AIChE

Improved Nonlinear Fault Detection Technique and Statistical Analysis

Yingwei Zhang
Key Laboratory of Integrated Automation of Process Industry, Ministry of Education, Northeastern University, Shenyang, Liaoning 110004, P. R. China

S. Joe Qin
The Mork Family, Dept. 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, 925 Bloom Walk, HED 210, Los Angeles, CA 90089

DOI 10.1002/aic.11617
Published online October 24, 2008 in Wiley InterScience (www.interscience.wiley.com).

In this article, first, some drawbacks of original Kernel Principal Component Analysis (KPCA) and Kernel Independent Component Analysis (KICA) are analyzed. Then the KPCA and KICA for multivariate statistical process monitoring (MSPM) are improved. The drawbacks of original KPCA and KICA are as follows: The data mapped into feature space become redundant; linear data introduce errors while the kernel trick is used; computation time increases with the number of samples. To solve the above problems, the original KPCA and KICA for MSPM are improved: similarity factors of the observed data in the input and feature space are defined; similar characteristics are measured; similar data are removed according to the similarity measurements; and k-means clustering in feature space is used to isolate different classes. Specifically, the similarity concept of data in one group is first proposed. Applications of the proposed approach indicate that improved KPCA and KICA effectively capture the nonlinearities.
© 2008 American Institute of Chemical Engineers *AIChE J,* 54: 3207–3220, 2008
Keywords: Kernel principal component analysis (KPCA), Kernel independent component analysis (KICA), nonlinear process monitoring, fault detection

Introduction

Application of linear monitoring approaches to nonlinear processes may lead to unreliable process monitoring, because a linear method is inappropriate to extract the nonlinearities within the process variables.[1–5] To solve the nonlinear problem of observer data, some nonlinear PCA approaches have been developed.[6–16] Hiden proposed nonlinear PCA methods based on genetic programming and input-training neural networks.[6] Chen proposed nonlinear monitoring scheme by combining PCA and neural networks.[7] Dong and McAvoy proposed a nonlinear PCA based on principal curves and neural networks and applied it to nonlinear process monitoring.[8] They used a neural network model to map the original data into the corresponding scores and to map these scores into the original variables. Jia et al. proposed an approach that combined linear PCA and input-training neural network in order to consider linear and nonlinear data correlations separately.[9] Kramer proposed a nonlinear PCA based on an autoassociative neural network consisting of five layers, specifically the input, mapping, bottleneck, demapping, and output layers.[10] In this formalism, the nonlinear extension of PCA was achieved by inserting nonlinear functions into the nodes of the mapping and demapping layers. Tan and Mavrovouniotis proposed a nonlinear PCA scheme based on an input-training neural network.[11] NLPCA also are based on

Correspondence concerning this article should be addressed to Y. Zhang at zhangyingwei@ise.neu.edu.cn (or) S. J. Qin at sqin@usc.edu.

© 2008 American Institute of Chemical Engineers

the neural networks.[12] Most existing nonlinear PCA approaches are based on neural networks, thus, it is necessary to train offline and a nonlinear optimization problem has to be solved to compute the principal components and the number of principal components must be specified in advance before training the neural networks.[13] Artificial neural networks (ANNs) have poor process interpretability and are hindered by problems that are associated with weight optimization, such as slow learning and local minimization.[14] As an alternative learning strategy, Schölkopf et al. adopted support vector machine (SVM) learning.[15] However, SVM learning is supervised. Lee and coworkers used multilayer perceptrons (MLPs) to show a global approximation of a nonlinear mapping, whereas radial basis function networks (RBFNs) construct a local approximation using radial basis functions that are exponentially decaying nonlinear functions.[16]

Recently, kernel principal component analysis (KPCA) first presented by Schölkopf et al. is a new nonlinear extension of PCA.[17] KPCA computes the principal components in a high dimensional feature space, which is nonlinearly related to the input space. Lee and coworkers proposed a new nonlinear process monitoring technique using kernel PCA (KPCA) to monitor continuous process and demonstrated its superiority to the PCA monitoring method.[18] KPCA is to first map the input space into a feature space via a nonlinear map, and then to extract the principal components in the feature space.[18–21] KPCA has the main advantages:

(1) It can avoid both performing the nonlinear mappings and computing inner products in the feature space by introducing a kernel function.

(2) KPCA and SVM utilize Mercer kernels to generalize a linear algorithm to a nonlinear setting; moreover, both use the same regularizer, however, they are in different domains of learning: SVM is supervised and KPCA is unsupervised

(3) For the RBF kernels, $k(\boldsymbol{x},\boldsymbol{x})$ takes the same constant value for all $\boldsymbol{x}$. Therefore, in feature space, all $\boldsymbol{\Phi}(\boldsymbol{x})$ lie on a hyper-dimensional sphere. Thus, it is not necessary to scale the standard deviation.

More recently, several multivariate statistical process monitoring (MSPM) methods based on ICA have been proposed.[3,22–28] This ICA-based monitoring method has been extended to dynamic process monitoring and batch process monitoring, respectively.[25,26] The goal of ICA is to decompose observed data into linear combinations of statistically independent components (ICs).[27,28] ICA involves higher-order statistics, that is, not only does it decorrelate the data based on second order statistics but also reduces higher order statistical dependencies.[29] Kernel Independent Component Analysis (KICA) performs two phases: whitened KPCA and ICA iteration in the KPCA transformed space. When compared with other nonlinear approaches, KICA combine the advantages of KPCA and ICA to develop a nonlinear dynamic approach to detect fault online.[30] KICA can also be applied to batch processes.[31] However, KPCA and KICA have disadvantages as follows:

(1) The data mapped into feature space become redundant.

(2) Linear data introduce errors while the kernel trick is used.

(3) The size of the kernel matrix is the square of the number of samples. The computation time may increase with the number of samples.

Up to now, the above mentioned disadvantages have never been investigated very well.

We improve KPCA and KICA for fault detection in this article. After similarity analysis, k mean cluster is used to reduce the computational load when the number of training samples is large. The similarity analysis approach of the same variables in two different groups or more than two groups has been presented based on PCA.[32] The similarity concept of data in one group is first proposed in this article.

The remainder of the article is organized as follows. The KICA algorithm is introduced, in the "Original KICA" section. Then the similarity concept of data in one group is proposed in the "Similarity analysis" section. In the "k-means clustering in feature space" section, k mean cluster in the feature space is proposed. The fault detection strategies using improved KICA are introduced in the "Fault detection based on improved KPCA and KICA" section. The illustrative example is given to demonstrate the effectiveness of the proposed methods in the "Simulation result" section. Finally, conclusions are drawn.

Original KICA

Whitened KPCA

The goal of this step is to map the input space into a feature space via nonlinear mapping and then to extract whitened principal components in that feature space such that their covariance structure is an identity matrix. Consider the observed data $\mathbf{x}_k \in R^m$, $k = 1,\ldots,N$ where N is the number of observations. Using the nonlinear mapping $\boldsymbol{\Phi}:R^m \rightarrow F$. Then, the covariance matrix in feature space will be

$$\mathbf{C}^F = \frac{1}{N}\sum_{j=1}^{N} \boldsymbol{\Phi}(\mathbf{x}_j)\boldsymbol{\Phi}(\mathbf{x}_j)^T \tag{1}$$

where $\boldsymbol{\Phi}(\mathbf{x}_k)$ for $k = 1,\ldots,N$ is assumed to be zero-mean and unit variance. Let $\boldsymbol{\Theta} = [\boldsymbol{\Phi}(\mathbf{x}_1), \ldots, \boldsymbol{\Phi}(\mathbf{x}_N)]$, then $\boldsymbol{C}^F$ can be expressed by $\boldsymbol{C}^F = \frac{1}{N}\boldsymbol{\Theta}\boldsymbol{\Theta}^T$. Instead of Eigen-decomposing $\boldsymbol{C}^F$ directly, we can alternatively find principal components using the "Kernel tricks." The Kernel trick is for algorithms in which the data points occur only within dot products, and the dot product can be replaced with a Kernel function. The Kernel trick allow us to compute the value of the dot product in $\boldsymbol{F}$ without having to carry out the map $\boldsymbol{\Phi}$. The kernel function k determines directly the mapping $\boldsymbol{\Phi}$ and the feature space $\boldsymbol{F}$. We do not need to compute the dot products explicitly between vectors in $\boldsymbol{F}$, as we know that this can be done directly in the input space, using Kernel functions. Defining an $N \times N$ Gram Kernel matrix $\boldsymbol{K}$ by

$$[\boldsymbol{K}]_{ij} = K_{ij} = \langle \boldsymbol{\Phi}(\mathbf{x}_i), \boldsymbol{\Phi}(\mathbf{x}_j) \rangle = k(\mathbf{x}_i, \mathbf{x}_j), \tag{2}$$

we have $\boldsymbol{K} = \boldsymbol{\Theta}^T\boldsymbol{\Theta}$. The mean centered Kernel matrix $\tilde{\boldsymbol{K}}$ can be easily obtained from

$$\tilde{\boldsymbol{K}} = \boldsymbol{K} - 1_N\boldsymbol{K} - \boldsymbol{K}1_N + 1_N\boldsymbol{K}1_N \tag{3}$$

where $1_N = \frac{1}{N}\begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix} \in R^{N\times N}$.

If we apply Eigenvalue decomposition to $\tilde{\boldsymbol{K}}$

$$\lambda\boldsymbol{\alpha} = \tilde{\boldsymbol{K}}\boldsymbol{\alpha}, \tag{4}$$

we can obtain the d largest positive Eigenvalues of $\boldsymbol{C}^F$ are $\frac{\lambda_1}{N}, \frac{\lambda_2}{N}, ..., \frac{\lambda_d}{N}$, and the associated eigenvectors $\boldsymbol{v}_1,\boldsymbol{v}_2,\ldots,\boldsymbol{v}_d$ can be expressed as

$$\boldsymbol{v}_j = \frac{1}{\sqrt{\lambda_j}}\boldsymbol{\Theta}\boldsymbol{\alpha}_j \quad j = 1, ..., d \tag{5}$$

The eigenvector matrix $\boldsymbol{V} = [\boldsymbol{v}_1,\boldsymbol{v}_2,\ldots,\boldsymbol{v}_d]$ can be briefly expressed by the following matrix

$$\boldsymbol{V} = \boldsymbol{\Theta}\boldsymbol{H}\boldsymbol{\Lambda}^{-1/2} \tag{6}$$

where $\boldsymbol{H} = [\alpha_1,\alpha_2,\ldots,\alpha_d]$ and $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1,\lambda_2,\ldots,\lambda_d)$. The whitening matrix $\boldsymbol{P}$ and the mapped data in the feature space can be whitened by the following transformation:

$$\mathbf{z} = \mathbf{P}^T\boldsymbol{\Phi}(\mathbf{x}) \tag{7}$$

In detail,

$$\begin{aligned}\mathbf{z} = \mathbf{P}^T\boldsymbol{\Phi}(\mathbf{x}) &= \sqrt{N}\boldsymbol{\Lambda}^{-1}\mathbf{H}^T\boldsymbol{\Theta}^T\boldsymbol{\Phi}(\mathbf{x}) \\ &= \sqrt{N}\boldsymbol{\Lambda}^{-1}\mathbf{H}^T[\boldsymbol{\Phi}(\mathbf{x}_1),\ldots,\boldsymbol{\Phi}(\mathbf{x}_N)]^T\boldsymbol{\Phi}(\mathbf{x}) \\ &= \sqrt{N}\boldsymbol{\Lambda}^{-1}\mathbf{H}^T\left[\tilde{k}_{\mathrm{scl}}(\mathbf{x}_1,\mathbf{x}),\ldots,\tilde{k}_{\mathrm{scl}}(\mathbf{x}_N,\mathbf{x})\right]^T\end{aligned} \tag{8}$$

where $\mathbf{z}$ is the same as the whitened KPCA score vector satisfying $E\{\mathbf{z}\mathbf{z}^T\} = \mathbf{I}$.

ICA

The following task is to find the mixing matrix in the KPCA transformed space to recover independent components. ICA goes one step further so that it transforms the whitened data into a set of statistically independent components. In this section, a ICA algorithm is introduced.[29,30]

The modified ICA finds m ($\leq d$) dominant source signals or ICs satisfying $E(\mathbf{y}\mathbf{y}^T) = \mathbf{D} = \mathrm{diag}\{\lambda_1,\ldots,\lambda_m\}$ such that the elements of $\mathbf{y}$ become as independent of each other as possible, using

$$\mathbf{y} = \mathbf{C}^T\mathbf{z} \tag{9}$$

where $\mathbf{C} \in R^{d\times m}$ and $\mathbf{C}^T\mathbf{C} = \mathbf{D}$. The requirement $E(\mathbf{y}\mathbf{y}^T) = \mathbf{D}$ reflects that the variance of each element of $\mathbf{y}$ is the same as that of scores in PCA, and hence, we can order the ICs according to their variances.

Defining the normalized ICs as:

$$\mathbf{y}_n = \mathbf{D}^{-1/2}\mathbf{y} = \mathbf{D}^{-1/2}\mathbf{C}^T\mathbf{z} = \mathbf{C}_n^T\mathbf{z} \tag{10}$$

It is clear that $\mathbf{D}^{-1/2}\mathbf{C}^T = \mathbf{C}_n^T$, $\mathbf{C}_n^T\mathbf{C}_n = \mathbf{I}$, and $E(\mathbf{y}_n\mathbf{y}_n^T) = \mathbf{I}$. We can set the first m components of $\mathbf{z}$ to be the initial components of $\mathbf{y}_n$. Let

$$\mathbf{C}_n^T = [\mathbf{I}_m\vdots\mathbf{0}] \tag{11}$$

where $\mathbf{I}_m$ is the m-dimensional identity matrix and $\mathbf{0}$ is $m \times (d-m)$ zero matrix. Denoting $\mathbf{c}_{n,i}$ as the i-th column of $\mathbf{C}_n$, the modified ICA algorithm obtains the i-th independent component $y_{n,i} = (\mathbf{c}_{n,i})^T\mathbf{z}$ with maximized non-Gaussianity or negentropy.

Once $\mathbf{C}_n$ is found in FastICA, then we can obtain some dominant independent components from

$$\mathbf{y} = \mathbf{D}^{1/2}\mathbf{C}_n^T\mathbf{z} \tag{12}$$

Similarity Analysis

Kernel theory has found increasing numbers of applications in the nonlinear chemical processes. The Kernel functions show superiority of nonlinearization for observed data in the chemical industry. The data similarity analysis before modeling is needed because of the following reasons:

(1) The data mapped into feature space become linearly redundant.

(2) If the data are linear, then errors will be introduced while the Kernel trick is used to nonlinearize.

(3) In the training process of KPCA, the size of the Kernel matrix is the square of the number of samples. When the sample number becomes large, the calculation of Eigenvalues and Eigenvectors will be time consuming.

In this article, the similarity factors of the observed data are defined and similar characteristics are measured in the input and feature space. Some similar data are removed according to the similarity factors. The observed data are analyzed by exploring the linear dependency of samples in input space and feature space, respectively. The similarity concept is introduced to define similarity factor between new data and the note set.

Similarity analysis in input space

The note set at time i is denoted as $N_i = \{\tilde{\mathbf{x}}_1,\ldots,\tilde{\mathbf{x}}_{n_i}\}$, $i = 1,\ldots,t_1$, and satisfying $n_i < t_1$. The subspace spanned by the vectors in set N_i is denoted as F_i. In the beginning of learning, there is only two data point that is $N_1 = \{\tilde{\mathbf{x}}_1,\tilde{\mathbf{x}}_2\} = \{\mathbf{x}_1,\mathbf{x}_2\}$. The new data point will be added into the set according to their dependencies on the space F_i using the following proposed similarity factor:

$$S_e = \left(\frac{\mathrm{cov}(\mathbf{x}_{t_1} - \tilde{\mathbf{x}}_{k_1}, \mathbf{x}_{t_1} - \tilde{\mathbf{x}}_{l_1})}{\mathrm{std}(\mathbf{x}_{t_1} - \tilde{\mathbf{x}}_{k_1})\mathrm{std}(\mathbf{x}_{t_1} - \tilde{\mathbf{x}}_{l_1})}\right)^2 \tag{13}$$

In fact, if three data points $\mathbf{x}_{t_1}$, $\tilde{\mathbf{x}}_{k_1}$, and $\tilde{\mathbf{x}}_{l_1}$ satisfy

$$\cos\theta = \left(\frac{\mathrm{cov}(\mathbf{x}_{t_1} - \tilde{\mathbf{x}}_{k_1}, \mathbf{x}_{t_1} - \tilde{\mathbf{x}}_{l_1})}{\mathrm{std}(\mathbf{x}_{t_1} - \tilde{\mathbf{x}}_{k_1})\mathrm{std}(\mathbf{x}_{t_1} - \tilde{\mathbf{x}}_{l_1})}\right) = 1$$

Then $\mathbf{x}_{t_1}$, $\tilde{\mathbf{x}}_{k_1}$, and $\tilde{\mathbf{x}}_{l_1}$ are linear. $0 \leq S_e \leq 1$

The similarity is strongest when $S_e = 1$. Hence, if $S_e < \gamma_0$, where γ_0 is a predefined small value satisfying $0 \leq \gamma_0 \leq 1$, the new data $\tilde{\mathbf{x}}_{t_1}$ will be introduced to constitute the expanded node set; otherwise, this data point will be rejected without the node set expansion, that is $N_{t_1} = N_{t_1} - 1$. After the similarity analysis in the input space, we obtain first data subset Ω_1.

Similarity analysis in feature space

For the first data subset Ω_1, let $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ are the three data points in the input space. Mapping the three data points

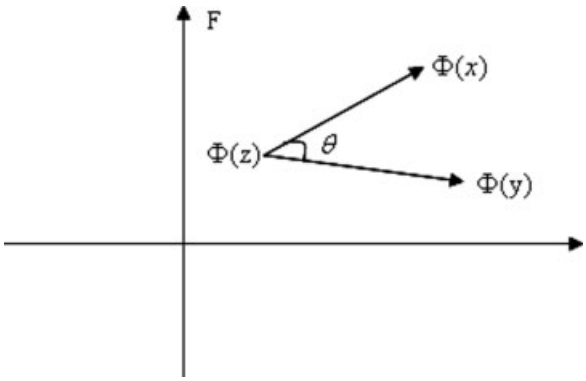

Figure 1. Angle of three data points in feature space.

into the feature space, they become $\mathbf{\Phi}(\mathbf{x})$, $\mathbf{\Phi}(\mathbf{y})$, and $\mathbf{\Phi}(\mathbf{z})$, shown in the Figure 1.

If

$$\cos\theta = \frac{(\mathbf{\Phi}(\mathbf{z}) - \mathbf{\Phi}(\mathbf{x}), \mathbf{\Phi}(\mathbf{z}) - \mathbf{\Phi}(\mathbf{y}))}{\|\mathbf{\Phi}(\mathbf{z}) - \mathbf{\Phi}(\mathbf{x})\|\|\mathbf{\Phi}(\mathbf{z}) - \mathbf{\Phi}(\mathbf{y})\|} = 1 \tag{14}$$

then

$$\theta = 0$$

that is, $\mathbf{\Phi}(\mathbf{x})$, $\mathbf{\Phi}(\mathbf{y})$, and $\mathbf{\Phi}(\mathbf{z})$ in feature space are linear.

The note set at time j in feature space is denoted as $L_j = \{\mathbf{\Phi}(\tilde{\mathbf{x}}_1), \ldots, \mathbf{\Phi}(\tilde{\mathbf{x}}_{n_j})\}$, $j = 1,\ldots,t_2$, and satisfying $n_j < t_2$. The subspace spanned by the vectors in set L_j is denoted as $\mathbf{E}_j$. In the beginning of learning, there is only two data point that is $N_1 = \{\mathbf{\Phi}(\tilde{\mathbf{x}}_1), \mathbf{\Phi}(\tilde{\mathbf{x}}_2)\} = \{\mathbf{\Phi}(\mathbf{x}_1), \mathbf{\Phi}(\mathbf{x}_2)\}$. The new data point will be added into the set according to their dependencies on the space $\mathbf{E}_j$ using the following proposed similarity factor:

$$S_f = \left(\frac{\mathrm{cov}\left(\mathbf{\Phi}(\mathbf{x}_{t_2}) - \tilde{\mathbf{\Phi}}(\mathbf{x}_{k_2}), \mathbf{\Phi}(\mathbf{x}) - \tilde{\mathbf{\Phi}}(\mathbf{x}_{l_2})\right)}{\mathrm{std}\left(\mathbf{\Phi}(\mathbf{x}_{t_2}) - \tilde{\mathbf{\Phi}}(\mathbf{x}_{k_2})\right)\mathrm{std}\left(\mathbf{\Phi}(\mathbf{x}_{t_2}) - \tilde{\mathbf{\Phi}}(\mathbf{x}_{l_2})\right)}\right)^2 \tag{15}$$

Thus, $0 \le S_f \le 1$. The similarity is strongest when $S_f = 1$. Hence, if $S_f < \gamma_1$, where γ_1 is a predefined small value satisfying $0 \le \gamma_1 \le 1$, the new data $\tilde{\mathbf{x}}_t$ will be introduced to constitute the expanded node set; otherwise, this data point will be rejected without the node set expansion, that is $L_{t_2} = L_{t_2-1}$.

The simply alternative similarity factors of S_f can be determined in the following process:

Eq. 14 is equivalent to Eq. 15

$$\frac{f(\mathbf{x},\mathbf{y},\mathbf{z})}{g(\mathbf{x},\mathbf{y},\mathbf{z})} = 1 \tag{16}$$

where

$$f(\mathbf{x},\mathbf{y},\mathbf{z}) = (\mathbf{\Phi}(\mathbf{x}), \mathbf{\Phi}(\mathbf{y})) - (\mathbf{\Phi}(\mathbf{y}), \mathbf{\Phi}(\mathbf{z})) - (\mathbf{\Phi}(\mathbf{x}), \mathbf{\Phi}(\mathbf{z})) + (\mathbf{\Phi}(\mathbf{z}), \mathbf{\Phi}(\mathbf{z}))$$

$$g(\mathbf{x},\mathbf{y},\mathbf{z}) = \Big((\mathbf{\Phi}(\mathbf{x}), \mathbf{\Phi}(\mathbf{x})) - 2(\mathbf{\Phi}(\mathbf{x}), \mathbf{\Phi}(\mathbf{z})) + (\mathbf{\Phi}(\mathbf{z}), \mathbf{\Phi}(\mathbf{z}))\Big)^{1/2} \times \Big((\mathbf{\Phi}(\mathbf{y}), \mathbf{\Phi}(\mathbf{y})) - 2(\mathbf{\Phi}(\mathbf{y}), \mathbf{\Phi}(\mathbf{z})) + (\mathbf{\Phi}(\mathbf{z}), \mathbf{\Phi}(\mathbf{z}))\Big)^{1/2} \tag{17}$$

using the Kernel trick, Eq. 15 becomes as follows

$$\frac{k(\mathbf{x},\mathbf{y}) - k(\mathbf{y},\mathbf{z}) - k(\mathbf{x},\mathbf{z}) + k(\mathbf{z},\mathbf{z})}{(k(\mathbf{x},\mathbf{x})) - 2k(\mathbf{x},\mathbf{z}) + k(\mathbf{z},\mathbf{z}))^{1/2}(k(\mathbf{y},\mathbf{y}) - 2k(\mathbf{y},\mathbf{z}) + k(\mathbf{z},\mathbf{z}))^{1/2}} = 1 \tag{18}$$

after using Kernel function $k(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{c}\right)$, Eq. 18 becomes

$$\frac{k(\mathbf{x},\mathbf{y}) - k(\mathbf{y},\mathbf{z}) - k(\mathbf{x},\mathbf{z}) + 1}{(1 - k(\mathbf{x},\mathbf{z}))^{1/2}(1 - k(\mathbf{y},\mathbf{z}))^{1/2}} = 2 \tag{19}$$

Equation 19 is equivalent to Eq. 20

$$\frac{k(\mathbf{x},\mathbf{y}) - k(\mathbf{y},\mathbf{z}) - k(\mathbf{x},\mathbf{z}) + 1}{(1 - k(\mathbf{x},\mathbf{z}) - k(\mathbf{y},\mathbf{z}) + k(\mathbf{x},\mathbf{z})k(\mathbf{y},\mathbf{z}))^{1/2}} = 2 \tag{20}$$

Let

$$h = 1 - k(\mathbf{y},\mathbf{z}) - k(\mathbf{x},\mathbf{z}) \tag{21}$$

Then from Eq. 20

$$\frac{h + k(\mathbf{x},\mathbf{y})}{(h + k(\mathbf{x},\mathbf{z})k(\mathbf{y},\mathbf{z}))^{1/2}} = 2 \tag{22}$$

So if the data points $\mathbf{x}$,$\mathbf{y}$,$\mathbf{z}$ satisfy

$$k(\mathbf{x},\mathbf{y}) = 2(h + k(\mathbf{x},\mathbf{z})k(\mathbf{y},\mathbf{z}))^{1/2} - h \tag{23}$$

then $\theta = 0$, that is, the data points $\mathbf{x}$,$\mathbf{y}$,$\mathbf{z}$ are linear in the feature space.

And we know that if

$$\begin{aligned} k(\mathbf{x},\mathbf{z}) &= k(\mathbf{x},\mathbf{y}) \\ k(\mathbf{y},\mathbf{z}) &= 4k(\mathbf{x},\mathbf{y}) - 3 \end{aligned} \tag{24}$$

then Eq. 23 is satisfied.

Hence, the three data points $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ are linear in the feature space when they satisfy Eq. 24. From Eq. 24, the alternative similarity factors in feature space can be defined as follows:

$$S_{f1} = 1 - \frac{\left|k(\mathbf{x}_{j+1},\mathbf{x}_{k_2}) - k(\mathbf{x}_{k_1},\mathbf{x}_{l_2})\right|}{k(\mathbf{x}_{j+1},\mathbf{x}_{k_2})} \tag{25}$$

and

$$S_{f2} = 1 - \frac{\left|k(\mathbf{x}_{j+1},\mathbf{x}_{l_2}) - 4k(\mathbf{x}_{k_2},\mathbf{x}_{l_2}) + 3\right|}{k(\mathbf{x}_{j+1},\mathbf{x}_{l_2})} \tag{26}$$

the similarity is strongest when $S_{f1} = 1$ and $S_{f2} = 1$.

After the similarity analysis in the feature space, the number the retained data point is denoted as n. And we obtain the second data subset Ω_2.

k-Means Clustering in Feature Space

Although there is a large amount of literature available concerning clustering methodologies and applications, only a few applications have been reported that cluster time-series

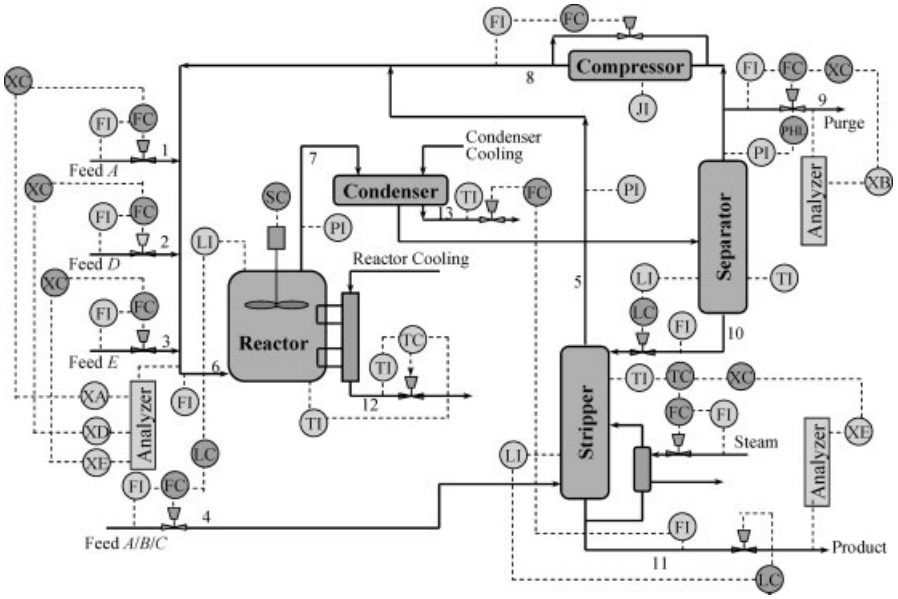

Figure 2. Process layout of the Tennessee Eastman process.

data. And the papers on clustering time-series data in feature space are much less.

In this work, *k*-means clustering in feature space is used to isolate different classes of time-series data. When compared with clustering methods in input space, *k*-means clustering in feature space is more effective for clustering large amounts of time-series data. The idea behind *k*-means clustering is to select the center data representing the characteristics in every cluster.

Suppose $\boldsymbol{x} \in \mathrm{R}^{m\times n}$, a set of m samples in n-dimensional input space R^n, and an integer k, and the problem is to determine a set of k centroids $\boldsymbol{\Phi}(\mu_{1,t_1})$, $\boldsymbol{\Phi}(\mu_{2,t_2})$, …, $\boldsymbol{\Phi}(\mu_{k,t_k})$ in F, so as to minimize the sum-of-squares criterion

$$J = \sum_{c=1}^{k}\sum_{j=1}^{n} \left\|\boldsymbol{\Phi}(\boldsymbol{x}_{j,t_j}) - \boldsymbol{\Phi}(\mu_{c,t_c})\right\|^2 \tag{27}$$

where

$$\begin{aligned}\left\|\boldsymbol{\Phi}(\boldsymbol{x}_{j,t_j}) - \boldsymbol{\Phi}(\mu_{c,t_c})\right\|^2 &= \left(\boldsymbol{\Phi}(\boldsymbol{x}_{j,t_j}) - \boldsymbol{\Phi}(\mu_{c,t_c}), \boldsymbol{\Phi}(\boldsymbol{x}_{j,t_j}) - \boldsymbol{\Phi}(\mu_{c,t_c})\right)\\ &= \left(\boldsymbol{\Phi}(\boldsymbol{x}_{j,t_j}), \boldsymbol{\Phi}(\boldsymbol{x}_{j,t_j})\right) - 2\left(\boldsymbol{\Phi}(\boldsymbol{x}_{j,t_j}), \boldsymbol{\Phi}(\mu_{c,t_c})\right) + \left(\boldsymbol{\Phi}(\mu_{c,t_c}), \boldsymbol{\Phi}(\mu_{c,t_c})\right)\\ &= k(\boldsymbol{x}_{j,t_j}, \boldsymbol{x}_{j,t_j}) - 2k(\boldsymbol{x}_{j,t_j}, \mu_{c,t_c})) + k(\mu_{c,t_c}, \mu_{c,t_c})\end{aligned} \tag{28}$$

A general algorithm is:

(1) Randomly pick k samples in the data set as the initial cluster centroids $\boldsymbol{\Phi}(\mu^0_{1,t_1})$, $\boldsymbol{\Phi}(\mu^0_{2,t_2})$, …, $\boldsymbol{\Phi}(\mu^0_{k,t_k})$ in F. Set iteration $i = 0$.

(2) Assign each sample $\boldsymbol{\Phi}(\mathbf{x}_{j,t_j})$ to the cluster with the nearest centroid $\boldsymbol{\Phi}(\mu^{(i)}_{c,t_c})$ according to the Eq. 28.

(3) When all samples have been assigned, recalculate the positions of the k centroids in F, that is, compute $E\{\boldsymbol{\Phi}(\boldsymbol{x}_{j,t_j})\} = \frac{1}{n_i}\sum_{j=1}^{n_i}\boldsymbol{\Phi}(\boldsymbol{x}_{j,t_j})$ with n_i is the number of data points in ith cluster. Thus, the distance from the data point to the centroid is

$$\begin{aligned}\left\|\boldsymbol{\Phi}(\boldsymbol{x}_{m,t_m}) - \frac{1}{n_i}\sum_{j=1}^{n_j}\boldsymbol{\Phi}(\boldsymbol{x}_{j,t_j})\right\| &= k(\boldsymbol{x}_{m,t_m}, \boldsymbol{x}_{m,t_m})\\ &\quad - \frac{2}{n_i}\sum_{j=1}^{n_i} k(\boldsymbol{x}_{m,t_m}, \boldsymbol{x}_{j,t_j}) + \frac{1}{n_i^2}\sum_{i=1}^{n_i}\sum_{j=1}^{n_j} k(\boldsymbol{x}_{i,t_i}, \boldsymbol{x}_{j,t_j})\end{aligned} \tag{29}$$

(4) Repeat Steps (2) and (3) until the centroids no longer move.

In the above step (3), it is difficult to compute the centroids of cluster in the feature space, so any data point in the cluster is

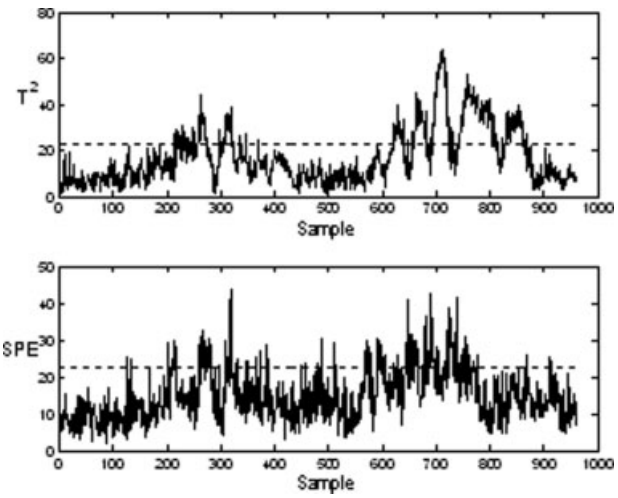

Figure 3. PCA monitoring results of the Tennessee Eastman process in the case of Fault 10.

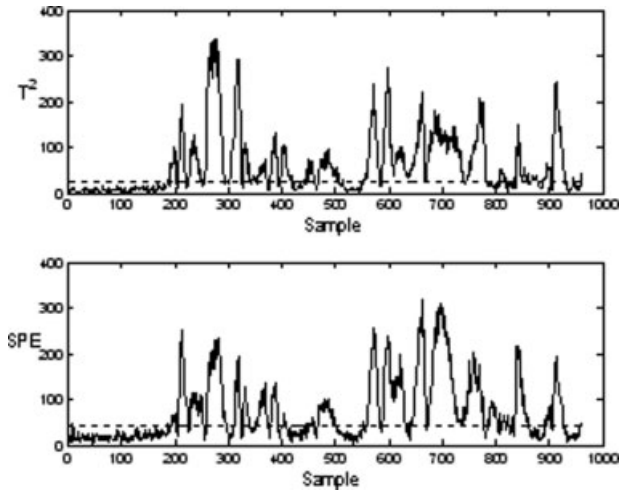

Figure 4. ICA monitoring results of the Tennessee Eastman process in the case of Fault 10.

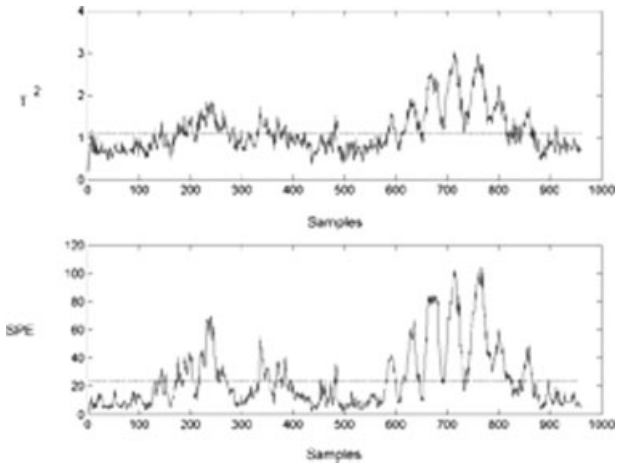

Figure 6. Monitoring results of the Tennessee Eastman process based on improved KICA (γ_0,γ_1 = 0.05 and n = 320) in the case of Fault 10.

chosen to be a centroids. In every cluster, there are n_i data points. For every point, compute the distance from other points to the centroid and compute the sum of all the distances. If the sum is minimum, then the data point is the centroid.

Fault Detection Based on Improved KPCA and KICA

Fault detection based on improved KPCA

In this section, KPCA is available for fault detection.[18] Here, Hotelling's T^2 statistic and squared prediction error (SPE) statistic are used for the control chart.

$$\boldsymbol{T}^2 = \boldsymbol{t}^T \boldsymbol{\Lambda}^{-1} \boldsymbol{t} = \left[\boldsymbol{t}_1, \boldsymbol{t}_2, ..., \boldsymbol{t}_p\right] \boldsymbol{\Lambda}^{-1} \left[\boldsymbol{t}_1, \boldsymbol{t}_2, ..., \boldsymbol{t}_p\right]^T \tag{30}$$

where $\mathbf{t}_k := \left(\mathbf{v}^k, \boldsymbol{\Phi}(\mathbf{x})\right) = \sum_{i=1}^{n} \alpha_i^k k(\mathbf{x}, \mathbf{x}_i)$ and $\boldsymbol{\Lambda}^{-1}$ are the diagonal matrix of the inverse of the Eigenvalues associated

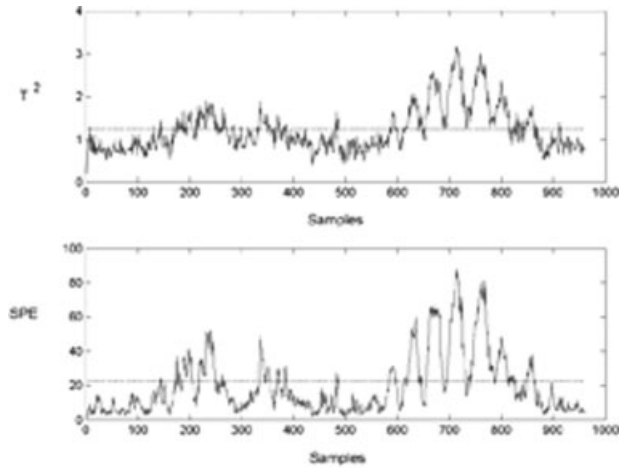

Figure 5. Monitoring results of the Tennessee Eastman process based KICA in the case of Fault 10.

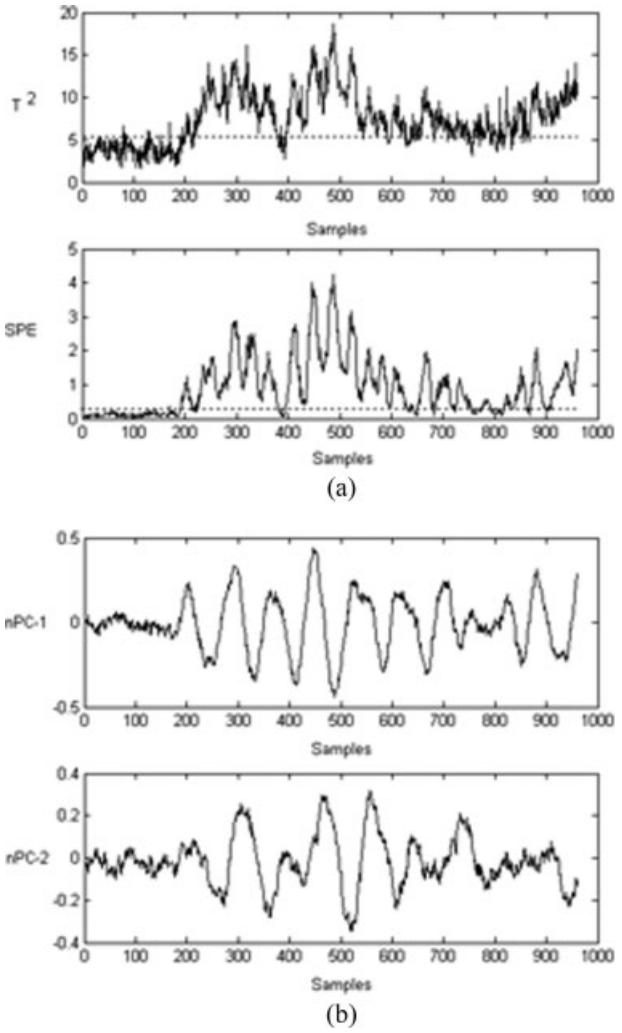

Figure 7. Monitoring results of the Tennessee Eastman process based on (a) KPCA, (b) Two nPCs in the case of Fault 8.

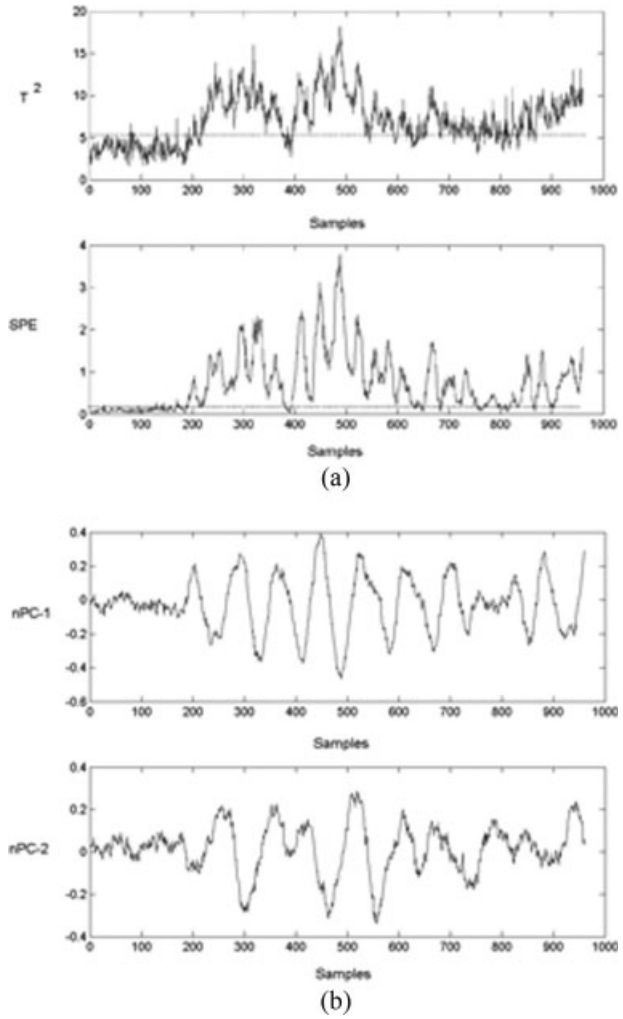

Figure 8. Monitoring results of the Tennessee Eastman process based on (a) Improved KPCA (b) Two nPCs (γ_0,γ_1 = 0.05 and n = 320) in the case of Fault 8.

with the retained principal components. The $100\alpha\%$ control limit for T^2 is obtained using the F-distribution

$$T^2_{\lim} = \frac{p(n-1)}{n-p} F(p, n-1, \alpha) \tag{31}$$

where $F(p,n-1,\alpha)$ is the upper $100\alpha\%$ critical point of the F-distribution with p and $n-p$ degrees of freedom.[1]

$$\mathrm{SPE} = \tilde{K}(x,x) - t^T t \tag{32}$$

The $100(1-\alpha)\%$ control limit for SPE is

$$\mathrm{SPE}_{\lim} = \theta_1 \left(\frac{c_\alpha \sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right)^{1/h_0} \tag{33}$$

where

$$\theta_i = \sum_{j=i+1}^{m} \lambda_j^i, \quad i = 1, 2, 3 \tag{34}$$

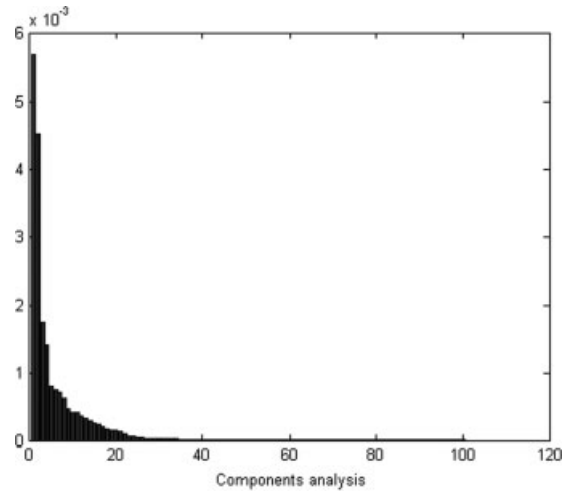

Figure 9. Components analysis in the feature space.

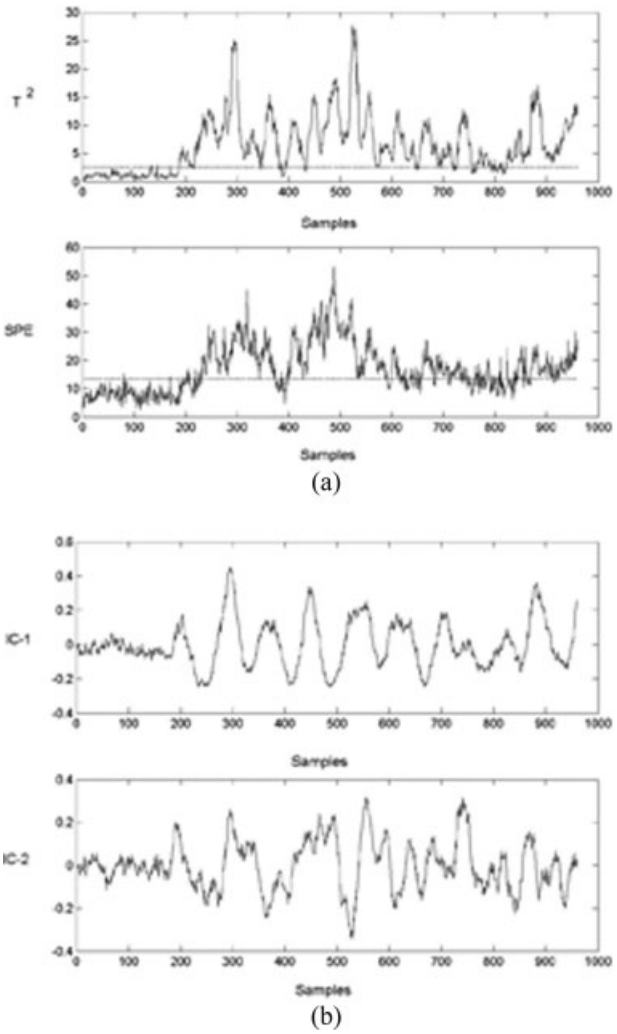

Figure 10. Monitoring results of the Tennessee Eastman process based (a) KICA, (b) Two ICs in the case of Fault 8.

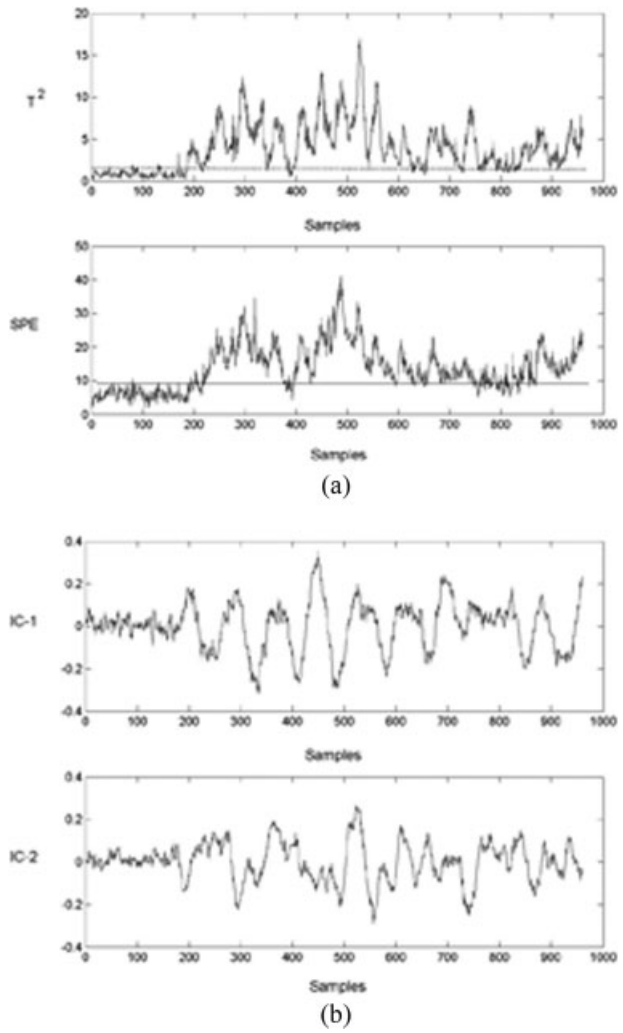

Figure 11. Monitoring results of the Tennessee Eastman process based on (a) improved KICA and (b) Two ICs (γ_0,γ_1 = 0.05 and n = 320) in the case of Fault 8.

$$h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2} \tag{35}$$

and c_α is the normal deviate corresponding to the upper $100(1-\alpha)\%$. m denotes the effective dimension of feature space.

Outline of fault detection based on improved KICA

In PCA and KPCA analysis, the changes in the score variable relation according to the PCs and nPCs reflect the changes in underlying process behavior. Likewise, in KICA analysis, the changes in the score variable relation according to the ICs reflect the changes in underlying process behavior.

Developing the Normal Operating Condition (NOC) Model

(1) Acquire an operating data set during normal operation.

(2) Similarities in the input space are analyzed to obtain first data subset Ω_1. The process is as follows:

(i) Set γ_0, $0 \le \gamma_0 \le 1$.

(ii) Calculate the similarity factor S_e according Eq. 13.

(iii) if $S_e < \sqrt{\gamma_0}$, the new data will be introduced to constitute the expanded node set; otherwise, this data point will be rejected without the node set expansion.

(3) Mapping the first sub-datasets Ω_1 to feature space, similarities in the feature space are analyzed to obtain second data subset Ω_2. The process is as follows:

(i) Set γ_1, $0 \le \gamma_1 \le 1$.

(ii) Calculate the similarity factor S_f (S_{f1} or S_{f2}) according Eq. 15 (Eq. 25 or 26).

(iii) if $S_f < \sqrt{\gamma_0}$, the new data will be introduced to constitute the expanded node set; otherwise, this data point will be rejected without the node set expansion.

(4) Normalize the data in the subset Ω_2 using the mean and standard deviation of each variable.

(5) For the second data subset Ω_2, compute the kernel matrix $\boldsymbol{K} \in \boldsymbol{R}^{n\times n}$ by Eq. 2. Calculate the mean centered kernel matrix $\tilde{\boldsymbol{K}}$ according to (3). (we can thus obtain the orthonormal Eigenvectors $\bar{\boldsymbol{\alpha}}_1, \bar{\boldsymbol{\alpha}}_2, ..., \bar{\boldsymbol{\alpha}}_d$ of $\tilde{\boldsymbol{K}}$ corresponding to largest positive eigenvalues $\bar{\lambda}_1 \ge \bar{\lambda}_2 \ge ... \ge \bar{\lambda}_d$).

(6) Choose q, the number of independent components to estimate. Set counter $i \leftarrow 1$.

(7) Take the initial vector c_i.

(8) Let $c_i \leftarrow E\{\mathbf{z}g(c_i^T\mathbf{z})\} - E\{g'(c_i^T\mathbf{z})\}c_i$, where g' is the first order derivative of g.

(9) Do the following orthogonalization: $c_i \leftarrow c_i - \sum_{j=1}^{i-1}(c_i^T c_i)c_i$. This orthogonalization excludes the information contained in the solutions already found.

(10) Normalize $c_i \leftarrow \frac{c_i}{\|c_i\|}$.

(11) If c_i has not converged, go back to Step (8).

(12) If c_i has converged, output the vector c_i. Then, set $i \leftarrow i+1$ and go back to step (8).

(13) Compute dominant components $\mathbf{y}$.

(14) Calculate the monitoring statistics (T^2 and SPE) of the normal operating data according to the Eqs. 36 and 37.

(15) Determine the control limits of T^2 and SPE charts.

Table 1. Fault Misdetection Rates of Methods in the Tennessee Eastman Process (PCA and Modified ICA)

	PCA		Modified ICA	
Faults	T^2	SPE	T^2	SPE
5	77	80	76	77
8	3	11	3	2
10	70	82	30	37
11	78	28	57	34
17	26	7	13	6
19	100	71	74	71

Table 2. Fault Misdetection Rates of Methods in the Tennessee Eastman Process (KPCA, Improved KPCA, KICA, and Improved KICA)

	KPCA		Improved KPCA		KICA		Improved KICA	
Faults	T^2	SPE	T^2	SPE	T^2	SPE	T^2	SPE
5	75	73	73	72	**71**	73	**71**	72
8	3	4	**2**	4	3	**2**	**2**	**2**
10	57	50	55	49	19	20	18	**15**
11	76	19	71	19	19	23	23	**18**
17	26	**5**	18	**5**	**5**	**5**	**5**	**5**
19	97	51	78	40	25	31	**23**	28

Table 3. Training Time for the Fault 8 Tennessee Eastman Process

Kernel Parameters	Original KPCA (s)	Improved KPCA (s)
$\sigma = 0.5$	107	92
$\sigma = 0.6$	114	99
$\sigma = 0.7$	103	84

From Lee and Qin,[30] we know the T^2 statistic is defined as follows:

$$T^2 = \mathbf{y}^T \mathbf{D}^{-1} \mathbf{y} \tag{36}$$

$$\text{SPE} = \mathbf{e}^T\mathbf{e} = (\mathbf{z} - \hat{\mathbf{z}})^T(\mathbf{z} - \hat{\mathbf{z}}) = \mathbf{z}^T\left(\mathbf{I} - \mathbf{C}_n\mathbf{C}_n^T\right)\mathbf{z}, \quad \mathbf{z} = \mathbf{P}^T\mathbf{\Phi}(\mathbf{x}) \tag{37}$$

where $\mathbf{e} = \mathbf{z} - \hat{\mathbf{z}}$ and $\hat{\mathbf{z}}$ can be found from:

$$\hat{\mathbf{z}} = \mathbf{C}_n\mathbf{C}_n^T\mathbf{z}. \tag{38}$$

The upper control limit for T^2 can be determined using the F-distribution because y does not follow Gaussian distribution. In this article, Kernel density estimation is used to define the control limit for T^2.

The control limits for the SPE are calculated from the following weighted χ^2 distribution[1]

$$\begin{aligned} \text{SPE} &\sim \mu\chi_h^2 \\ \mu = b/2a, \quad h &= 2a^2/b \end{aligned} \tag{39}$$

where a and b are the estimated mean and variance of the SPE from the normal operating data, respectively.

On-Line Monitoring

(1) For new scaled test data $\mathbf{x}_t \in \boldsymbol{R}^m$, compute the Kernel vector $k_t \in R^{1\times n}$ by $[k_t]_j = [k_t(\mathbf{x}_t,\mathbf{x}_j)]$, where $\mathbf{x}_j$ is the scaled normal operating. $\mathbf{x}_j \in \boldsymbol{R}^m$, $j = 1,2,\ldots,n$.

(2) The test Kernel vector k_t is mean centered as follows

$$\tilde{k}_t = k - 1_t\boldsymbol{K} - \boldsymbol{K}1_I + 1_t\boldsymbol{K}1_{\boldsymbol{I}}$$

where $\boldsymbol{K}$ are determined by step (2) of the NOC. $\boldsymbol{1}_t = (1/I)[1,\ldots,1] \in \boldsymbol{R}^{1\times n}$.

(3) For the test data $\mathbf{x}$, extract a nonlinear component via

$$\mathbf{z}_t = \mathbf{P}^T\mathbf{\Phi}(\mathbf{x}_t) = \sqrt{N}\mathbf{\Lambda}^{-1}\mathbf{H}^T[\tilde{k}(\mathbf{x}_l,\mathbf{x}_t),\ldots,\tilde{k}(\mathbf{x}_n,\mathbf{x}_t)]^T$$

(4) Further processing using modified ICA

$$\mathbf{y} = \mathbf{D}^{1/2}\mathbf{C}_n^T\mathbf{z}_t$$
$$\hat{\mathbf{z}} = \mathbf{C}_n\mathbf{C}_n^T\mathbf{z}_t$$

Table 4. Training Time for the Fault 10 Tennessee Eastman Process

Kernel Parameters	Original KICA (s)	Improved KICA (s)
$\sigma = 0.5$	132	101
$\sigma = 0.6$	145	113
$\sigma = 0.7$	124	106

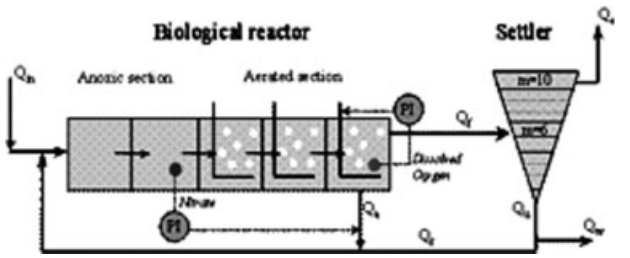

Figure 12. Flow diagram of the WWTP simulation benchmark (Q_{in}: influent flow rate, Q_a: internal recycle flow rate, Q_e: effluent flow rate, Q_f: feed layer flow rate, Q_r: return sludge flow rate, Q_u: under flow rate, Q_w: treated water output flow rate).

(5) Calculate the monitoring statistics (T^2 and SPE) of the normal operating data.

(6) Determine the control limits of T^2 and SPE.

Simulation Result

Tennessee Eastman process

In this section, the proposed method is applied to the Tennessee Eastman process simulation data and is compared with PCA, modified ICA, original KPCA, and KICA monitoring results. The control structure is shown schematically in Figure 2. The details on the process description are well explained in Chiang et al.[4] A total of 52 variables are used for monitoring in this study. The data are generated by Chiang et al. And can be downloaded from http://brahms.scs.uiuc.edu.

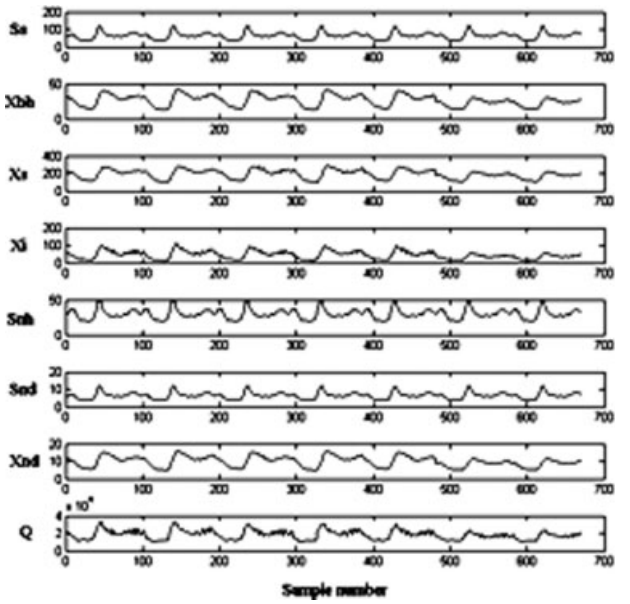

Figure 13. Variable patterns of normal operating condition data.

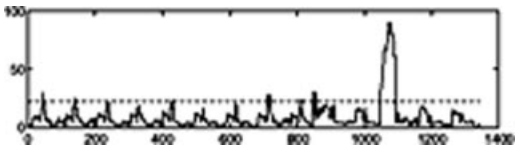

Figure 14. ICA monitoring results of WWTP.

All the data are applied to the PCA, modified ICA, KPCA, KICA, improved KPCA, and KICA. The dotted line in each figure is the threshold for the statistic, the statistic above the threshold indicates that a fault is detected (the statistic is shown as a solid line). To clearly illustrate the superiority of improved KPCA and KICA in process monitoring, two monitoring results in the case of Faults 10 and 8 are shown. Radial basis kernel $k(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{\sigma}\right)$ is selected. And $\sigma = 0.6$. For the data obtained after the fault occurrence, the percentage of the samples outside the 99% control limits was calculated in each simulation. In the case of Fault 10, the C feed temperature of stream 4 is randomly changed. When the fault occurs, stripper temperature also changes, which results in change in stripper pressure. To compensate for the changes of the stripper temperature and pressure, stripper steam valve is manipulated by a control loop and thereby stripper steam flow rate also changes.[4] In Figure 3, PCA detect the fault from about sample 200 and delayed time is 40 sample intervals, there are lots of samples below the control limit despite the presence of the fault. From Figure 4, ICA is able to detect the fault from about sample 180, which are more efficiently than PCA without false alarms. From Figures 5 and 6, it is evident that the KICA and the improved KICA give the better monitoring performance to detect the fault from about sample 170, giving no false alarms and a higher detection rate. The improved KICA monitoring charts show that statistics successfully detect more faults than the original KICA. For the Fault 8, the monitoring results based on KPCA and improve KPCA are shown in Figures 7 and 8. The nonlinear principal component analysis is shown in Figure 9. For this fault, KPCA can detect the fault from about sample 185 (Figure 8) and delayed time is 25 sample interval. For the improved KPCA,

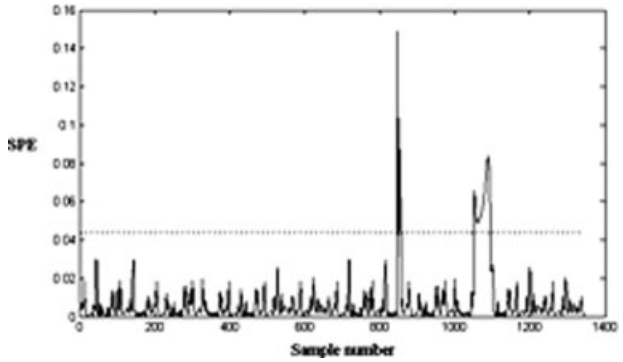

Figure 15. KPCA monitoring results of WWTP in the case of two storms.

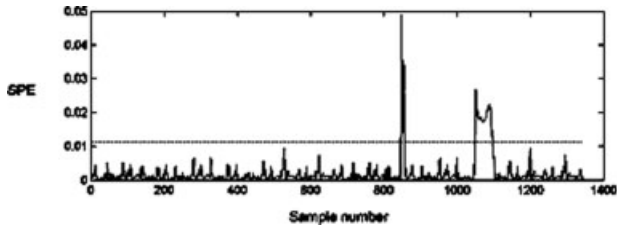

Figure 16. Improved KPCA monitoring results of WWTP based on $\gamma_0,\gamma_1 = 0.05$ and $n = 320$ in the case of two storms.

there are the less sample than KPCA below the control limit. In contrast to the original KPCA monitoring, the improved KPCA monitoring charts show that T^2 statistics successfully detect more faults. The monitoring results based on KICA and improved KICA in the case of Fault 8 are shown in Figures 10 and 11, respectively. The KICA monitoring charts show that statistics successfully detect the faults from sample 180 (Figure 10). The improved KICA monitoring charts show that statistics successfully detect more faults than the original KICA. The fault misdetection rates of the six multivariate methods, PCA, Modified ICA, KPCA, improved KPCA, KICA, and improved KICA for Faults 5, 8, 10, 11, 17, and 19 are computed and summarized in Tables 1 and 2. For most faults, the detection rate of improved KICA (or KPCA) is higher than that of KICA (or KPCA) because the former avoids the linear data errors introduced by the Kernel trick. Otherwise, the detection rate of KICA and improved KICA is higher than that of KPCA and improved KPCA because the former considers the non-Gaussianity of data. The minimum misdetection rates achieved are marked with a bold number. From Table 3, we found the training time of the improved KPCA is less than that of original KPCA. From Table 4, we found the training time of the improved KICA is less than that of original KICA. The missed detection rates for the Faults 3, 9, and 15 are very high for all the fault detection statistics. No observed change in the mean, the variance, and the higher order variance can be detected with Faults 3, 9, and 15 to the plots associated with the normal operated condition. It conjectured that any statistic will result in high missed detection rates for those fault, in other words, the Faults 3, 9, and 15 are unobservable from the test data.

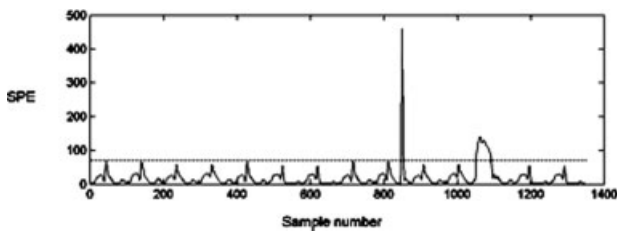

Figure 17. KICA monitoring results of KICA of WWTP in the case of two storms.

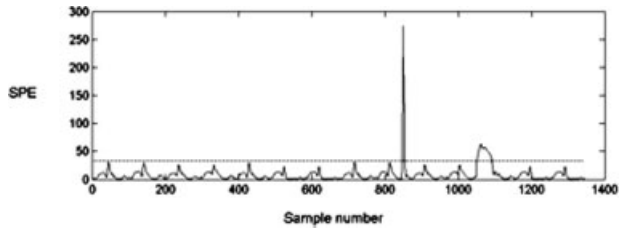

Figure 18. Improved KICA monitoring results based on $\gamma_0,\gamma_1 = 0.05$ and $n = 320$ of WWTP in the case of two storms.

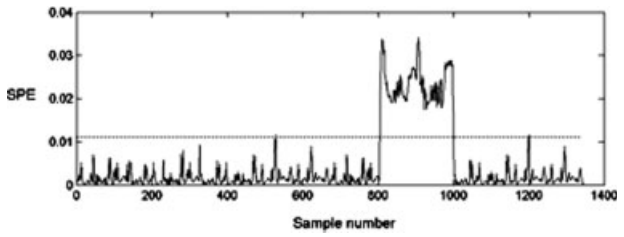

Figure 20. Improved KPCA monitoring results of WWTP based on $\gamma_0,\gamma_1 = 0.05$ and $n = 320$ in the case of long rain.

Wastewater treatment process (WWTP)

The proposed improved KPCA and KICA monitoring methods are tested for its ability to detect two storms and long rain in simulated data obtained from a "benchmark simulation" of the WWTP (Figure 12). Detail process description is well explained from http://www.Ensic.u-nancy.Fr/COSTWWTP/.

In this article, 1 week normal data with the sampling time of 15 min are generated from the simulator for training data. Eight variables are selected for process monitoring since they are important and typically monitored in real WWTP systems. The values of measurement variables during the normal weather are presented in Figure 13. WWTP is subject to large diurnal fluctuations in the flow rate and composition of the feed stream. The variables of such processes tend to fluctuate widely over a cycle, their mean and variance do not remain constant over time. Because of this, conventional MSPM, which implicitly assumes a stationary underlying process, may lead to numerous false alarms and missed faults. Nonlinear process monitoring approach KPCA is used to detect and diagnose the abnormal case.

Two types of disturbances are tested. We applied the proposed method to this process in which two sudden storm events occur after a long period of dry weather. First storm: samples 850–865 and second storm: 1050–1110. For two storms, ICA is able to detect one storm at sample 1050 in SPE charts (Figure 14). From this picture, we found it can only detect second storm. 95% confidence limit of the plot is plotted. KPCA can detect successfully two storms at around sample 860 and sample 1060, respectively (Figure 15). In contrast to the original KPCA monitoring, the improved KPCA monitoring charts show that SPE statistics also detect the two storms at around sample 855 and sample 1055, respectively (Figure 16). KICA can detect successfully two storms at around sample 860 and sample 1060, respectively (Figure 17). In contrast to the original KICA monitoring, the improved KICA monitoring charts show that SPE statistics also detect the two storms at around sample 855 and sample 1055, respectively (Figure 18). For long rain, KPCA can detect it successfully (Figure 19). Also, the improved KPCA monitoring charts show that SPE statistics successfully detect the long rain (Figure 20). The training time of the improved Kernel methods (improved KPCA and KICA) is less than original Kernel methods (KPCA and KICA).

Penicillin fermentation

In this section, the proposed method is applied to the monitoring of a well-known benchmark process, penicillin fermentation process. A flow diagram of the penicillin fermentation process is given in Figure 21. Trajectories of nine variables from a nominal batch run are shown in Figure 22. The production of secondary metabolites such as antibiotics has been the subject of many studies because of its academic and industrial importance. Here, we focus on the process to produce penicillin, which has nonlinear dynamics and multi-

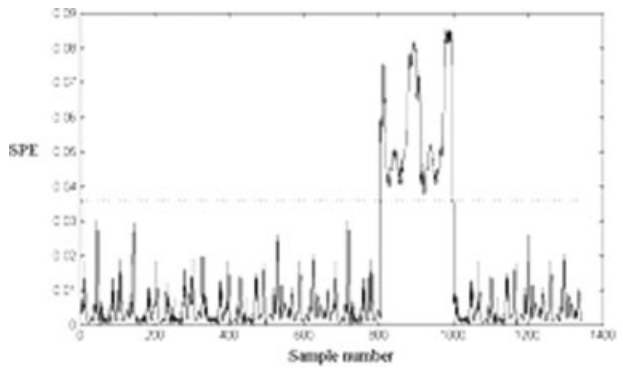

Figure 19. KPCA monitoring results of WWTP in the case of long rain.

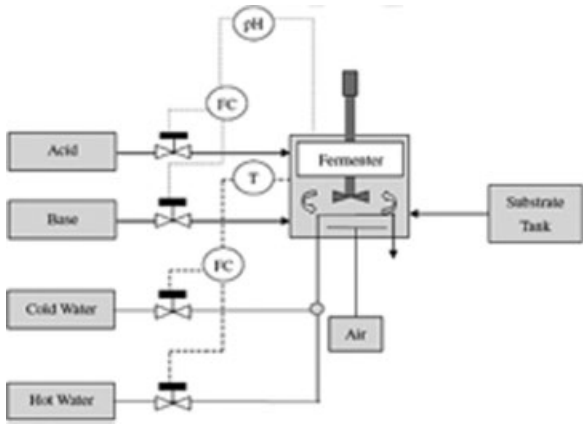

Figure 21. Penicillin fermentation process.

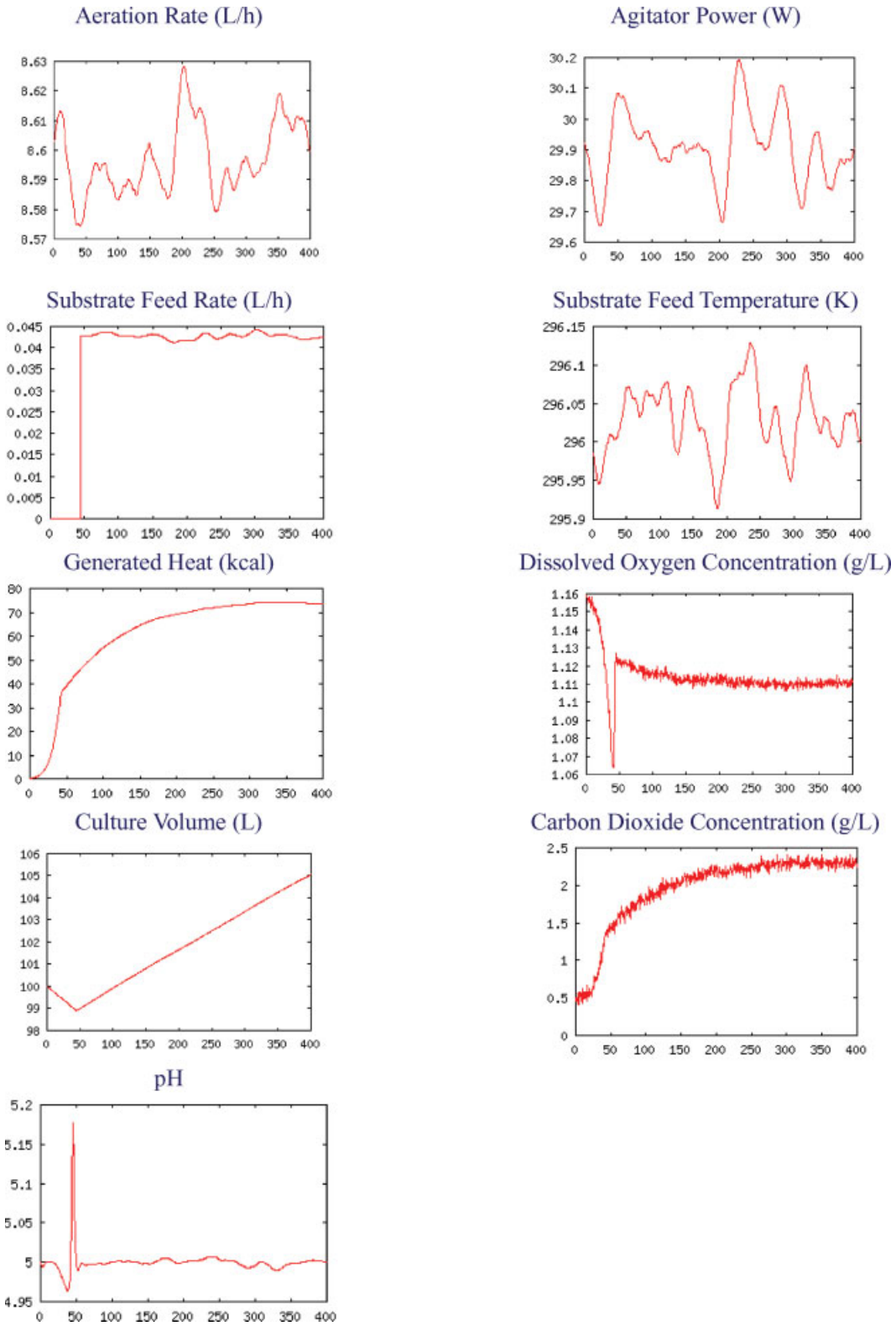

Figure 22. Trajectories of nine variables from a nominal batch run.

[Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

phase characteristics. In typical operating procedure for the modeled fed-batch fermentation, most of the necessary cell mass is obtained during the initial preculture phase. When most of the initially added substrate has been consumed by the microorganisms, the substrate feed begins. The penicillin starts to be generated at the exponential growth phase and continues to be produced until the stationary phase. A low substrate concentration in the fermentor is necessary for achieving a high product formation rate due to the catabolite repressor. Consequently, glucose is fed continuously during fermentation at the beginning. In the present simulation experiment, a total of 60 reference batches are generated using a simulator (PenSim v2.0 simulator). Detail process description is well explained from http://www.chee.iit.edu/~cinar/software.htm. These simulations are run under closed-loop control of pH and temperature, whereas glucose addition is performed under open-loop. Small variations are automatically added to mimic the real normal operating conditions under the default initial setting conditions. The duration of each batch is 400 h, consisting of a preculture phase of about 45 h and a fed-batch phase of about 355 h. The models are constructed using the proposed method. KPCA and improved KPCA are then tested against monitoring of fault batches. Fault is implemented by introducing a 25% step increase in the substrate feed rate at 100 h and retaining until 220 h. The monitoring results are shown in Figures 23 and 24, respectively. From the beginning of a batch to the time of about 45 h, both SPE move up and down because the operating conditions of startup have some deviations for each batch. In our study, notable result is that the improved method can detect faults successfully in all the cases.

Conclusions

In this article, the nonlinear process monitoring methods based on KPCA and KICA are improved. When compared with original KPCA and KICA, the proposed algorithm has the advantages: the computational error and loading are avoided by using similarity analysis and *k*-mean cluster in Kernel space. The proposed method is applied to the fault detection of the Tennessee Eastman process, the WWTP process, and the Penicillin process. The examples demonstrated that the proposed method detected various nonlinear process faults more efficiently than original KPCA and original KICA.

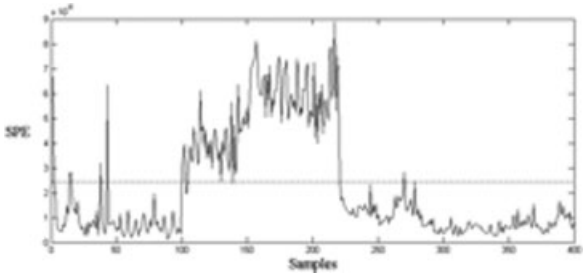

Figure 23. Improved KPCA monitoring results of penicillin fermentation process based on n = 320 in the case of 25% increase between 100s and 220s.

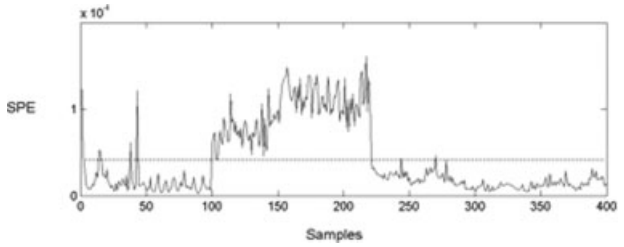

Figure 24. Improved KICA monitoring results of penicillin fermentation process based on γ_0,γ_1 = 0.05 and n = 320 in the case of 25% increase between 100s and 220s.

Acknowledgments

This work was supported by the Texas-Wisconsin-California Control Consortium. The second author acknowledges the financial support from the Chang-Jiang Professorship by the Ministry of Education of China.

Literature Cited

1. Qin SJ. Statistical process monitoring: basics and beyond. *J Chemom*. 2003;17:480–502.
2. Lee G, Han CH, Yoon ES. Multiple-fault diagnosis of the Tennessee Eastman process based on system decomposition and dynamic PLS. *Ind Eng Chem Res*. 2004;43:8037–8048.
3. Lee JM, Yoo CK, Lee IB. Statistical process monitoring with independent component analysis. *J Process Control*. 2004;14:467–485.
4. Chiang LH, Russell FL, Braatz RD. *Fault Detection and Diagnosis in Industrial Systems*. London: Springer, 2001.
5. MacGregor JF, Kourti T. Statistical process control of multivariate processes. *Control Eng Practice*. 1995;3:403–414.
6. Hiden HG, Willis MJ, Tham MT, Montague GA. Non-linear principal components analysis using genetic programming. *Comput Chem Eng*. 1999;23:413–425.
7. Chen JH, Liao CM. Dynamic process fault monitoring based on neural network and PCA. *J Process Control*. 2002;12:277–289.
8. Dong D, McAvoy TJ. Nonlinear principal component analysis-based on principal curves and neural networks. *Comput Chem Eng*. 1996; 20:65–78.
9. Jia F, Martin EB, Morris AJ. Non-linear principal component analysis for process fault detection. *Comput Chem Eng*. 1998;22:S851–S854.
10. Kramer MA. Non-linear principal component analysis using autoassociative neural networks. *AIChE J*. 1991;37:233–243.
11. Tan S, Mavrovouniotis ML. Reducing data dimensionality through optimizing neural networks inputs. *AIChE J*. 1995;41:1471–1480.
12. Geng ZQ, Zhu QX. Multiscale nonlinear principal component analysis (NLPCA) and its application for chemical process monitoring. *Ind Eng Chem Res*. 2005;44:3585–3593.
13. Kulkarni SG, Chaudhary AK, Nandi S, Tambe SS, Kulkarni BD. Modeling and monitoring of batch processes using principal component analysis (PCA) assisted generalized regression neural networks (GRNN). *Biochem Eng J*. 2004;18:193–210.
14. Walczak B, Massart DL. Local odeling with radial basis function networks. *Chem Intell Lab Syst*. 2000;50:179–198.
15. Schölkopf B, Sung KK, Burges CJC, Girosi F, Niyogi P, Poggio T, Vapnik V. Comparing support vector machines with Gaussian kernels to radial basis function classifiers. *IEEE Trans Signal Process*. 1997;45:2758–2765.
16. Choi SW, Lee D, Park JH, Lee IB. Nonlinear regression using RBFN with linear submodels. *Chem Intell Lab Syst*. 2003;65:191–208.
17. Schölkopf B, Smola AJ, Mjuller K. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput*. 1998;10:1299–1399.

18. Cho JH, Lee JM, Choi SW, Lee D, Lee IB. Fault identification for process monitoring using kernel principal component analysis. *Chem Eng Sci*. 2005;60:279–288.
19. Choi SW, Lee C, Lee JM, Park JH, Lee IB. Fault detection and identification of nonlinear processes based on KPCA. *Chem Intell Lab Syst*. 2005;75:55–67.
20. Mika S, Schölkopf B, Smola AJ, Mjuller KR, Scholz M, Rjatsch G. KPCA and de-noising in feature spaces. *Adv Neural Inf Process Syst*. 1999;11:536–542.
21. Romdhani S, Gong S, Psarrou A. A multi-view nonlinear active shape model using KPCA. *In Proceedings of BMVC*, Nottingham, UK. 1999:483–492.
22. Kano M, Tanaka S, Hasebe S, Hashimoto I, Ohno H. Monitoring independent components for fault detection. *AIChE J*. 2003;49:969–976.
23. Lee JM, Yoo CK, Lee IB. New monitoring technique with ICA algorithm in wastewater treatment process. *Water Sci Technol*. 2003;47:49–56.
24. Kano M, Tanaka S, Hasebe S, Hashimoto I, Ohno H. Evolution of multivariate statistical process control: independent component analysis and external analysis. *Comput Chem Eng*. 2004;28:1157–1166.
25. Lee JM, Yoo CK, Lee IB. Statistical monitoring of dynamic processes based on dynamic independent component analysis. *Chem Eng Sci*. 2004;59:2995–3006.
26. Yoo CK, Lee JM, Vanrolleghem PA, Lee IB. On-line monitoring of batch processes using multiway independent component analysis. *Chem Intell Lab Syst*. 2004;71:151–163.
27. Albazzaz H, Wang XZ. Statistical process control charts for batch operations based on independent component analysis. *Ind Eng Chem Res*. 2004;43:6731–6741.
28. Lee T. *Independent Component Analysis: Theory and Applications*. Boston: Kluwer Academic Publishers, 1998.
29. Lee JM, Qin SJ, Lee IB. Fault detection and diagnosis of multivariate processes based on modified independent component analysis. *AIChE J*. 2006;52:3501–3514.
30. Lee JM, Qin SJ, Lee IB. Fault detection of nonlinear processes using Kernel independent component analysis. *Can J Chem Eng*. 2007;85:526–536.
31. Zhang Y, Qin SJ. Fault detection of nonlinear processes using multiway Kernel independent analysis. *Ind Eng Chem Res*. 2007;46:7780–7787.
32. Krzanowski WJ. Between-groups comparison of principal components. *J Am Stat Assoc*. 1979;74:703–707.

Manuscript received Mar. 21, 2008, revision received May 18, 2008, and final revision received July 17, 2008.