Korean J. Chem. Eng., **30**(6), 1181-1186 (2013)
DOI: 10.1007/s11814-013-0034-7

Fault detection in nonlinear chemical processes based on kernel entropy component analysis and angular structure

Qingchao Jiang*, Xuefeng Yan*,†, Zhaomin Lv*, and Meijin Guo**

*Key Laboratory of Advanced Control and Optimization for Chemical Processes of Ministry of Education, East China University of Science and Technology, Shanghai 200237, P. R. China
**State Key Laboratory of Bioreactor Engineering, East China University of Science and Technology, Shanghai 200237, P. R. China
(*Received 12 October 2012 • accepted 3 March 2013*)

Abstract−Considering that kernel entropy component analysis (KECA) is a promising new method of nonlinear data transformation and dimensionality reduction, a KECA based method is proposed for nonlinear chemical process monitoring. In this method, an angle-based statistic is designed because KECA reveals structure related to the Renyi entropy of input space data set, and the transformed data sets are produced with a distinct angle-based structure. Based on the angle difference between normal status and current sample data, the current status can be monitored effectively. And, the confidence limit of the angle-based statistics is determined by kernel density estimation based on sample data of the normal status. The effectiveness of the proposed method is demonstrated by case studies on both a numerical process and a simulated continuous stirred tank reactor (CSTR) process. The KECA based method can be an effective method for nonlinear chemical process monitoring.

Key words: Kernel Entropy Component Analysis, Process Monitoring, Fault Detection, Angular Structure

INTRODUCTION

Chemical processes have become larger, more complex and more intelligent. Fault detection and diagnosis plays an increasingly important role in process safety and productivity. Recently, with the development of measurement, data storage and computing equipment, multivariate statistical process monitoring (MSPM) methods have been widely used and have achieved many favorable effects [1-9]. Among the MSPM methods, principal component analysis (PCA) process monitoring is the most widely used because of its effectiveness in handling high dimensional and correlated data. PCA projects data information into two subspaces, dominant subspace and residual subspace; and two statistics, represented by Mahalanobis and Euclidean distances, are constructed to detect the changes in the two subspaces [2,10]. However, for some complicated chemical and biological processes with particular nonlinear characteristics, PCA performs poorly due to its assumption that the process data are linear [4,11].

To deal with nonlinear processes, Kramer [12] developed a nonlinear PCA method based on auto-associative neural networks. However, it is difficult to determine the number of nodes in each layer and train the network with five layers proposed by Kramer. Dong and McAvoy [13] developed a nonlinear PCA approach based on principal curves and neural networks, which assumes that the nonlinear function can be approximated by a linear combination of several univariate functions. Such mappings can only be made for a limited class of nonlinear models, and the application of the principal curve algorithm is restricted to the identification of structures that exhibit additive-type behavior [14,15]. Furthermore, KPCA as a new nonlinear PCA technique has been developed very quickly in recent years [16-18]. KPCA first maps the input space into a feature space via nonlinear mapping and then computes the principal components (PCs) in the high-dimensional feature space. Compared to other nonlinear methods, the main advantage of KPCA is that the nonlinear mapping and the inner product computation is avoided by introducing a kernel function. Similar to conventional PCA, two monitoring statistics, known as T^2 and SPE, are constructed to monitor the dominant and residual subspaces separately. The confidence limit determinations of the statistics are based on the assumption that the obtained score variable follows Gaussian distribution, which can hardly be satisfied in nonlinear processes.

In both PCA and KPCA monitoring methods, the first several principal components (PCs) corresponding to the top eigenvalues are employed to construct the dominant subspace. The transformation preserves the maximum variances information of the original process data. In this paper, a new multivariate statistical data transformation method, kernel entropy component analysis (KECA), is introduced for nonlinear chemical processes monitoring. The KECA data transformation method was first proposed and employed in pattern analysis by Jenssen [19]. It is fundamentally different from other multivariate statistical analysis methods, such as PCA and KPCA, in three important ways: first, the data transformation reveals structure related to the Renyi entropy of input space data set; second, the method does not necessarily use the top eigenvalues and eigenvectors of the kernel matrix; and finally, KECA produces the transformed data with a angular structure in the feature subspace [19, 20]. It has been reported as a promising method in nonlinear data transformation and dimensionality reduction, and has achieved favorable effects in pattern analysis; however, to the best of our knowledge, it has not been used in process monitoring. The main contributions of this paper are as follows: (1) Introduce the KECA method

†To whom correspondence should be addressed.
E-mail: xfyan@ecust.edu.cn

to the process monitoring field for the first time and provide a useful alternative to kernel PCA for dealing with nonlinear chemical processes; (2) Propose an angle-based fault detection method for nonlinear chemical process monitoring and relax the Gaussian assumption in conventional PCA and KPCA methods.

The rest of the paper is organized as follows. The KECA method for data transformation is presented, and then an angle-based method for process monitoring is proposed. Next, a numerical process and the continuous stirred tank reactor (CSTR) process are employed to demonstrate the effectiveness of KECA based process monitoring. Finally, conclusions are presented.

METHOD

In this section, the basics of the KECA for data transformation are presented at first. And then, based on KECA, an angle-based statistic is designed for fault detection.

1. Kernel Entropy Component Analysis

Assuming that $p(\mathbf{x})$ is the probability density function generating the data set D: $\mathbf{x}_1, \cdots, \mathbf{x}_N$, the Renyi quadratic entropy can be given by [21]

$$H(p) = -\log \int p^2(\mathbf{x}) d\mathbf{x} \tag{1}$$

Because the logarithm is a monotonic function, attention could be focused on the quantity $V(p)=\int p^2(\mathbf{x})d\mathbf{x}$. This expression could also be formulated as $V_p=\varepsilon_p(p)$, in which $\varepsilon_p(\cdot)$ denotes expectation with regard to the density p(x). To estimate V(p), and hence H(p), a Parzen window density estimator is invoked, given by [22]

$$\hat{p}(x) = \frac{1}{N}\sum_{x_t \in D} k_\sigma(\mathbf{x}, \mathbf{x}_t) \tag{2}$$

Here, $k_\sigma(\mathbf{x}, \mathbf{x}_t)$ is the Parzen window, or kernel, centered at $\mathbf{x}_t$; the parameter σ is the window width. To guarantee $\hat{p}(x)$ a proper density function, $k_\sigma(\mathbf{x}, \bullet)$ must be a density function itself [23]. Using the sample mean approximation of the expectation operator, we can obtain [19]

$$\hat{V}(p) = \frac{1}{N}\sum_{x_t \in D} \hat{p}(\mathbf{x}_t) = \frac{1}{N}\sum_{x_t \in D}\frac{1}{N}\sum_{x_t' \in D} k_\sigma(\mathbf{x}_t, \mathbf{x}_t') = \frac{1}{N^2}\mathbf{1}^T\mathbf{K}\mathbf{1} \tag{3}$$

where element (t, t') of the (N×N) kernel matrix $\mathbf{K}$ equals $k_\sigma(\mathbf{x}_t, \mathbf{x}_t')$ and $\mathbf{1}$ is an (N×1) vector where each element equals one. Therefore, the Renyi entropy estimator obtained based on the available sample fully resides in the elements of the corresponding kernel matrix.

Moreover, the Renyi entropy estimator may be expressed in terms of the eigenvalues and eigenvectors of the kernel matrix, which may be eigen-decomposed as $\mathbf{K}=\mathbf{E}\mathbf{D}\mathbf{E}^T$, where $\mathbf{D}$ is a diagonal matrix storing the eigenvalues $\lambda_1, \cdots, \lambda_N$ and $\mathbf{E}$ is a matrix with the corresponding eigenvectors $\mathbf{e}_1, \cdots, \mathbf{e}_N$ as columns. Rewriting (3), we then obtain

$$\hat{V}(p) = \frac{1}{N^2}\sum_{i=1}^{N}(\sqrt{\lambda_i}\mathbf{e}_i^T\mathbf{1})^2 \tag{4}$$

This is the same with the decomposition in kernel PCA, and in this expression, each term will contribute to the entropy estimate. Certain eigenvalues and eigenvectors will contribute more to the entropy estimate than others because the terms depend on different eigenvalues and eigenvectors.

In kernel PCA, the nonlinear mapping from input space to feature space is given by ϕ, and that is $\mathbf{x}_t \rightarrow \phi(\mathbf{x}_t)$. Let $\Phi=[\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \cdots, \phi(\mathbf{x}_N)]$, the projection of Φ onto the *i*-th principal axis u_i in the kernel feature space is defined as $P_{u_i}\Phi=\sqrt{\lambda_i}\mathbf{e}_i^T$. Eq. (4) therefore reveals that, given a positive semidefinite (psd) window function, the Renyi entropy estimator is composed of projections onto all the kernel PCA axes. However, only a principal axis u_i for which $\lambda_i \neq 0$ and $\mathbf{e}_i^T\mathbf{1} \neq 0$ contributes to the entropy estimate, and a large λ_i cannot guarantee that the axis contributes more to the entropy estimate. Obviously, those principal axes contributing most to the Renyi entropy estimate carry most of the information regarding the shape of the probability distribution function generating the input space data set. Note that kernel PCA performs dimensionality reduction by selecting s eigenvalues and eigenvectors solely based on the size of the eigenvalues, and the resulting transformation may be based on uninformative eigenvectors from an entropy perspective [19].

Kernel entropy component analysis is defined as an *s*-dimensional data transformation obtained by projecting Φ onto a subspace U_s spanned by those s kernel PCA axes contributing most to the Renyi entropy estimate of the data [19]. Not necessarily those corresponding to the top s eigenvalues, U_s can be composed of a subset of the kernel PCA axes. The transformation is as follows [19]

$$\Phi_{eca} = P_{U_s}\Phi = \mathbf{D}_s^{\frac{1}{2}}\mathbf{E}_s^T \tag{5}$$

This is the solution to the minimization problem, which is denoted by

$$\Phi_{eca} = \mathbf{D}_s^{\frac{1}{2}}\mathbf{E}_s^T : \min_{\lambda_1, e_1, \ldots, \lambda_N, e_N} \hat{V}(p) - \hat{V}_s(p) \tag{6}$$

where the entropy estimate associated with Φ_{eca} is expressed by

$$\hat{V}_s(p) = \frac{1}{N^2}\mathbf{1}^T\mathbf{E}_s\mathbf{D}_s\mathbf{E}_s^T\mathbf{1} = \frac{1}{N^2}\mathbf{1}^T\mathbf{K}_{eca}\mathbf{1} \tag{7}$$

in which $\mathbf{K}_{eca}=\mathbf{E}_s\mathbf{D}_s\mathbf{E}_s^T$. An alternative expression yields

$$\Phi_{eca} = \mathbf{D}_s^{\frac{1}{2}}\mathbf{E}_s^T : \min_{\lambda_1, e_1, \ldots, \lambda_N, e_N} \frac{1}{N^2}\mathbf{1}^T(\mathbf{K} - \mathbf{K}_{eca})\mathbf{1} \tag{8}$$

The minimum value obtained as the solution to (8) is given by

$$\min_{\lambda_1, e_1, \ldots, \lambda_N, e_N} \frac{1}{N^2}\mathbf{1}^T(\mathbf{K} - \mathbf{K}_{eca})\mathbf{1} = \frac{1}{N^2}\sum_{j=s+1}^{N}\psi_j \tag{9}$$

where ψ_j corresponds to the *j*-th largest term of (4).

Since kernel ECA selects components from the kernel future space, out of sample data points represented by Φ' are projected onto U_s, that is,

$$\Phi'_{eca} = P_{U_s}\Phi' = \mathbf{D}_s^{-\frac{1}{2}}\mathbf{E}_s^T\mathbf{K}'^T \tag{10}$$

Note that kernel ECA outputs the new kernel matrix $\mathbf{K}_{eca}$, and hence, –log of the quantity $\hat{V}_s(p)=(1/N^2)\mathbf{1}^T\mathbf{K}_{eca}\mathbf{1}$ can be interpreted as the Renyi entropy estimator of some data set, which can be denoted by $\mathbf{x}_1', \ldots, \mathbf{x}_N'$ [19]. $\mathbf{K}_{eca}$ could be associated with an input space transformation $\mathbf{x} \rightarrow \mathbf{x}'$ in terms of the entropy of $\mathbf{x}_1', \ldots, \mathbf{x}_N'$ is maximally similar to the entropy of $\mathbf{x}_1, \ldots, \mathbf{x}_N$, because $\hat{V}_s(p)$ preserves as much as possible of the Renyi entropy estimate $\hat{V}(p)$ of the original data $\mathbf{x}_1, \ldots, \mathbf{x}_N$. Moreover, it should be noted that KECA does not center

the kernel feature space data because centering does not make sense in KECA. More detail could be seen in [19].

2. Fault Detection Using Kernel ECA

It has been shown in [19] that KECA usually leads to a data set with a distinct angular structure, where different clusters are distributed more or less in different angular directions with respect to the origin of the kernel feature space. Therefore, the angle-based method is introduced in this section for fault detection. For a normal training data set $\mathbf{X} \in \mathbf{D} \in R^{d \times N}$, where d is the number of measurements and N is the number of observations, the overall kernel space mean vector $\mathbf{m}$ of the data is as follows:

$$\mathbf{m} = \frac{1}{N} \sum_{x_i \in D} \phi(\mathbf{x}_i) \tag{11}$$

where i=1, …, N, $\phi(\mathbf{x}_i)$ is the entropy component score of $\mathbf{x}_i$ in kernel spaces, which could be denoted as $\mathbf{m}_i$, that is $\mathbf{m}_i=\phi(\mathbf{x}_i)$. For a currently sampled point $\mathbf{x}_j$, the cosine value CV of the angle between the current point and normal data in feature kernel space is defined as follows:

$$CV = \cos\angle(\mathbf{m}_j, \mathbf{m}) = \frac{\mathbf{m}^T \mathbf{m}_j}{\|\mathbf{m}\| \|\mathbf{m}_j\|} \tag{12}$$

Here $\|\mathbf{m}\|$ is the L_2 norm of vector $\mathbf{m}$. Note that the CV values do not follow a particular distribution and the confidence limit cannot be determined directly from an approximate distribution. We use the kernel density estimation (KDE) in calculating the confidence limit of CV because the normal observations are easy to obtain and the KDE is effective in non-parametric estmation.

A univariate kernel estimator with kernel K is defined as follows [24]:

$$f(v) = \frac{1}{nh} \sum_{j=1}^{n} K\left\{\frac{v - v(j)}{h}\right\} \tag{13}$$

where v is the data point under consideration; v(j) is an observation value from the data set; h is the window width (also known as the smoothing parameter); n is the number of observations, and K is the kernel function. The kernel function K determines the shape of the smooth curve and satisfies the condition

$$\int_{-\infty}^{+\infty} K(v)dv = 1 \tag{14}$$

There are a number of possible kernel functions and the Gaussian kernel function is the most commonly used. A Gaussian kernel function is used as the kernel function in the present work, and details on kernel density estimation could be seen in [24,25]. First, we calculate the cosine values of a normal observation set. Second, we determine the confidence limit CL of CV that occupies the 99% area of the density function. Because the cosine values could be either positive or negative, the absolute values are calculated and the CL should include both up confidence limit and down confidence limit.

The KECA based fault detection method is illustrated in Fig. 1.

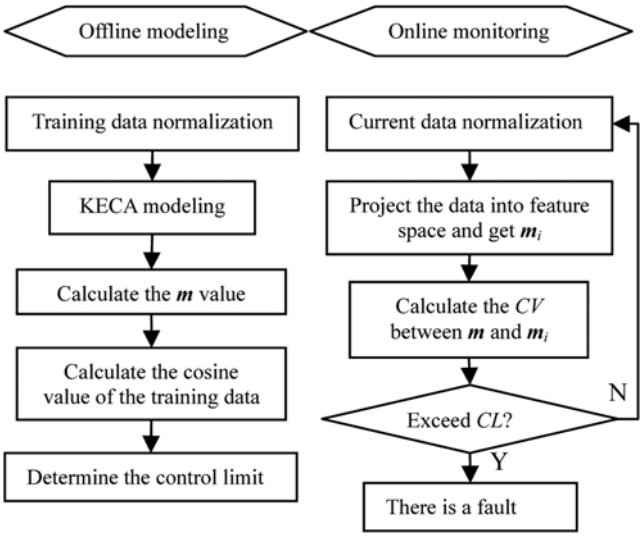

Fig. 1. Process monitoring scheme of the proposed KECA method.

APPLICATION

In this section, the proposed KECA fault detection method is applied to both a numerical process and the nonisothermal continuous stirred tank reactor model. The following simulations are in Matlab 7.12.0 environment.

1. Case Study on a Numerical Process

Consider the following system with two variables but only one factor, originally suggested by Dong and McAvoy [4,13].

$$x_1 = r + \varepsilon 1 \tag{15}$$

$$x_2 = r^2 - 3r + \varepsilon 2 \tag{16}$$

where $\varepsilon 1$ and $\varepsilon 2$ are independent noise variables N(0, 0.01), and

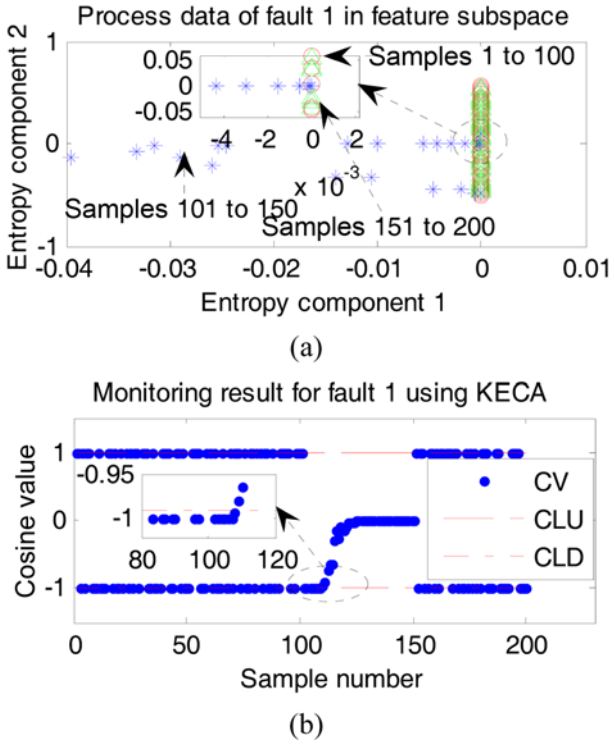

Fig. 2. Monitoring performance for fault 1 using KECA (a) Fault data in feature subspace (b) Fault detection results.

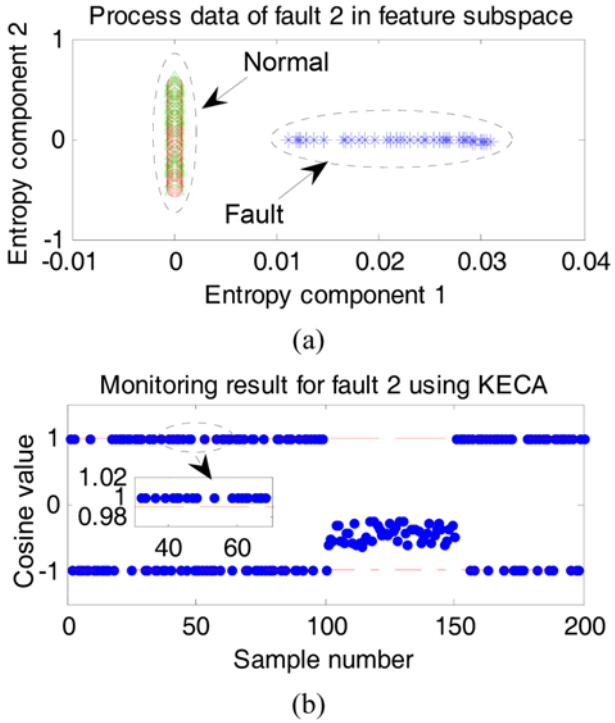

Fig. 3. Monitoring performance for fault 2 using KECA (a) Fault data in feature subspace (b) Fault detection results.

r∈[0.01, 0.2]. Normal data comprising 100 samples were generated according to these equations. Two sets of test data comprising 200 samples each were also generated. The following two faults were applied separately during generation of the test data sets:

Fault 1: x_1 was linearly increased from sample 101 to 150 by adding $0.05\times(n-100)$ to the x_1 value of each sample in this range, where n is the sample number.

Fault 2: a step change of x_2 by −2 was introduced starting from sample 101 and ended at sample 150.

The monitoring performance of KECA for the two faults is presented in Fig. 2 and Fig. 3. Fig. 2(a) shows the fault data reflected on the first two kernel entropy components in the kernel feature subspace. As is shown, the KECA transformation provides the fault data (samples 101 to 150, represented by star) a vertical angular structure with the normal process data (samples 1 to 100, circle, and samples 151 to 200, triangle). Based on the angular structure, the fault detection results for fault 1 are shown in Fig. 2(b), in which the CLU and CLD stand for the up confidence limit and down confidence limit, respectively. From Fig. 2(b), we can see the CVs of fault data exceed the confidence limits, indicating that there is a fault in the process. The angular structure and fault detection results for fault 2 are shown in Fig. 3(a) and Fig. 3(b). From Fig. 3(a) and 3(b), we can see that the fault data in kernel feature subspace also present an angular structure and the fault can be detected successfully.

2. Case Study on CSTR

In this application case study, the simulation model parameters and its simulation conditions used in [26] are adopted. The diagram of the process is shown in Fig. 4. This reactor is based on three assumptions: perfect mixing, constant physical properties, and negligible shaft work. The simulation is performed according to the following model:

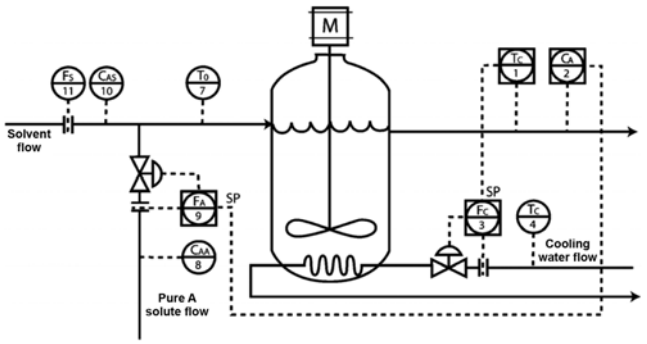

Fig. 4. Diagram of the CSTR process.

$$\frac{dC_A}{dt}=\frac{F}{V}C_{A0}-\frac{F}{V}C_A-k_0e^{-E/(RT)}C_A \tag{17}$$

$$V\rho c_p\frac{dT}{dt}=\rho c_pF(T_0-T)-\frac{aF_C^{b+1}}{F_C+aF_C^b/(2\rho_C c_{pc})}(T-T_{C,in})+(-\Delta H_{rxn})Vk_0e^{-E/(RT)}C_A \tag{18}$$

The process is monitored measuring the cooling water temperature T_C, the inlet temperature T_0, the inlet concentrations C_{AA} and C_{AS}, the solvent flow F_S, the cooling water flow F_C, the outlet concentration C_A, the temperature T, and the reactant flow F_A. The nine variables form the measurement vector [27]

$$\mathbf{x}=[T_C\ T_0\ C_{AA}\ C_{AS}\ F_S\ F_C\ C_A\ T\ F_A]^T \tag{19}$$

The variables are sampled every minute and 500 samples, taken under normal conditions, are used as the training set. Five different faults are introduced into the system. The first four faults are sensor faults similar to the ones shown in [16]. The first simulated fault, Fault 1, is a bias in the sensor of the output temperature T; the bias magnitude is 1(K). Since T is a controlled variable, the effect of the fault will be removed by the PI controller, and its effect will propagate to other variables. This fault is considered a complex fault since it affects several variables. The second and third faults, faults 2 and 3, are biases in the sensors of the inlet temperature T_0 and inlet reactant concentration C_{AA}, respectively; the bias magnitude for T_0 is 1.5(K) and for C_{AA} is 1.0 (kmol/m^3). These faults are considered simple faults since they only affect one variable. Fault 4 is a drift in the sensor of C_{AA} and its magnitude is $dC_{AA}/dt=0.2$ (kmol/(m^3 min)); this is also a simple fault. Fault 5 is a slow drift in the reaction kinetics. The fault has the form of an exponential degradation of the reaction rate caused by catalyst poisoning. In this case, the reaction rate coefficient will change with time as $k_0(t+1)=0.996\times k_0(t)$. This process fault is a complex fault that affects several variables such as the output temperature T, concentration C_A, and the cooling water F_C.

For each of the five scenarios mentioned above, 1000 observations are simulated, and the fault is introduced at the 501-th measurement. The proposed algorithm of fault detection was applied to the five faults. During the KECA transformation, the largest three kernel entropy components preserving 90% Reni entropy are retained. Fig. 5 shows the monitoring performance of KECA on fault 1. The fault 1 data reflected on the first three entropy components in kernel

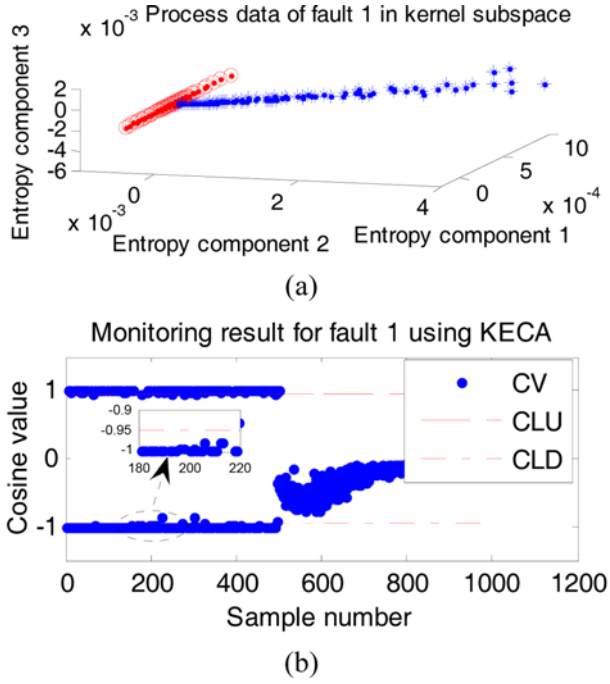

Fig. 5. Monitoring performance for fault 1 using KECA (a) Fault data in feature subspace (b) Fault detection results.

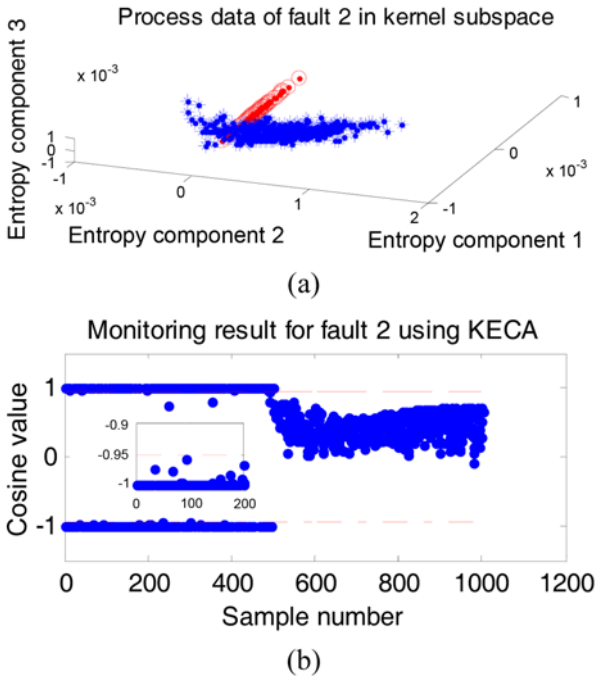

Fig. 6. Monitoring performance for fault 2 using KECA (a) Fault data in feature subspace (b) Fault detection results.

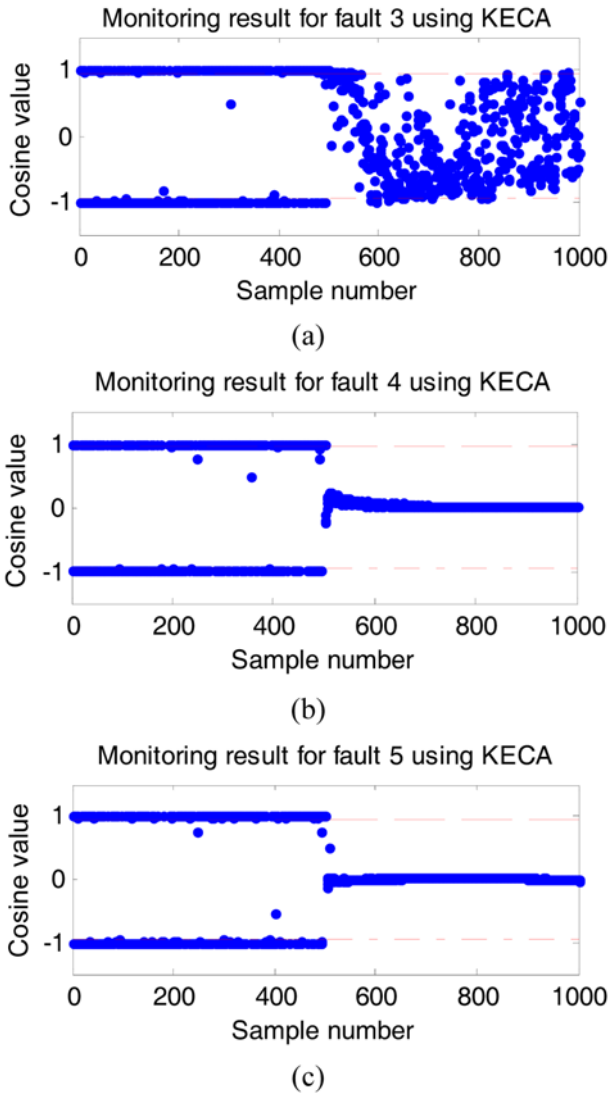

Fig. 7. The fault detection results using KECA for (a) Fault 3 (b) Fault 4 (c) Fault 5.

Table 1. Detection results for the faults using PCA, KPCA and KECA

Fault no.	PCA T^2		KPCA T^2		KECA CV	
	MR	DD	MR	DD	MR	DD
F 1	0	1	0	1	0	1
F 2	0	1	0	1	0	1
F 3	0.116	28	0.06	2	0.05	2
F 4	0	1	0	1	0	1
F 5	0.006	9	0.003	2	0.002	2

feature subspace are illustrated in Fig. 5(a) and the monitoring results with the three entropy components is shown in Fig. 5(b). From Fig. 5(a), we can see that the process data presented an angular structure in the first three components, and it could be seen from the angular monitoring results in Fig. 5(b). Fig. 6(a) shows the angular structure of fault 3 in kernel subspace, and Fig. 6(b) illustrates the fault detection results using KECA. The fault detection results for the other three faults are illustrated in Fig. 7(a), 7(b) and 7(c). From Figs. 5-7, we can see that the faults can be detected successfully.

To compare the performance of KECA monitoring, PCA and KPCA methods were applied to the same data sets. The monitoring results of PCA, KPCA and KECA for the five faults in CSTR are listed in Table 1, including missed detection rates (MR) and detection delays (DD/minutes). From Table 1, we can see that KPCA

and KECA have shown advantages when monitoring fault 3 and fault 5. The KECA is an effective alternative to KPCA in nonlinear chemical processes monitoring.

CONCLUSIONS

This paper presents a useful alternative to traditional multivariate statistical analysis methods such as PCA, KPCA, etc., for nonlinear chemical process monitoring. The kernel entropy component analysis transforms the input data set to preserve the maximum Renyi entropy of the input space data set and provides an angular structure in kernel feature subspace, which is fundamentally different from the kernel PCA. Based on KECA, a fault detection method which monitors the change of the angle-structure in feature subspace is proposed. To best of our knowledge, this is the first time to introduce the KECA into process monitoring and use the angle-structure in feature subspace for fault detection. First, the process data are reflected into kernel feature subspace through KECA transformation; second, the angle between normal condition and current point in feature subspace is calculated to determine whether there is a fault in the process or not. The proposed method is applied to both a numerical process and the CSTR process, and the monitoring results indicate that it is effective. Moreover, we note that the use of KECA may be explored in any process monitoring method previously based on kernel PCA and this is a task for future work. Furthermore, future work could also be focused on fault diagnosis using KECA and angular structure.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support from the following foundations: 973 project of China (2013CB733600), National Natural Science Foundation of China (21176073), Program for New Century Excellent Talents in University (NCET-09-0346), "Shu Guang" project (09SG29) and the Fundamental Research Funds for the Central Universities.

NOMENCLATURE

C_A : outlet concentration in CSTR
C_{AA} C_{AS} : inlet concentrations in CSTR
CL : the confidence limits of CV
CV : Cosine value between vectors
D : eigenvalue matrix
d : number of the variables
E : eigenvector matrix
e : eigenvector of kernel matrix
F_A : reactant flow in CSTR
F_C : cooling water flow in CSTR
F_S : solvent flow in CSTR
f : kernel estimator in kernel density estimation
h : the window width in kernel estimator
K : kernel matrix in kernel transformation
K : kernel function in kernel density estimation
k_σ : Parzen window in kernel transformation
m : data vector in kernel feature subspace
N : number of the observations
P_u : kernel projection
s : number of kernel entropy components
T : temperature in CSTR
T_C : cooling water temperature in CSTR
T_0 : inlet temperature in CSTR
U_s : subspace spanned by the *s* KEC axes
X : process sample set
λ : eigenvalue of kernel matrix
σ : width parameter in Parzen window
$\varepsilon 1$, $\varepsilon 2$: independent noise variables

REFERENCES

1. J. E. Jackson, *Technometrics*, **1**, 359 (1959).
2. S. J. Qin, *J. Chemometr.*, **17**, 480 (2003).
3. J. V. Kresta, J. F. Macgregor and T. E. Marlin, *Can. J. Chem. Eng.*, **69**, 35 (2009).
4. J. M. Lee, C. K. Yoo, S. W. Choi, P. A. Vanrolleghem and I. B. Lee, *Chem. Eng. Sci.*, **59**, 223 (2004).
5. X. Liu, L. Xie, U. Kruger, T. Littler and S. Wang, *AIChE J.*, **54**, 2379 (2008).
6. P. Nomikos and J. F. MacGregor, *AIChE J.*, **40**, 1361 (1994).
7. X. Pei, Y. Yamashita, M. Yoshida and S. Matsumoto, *JCEJ*, **41**, 25 (2008).
8. M. H. Kim and C. K. Yoo, *Korean J. Chem. Eng.*, **25**, 947 (2008).
9. K. Han, K. J. Park, H. Chae and E. S. Yoon, *Korean J. Chem. Eng.*, **25**, 13 (2008).
10. L. H. Chiang, E. Russell and R. D. Braatz, *Fault detection and diagnosis in industrial systems*, Springer Verlag (2001).
11. Z. Ge, C. Yang and Z. Song, *Chem. Eng. Sci.*, **64**, 2245 (2009).
12. M. A. Kramer, *AIChE J.*, **37**, 233 (1991).
13. D. Dong and T. J. McAvoy, *Comput. Chem. Eng.*, **20**, 65 (1996).
14. P. Cui, J. Li and G. Wang, *Expert Syst. Appl.*, **34**, 1210 (2008).
15. F. Jia, E. Martin and A. Morris, *Int. J. Syst. Sci.*, **31**, 1473 (2000).
16. S. W. Choi, C. Lee, J. M. Lee, J. H. Park and I. B. Lee, *Chemom. Intell. Lab. Syst.*, **75**, 55 (2005).
17. V. H. Nguyen and J. C. Golinval, *Eng. Struct.*, **32**, 3683 (2010).
18. B. Schölkopf, A. Smola and K. R. Müller, *Neural Computation*, **10**, 1299 (1998).
19. R. Jenssen, *IEEE PAMI.*, **32**, 847 (2010).
20. R. Jenssen and T. Eltoft, *Neurocomputing*, **72**, 23 (2008).
21. A. Renyi, *On measures of entropy and information*, in Proc. Fourth Berkeley Symp. on Math. Statist. and Prob., Univ. of Calif. Press, **1**, 547 (1961).
22. E. Parzen, *The Annals of Mathematical Statistics*, **33**, 1065 (1962).
23. K. Dehnad, *Technometrics*, **29**, 495 (1987).
24. D. W. Scott, *Density estimation*, Encyclopedia of Biostatistics (2005).
25. A. R. Webb, *Statistical pattern recognition*, Wiley (2003).
26. S. Yoon and J. F. MacGregor, *J. Process Control*, **11**, 387 (2001).
27. H. H. Yue and S. J. Qin, *Ind. Eng. Chem. Res.*, **40**, 4403 (2001).