Decentralized Fault Diagnosis of Large-scale Processes Using Multiblock Kernel Principal Component Analysis

ZHANG Ying-Wei[1] ZHOU Hong[1] QIN S. Joe[2]

Abstract In this paper, a multiblock kernel principal component analysis (MBKPCA) algorithm is proposed. Based on MBKPCA, a new fault detection and diagnosis approach is proposed to monitor large-scale processes. In particular, definitions of nonlinear block contributions to T^2 and the squared prediction error (SPE) statistics are first proposed in order to diagnose nonlinear faults. In addition, the relative contribution, which is the ratio of the contribution to the corresponding upper control limit, is considered to find process variables or blocks responsible for faults. The proposed method is applied to fault detection and diagnosis in the Tennessee Eastman process. The proposed decentralized nonlinear approach effectively captures the nonlinear relationship in the block process variables and shows superior fault diagnosis ability compared with other methods.

Key words Multiblock kernel methods, nonlinear component analysis, process monitoring, fault detection, principal component analysis (PCA)

DOI 10.3724/SP.J.1004.2010.00593

Multivariate statistical approaches based on principal component analysis (PCA) have been widely applied in industry for process monitoring[1−7]. PCA divides data systematically into two parts; the first part is the data with wide variation and the second part is the data with the least variance, which is noisy. Two statistics, T^2 and the squared prediction error (SPE) are used for process monitoring in the model and residual subspaces, respectively. Due to large-scale process dimensionalities and nonlinearities, process monitoring and diagnosis become difficult and the results obtained by PCA methods are hard to interpret for reaching a proper decision[8−12].

For a complicated large-scale process, consensus PCA (CPCA) models the process by dividing all measured variables into several blocks, which implements decentralized monitoring[9]. The multiblock approach effectively captures the relationship in the complex process variables and shows superior fault diagnosis ability in large-scale processes compared with other methods. Advantages of the multiblock approach are as follows: 1) Multiblock approaches can reduce the complexity of process analysis; 2) Multiblock process diagnosis monitors processes in a decentralized manner. However, CPCA performs poorly because it uses second-order statistics and assumes linearity when it is applied to large-scale process data having nonlinear characteristics.

To solve the problem posed by nonlinear data, nonlinear PCA approaches have been developed. Mark[13] developed auto-associative neural networks having five layers (input, mapping, bottleneck, damping, and output layers). Dong and McAvoy[14] proposed a nonlinear PCA based on principal curves and neural networks and applied it to nonlinear process monitoring. Alternative nonlinear PCA methods based on input-training neural networks[15] have been also developed. However, most of the existing nonlinear PCA approaches are based on neural networks; thus, a nonlinear optimization problem has to be solved to compute principal components, and the number of principal components must be specified in advance before training the neural network. Recently, kernel theory has found increasing numbers of applications in nonlinear processes[16−22]. Kernel principal component analysis (KPCA) was introduced for nonlinear process monitoring and fault detection[23−24]. KPCA computes principal components in a high-dimensional feature space, which is nonlinearly related to the input space. However, fault diagnosis is difficult since the nonlinear mapping function from input space to feature space is unknown.

In this paper, a new fault detection and diagnosis approach based on multiblock kernel principal component analysis,(MBKPCA) is proposed to monitor large-scale processes. In Section 1, an iterative KPCA based on nonlinear iterative partial least squares (NIPALS)[9] is proposed; also, MBKPCA algorithm is proposed and discussed in this section. In Section 2, MBKPCA for diagnosis is described. Then, the superiority of MBKPCA and its application to process monitoring are illustrated through the Tennessee Eastman process in Section 3. Finally, Conclusions are given in Section 4.

1 Multiblock kernel principal component analysis (MBKPCA)

1.1 An iterative algorithm for KPCA

KPCA is an extension of PCA, and it can be solved as an eigenvalues problem of its kernel matrix. NIPALS[9] algorithm is used for the computation of PCA. NIPALS offers the same result as the eigenvalue method proposed by Scholkopf[19], but with NIPALS algorithm, the nonlinear principal components (PCs) are obtained one by one. The NIPALS algorithm is proposed in Table 1.

Table 1 NIPALS for KPCA

Step	For comprehension	For computation
1	Scale K	Scale K
2	Initialize $\boldsymbol{t}_i$	Initialize $\boldsymbol{t}_i$
3	$\boldsymbol{p}_i = \Phi_i^{\mathrm{T}}\boldsymbol{t}_i/\|\Phi_i^{\mathrm{T}}\boldsymbol{t}_i\|$ $\boldsymbol{t}_i = \Phi_i\boldsymbol{p}_i$	$\boldsymbol{t}_i = K_i\boldsymbol{t}_i/\sqrt{\boldsymbol{t}_i^{\mathrm{T}}K_i\boldsymbol{t}_i}$
	Loop until $\boldsymbol{t}_i$ converges	Loop until $\boldsymbol{t}_i$ converges
4	$\Phi_{i+1} = (I - \boldsymbol{t}_i\boldsymbol{t}_i^{\mathrm{T}}/\boldsymbol{t}_i^{\mathrm{T}}\boldsymbol{t}_i)\Phi_i$	$K_{i+1} = (I - \boldsymbol{t}_i\boldsymbol{t}_i^{\mathrm{T}}/\boldsymbol{t}_i^{\mathrm{T}}\boldsymbol{t}_i)\times$ $K_i(I - \boldsymbol{t}_i\boldsymbol{t}_i^{\mathrm{T}}/\boldsymbol{t}_i^{\mathrm{T}}\boldsymbol{t}_i)$
	Go to Step 3	Go to Step 3

In the first step of iteration, loadings vector $\boldsymbol{p}_i$ is obtained as:

$$\boldsymbol{p}_i = \Phi_i^{\mathrm{T}}\boldsymbol{t}_i \tag{1}$$

where Φ is the mapping of x from the input space into the feature space and $\boldsymbol{t}$ in the score vector. Then, $\boldsymbol{p}_i$ is normalized to unit length by

$$\boldsymbol{p}_i = \frac{\boldsymbol{p}_i}{\|\boldsymbol{p}_i\|} = \frac{\boldsymbol{p}_i}{\sqrt{\boldsymbol{t}_i^{\mathrm{T}}K_i\boldsymbol{t}_i}} \tag{2}$$

In the second step of iteration, the scores vector is updated

Manuscript received December 25, 2008; accepted March 13, 2009
Supported by National Basic Research Program of China (973 Program) (2009CB320604) and the 111 Project
1. Key Laboratory of Integrated Automation of Process Industry, Ministry of Education, Shenyang 110004, P. R. China 2. Mork Family Department of Chemical Engineering and Materials Science, Ming Hsieh Department of Electrical Engineering, Daniel J. Epstein Department of Industrial and Systems Engineering, University of Southern California, Los Angeles CA 90089-1211, United States of America

by

$$\boldsymbol{t}_i = \Phi_i \boldsymbol{p}_i \tag{3}$$

By combining (2) and (3),

$$\boldsymbol{t}_i = \Phi_i \boldsymbol{p}_i = \frac{\Phi_i \Phi_i^{\mathrm{T}} \boldsymbol{t}_i}{\|\Phi_i^{\mathrm{T}} \boldsymbol{t}_i\|} = \frac{K_i \boldsymbol{t}_i}{\sqrt{\boldsymbol{t}_i^{\mathrm{T}} K_i \boldsymbol{t}_i}} \tag{4}$$

After the convergence of $\boldsymbol{t}_i$, residual of Φ_i is deflated by

$$\Phi_{i+1} = \left(I - \frac{\boldsymbol{t}_i \boldsymbol{t}_i^{\mathrm{T}}}{\boldsymbol{t}_i{}^{\mathrm{T}} \boldsymbol{t}_i}\right) \Phi_i \tag{5}$$

In the process of computation, $K = \Phi\Phi^{\mathrm{T}}$ converts (5) to the following:

$$K_{i+1} = \left(I - \frac{\boldsymbol{t}_i \boldsymbol{t}_i^{\mathrm{T}}}{\boldsymbol{t}_i^{\mathrm{T}} \boldsymbol{t}_i}\right) K_i \left(I - \frac{\boldsymbol{t}_i \boldsymbol{t}_i^{\mathrm{T}}}{\boldsymbol{t}_i^{\mathrm{T}} \boldsymbol{t}_i}\right) \tag{6}$$

This residual is used for computing the next PC.

1.2 Algorithm derivation

1.2.1 Dividing the kernel matrix into blocks

Radial basis function is selected to build the kernel matrix. The element belonging to i-th row j-th column of kernel matrix is $K_{i,j} = \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/c)$. Note that in 2-norm there exists the relation $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 = \sum_{b=i}^{B} \|x_{b,\,i} - x_{b,\,j}\|^2$. This can lead to the following property.

$$K_{i,\,j} = \prod_{b=1}^{B} \exp\left(\frac{-\|\boldsymbol{x}_{b,\,i} - \boldsymbol{x}_{b,\,j}\|}{c}\right) = \prod_{b=1}^{B} K_{i,\,j}^{b}$$

Due to this property, dividing $\boldsymbol{x}$ into B blocks, one can compute the kernel matrix of each block $K_b = \{\boldsymbol{k}_{i,\,j}^{b}\}$. The kernel matrix of each block should be centered by

$$\bar{K_b} = K_b - 1_N K_b - K_b 1_N + 1_N K_b 1_N \tag{7}$$

where matrix $1_N = (1/N)I$ and $I \in \mathbf{R}^{N \times N}$ is identity matrix.

1.2.2 Algorithm derivation

MBKPCA first uses super scores $\boldsymbol{t}_{T,\,i}$ to compute block loadings $\boldsymbol{p}_{b,\,i}$

$$\boldsymbol{p}_{b,\,i} = \Phi(X_{b,\,i})^{\mathrm{T}} \boldsymbol{t}_{T,\,i} \tag{8}$$

Normalize the block loadings

$$\boldsymbol{p}_{b,\,i} = \frac{\boldsymbol{p}_{b,\,i}}{\|\boldsymbol{p}_{b,\,i}\|} = \frac{\Phi(X_{b,\,i})^{\mathrm{T}} \boldsymbol{t}_{T,i}}{\sqrt{\boldsymbol{t}_{T,\,i}{}^{\mathrm{T}} K_{b,\,i} \boldsymbol{t}_{T,\,i}}} \tag{9}$$

Block scores from block loadings are computed by

$$\boldsymbol{t}_{b,\,i} = \Phi(X_{b,\,i}) \boldsymbol{p}_{b,\,i} = \frac{K_{b,\,i} \boldsymbol{t}_{T,\,i}}{\sqrt{\boldsymbol{t}_{T,\,i}{}^{\mathrm{T}} K_{b,i} \boldsymbol{t}_{T,\,i}}} \tag{10}$$

Arrange all block scores into a single matrix

$$T_i = [\boldsymbol{t}_{1,\,i} \ \cdots \ \boldsymbol{t}_{B,\,i}] \tag{11}$$

Super loadings are obtained by regressing T_i on $\boldsymbol{t}_{T,\,i}$ and then normalize it

$$\boldsymbol{p}_{T,\,i} = \frac{T_i^{\mathrm{T}} \boldsymbol{t}_{T,1}}{\|T_i^{\mathrm{T}} \boldsymbol{t}_{T,\,i}\|} \tag{12}$$

Update the super loadings $\boldsymbol{t}_{T,\,i}$ using

$$\boldsymbol{t}_{T,\,i} = T_i \boldsymbol{p}_{T,\,i} \tag{13}$$

Repeat these steps until $\boldsymbol{t}_{T,\,i}$ converges to a predefined precision. The residual is used to compute the next PC. The residual is deflated using $\boldsymbol{t}_{T,1}$

$$K_{b,\,i+1} = \left(I - \frac{\boldsymbol{t}_{T,\,i} \boldsymbol{t}_{T,\,i}{}^{\mathrm{T}}}{\boldsymbol{t}_{T,\,i}{}^{\mathrm{T}} \boldsymbol{t}_{T,\,i}}\right) K_{b,\,i} \left(I - \frac{\boldsymbol{t}_{T,\,i} \boldsymbol{t}_{T,\,i}{}^{\mathrm{T}}}{\boldsymbol{t}_{T,\,i}{}^{\mathrm{T}} \boldsymbol{t}_{T,\,i}}\right) \tag{14}$$

The MBKPCA algorithm is summarized as follows.

Algorithm for MBKPCA

Step 1. Scale each block data to 0 means;
Step 2. Initialize $\boldsymbol{t}_{T,\,i}$;
Step 3. For each block, compute $\boldsymbol{t}_{b,\,i} = K_{b,\,i} \boldsymbol{t}_{T,\,i} / \sqrt{\boldsymbol{t}_{T,\,i}{}^{\mathrm{T}} K_{b,\,i} \boldsymbol{t}_{T,\,i}}$;
Step 4. $T_i = [\boldsymbol{t}_{1,\,i} \ \cdots \ \boldsymbol{t}_{B,\,i}]$;
Step 5. $\boldsymbol{p}_{T,\,i} = T_i^{\mathrm{T}} \boldsymbol{t}_{T,\,i} / \|T_i^{\mathrm{T}} \boldsymbol{t}_{T,\,i}\|$;
Step 6. $\boldsymbol{t}_{T,\,i} = T_i \boldsymbol{p}_{T,\,i}$;
Step 7. If $\boldsymbol{t}_{T,\,i}$ is not converging, go to Step 3; else, go to Step 8;
Step 8. For each block, deflate residual $K_{b,\,i+1} = (I - \boldsymbol{t}_{T,\,i} \boldsymbol{t}_{T,\,i}{}^{\mathrm{T}} / (\boldsymbol{t}_{T,\,i}{}^{\mathrm{T}} \boldsymbol{t}_{T,\,i})) K_{b,\,i} (I - \boldsymbol{t}_{T,\,i} \boldsymbol{t}_{T,\,i}{}^{\mathrm{T}} / (\boldsymbol{t}_{T,\,i}{}^{\mathrm{T}} \boldsymbol{t}_{T,\,i}))$;
Step 9. Go to Step 2 to get the next PC.

2 MBKPCA for fault diagnosis

PCA performs well in many cases, but it lacks the ability to exhibit significant nonlinear characteristics since PCA assumes that process data are linear. To solve the issue of data nonlinearity, KPCA have been used to process monitoring. It is difficult to diagnose faults for KPCA since the nonlinear mapping function is unknown. MBKPCA has superior fault diagnosis ability since variables are grouped compared with KPCA. The main advantage of MBKPCA over KPCA on fault diagnosis is that MBKPCA provides the block statistics.

Both T^2 statistic and SPE statistic can be used for monitoring the process. For a new sample x_{new} , divide it into B blocks and then map them into feature space: $\Phi(x_{b,\,\mathrm{new}})$ for $b = 1, \cdots, B$. For each block, compute the kernel vector $\boldsymbol{k}_{b,\,\mathrm{new}} = \Phi(x_{b,\,\mathrm{new}}) \Phi(X_b)^{\mathrm{T}}$.

$$\bar{\boldsymbol{k}}_{b,\,\mathrm{new}} = \boldsymbol{k}_{b,\,\mathrm{new}} - \mathbf{1}'_N K_b - 1_N \boldsymbol{k}_{b,\,\mathrm{new}} + \mathbf{1}'_N K_b 1_N \tag{15}$$

where $\mathbf{1}'_N = \frac{1}{N}[1 \ \cdots \ 1]$ in $\mathbf{R}^{1 \times N}$. The block coefficient matrix A_b is denoted by

$$A_{b,\,i} = \frac{\boldsymbol{t}_{T,\,i}}{\sqrt{\boldsymbol{t}_{T,\,i} K_b \boldsymbol{t}_{T,\,i}}} \tag{16}$$

as given in (10). It can be used to compute block scores of the new sample

$$\boldsymbol{t}_{b,\,\mathrm{new}} = \boldsymbol{k}_{b,\,\mathrm{new}} A_b \tag{17}$$

where $\boldsymbol{t}_{b,\,\mathrm{new}}$ is a row vector, which is the score of one sample. For each column i, use $T_{\mathrm{new}}^{i} = [\boldsymbol{t}_{1,\,\mathrm{new}}^{i} \ \cdots \ \boldsymbol{t}_{B,\mathrm{new}}^{i}]$ and super loadings to calculate the i-th super scores element of the sample

$$\boldsymbol{t}_{b,\,\mathrm{new}}^{i} = T_{\mathrm{new}}^{i} \boldsymbol{p}_T \tag{18}$$

The super scores $\boldsymbol{t}_{T,\,\mathrm{new}} = [\boldsymbol{t}_{b,\,\mathrm{new}}^{1} \ \cdots \ \boldsymbol{t}_{b,\,\mathrm{new}}^{i}]$. The super T^2 statistic can be calculated by

$$T_{\mathrm{new}}^{2} = \boldsymbol{t}_{T,\,\mathrm{new}} \Lambda^{-1} \boldsymbol{t}_{T,\,\mathrm{new}}{}^{\mathrm{T}} \tag{19}$$

where Λ^{-1} is the inverse of the covariance matrix of super scores. Block T^2 statistic is computed by

$$T^2_{b,\,\mathrm{new}} = \boldsymbol{t}_{b,\,\mathrm{new}} \Lambda_b^{-1} \boldsymbol{t}_{b,\,\mathrm{new}}{}^{\mathrm{T}} \tag{20}$$

where Λ_b^{-1} is the inverse of the covariance matrix of block scores. The block SPE statistic is defined as

$$\begin{aligned} SPE_b = \bar{\Phi}(x_{b,\,\mathrm{new}})(I - PP^{\mathrm{T}})\bar{\Phi}^{\mathrm{T}}(x_{b,\,\mathrm{new}}) = \\ \bar{k}(x_{b,\,\mathrm{new}}, x_{b,\,\mathrm{new}}) - \bar{k}^{\mathrm{T}}_{b,\,\mathrm{new}} A_b A_b^{\mathrm{T}} \bar{k}_{b,\,\mathrm{new}} \end{aligned} \tag{21}$$

and

$$\begin{aligned} \bar{k}(x_{b,\,\mathrm{new}}, x_{b,\,\mathrm{new}}) = k(x_{b,\,\mathrm{new}}, x_{b,\,\mathrm{new}}) - \\ \frac{2}{N}\sum_{i=1}^{N} k(x_{b,\,i}, x_{b,\,\mathrm{new}}) + \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} k(x_{b,\,i}, x_{b,\,j}) \end{aligned} \tag{22}$$

where $k(x, y) = \exp(-\|x - y\|^2/c)$ is kernel function. The super SPE statistic is directly cumulated by

$$SPE = \sum_{b=1}^{B} SPE_b \tag{23}$$

3 Simulation results

3.1 An artificial nonlinear case

To illustrate the performance of the MBKPCA over CPCA, we apply these methods to a simple example system. Let x_1 increase from 1 to 500 and x_4 be the random variable which has 500 samples. Assume that x_1 and x_4 are input variables. Variables x_2 and x_3 are outputs of x_1, and x_5 is output of x_4 . These variables are related as: $x_2 = \cos(2\pi x_1^2/500)$, $x_3 = x_2^2 - \sqrt{\sin(x_1)^2}$, and $x_5 = \sin x_4^2$. Noises with zero mean and standard deviation 0.01 are added to each variable. At the 250-th sample, if x_1 deviates from its "normal" value, then x_2 and x_3 will be affected, as shown in Fig. 1. Fig. 1 (a) depicts the normal data, and Fig. 1 (b) depicts the faulty data. Based on the structure of this system, the variables are divided into 2 groups: $\boldsymbol{X}_1 = [x_1, x_2, x_3]$ and $\boldsymbol{X}_2 = [x_4, x_5]$, MBKPCA can detect the fault using the SPE chart, as shown in Fig. 2.

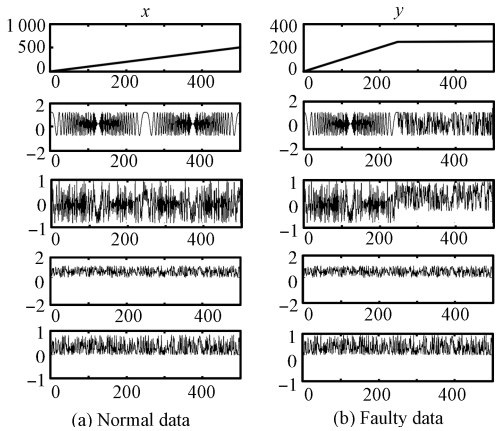


Fig. 1 Normal and faulty data

Furthermore, MBKPCA can diagnose whether the block is affected by the fault or not, i.e., it can detect that the first block is affected by the fault and the second block is not affected. For comparison, CPCA is used to monitor the process. As shown in Fig. 3, CPCA cannot detect the fault. In Figs. 3 and 4, the two lines denote the control limits with 99 % and 95 % confidence, respectively.

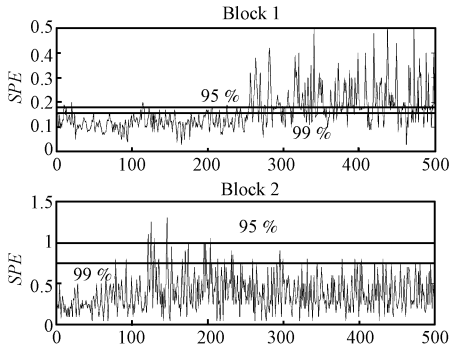


Fig. 2 Block SPE by MBKPCA

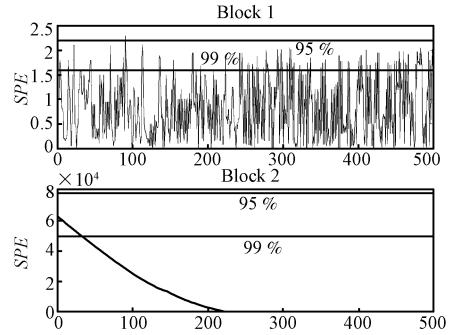
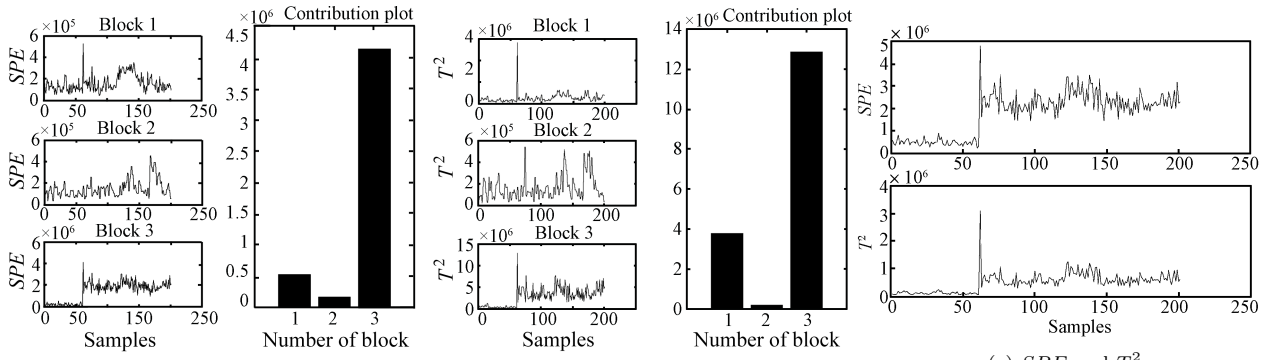


Fig. 3 Block SPE by CPCA

3.2 The Tennessee Eastman process

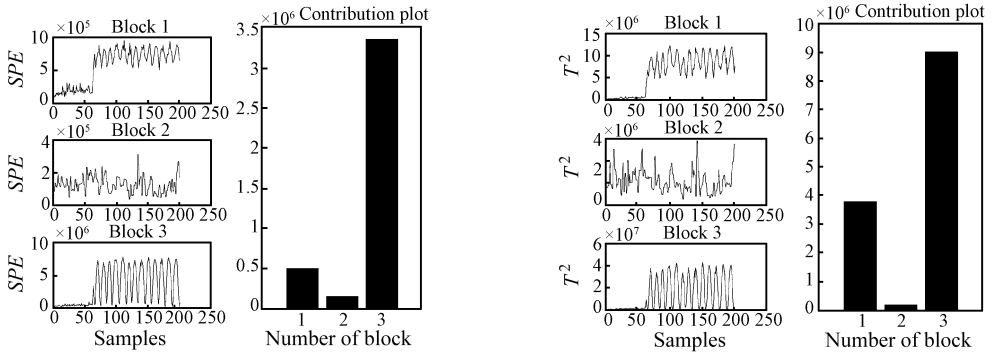
In this section, the proposed method is applied to the Tennessee Eastman process simulation data. The Tennessee Eastman process is a complex nonlinear process, which was created by Eastman Chemical Company to provide a realistic industrial process for evaluating process control and monitoring methods. The test process is based on a simulation of an actual industrial process where the components, kinetics, and operating conditions have been modified for proprietary reasons. There are five major unit operations in the process: a reactor, a condenser, a recycle compressor, a separator, and a stripper. The four reactants A, C, D, and E and the inert B are fed to the reactor where the products G and H are formed and a by product F is also produced. The process has 22 continuous process measurements, 12 manipulated variables, and 19 composition measurements sampled less frequently. In this study, a total of 52 variables are used for monitoring. A sampling interval of 3 minutes was used to collect the simulated data for the training and testing sets. The data can be downloaded from http://brahms.scs.uiuc.edu. Variables are divided into 3 groups: continuous process measurements, manipulated variables, and composition measurements.

To show the performance of the proposed MBKPCA, select faults 4 and 14 to test MBKPCA. The 100-th to 300-th samples were used to test the proposed method since faults occurred at about 160-th samples of test data set. First, the model was built from a training data set of 500 normal samples where the parameter of radial basis function was set to 500. Second, a test data set is used to test the fault



(a) Block SPE (b) Block T^2 (c) SPE and T^2

Fig. 4 Monitoring and diagnosis results for fault 4



(a) Block SPE (b) Block T^2

Fig. 5 Monitoring and diagnosis results for fault 14

diagnosis ability of the proposed method. The 51-st variable in Block 3 is most related to both fault 4 and fault 14. In Fig. 4 (a), block SPE plots indicate that Block 3 is affected by the fault 4; the contribution plot of block SPE at the 161-st samples shows Block 3 is suspect. The same situation is shown in Fig. 4 (b) through block T^2. In Fig. 4 (c), although SPE and T^2 plots generated by KPCA can detect the fault, one could not diagnose which part of the process is most related to the fault. From Fig. 5, one can see that fault 14 mainly affects Block 1 and Block 3. These figures show that the MBKPCA can locate the block most related to the fault.

4 Conclusion

A new approach to complex process monitoring based on MBKPCA is proposed in this paper. The multiblock approach effectively captured the relationship in the complex process blocks and showed fault detection and diagnosis ability in large-scale processes compared with other kernel methods. In the future, the following research work is worth studying: 1) Find a way to form blocks in feature space to implement consensus KPCA using the result of KPCA; 2) Find the relation of each K_b in MBKPCA to enhance the interpretability; 3) Determine the optimal number of principal components in the kernel space and identify which variable causes the process fault; 4) Explore different nonlinear relationships in different blocks, i.e., the kernel function and parameters may be distinct for different blocks.

References

1 Kano M, Nakagawa Y. Data-based process monitoring, process control, and quality improvement: recent developments and applications in steel industry. *Computers and Chemical Engineering*, 2008, **32**(1-2): 12−24

2 Yoo C, Kim M, Hwang S, Jo Y, Oh J. Online predictive monitoring and prediction model for a periodic process through multiway non-Gaussian modeling. *Chinese Journal of Chemical Engineering*, 2008, **16**(1): 48−51

3 Zhao C H, Wang F L, Lu N Y, Jia M X. Stage-based soft-transition multiple PCA modeling and on-line monitoring strategy for batch processes. *Journal of Process Control*, 2007, **17**(9): 728−741

4 Xiong L, Liang J, Qian J X. Multivariate statistical process monitoring of an industrial polypropylene catalyzer reactor with component analysis and kernel density estimation. *Chinese Journal of Chemical Engineering*, 2007, **15**(4): 524−532

5 Kruger U, Kumar S, Littler T. Improved principal component monitoring using the local approach. *Automatica*, 2007, **43**(9): 1532−1542

6 Aguado D, Rosen C. Multivariate statistical monitoring of continuous wastewater treatment plants. *Engineering Applications of Artificial Intelligence*, 2008, **21**(7): 1080−1091

7 Qin S J. Statistical process monitoring: basics and beyond. *Journal of Chemometrics*, 2003, **17**(8-9): 480−502

8 Westerhuis J A, Kourti T, MacGregor J F. Analysis of multiblock and hierarchical PCA and PLS models. *Journal of Chemometrics*, 1998, **12**(5): 301−321

9 Qin S J, Valle S, Piovoso M J. On unifying multiblock analysis with application to decentralized process monitoring. *Journal of Chemometrics*, 2001, **15**(9): 715−742

10 Choi S W, Lee I B. Multiblock PLS–based localized process diagnosis. *Journal of Process Control*, 2005, **15**(3): 295−306

11 MacGregor J F, Jaeckle C, Kiparissides C, Koutoudi M. Process monitoring and diagnosis by multiblock PLS methods. *American Institute of Chemical Engineers Journal*, 1994, **40**(5): 826−838

12 Kourti T, Nomikos P, MacGregor J F. Analysis, monitoring and fault diagnosis of batch processes using multiblock and multiway PLS. *Journal of Process Control*, 1995, **5**(4): 277−284

13 Mark A K. Nonlinear principal component analysis using autoassociative neural networks. *American Institute of Chemical Engineers Journal*, 1991, **37**(2): 233−243

14 Dong D, McAvoy T J. Nonlinear principal component analysis-based on principal curves and neural networks. *Computers and Chemical Engineering*, 1996, **20**(1): 65−78

15 Tan S F, Mavrovouniotis M L. Reducing data dimensionality through optimizing neural network inputs. *American Institute of Chemical Engineers Journal*, 1995, **41**(6): 1471−1480

16 Kocsor A, Toth L. Kernel-based feature extraction with a speech technology application. *IEEE Transactions on Signal Processing*. 2004, **52**(8): 2250−2263

17 Mika S, Scholkopf B, Smola A, Muller K R, Scholz M, Ratsch G. Kernel PCA and de-noising in feature spaces. In: Proceedings on Neural Information Processing Systems. Berlin, UK: Springer, 1991. 536−542

18 Romdhani S, Gong S, Psarrou A. A multi-view nonlinear active shape model using kernel PCA. In: Proceedings of British Machine Vision Conference. Nottingham, UK: Springer, 1999. 483−492

19 Scholkopf B, Smola, A, Muller K R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 1998, **10**(5): 1299−1319

20 Scholkopf B, Mika S, Burges C J C, Knirsch P, Muller K R, Ratsch G. Input space versus feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, 1999, **10**(5): 1000−1017

21 Zhang Y W, Qin S J. Fault detection of nonlinear processes using multiway kernel independent component analysis. *Industrial and Engineering Chemistry Research*, 2007, **46**(23): 7780−7787

22 Zhang Y W, Qin S J. Improved nonlinera fault detection technique and statistical analysis. *American Institute of Chemical Engineers Journal*, 2008, **54**(12): 3207−3220

23 Lee J M, Yoo C K, Lee I B. Fault detection of batch processes using multiway kernel principal component analysis. *Computers and Chemical Engineering*, 2004, **28**(9): 1837−1847

24 Lee J M, Yoo C K, Choi S W, Vanrolleghem P A, Lee I B. Nonlinear process monitoring using kernel principal component analysis. *Chemical Engineering Science*, 2004, **59**(1): 223−234

ZHANG Ying-Wei Received her double B. S. degree in both automation and mathematics from Haerbin Institute of Technology, the master and Ph. D. degrees in control theory and control engineering from Northeastern University, P. R. China, in 1993, 1998, and 2000, respectively. From August 2006 to August 2007, she worked as a visiting scholar in the Department of Chemical Engineering, University of Texas, USA. Her research interest covers networked control systems, process monitoring, and fault tolerant control. E-mail: zhangyingwei@mail.neu.edu.cn

ZHOU Hong Received his B. S. degree in automation from Central South University in 2007. Now he is a master student in control theory and control engineering from Northeastern University. His research interest covers process monitoring and fault diagnosis. E-mail: zh0906@qq.com

QIN S. Joe Professor in the Mork Family Department of Chemical Engineering and Materials Science, Ming Hsieh Department of Electrical Engineering, Daniel J. Epstein Department of Industrial and Systems Engineering, University of Southern California, USA. He obtained his B. S. and M. S. degrees in automatic control from Tsinghua University, P. R. China, in 1984 and 1987, respectively. He received his Ph. D. degree in chemical engineering from University of Maryland in 1992. He worked as a principal engineer at Fisher-Rosemount from 1992 to 1995 and then joined University of Texas as a professor. His research interest covers system identification, process monitoring and fault diagnosis, model predictive control, run-to-run control, microelectronics process control, and control performance monitoring. Corresponding author of this paper. E-mail: sqin@usc.edu