeling. During the recursive updating, wafer 61 is outside the SPE limit and wafer 71 is outside the T^2 limit. For all other wafers, both indices are within their confidence limits, significantly eliminating false alarms.

The second experiment proceeds with the monitoring of the entire 1000 wafers. From here on, the RPCA model uses a forgetting factor of 0.995 to ignore old data. The first 220 wafers are used to build the initial PCA model, then the model is updated using the block-wise Lanczos algorithm, every five wafers based on the remaining 780 wafers. The monitoring results without model updating appear in Fig. 5, and those with RPCA model updating are shown in Fig. 6. Without model updating, the SPE and T^2 demonstrate an obvious trend that tends to violate

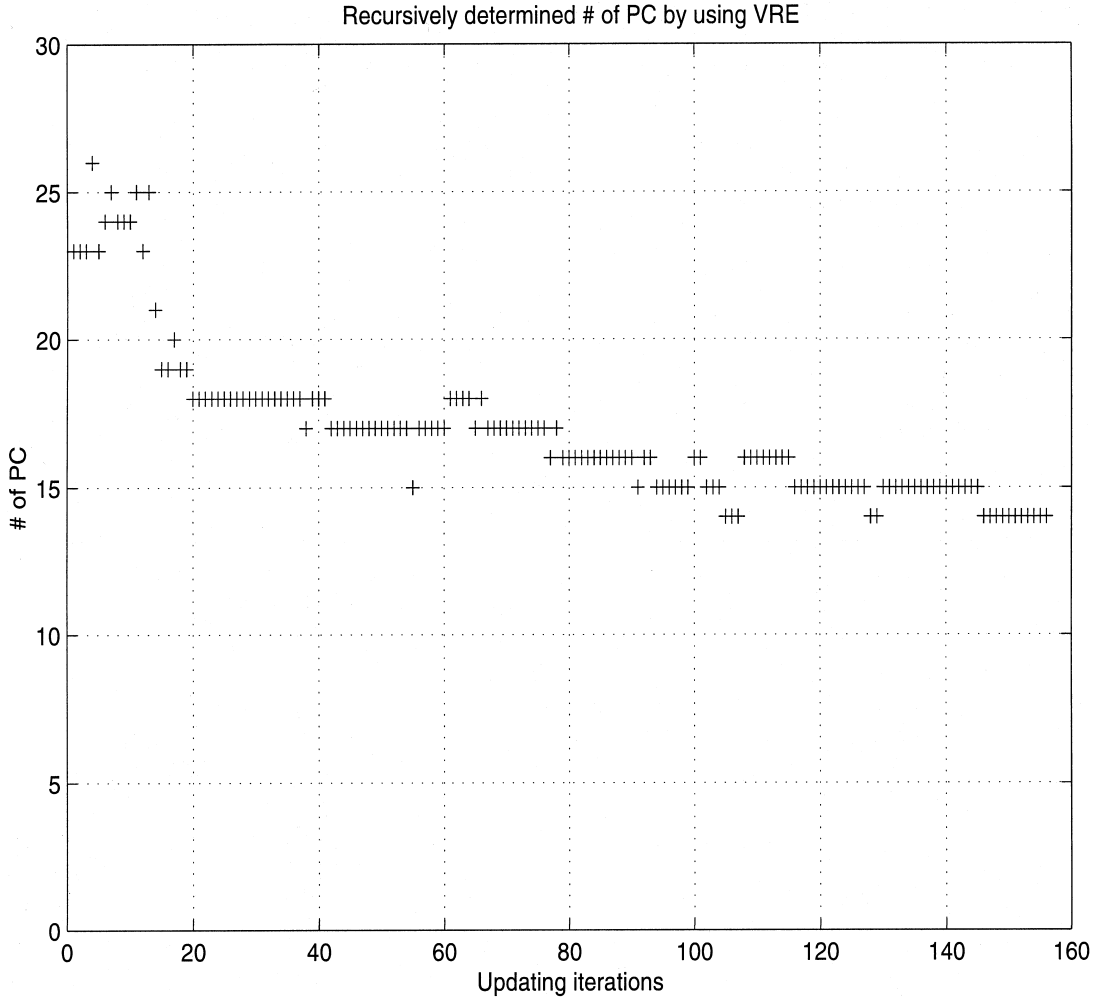

Fig. 7. The recursively calculated number of Pcs using the VRE method.

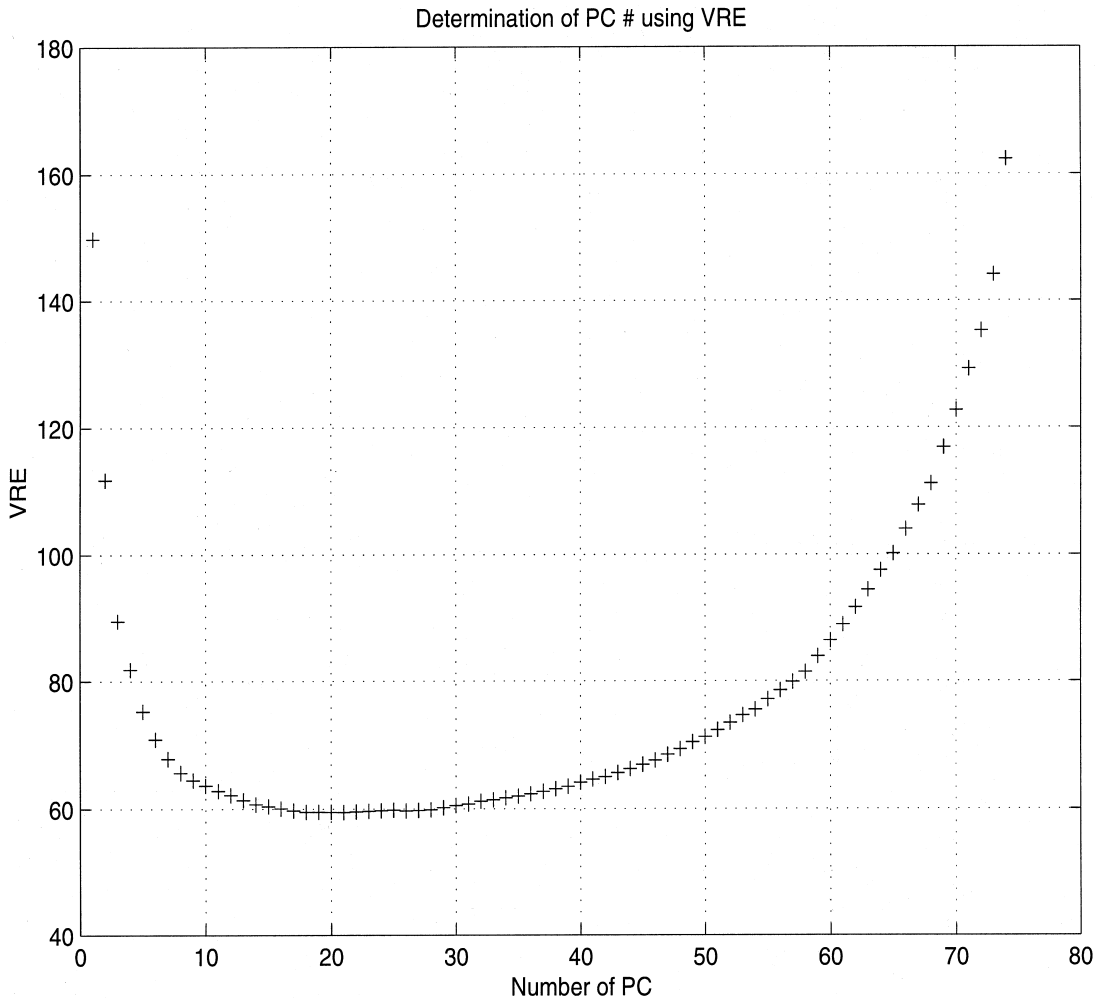

Fig. 8. The number of PCs is calculated using the VRE method.

the control limits. The results with model updating effectively capture this trend in the model and significantly reduce false alarms. Notice that in Fig. 6 the control limits are also updated based on new data. The number of significant principal components is also calculated recursively using the VRE method and is shown in Fig. 7, where 156 model updating iterations (780 wafers) are used. The number of PCs calculated during the first 20 iterations fluctuates because the associated data (from wafer 220 to 300) change abruptly. A snapshot of the VRE in one model iteration is shown in Fig. 8. Although the VRE value calculated by Eq. (38) reaches a minimum when the number of PCs is equal to 21, the VRE is rather flat near the minimum. Therefore, slight fluctuations of VRE will change the number of PCs.

9. Conclusions

Two recursive PCA algorithms based on rank-one modification and Lanczos tridiagonalization are proposed to adapt for normal process changes such as drifting. The number of PCs and the confidence limits for process monitoring are also calculated recursively. The application of the proposed algorithms to a rapid thermal process demonstrates the feasibility and effectiveness of the recursive algorithms for adaptive process monitoring. As most industrial processes experience slow and normal changes such as equipment aging, sensor drifting, catalyst deactivation, and periodic cleaning, the adaptive monitoring scheme is expected to have broad applicability in industry.

Acknowledgements

Financial support for this project from National Science Foundation under CTS-9814340, DuPont and Advanced Micro Devices is gratefully acknowledged.

Appendix A. Updating the standard deviation

We follow the notations used in Section 2.2. All the $k+1$ blocks of data, after being mean conferred, are described by

$$\mathbf{X}^0_{k+1} - 1_{k+1}\mathbf{b}^T_{k+1} = \begin{bmatrix} \mathbf{X}^0_k - 1_k\Delta\mathbf{b}^T_{k+1} - 1_k\mathbf{b}^T_k \\ \mathbf{X}^0_{n_{k+1}} - 1_{n_{k+1}}\mathbf{b}^T_{k+1} \end{bmatrix}$$

The standard deviation for the ith variable is

$$\sigma^2_{k+1\cdot i} - \frac{\left\| \begin{matrix} \mathbf{X}^0_k(:,i) - 1_k\Delta\mathbf{b}_{k+1}(i) - 1_k\mathbf{b}_k(i) \\ \mathbf{X}^0_{n_{k+1}}(:,i) - 1_{n_{k+1}}\mathbf{b}_{k+1}(i) \end{matrix} \right\|^2}{N_{k+1}-1}$$

and for the ith standard deviation, we are led to

$$\begin{aligned}(N_{k+1}-1)\sigma^2_{k+1\cdot i} &= \left\|\mathbf{X}^0_k(:,i) - 1_k\mathbf{b}_k(i)\right\|^2 + N_k\Delta\mathbf{b}^2_{k+1}(i) \\ &\quad + \left\|\mathbf{X}^0_{n_{k+1}}(:,i) - 1_{n_{k+1}}\mathbf{b}_{k+1}(i)\right\|^2\end{aligned}$$

where two relationships, $1^T_k\left(\mathbf{X}^0_k(:,i) - 1_k\mathbf{b}_k(i)\right) = 0$ and $1^T_k 1_k = N_k$ have been applied.

Further, consider the definition that

$$\sigma^2_{k\cdot i}(N_k - 1) = \left\|\mathbf{X}^0_k(:,i) - 1_k\mathbf{b}_k(i)\right\|^2$$

we have Eq. (12) instantly.

Appendix B. Updating the correlation matrix

Substituting $\mathbf{X}_{k+1}$ given by Eq. (10) into the definition

$$\mathbf{R}_{k+1} = \frac{1}{N_{k+1}-1}\mathbf{X}^T_{k+1}\mathbf{X}_{k+1}$$

directly yields

$$\begin{aligned}(N_{k+1}-1)\mathbf{R}_{k+1} &= \Sigma^{-1}_{k+1}\Sigma_k\mathbf{X}^T_k\mathbf{X}_k\Sigma_k\Sigma^{-1}_{k+1} \\ &\quad + \mathbf{1}^T_k\mathbf{1}_k\Sigma^{-1}_{k+1}\Delta\mathbf{b}_{k+1}\Delta\mathbf{b}^T_{k+1}\Sigma^{-1}_{k+1} \\ &\quad - 2\Sigma^{-1}_{k+1}\Delta\mathbf{b}_{k+1}1^T_k\mathbf{X}_k\Sigma_k\Sigma^{-1}_{k+1} + \mathbf{X}^T_{n_{k+1}}\mathbf{X}_{n_{k+1}}\end{aligned}$$

Using the facts that

$$\begin{aligned}1^T_k\mathbf{X}_k &= 0 \\ 1^T_k 1_k &= N_k \\ (N_k-1)\mathbf{R}_k &= \mathbf{X}^T_k\mathbf{X}_k\end{aligned}$$

in the above equation gives Eq. (13) immediately.

Appendix C. Computing the principal eigenvalues of Γ

Given a tridiagonal matrix $\Gamma \in \mathcal{R}^{m_1\times m_1}$ and an error limit ε, e.g. $\varepsilon = 10^{-10}$, the following algorithm calculates the l_k largest eigenvalues of Γ with $\lambda_1 \geqslant \lambda_2 \geqslant \ldots, \geqslant \lambda_{l_k}$

$\lambda_{\min} = \max\{\min(\alpha_i - |\beta_i| - |\beta_{i-1}|), 0\}$
$\lambda_{\max} = \min\{\max(\alpha_i + |\beta_i| + |\beta_{i-1}|)\}$,
trace (Γ)}, with $1 \leqslant i \leqslant m_1$
for $i = 1$ to l_k
  $x_1 = \lambda_{\min}$, $x_2 = \lambda_{\max}$
  while $|x_1 - x_2| > \varepsilon(|x_1| + |x_2|)$
    $\lambda - (x_1 + x_2)/2$
    calculate $a(\lambda)$
    if $a(\lambda) > m_1 - i$
      $x_2 = \lambda$
    else
      $x_1 = \lambda$
    end
  end
  $\lambda_{\max} = \lambda$
end

In this algorithm, the bisection approach is used to search for an eigenvalue, and the search process will be terminated after s steps if

$$\frac{\lambda_{\max} - \lambda_{\min}}{2^s} = \varepsilon$$

or equivalently

$$s = \frac{\log(\lambda_{\max} - \lambda_{\min}) - \log\varepsilon}{\log 2}$$

Since $\lambda_{\max} - \lambda_{\min} \leqslant \lambda_{\max} \leqslant \text{trace}\,(\mathbf{R}_{k+1}) = m$, it is clear that

$$s \leqslant \frac{\log(m) - \log\varepsilon}{\log 2}.$$

References

[1] B.M. Wise, N.L. Ricker, Recent advances in multivariate statistical process control: Improving robustness and sensitivity, in: Proceedings of the IFAC, ADCHEM Symposium, 1991, pp. 125–130.
[2] J.F. MacGregor, M. Koutodi, Statistical process control of multivariate process, Control Engineering Practice 3 (1995) 403–414.
[3] P. Nomikos, J.F. MacGregor, Multivariate SPC charts for monitoring batch processes, Technometrics 37 (1) (1995) 41–59.
[4] V.B. Gallagher, R.M. Wise, S.W. Butler, D.D. White, G.G. Barna, Development and benchmarking of multivariate statistical process control tools for a semiconductor etch process; improving robustness through model updating, in: Proc. of ADCHEM 97, Banff, Canada, 9–11 June 1997, pp. 78–83.
[5] S. Wold, Exponentially weighted moving principal component analysis and projection to latent structures, Chemometrics and Intelligent Laboratory systems 23 (1994) 149–161.
[6] A. Rigopoulos, Y. Arkun, F. Kayihan, E. Hanezyc, Identification of paper machine full profile disturbances models using adaptive principal component analysis, in: Chemical Process Control — V. Tahoe, 7–12 January 1996, pp. 275–279.
[7] S. Rannar, J.F. MacGregor, S. Wold, Adaptive batch monitoring using hierarchical PCA, in: AIChE Annual Meeting, Los Angeles, 16–21 November 1997.

[8] S.Joe Qin, Recursive PLS algorithms for adaptive data modeling, Comput. Chem. Eng. 22 (1998) 503–514.
[9] G. Xu, T. Kailath, Fast estimation of principal eigenspace using Lanczos algorithm, SIAM J. Matrix Anal. Appl 15 (1994) 974–994.
[10] J.E. Jackson, A User's Guide to Principal Components, Wiley-Interscience, New York, 1991.
[11] J.R. Bunch, C.P. Nielsen, D.C. Sorensen, Rank-one modification of the symmetric eigenproblem, Num. Math 31 (1978) 31–48.
[12] G.H. Golub, Some modified matrix eigenvalue problems, SIAM Rev. 15 (1973) 318–334.
[13] Golub, G.H., Van Loan, C.F. Matrix Computations, 3rd Edition, The Johns Hopkins University Press, 1996.
[14] C.C. Paige, Accuracy and effectiveness of the Lanczos algorithm for the symmetric eigenproblem, Linear Algebra and Its Appl 34 (1980) 235–258.
[15] B.N. Parlett, The Symmetric Eigenvalue Problem, Prentice Hall, Englewood Cliffs, NJ, 1980.
[16] J. Cullum, R.A. Willoughby, Lanczos algorithms for large symmetric Eigenvalue Computations, Vol. I, Theory, Birkhauser, Boston, 1985.
[17] R.I. Shrager, R.W. Hendler, Titration of individual components in a mixture with resolution of difference spectra, pks, and redox transition, Analytical Chemistry 54 (7) (1982) 1147–1152.
[18] S. Wold, Cross validatory estimation of the number of components in factor and principal component analysis, Technometrics 20 (1978) 397–406.
[19] D.W. Osten, Selection of optimal regression models via cross-validation, Journal of Chemometrics 2 (1988) 39–48.
[20] F.R. Malinowski, Factor Analysis in Chemistry, Wiley-Interscience, New York, 1991.
[21] R.B. Cattell, The scree test for the number of factors, Multivariate Behav. Res. 1 (1966) 245.
[22] H. Akaike, A new look at the statistical model identification, IEEE Trans. Auto. Con 19 (1974) 716–723.
[23] M. Wax, T. Kailath, Detection of signals by information theoretic criterial, IEEE Trans. Acoust. Speech, Signal Processing, ASSP 33 (1985) 387–392.
[24] J. Rissanen, Modeling by shortest data description, Automatica 14 (1978) 465–471.
[25] R. Dunia, J. Qin, A subspace approach to multidimensional fault identification and reconstruction, AIChE Journal 44 (1998) 1813–1831.
[26] R. Dunia, J. Qin, T.F. Edgar, T.J. McAvoy, Sensor fault identification and reconstruction using principal component analysis, in: 13th IFAC World Congress, San Francisco, 1996, pp. N2959–264.
[27] J.E. Jackson, G. Mudholkar, Control procedures for residuals associated with principal component analysis, Technometrics 21 (1979) 341–349.