Sensor Fault Detection and Identification using Kernel PCA and Its Fast Data Reconstruction

PENG Hong-xing, WANG Rui, HAI Lin-peng

School of Computer Science & Technology，Henan Polytechnic University，Jiaozuo Henan 454000
E-mail: scipaper@foxmail.com

Abstract: In this paper, a novel sensor fault detection and identification technique based on kernel principal component analysis (KPCA) and its fast data reconstruction is presented. Although it has been proved that KPCA shows a better performance for sensor fault detection, the fault identification method has rarely been developed. Using the fast data reconstruction based on distance constraint, we employ the residuals of variables to identify the faulty sensor. Since the proposed method does not include iterative calculation, it has a lower calculation burden and is more suitable for online application. The simulation results show that the proposed method effectively identifies the source of typical sensor faults.

Key Words: Kernel principal component analysis; Sensor fault detection and identification; Data reconstruction; Distance constraint

1 INTRODUCTION

In industrial processes, accurate measurements are necessary in providing controllers and operators with a view of the system status. Sensor readings play a key role in assessments of the system status. The safety, reliability and performance of complex systems with many sensors are largely dependent on the accuracy and reliability of the sensors. However, the sensors are not significantly more reliable than the systems being monitored, the indication of an abnormal state may be the result of a sensor failure rather than a system failure. Failure to identify the source of the indication of an “abnormal state” and take appropriate corrective actions could result in expensive and unnecessary shutdowns or, worse still, accidents that endanger both system and personal safety. Thus, it is quite important for monitoring and diagnosis of sensors fault.

Principal component analysis (PCA) the most widely used data-driven technique for sensor fault diagnosis [1]. It divides data in formation into the significant patterns, such as linear tendencies or directions in model subspace, and the uncertainties, such as noises or outliers located in residual subspace. Two different statistics, Hotelling T^2 and Q, represented by Mahalanobis and Euclidean distances, are used to elucidate the pattern variations in the model and residual subspaces, respectively [2]. PCA methodology in combination with T^2 statistics, Q statistics and contribution plots has been applied in several fields [3, 4].

However, for some complicated cases in industrial chemical and biological processes with particularly nonlinear characteristics, PCA performs poorly due to its assumption that the process data are linear [5]. To handle the problem posed by nonlinear data, several nonlinear PCA, such as nonlinear PCA method based on auto-associative neural networks [6], nonlinear PCA approach based on principal curves and neural networks [5], etc. had been developed in recent years. But these methods employ neural networks to deal with nonlinear characteristics and have some detectives including the determination of the number of nodes and layers and approximation for the nonlinear function [7].

kernel PCA (KPCA) is a novel nonlinear PCA technique which is developed in recent years. It can efficiently compute principal components (PCs) in high-dimensional feature spaces by means of integral operators and nonlinear kernel functions. The basic idea of KPCA is to first map the input space into a feature space via nonlinear mapping and then to compute the PCs in that feature space. Compared to the above nonlinear methods, the main advantage of KPCA is that it does not involve nonlinear optimization[8]. In addition, KPCA does not require that the number of components to be extracted be specified prior to modeling. Due to these merits, KPCA has shown better performance than linear PCA in process monitoring [2, 7] and sensor fault detection [9] in nonlinear systems.

Fault identification is a key step of KPCA application in fault diagnosis. Choi(2004) developed a robust reconstruction-based fault identification method for KPCA-based monitoring, which has been shown to be useful for discriminating process faults. The robust data reconstruction algorithm he used [10] contributes a more precise result. However, its iteration scheme is numerically unstable and one has to try a number of guesses for initial values[11]. Adopting the derivative of kernel function proposed by Rakotomamonjy, Cho(2004) propose a method for sensor fault identification by defining the contribution of each variable to the monitoring statistics, Hotelling's T^2 and squared prediction error (SPE), known as Q statistics. There are a lot of complicated kernel functions derivative calculation when one obtains the contribution in this method. It is not convenient to perform and has no sensor data reconstruction.

In this study, we propose a new sensor fault detection and

This work is supported by Doctoral Fund of Henan Polytechnic University

identification technique based on KPCA and its fast data reconstruction. The monitoring statistics, T^2 and Q statistics, for fault detection are calculated in feature space, while the fault identification is accomplished in input space. The rest of this paper is organized as follows.. In Section 2, the online sensor fault detection scheme of KPCA is presented. Sensor data reconstruction and fault identification is described in detail in Section 3. In Section 4, a typical simulation example is put forward. Finally, we present our conclusions in Section 5.

2 ONLINE SENSOR FAULT DETECTION SCHEME OF KPCA

The fault detection procedure of KPCA is accomplished by two monitoring statistics, Hotelling T^2 and SPE, in the feature space. A measure of the variation of KPCA model is given by the T^2 statistics. T^2 is the sum of the nomalized squared scores. For a new observation $\boldsymbol{x}_{new}$, it is defined as

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

Where $\boldsymbol{t}_{new}$ denotes the corresponding PCs of $\boldsymbol{x}_{new}$. Λ^{-1} is the diagonal matrix of the eigenvalues associated with the retained PCs. The control limit for T^2 is obtained using the F-distribution [7]

$$T_{p,N,\alpha}^2 \sim \frac{p(N-1)}{N-p}F_{p,N-1,\alpha} \tag{2}$$

Where $F_{p,N-1,\alpha}$ represents a F-distribution with the degree of freedom of p and $N-1$ with the level of significance $100(1-\alpha)\%$; N is the number of samples in the model and p is the number of PCs.

Q statistics implies the Euclidean distance of an observation from the model space [9]. In this paper, we adopt a simple calculation of Q statistics proposed by Lee (2004). The conceptual framework of the calculation for SPE in KPCA method is shown schematically in Fig.1.

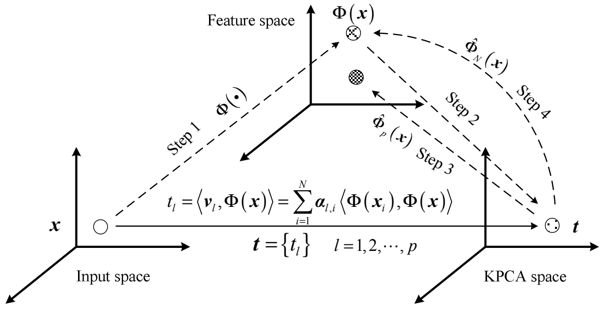
Feature space
$\Phi(\boldsymbol{x})$
$\hat{\Phi}_N(\boldsymbol{x})$
Step 4
Step 1
$\Phi(\bullet)$
Step 2
$\hat{\Phi}_p(\boldsymbol{x})$
Step 3
$t_l = \langle \boldsymbol{v}_l, \Phi(\boldsymbol{x})\rangle = \sum_{i=1}^{N}\boldsymbol{\alpha}_{l,i}\langle \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x})\rangle$
$\boldsymbol{x}$
$\boldsymbol{t}$
$\boldsymbol{t} = \{t_l\}$ $l = 1,2,\cdots,p$
Input space
KPCA space

Fig.1 Conceptual diagram for SPE calculation

First, for an observation vector $\boldsymbol{x}$ KPCA performs a nonlinear mapping $\Phi(\bullet)$ from an input space to a feature space(Step 1). Then, a linear PCA is performed in this high dimensional space, which gives rise to a score vector $\boldsymbol{t}$ in a lower p-dimensional KPCA space(Step 2). In order to compute SPE in feature space, we need to reconstruct a feature vector $\Phi(\boldsymbol{x})$ from $\boldsymbol{t}$. This is done by projecting $\boldsymbol{t}$ into the feature space via eigenvectors $\boldsymbol{v}$ (Step 3). Thus the reconstructed feature vector can be written as

$$\hat{\Phi}_p(\boldsymbol{x}) = \sum_{l=1}^{p} t_l \boldsymbol{v}_l, \tag{3}$$

Therefore, SPE in F is defined as

$$Q = \left\|\Phi(\boldsymbol{x}) - \hat{\Phi}_p(\boldsymbol{x})\right\|^2 \tag{4}$$

Note that $\Phi(\boldsymbol{x})$ is identical to $\hat{\Phi}_N(\boldsymbol{x}) = \sum_{l=1}^{N} t_l \boldsymbol{v}_l$ (Step 4). Hence, Eq.(4) can be rewritten as

$$\begin{aligned} Q &= \left\|\hat{\Phi}_N(\boldsymbol{x}) - \hat{\Phi}_p(\boldsymbol{x})\right\|^2 \\ &= \hat{\Phi}_N(\boldsymbol{x})^{\mathrm{T}}\hat{\Phi}_N(\boldsymbol{x}) - 2\hat{\Phi}_N(\boldsymbol{x})^{\mathrm{T}}\hat{\Phi}_p(\boldsymbol{x}) + \hat{\Phi}_p(\boldsymbol{x})^{\mathrm{T}}\hat{\Phi}_p(\boldsymbol{x}) \\ &= \sum_{j=1}^{N} t_j \boldsymbol{v}_j^{\mathrm{T}} \sum_{l=1}^{N} t_l \boldsymbol{v}_l^{\mathrm{T}} - 2\sum_{j=1}^{N} t_j \boldsymbol{v}_j^{\mathrm{T}} \sum_{l=1}^{p} t_l \boldsymbol{v}_l^{\mathrm{T}} + \sum_{j=1}^{p} t_j \boldsymbol{v}_j^{\mathrm{T}} \sum_{l=1}^{p} t_l \boldsymbol{v}_l^{\mathrm{T}} \\ &= \sum_{j=1}^{N} t_j^2 - 2\sum_{j=1}^{p} t_j^2 + \sum_{j=1}^{p} t_j^2 = \sum_{j=1}^{N} t_j^2 - \sum_{j=1}^{p} t_j^2 \end{aligned} \tag{5}$$

Where $\boldsymbol{v}_j^{\mathrm{T}}\boldsymbol{v}_l = 1$ when $j = l$, $\boldsymbol{v}_j^{\mathrm{T}}\boldsymbol{v}_l = 0$ otherwise.

For a new observation $\boldsymbol{x}_{new}$ of online detection, the Q statistics is calculated as follow s

$$Q_{new} = \sum_{j=1}^{N} t_{new,j}^2 - \sum_{j=1}^{p} t_{new,j}^2 \tag{6}$$

Where $\boldsymbol{t}_{new}$ denotes the PCs of $\boldsymbol{x}_{new}$, $t_{new,j}$ is the jth element of $\boldsymbol{t}_{new}$.The control limits of Q statistics can be computed from its appropriate distribution

$$Q_\alpha \sim g\chi_h^2 \tag{7}$$

Where g is a weighting parameter included to account for the magnitude of SPE and h is the degree of freedom. If a and b denote the estimated mean and variance of the SPE in KPCA modeling samples set respectively, then g and h can be calculated by

$$g = b/2a,\ h = 2a^2/b \tag{8}$$

The online detection procedure is described in detail as follows.

Step 1 Obtain an observation and scale it with the mean and variance from the modeling sample set to get the new vector $\boldsymbol{x}_{new} \in \mathrm{R}^M$;

Step 2 Compute the kernel vector $\boldsymbol{k}_{new} \in \mathrm{R}^{1\times N}$, the ith element of $\boldsymbol{k}_{new}$ is given by

$$\boldsymbol{k}_{new,i} = k(\boldsymbol{x}_{new}, \boldsymbol{x}_i) \tag{9}$$

Where $\boldsymbol{x}_i$ is the ith sample of the modeling dataset, $i = 1,\cdots,N$.

Step 3 Calculate the mean centered kernel vector as follows

$$\tilde{\boldsymbol{k}}_{new} = \boldsymbol{k}_{new} - 1_{new}\mathbf{K} - \boldsymbol{k}_{new}1_N + 1_{new}\mathbf{K}1_N \tag{10}$$

Where $1_{new} = (1/N)[1,1,\cdots,1] \in \mathrm{R}^{1\times N}$.

Step 4 Extract the nonlinear PCs of $\boldsymbol{x}_{new}$ as follows

$$\boldsymbol{t}_{new} = \left[t_{new,1}, t_{new,2}, \cdots, t_{new,p}\right] \tag{11}$$

$$
\begin{aligned}
t_{new,l} &= \left\langle \boldsymbol{v}_l, \tilde{\Phi}(\boldsymbol{x}_{new}) \right\rangle \\
&= \sum_{i=1}^{N} \boldsymbol{\alpha}_{l,i} \left\langle \tilde{\Phi}(\boldsymbol{x}_i), \tilde{\Phi}(\boldsymbol{x}_{new}) \right\rangle \\
&= \sum_{i=1}^{N} \boldsymbol{\alpha}_{l,i} \tilde{\boldsymbol{k}}_{new}(\boldsymbol{x}_i, \boldsymbol{x}_{new})
\end{aligned} \tag{12}
$$

Where $\tilde{\Phi}(\bullet)$ represents mean centered feature vector and $l = 1, 2, \cdots, p$.

Step 5 Calculate the two statistics, T^2_{new} and Q_{new}, and judge whether either of them exceed its control limit.

3 SENSOR DATA RECONSTRUCION AND FAULT IDENTIFICATION

It is more practical to reconstruct sensor data based on the KPCA model as soon as possible after a fault is detected. Residuals between reconstruction data and measurements include mainly white noise under normal operation situation and its statistical characteristic changes when there is a sensor fault. Hence, residual can be used to identify the source of a fault.

In essence, sensor data reconstruction is the same as to find the pre-image of a feature vector using KPCA model. However, the exact pre-image often does not exist and an appropriate solution can be obtained. This is called as data reconstruction in the following in this paper.

3.1 Data reconstruction based on distance constraint

3.1.1 Basic idea of fast data reconstruction

For any two sample vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, we can obtain their Euclidean distance $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Similarly, we can also obtain the feature space distance $\tilde{d}[\Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_j)]$. For many commonly used kernel, there exists a simple relationship between $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $\tilde{d}[\Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_j)]$ [14]. The basic idea of data reconstruction based on distance constraint is as follows(Fig.2).

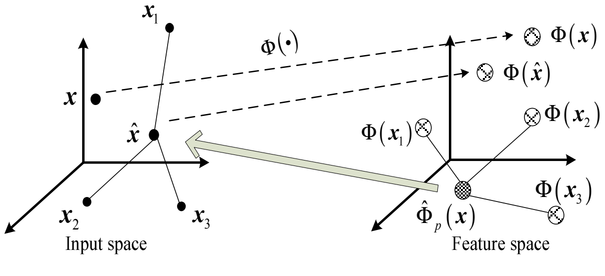

Fig.2 Schematic diagram for data reconstruction based on distance constraint

For each training sample $\boldsymbol{x}_i$, there is a distance between $\Phi(\boldsymbol{x}_i)$ and $\hat{\Phi}_p(\boldsymbol{x})$. Using the simple relationship mentioned before, we can obtain the corresponding input space distance between $\hat{\boldsymbol{x}}$ and each $\boldsymbol{x}_i$. Then using the Metric Multidimensional Scaling idea [14], one can embed $\hat{\Phi}_p(\boldsymbol{x})$ to input space. When the exact $\hat{\boldsymbol{x}}$ exists, it would have exactly satisfied these input space distance constraints. In most cases, the exact $\hat{\boldsymbol{x}}$ does not exist, we will require the solution $\hat{\boldsymbol{x}}$ to satisfy these constraints approximately, to be more precise, in the least-square sense.

3.1.2 Distance in the feature space

For any two sample vectors $\boldsymbol{x}$ and $\boldsymbol{x}_i$, the squared feature space distance between $\hat{\Phi}_p(\boldsymbol{x})$ and $\Phi(\boldsymbol{x}_i)$ is given by:

$$
\begin{aligned}
\tilde{d}\left[\hat{\Phi}_p(\boldsymbol{x}), \Phi(\boldsymbol{x}_i)\right] = \left\|\hat{\Phi}_p(\boldsymbol{x})\right\|^2 + \left\|\Phi(\boldsymbol{x}_i)\right\|^2 \\
-2\hat{\Phi}_p(\boldsymbol{x})^{\mathrm{T}} \Phi(\boldsymbol{x}_i)
\end{aligned} \tag{13}
$$

From Eq.(31), we have

$$\left\|\hat{\Phi}_p(\boldsymbol{x})\right\|^2 = \left(\sum_{l=1}^{p} t_l \boldsymbol{v}_l\right)^{\mathrm{T}} \left(\sum_{l=1}^{p} t_l \boldsymbol{v}_l\right) = \sum_{l=1}^{p} t_l^2$$

And

$$\hat{\Phi}_p(\boldsymbol{x})^{\mathrm{T}} \Phi(\boldsymbol{x}_i) = \left(\sum_{l=1}^{p} t_l \boldsymbol{v}_l\right)^{\mathrm{T}} \Phi(\boldsymbol{x}_i) = \left(\sum_{l=1}^{p} t_l \boldsymbol{v}_l\right)^{\mathrm{T}} \sum_{l=1}^{N} t_{i,l} \boldsymbol{v}_l$$

Where t_l is the lth PCs of $\boldsymbol{x}$ and $t_{i,l}$ is the lth of $\boldsymbol{x}_i$. Thus, Eq.(13) becomes

$$
\begin{aligned}
&\tilde{d}\left[\hat{\Phi}_p(\boldsymbol{x}), \Phi(\boldsymbol{x}_i)\right] \\
&= \sum_{l=1}^{p} t_l^2 - 2\left(\sum_{l=1}^{p} t_l \boldsymbol{v}_l\right)^{\mathrm{T}} \sum_{l=1}^{N} t_{i,l} \boldsymbol{v}_l + \tilde{k}(\boldsymbol{x}_i, \boldsymbol{x}_i)
\end{aligned} \tag{14}
$$

3.1.3 Distance in the input space

Because the distances with the nearest neighbors are the most important in determining the location of any point. Hence, we will only consider the squared input space distances between $\hat{\Phi}_p(\boldsymbol{x})$ and its n nearest neighbors in the following, i.e.

$$\boldsymbol{d}^2 = \left[d_1^2, d_2^2, \cdots, d_n^2\right]^{\mathrm{T}} \tag{15}$$

Consider isotropic kernels of the form $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = k\left(\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2\right)$, which only depends on the distance $\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2$. According to the simple relationship between the feature space distance $\tilde{d}_{ij}$ and the input distance d_{ij} [14],

$$
\begin{aligned}
\tilde{d}_{ij}^2 = \tilde{d}(\boldsymbol{x}_i, \boldsymbol{x}_j) = K_{ii} + K_{jj} - 2k\left(\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2\right) \\
= K_{ii} + K_{jj} - 2k\left(d_{ij}^2\right)
\end{aligned} \tag{16}
$$

Where $K_{ii} = k(\boldsymbol{x}_i, \boldsymbol{x}_i)$, and hence

$$k\left(d_{ij}^2\right) = \frac{1}{2}\left(K_{ii} + K_{jj} - \tilde{d}_{ij}^2\right) \tag{17}$$

Generally, $k(\bullet)$ is invertible. For the Gaussian kernel Eq.(17), we have

$$d_{ij}^2 = -\left(2\sigma^2\right)\log\left[\frac{1}{2}\left(K_{ii} + K_{jj} - \tilde{d}_{ij}^2\right)\right] \tag{18}$$

3.1.4 Data reconstruction using distance constraints

Suppose the corresponding n neighbors in input space be $\{\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n\}$, we will first center them in their centroid $\bar{\boldsymbol{x}}=(1/n)\sum_{i=1}^{n}\boldsymbol{x}_i$ and define a coordinate system in their span.

First, construct $m\times n$ matrix $\mathbf{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n]$ and let $\bar{\mathbf{X}}$ be the corresponding centered matrix (i.e., the column of $\bar{\mathbf{X}}$ are zero). Assuming that the sample dataset spans a q-dimensional space (i.e., $\mathbf{X}$ is of rank q), we can obtain the singular value decomposition of the $q\times n$ matrix $\bar{\mathbf{X}}$ as

$$\bar{\mathbf{X}}=\mathbf{U\Lambda V}^{\mathrm{T}}=\mathbf{UZ} \tag{19}$$

Where $\mathbf{U}=[\boldsymbol{e}_1,\cdots,\boldsymbol{e}_q]$ is a $d\times q$ matrix with orthonormal columns $\boldsymbol{e}_j$, $j=1,\cdots,q$, and $\mathbf{Z}=[\boldsymbol{z}_1,\cdots,\boldsymbol{z}_q]$ is a $q\times n$ matrix with columns $\boldsymbol{z}_i$ being the projections of $\boldsymbol{x}_i$ onto $\boldsymbol{e}_j$, $i=1,\cdots,n$. Thus, the squared distance of $\boldsymbol{x}_i$ to the origin is equal to $\|\boldsymbol{z}_i\|^2$. Collecting them into a n-dimensional vector, we obtain

$$\boldsymbol{d}_0^2=\left[\|\boldsymbol{z}_1\|^2,\cdots,\|\boldsymbol{z}_n\|^2\right] \tag{20}$$

Because the reconstruction vector $\hat{\boldsymbol{x}}$ is in the span of the n neighbors, we can obtain the location of $\hat{\boldsymbol{x}}$ (specifically, coordinates of $\hat{\boldsymbol{x}}$) by requiring $d^2(\hat{\boldsymbol{x}},\boldsymbol{x}_i)$ to be close to the d_i^2 as possible, i.e.,

$$d^2(\hat{\boldsymbol{x}},\boldsymbol{x}_i)=d_i^2,\ i=1,\cdots,n \tag{21}$$

Generally, the least square solution $\hat{\boldsymbol{z}}$ for $\hat{\boldsymbol{x}}$ can be obtained by satisfying [15]

$$-2\mathbf{Z}^{\mathrm{T}}\hat{\boldsymbol{z}}=(\boldsymbol{d}^2-\boldsymbol{d}_0^2)-\frac{1}{n}\boldsymbol{1}\boldsymbol{1}^{\mathrm{T}}(\boldsymbol{d}^2-\boldsymbol{d}_0^2) \tag{22}$$

Where $\boldsymbol{1}=[1,1,\cdots,1]^{\mathrm{T}}\in\mathrm{R}^{N\times 1}$. Because of the centering, $\mathbf{Z}\boldsymbol{1}\boldsymbol{1}^{\mathrm{T}}=\mathbf{0}$, and the least square solution $\hat{\boldsymbol{z}}$ is given by

$$\hat{\boldsymbol{z}}=-\frac{1}{2}(\mathbf{ZZ}^{\mathrm{T}})\mathbf{Z}(\boldsymbol{d}^2-\boldsymbol{d}_0^2)=-\frac{1}{2}\Lambda^{-1}\mathbf{V}^{\mathrm{T}}(\boldsymbol{d}^2-\boldsymbol{d}_0^2) \tag{23}$$

Here, $\hat{\boldsymbol{z}}$ is represented in terms of the coordinate system defined by $\{\boldsymbol{e}_j\}$. Transforming back to the original coordinate system we have

$$\hat{\boldsymbol{x}}=\mathbf{U}\hat{\boldsymbol{z}}+\bar{\boldsymbol{x}} \tag{24}$$

3.2 Online reconstruction procedure

Let $\boldsymbol{x}_{new}$ be the new sample vector and $\hat{\boldsymbol{x}}_{new}$ be the reconstruction vector to be determined. The online reconstruction procedure is described in detail as follows.

Step 1 Find the n nearest neighbors of $\hat{\boldsymbol{\Phi}}_p(\boldsymbol{x}_{new})$ in the feature space according to the KPCA model;

Step 2 Compute n squared feature space $\tilde{d}_i^2$ and transform them to d_i^2 using Eq.(18);

Step 3 Determine the n samples in input space based on the n nearest neighbors in feature space and construct matrix $\mathbf{X}$. After centering and singular value decomposition, we can obtain $\boldsymbol{d}_0^2$ using Eq.(20);

Step 4 calculate $\hat{\boldsymbol{z}}_{new}$ using Eq.(23), then transform it to $\hat{\boldsymbol{x}}_{new}$.

4 SIMULATION AND DISCUSSION

In this section, the KPCA sensor fault detection and identification based on fast data reconstruction is applied in a simple example. The iterative data reconstruction and the proposed method are compared for two cases. is selected as kernel function in this paper.

When developing the KPCA model, we must determine the number of PCs and the parameters of kernel function, i.e., σ for this study, at the same time. Our approach is as follows. To begin with, we empirically determine the number of PCs, p, in an interval $(5\%\sim 15\%)N$. Then find an appropriate σ with which the cumulative sum of the first p eigenvalues is above 95% of the sum of all eigenvalues. If such σ cannot be obtained, p is modified and the above procedure is performed again. Although this is a typically empirical approach, it is shown convenient and practical by the following experiments.

We adopt a system with 5 variables but only on factor, which is originally suggested by Dong and McAvoy [5] and has been slightly modified.

$$\begin{cases} x_1=t+\xi_1 \\ x_2=t^2-3t+\xi_2 \\ x_3=-t^3+3t^2+\xi_3 \\ x_4=\dfrac{1}{2}x_2-t^{\frac{1}{2}}+4+\xi_4 \\ x_5=x_4-x_3+t^2+\xi_5 \end{cases} \tag{25}$$

Where $\xi_i\sim N(0,0.01)$, $i=1,2,\cdots,5$, are independent noise variables and $t\in[0.01,1]$. Simulation data generated by these equations are scaled to zero mean and unit variance. Normal data set including 100 samples is used to build the PCA and KPCA model. One PC is selected to model the linear PCA due to one factor. According to the above method, 10 PCs are chosen for KPCA and then σ is determined to 0.5, which makes the cumulative sum of the first 10 eigenvalues is above 95%.

Two test data set comprising 300 samples each are also generated. The following two typical faults are added separately during the generation of the two sets.

Scenario 1: A step change of x_2 by -0.35 is introduced starting from sample 101 to simulate the step bias fault of sensor.

Scenario 1: x_2 is linearly increased from sample 101 to 208 by adding $0.005(k-100)$ to the value of each sample to simulate the zero drift fault, where k is sample number.

The T^2 and Q charts of PCA fault detection for scenario 1 are shown in Fig.4 (a). The 95% control limits are also plotted in this figure. Obviously, PCA does not detect the step fault. However, KPCA gives relatively satisfactory results in comparison to PCA, as shown in fig.4(b). Although no fault is detected from sample 101 to the end in the T^2 chart, Q chart surpasses the control limit from sample 101. Moreover, almost no sample among the first 100 normal operation data exceeds the control limit, indicating that a $g\chi_h^2$ distribution provides a good approximation for the 95% control limit of the Q chart.

The T^2 and Q charts of PCA fault detection for scenario 2 are shown in Fig.5 (a). The T^2 chart does not detect any abnormalities whereas the Q chart detects fault from about sample 185. The Q value rapidly decreases to normal operation level after the zero drift fault disappears at sample 280. In contrast to the PCA simulation results, both the T^2 and Q charts of KPCA detect the zero drift fault, as shown in Fig.5 (b). In addition, the KPCA charts detect the zero drift fault earlier than does the Q charts of PCA. Specifically, in the KPCA detection, the zero drift fault is detected from about sample 145 in the Q chart, implying that KPCA method detects such fault earlier than the PCA method. Moreover, the Q value also rapidly decreases to normal operation level after the fault disappears at sample 280.

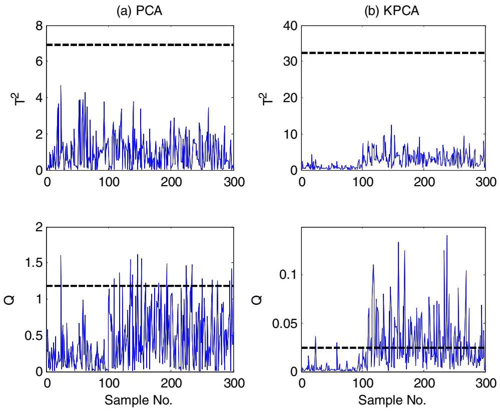

Fig.4 (a) PCA detection charts
(a) KPCA detection charts with scenario 1

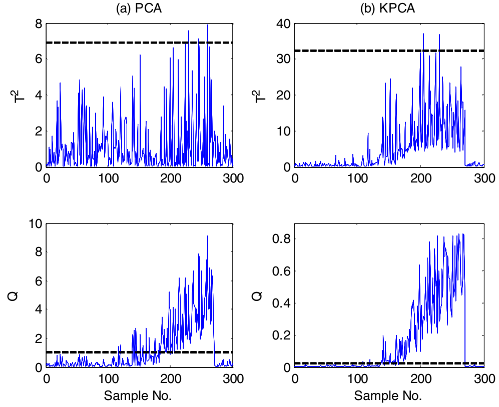

Fig.5 (a) PCA detection charts
(b) KPCA detection charts with scenario 2

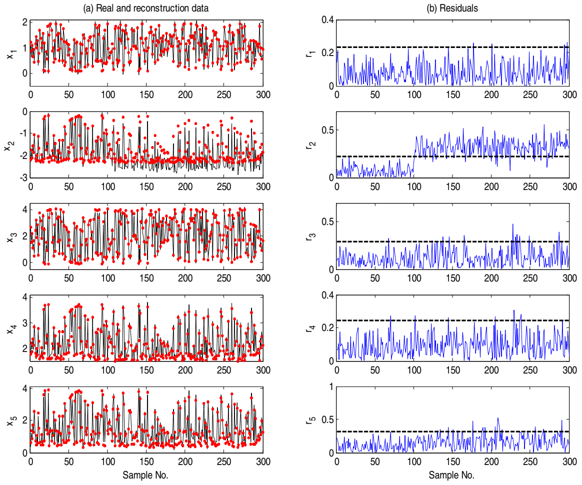

Fig.7 (a) Real and reconstruction data
(b) residuals for scenario 1 (simple example)

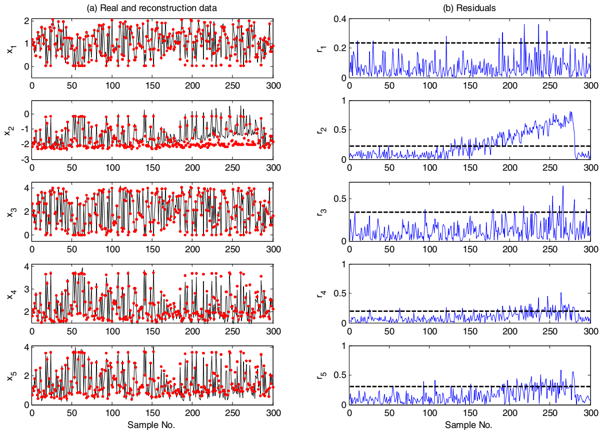

Fig.8 (a) Real and reconstruction data
(b) residuals for scenario 2 (simple example)

Fig.7 and Fig.8 shows the data reconstruction and fault identification results of KPCA for the two fault respectively. The real data and corresponding reconstruction data (dots) of five variables are plotted in Fig.7 (a) and the residuals are plotted in Fig.7 (b). The thresholds of residuals for various variables are determined by the following method. Suppose μ_i and δ_i are the mean and standard variance of a residual r_i , $i=1,2,\cdots,5$, then the ith threshold for fault identification is defined as

$$th_i = \mu_i + 3\delta_i \tag{26}$$

The five thresholds (dotted line) are also plotted in Fig.7 (b). It is clearly that only the residual of x_2, r_2 exceeds its threshold after sample 101, indicating that the fault variable is x_2, as we assumed exactly.

Fig.8 is quite similar to Fig.7 and shows that x_2 is the faulty one. From Fig.8 (b), we can see that other reconstruction residuals slightly surpass their thresholds around sample 250. That is because the reconstructions of other variables are affected by the faulty variable. However, we can definitely conclude the faulty variable from an overall perspective. In addition, the residual r_2 rapidly decreases to normal operation level after the zero drift fault is stopped at sample 280.

The same experiments are also carried out by using Takahashi's method [10]. The comparison results between the proposed method and Takahashi's method are shown in Table 1. We compare the reconstruction accuracy and speed by adopting Mean Square Error, Maximum Relative Error and Running Time of 300 samples with step bias fault. For simplicity, only one variable (x_1) is adopted here.

Table 1 Results comparison for two methods

	The proposed method	Takahashi's method
MSE	0.152	0.142
MRE	12%	7%
Running Time	1.983s	4.745s

Table 1 shows that Takahashi's method has a higher reconstruction accuracy than the proposed method. However, the former spends more than double times as the later in accomplishing the 300 samples reconstruction because of its iterative calculation. Therefore, the proposed method has a better performance for online identification.

5 Conclusions

In this research, we propose a new sensor fault detection and identification approach using KPCA and its fast data reconstruction. KPCA has shown its better performance in sensor fault diction and process monitoring, however, appropriate fault identification scheme has barely found. We develop a new sensor fault identification method based on fast data reconstruction of distance constraint. It does not require iterative calculation and in turn alleviate the reconstruction calculation burden. As shown in the simulation study, the proposed method gives satisfactory results for two typical sensor faults, step bias fault and zero drift fault. Furthermore, it can effectively discriminate a faulty sensor at a relatively higher speed.

REFERENCES

[1] Dunia R, Qin S J, Edgar T F, et al. Identification of faulty sensors using principal component analysis[J]. AIChE Journal. 1996, 42(10): 2797-2812.

[2] Choi S W, Lee C, Lee J M, et al. Fault detection and identification of nonlinear processes based on kernel PCA[J]. Chemometrics and Intelligent Laboratory Systems. 2005, 75(1): 55-67.

[3] Lennox J, Rosen C. Adaptive multiscale principal components analysis for online monitoring of wastewater treatment[J]. Water Science and Technology. 2002, 45(4-5): 227-235.

[4] Xiao F, Wang S, Xu X, et al. An isolation enhanced PCA method with expert-based multivariate decoupling for sensor FDD in air-conditioning systems[J]. Applied Thermal Engineering. 2009(29): 712-722.

[5] Dong D, Mcavoy T J. Nonlinear principal component analysis--Based on principal curves and neural networks[J]. Computers & Chemical Engineering. 1996, 20(1): 65-78.

[6] Mark A K. Nonlinear principal component analysis using autoassociative neural networks[J]. AIChE Journal. 1991, 37(2): 233-243.

[7] Lee J M, Yoo C K, Choi S W, et al. Nonlinear process monitoring using kernel principal component analysis[J]. Chemical Engineering Science. 2004, 59(1): 223-234.

[8] Scholkopf B, Smola A, Klaus-robertMüller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem[J]. Neural Computation. 1998, 10(5): 1299-1399.

[9] Cho I, Lee J, Choi S W, et al. Sensor fault identification based on kernel principal component analysis[C]. Taipei,Taiwan: 2004.

[10] Takahashi T, Kurita T. Robust De-noising by Kernel PCA[J]. Lecture Notes in Computer Science. 2002, 2415: 739-744.

[11] Kwok J T, Tsang I W. The pre-image problem in kernel methods[J]. IEEE Transactions On Neural Networks. 2004, 15(6): 1517-1525.

[12] Christianini N, Shawe-taylor J. An introduction to support vector machines and other kernel-based learning methods[M]. UK: Cambridge University Press, 2000.

[13] Mika S, Scholkopf B, Smola A, et al. Kernel PCA and de-noising in feature spaces[C]. MIT Press, 1999.

[14] Williams C K I. On a Connection between Kernel PCA and Metric Multidimensional Scaling[J]. Machine Learning. 2002, 46(1-3): 11-19.

[15] Gower J C. Adding a point to vector diagrams in multivariate analysis[J]. Biometrika. 1968, 55(3): 582-585.