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

ZHANG Ying-Wei1 ZHOU Hong1 QIN S. Joe2

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[13]. 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[14] developed auto-associative neural networks having five layers (input, mapping, bottleneck, damping, and output layers). Dong and McAvoy[15] 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[16,17] 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[18−22]. 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[10], but with NIPALS algorithm, the nonlinear principal components (PCs) are obtained one by one. The NIPALS algorithm is proposed in Table 1.

<table>
<thead>
<tr>
<th>Step</th>
<th>For comprehension</th>
<th>For computation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Scale $K$</td>
<td>Scale $K$</td>
</tr>
<tr>
<td>2</td>
<td>Initialize $t_i$</td>
<td>Initialize $t_i$</td>
</tr>
<tr>
<td>3</td>
<td>$p_i = \Phi_i^T t_i/|\Phi_i^T t_i|$</td>
<td>$t_i = K_i t_i/\sqrt{\phi_i}$</td>
</tr>
<tr>
<td>4</td>
<td>Loop until $t_i$ converges</td>
<td>Loop until $t_i$ converges</td>
</tr>
<tr>
<td>5</td>
<td>$\Phi_{i+1} = (I - t_i^2/|t_i^2|)\Phi_i$</td>
<td>$K_{i+1} = (I - t_i^2/|t_i^2|) K_i (I - t_i^2/|t_i^2|)$</td>
</tr>
<tr>
<td>6</td>
<td>Go to Step 3</td>
<td>Go to Step 3</td>
</tr>
</tbody>
</table>

In the first step of iteration, loadings vector $p_i$ is obtained as:

$$p_i = \Phi_i^T t_i$$

(1)

where $\Phi$ is the mapping of $x$ from the input space into the feature space and $t$ in the score vector. Then, $p_i$ is normalized to unit length by

$$p_i = \frac{p_i}{\|p_i\|} = \frac{p_i}{\sqrt{t_i^2 K_i t_i}}$$

(2)

In the second step of iteration, the scores vector is updated.
by
\[ t_i = \Phi_i p_i, \]  
(3)

By combining (2) and (3),
\[ t_i = \Phi_i p_i = \frac{\Phi_i \Phi_i^T t_i}{\| \Phi_i^T t_i \|} = \frac{K_i t_i}{\sqrt{\| t_i \|^2 \cdot K_i}}, \]  
(4)

After the convergence of \( t_i \), residual of \( \Phi_i \) is deflated by
\[ \Phi_{i+1} = \left( I - \frac{t_i t_i^T}{t_i^T t_i} \right) \Phi_i \]  
(5)

In the process of computation, \( K = \Phi \Phi^T \) converts (5) to the following:
\[ K_{i+1} = \left( I - \frac{t_i t_i^T}{t_i^T t_i} \right) K_i \left( I - \frac{t_i t_i^T}{t_i^T t_i} \right) \]  
(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 \( K_{i,j} = \exp(-\| x_i - x_j \|^2/c) \). Note that in 2-norm there exists the relation \( \| x_i - x_j \|^2 = \sum_{b=1}^{B} \| x_{b,i} - x_{b,j} \|^2 \). This can lead to the following property.

\[ K_{i,j} = \prod_{b=1}^{B} \exp\left(-\frac{\| x_{b,i} - x_{b,j} \|^2}{c} \right) = \prod_{b=1}^{B} K_{b,i,j}^b \]

Due to this property, dividing \( x \) into \( B \) blocks, one can compute the kernel matrix of each block \( K_b = \{ K_{b,i,j}^b \} \). The kernel matrix of each block should be centered by
\[ \bar{K}_b = K_b - \frac{1}{N_b} K_b - K_b 1_N^T + \frac{1}{N_b} K_b 1_N \]  
(7)

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

1.2.2 Algorithm derivation

MBKPCA first uses super scores \( t_{T,i} \) to compute block loadings \( p_{b,i} \)
\[ p_{b,i} = \Phi(X_{b,i})^T t_{T,i} \]  
(8)

Normalize the block loadings
\[ p_{b,i} = \frac{p_{b,i}}{\| p_{b,i} \|} = \frac{\Phi(X_{b,i})^T t_{T,i}}{\sqrt{t_{T,i}^T K_b t_{T,i}}} \]  
(9)

Block scores from block loadings are computed by
\[ t_{b,i} = \Phi(X_{b,i})p_{b,i} = \frac{K_{b,i} t_{T,i}}{\sqrt{t_{T,i}^T K_b t_{T,i}}} \]  
(10)

Arrange all block scores into a single matrix
\[ T_i = [t_{1,i}, \cdots, t_{b,i}] \]  
(11)

Super loadings are obtained by regressing \( T_i \) on \( t_{T,i} \) and then normalize it
\[ p_{T,i} = \frac{T_i^T t_{T,i}}{\| t_{T,i} \|^2} \]  
(12)

Update the super loadings \( t_{T,i} \) using
\[ t_{T,i} = \frac{T_i p_{T,i}}{\sqrt{T_i^T K_b t_{T,i}}} \]  
(13)

Repeat these steps until \( t_{T,i} \) converges to a predefined precision. The residual is used to compute the next PC. The residual is deflated using \( t_{T,i} \)
\[ K_{b,i+1} = \left( I - \frac{t_{T,i} t_{T,i}^T}{t_{T,i}^T t_{T,i}} \right) K_{b,i} \left( I - \frac{t_{T,i} t_{T,i}^T}{t_{T,i}^T t_{T,i}} \right) \]  
(14)

The MBKPCA algorithm is summarized as follows.

Algorithm for MBKPCA

Step 1. Scale each block data to 0 means;
Step 2. Initialize \( t_{T,i} \);
Step 3. For each block, compute \( t_{b,i} = K_b t_{T,i} / \sqrt{t_{T,i}^T K_b t_{T,i}} \);
Step 4. \( t_i = [t_{1,i}, \cdots, t_{b,i}] \);
Step 5. \( p_{T,i} = T_i^T t_{T,i} / \| t_{T,i} \|^2 \);
Step 6. \( t_{T,i} = T_i p_{T,i} \);
Step 7. If \( t_{T,i} \) is not converging, go to Step 3; else, go to Step 8;
Step 8. For each block, deflate residual \( K_{b,i+1} = (I - t_{T,i} t_{T,i}^T / (t_{T,i}^T t_{T,i})) K_{b,i} (I - t_{T,i} t_{T,i}^T / (t_{T,i}^T t_{T,i})) \);
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: \( \Phi(x_{new}) \) for \( b = 1, \cdots, B \). For each block, compute the kernel vector \( k_{b,new} = \Phi(x_{b,new}) \Phi(X_{b,new})^T \).

\[ k_{b,new} = k_{b,new} - 1_N K_b - 1_N k_{b,new} + 1_N K_b 1_N \]  
(15)

where \( 1_N = \frac{1}{N}[1 \cdots 1] \) in \( \mathbb{R}^{1 \times N} \). The block coefficient matrix \( A_b \) is denoted by
\[ A_{b,i} = \frac{t_{T,i}^T K_b t_{T,i}}{\sqrt{t_{T,i}^T K_b t_{T,i}}} \]  
(16)

as given in (10). It can be used to compute block scores of the new sample
\[ t_{b,new} = k_{b,new} A_b \]  
(17)

where \( t_{b,new} \) is a row vector, which is the score of one sample. For each column \( i \), use \( T_{new} = [t_{1,new} \cdots t_{b,new}] \) and super loadings to calculate the \( i \)-th super scores element of the sample
\[ t_{b,new}^T = T_{new} p_{T,i} \]  
(18)

The super scores \( t_{T,new} = [t_{1,new} \cdots t_{b,new}] \). The super \( T^2 \) statistic can be calculated by
\[ T_{new}^2 = t_{T,new} A_{new}^{-1} t_{T,new}^T \]  
(19)
where $\Lambda_b^{-1}$ is the inverse of the covariance matrix of block $b$. The statistic is computed by

$$T_{b, \text{new}}^2 = t_{b, \text{new}}\Lambda_b^{-1}t_{b, \text{new}}^T$$ (20)

where $\Lambda_b^{-1}$ is the inverse of the covariance matrix of block $b$. The block SPE statistic is defined as

$$SPE_b = \hat{\Phi}(x_{b, \text{new}})(I - PP^T)\hat{\Phi}^T(x_{b, \text{new}}) = k(x_{b, \text{new}}, x_{b, \text{new}}) - k_b^{\text{new}}\Lambda_b A_k^{\text{new}}k_b, \text{new}$$ (21)

and

$$k(x_{b, \text{new}}, x_{b, \text{new}}) = k(x_{b, \text{new}}, x_{b, \text{new}}) -$$

$$\frac{2}{N} \sum_{i=1}^{N} k(x_{b, i}, x_{b, \text{new}}) + \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} k(x_{b, i}, x_{b, j})$$ (22)

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

$$SPE = \sum_{b=1}^{B} SPE_b$$ (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 $x_5 = \sin x_3^2$. Noise 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: $X_1 = [x_1, x_2, x_3]$ and $X_2 = [x_4, x_5]$. MBKPCA can detect the fault using the SPE chart, as shown in Fig. 2.

![Fig. 1 Normal and faulty data](image)

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.

#### 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 100-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.
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 $\text{SPE}$ plots indicate that Block 3 is affected by the fault 4; the contribution plot of block $\text{SPE}$ 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 $\text{SPE}$ 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


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: zhangyingwei@neu.edu.cn

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: sinq@usc.edu