Article

A Multi-Fault Diagnosis Method for Sensor Systems Based on Principle Component Analysis

Daqi Zhu [1],*****, **Jie Bai** [1] **and Simon X. Yang** [2]

[1] Laboratory of Underwater Vehicles and Intelligent Systems, Shanghai Maritime University, Shanghai, 200135, China

[2] The Advanced Robotics and Intelligent Systems Laboratory, School of Engineering, University of Guelph, Guelph, ON. N1G 2W1, Canada; E-Mail: syang@uoguelph.ca

* Author to whom correspondence should be addressed; E-Mail: zdq367@yahoo.com.cn; Tel.: +86-21-3828-2874.

Abstract: A model based on PCA (principal component analysis) and a neural network is proposed for the multi-fault diagnosis of sensor systems. Firstly, predicted values of sensors are computed by using historical data measured under fault-free conditions and a PCA model. Secondly, the squared prediction error (SPE) of the sensor system is calculated. A fault can then be detected when the SPE suddenly increases. If more than one sensor in the system is out of order, after combining different sensors and reconstructing the signals of combined sensors, the SPE is calculated to locate the faulty sensors. Finally, the feasibility and effectiveness of the proposed method is demonstrated by simulation and comparison studies, in which two sensors in the system are out of order at the same time.

Keywords: multi-fault diagnosis; principal component analysis; signal reconstruction; fault detection; fault isolation

1. Introduction

Since the principal component analysis (PCA) model can effectively reduce the dimension of input data, extract useful information and improve the efficiency of data analysis, it has been widely used in pattern recognition, signal processing, data compression, fault monitoring and many other fields [1-4]. The PCA model has been studied in depth in the field of fault diagnosis, especially in some

complicated systems [4-7]. With the development of neural networks, some PCA diagnosis models based on artificial neural networks such as the Error Back Propagation (BP) neural network [8], the radial basis function (RBF) neural network [9,10] and the self organizing mapping (SOM) neural network [11] have been extensively studied in recent years. However, all the current fault diagnosis models aim at studying single-fault sensor systemd and rarely discuss multi-fault diagnosis (which is a common issue in the practical application). In addition, regarding the mentioned neural network PCA models, the structures are complicated, the parameters are difficult to design and the convergences are not so perfect. All of these hamper the practical use of the models. Compared to single-fault sensor diagnosis system, the real-time demand for multi-fault sensor diagnosis is higher and the calculations are more complicated, and consequently the PCA fault diagnosis method has rarely been applied to multi-fault sensor systems.

A rapid PCA fault diagnosis model for multi-fault sensor diagnosis systems based on a credit assigned cerebellar model articulation controller (CA-CMAC) is presented in this paper. As a result of the rapid convergence characteristics of the CA-CMAC, the real-time characteristics of the PCA fault diagnosis model are improved. The faulty sensors can be isolated by reconstructing the combined sensors' signals, and thus a rapid diagnosis of multi-fault sensor systems is realized.

This paper is organized as follows. In Section 2, the conventional CMAC algorithm and CA-CMAC are introduced. The proposed PCA-based multi-fault sensor system diagnosis model is presented in Section 3. To illustrate the effectiveness of the proposed method, a simulation example is given in Section 4 and a comparison study is shown in Section 5. Finally, some concluding remarks are made in Section 6.

2. CMAC Algorithms

2.1. Conventional CMAC Algorithm

The CMAC neural network algorithm [12,13], which is built by imitating the behavior by which cerebella control human limbs, was proposed by Albus. It is a local approximating network with characteristics such as linear structure, simple algorithm, rapid learning speed and capability of dealing with uncertain knowledge. Also it has a generalization capacity. The basic idea of CMAC is storing the learning data (knowledge) in the overlapped memory cells (memory space). It contains two operations. The first is to output the network results, and the second is to adjust the weights (stored knowledge).

Its basic structure is shown as Figure 1, S is n-dimensional input space, A is the address of memory cells and w is the stored weights in memory cells. Each input s_i activates the unit of memory space A to A^*. The sum of connected weights which A^* corresponds to is the output:

$$y_i = \sum_{j=1}^{N} w_j a_j(x), \qquad i = 1,...,m \qquad (1)$$

In Equation (1), a_j is the memory cell activation flag. For the activated units, $a_j = 1$, otherwise, $a_j = 0$. For weight learning and adjusting of the network, supposed S is a state, $w_j(t)$ in Equation (1) is the weight stored in the j-th memory cell after k-th iteration. In conventional CMAC algorithm, errors are distributed to all the activated units averagely, so $w_j(k)$ can be updated as follows:

$$w_j(t) = w_j(t-1) + \frac{\alpha}{m} a_j(\overline{y_s} - \sum_{j=1}^{N} a_j w_j(t-1)) \tag{2}$$

In Equation (2), $\overline{y_s}$ is the desired outputs of state S, $\sum_{j=1}^{N} a_j w_j(t-1)$ is the actual outputs of state S, α is a learning constant and m is the number of activated memory cells in some state. Only the weights of memory cells that have been activated will be updated.

Figure 1. Basic structure diagram of a CMAC.
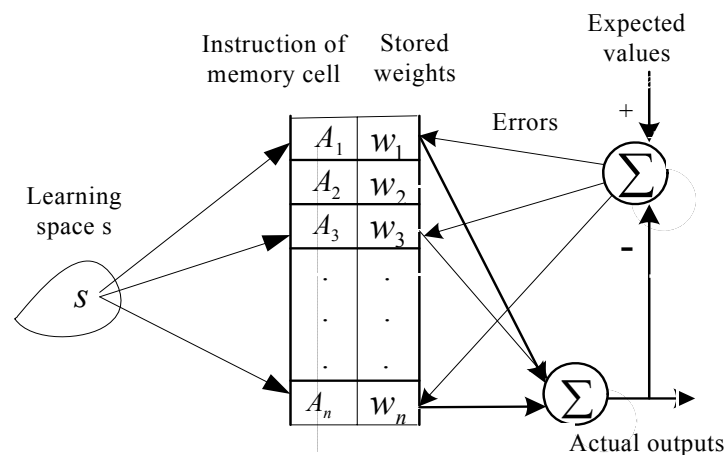
2.2. CA-CMAC Algorithm

In the conventional CMAC algorithm, the errors are averagely distributed into all the activated memory cells. We know that the weights of CMAC has included the former learned knowledge after k-1 times of iteration, however not each addressed hypercube has the same learning experience, which leads to the differences in the reliability of each addressed memory cell weight and each addressed cell has a different contribution to errors. That's to say, the same credit assignment should not exist in the m memory cells. If these differences are ignored and all the memory cells acquire the same amounts of errors, the errors produced by the state that has not been learned will cause "corrosion" to the former learned information, and in the network learning process, the desired data can only be gained after many learning cycles.

In order to improve the learning efficiency of CMAC and avoid the "corrosion" effect, the errors should be distributed in accordance with the memory cells' credibility. However, no effective methods have been developed to decide which cell should take more responsibilities for the current errors. In other words, no good methods have been proposed to decide the credibility of the memory cells' weights. The only available information are the current weights updating times. The more the numbers of updating for memory cells are, the more reliable the stored values are. As a result, the learning numbers of memory cells are regarded as its credibility. The credibility is higher if the weights' amendment is smaller. Supposed $f(j)$ is the learning number of j-th memory cell and m is the number of activated memory cells in some state, the idea of CA-CMAC is that adjusting errors must be in contrast with the learning numbers of activated memory cells, that is, $1/m$ in Equation (2) can be

replaced with $(f(j)+1)^{-1}/\sum_{l=1}^{m}(f(j)+1)^{-1}$. Thus learning performance can be improved effectively [14,15]. The specific algorithm can be expressed as:

$$w_j(t) = w_j(t-1) + \alpha a_j \frac{(f(j)+1)^{-1}}{\sum_{j=1}^{m}(f(j)+1)^{-1}}(\overline{y_s} - \sum_{j=1}^{m} a_j w_j(t-1)) \tag{3}$$

3. The Proposed Multi-Fault Sensor System Diagnosis Model Based on PCA

3.1. Principle of PCA and Signal Forecasting Model

PCA is a statistical analysis technique that can deal with relevance of data. It can reduce the dimensions of original variable X, bring effective information in original variable to the full and realize efficient compression of data. Suppose an original variable $X = [X_1, X_2, \ldots, X_m]^T$ is an m-dimensional random vector and the mean average of each component is zero, that is, $E(X_i) = 0$, (I = 1,2,…,m), its covariance matrix can be given by:

$$A = E[XX^T] = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots\vdots & a_{mm} \end{bmatrix} \tag{4}$$

In Equation (4), $a_{ij} = E(X_iX_j)$ is the covariance of X_i and X_j. A is a positive matrix, and $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m \geq 0$ are supposed as the eigenvalues of it. P is an orthogonal matrix, $P^TAP = \Lambda$, $\Lambda = diag(\lambda_1, \lambda_2, \ldots, \lambda_m)$ (Λ is a diagonal matrix). If $Y = P^TX$, then $E(YY^T) = E[P^TYY^TP] = P^TE[XX^T]P = \Lambda$, that is, $E[Y_iY_j] = \begin{cases} \lambda_i & i = j \\ 0 & i \neq j \end{cases}$. Therefore, Y_i is irrelevant to Y_j. If X is a normal random vector, Y_i and Y_j are also independent. $\lambda_i = E(Y_i^2)$. The traces of similar matrix are equal, then we can get the equation:

$$\lambda_1 + \lambda_2 + \cdots + \lambda_m = a_{11} + a_{22} + \cdots + a_{mm} \tag{5}$$

$$E(Y_1^2) + E(Y_2^2) + \cdots + E(Y_m^2) = E(X_1^2) + E(X_2^2) + \cdots + E(X_m^2) \tag{6}$$

where $\lambda_1 + \lambda_2 + \ldots + \lambda_m$ is the average energy of X. Supposed a is a constant between 0 and 1, $\beta_i = \lambda_1/(\lambda_1 + \lambda_2 + \ldots + \lambda_m)$, an integer s is chosen to satisfy the equation: $\beta_1 + \beta_2 + \ldots + \beta_s \geq a$. The proportion of energy brought by $Y_1, Y_2, \ldots, Y_s$ has exceeded a. Therefore $Y_{s+1}, Y_{s+2}, \ldots, Y_m$ can be regarded as random disturbance. Generally, because PCA model can effectively reduce the dimension of input data, s is less than m and $Y_1, Y_2, \ldots, Y_s$ are the components of $Y_1, Y_2, \ldots, Y_m$ whose significance level is a. Each principal component concentrates the common features of all the components in random variable X. Obviously, inverse transformation can be expressed as follows:

$$X = PY \tag{7}$$

According to the theory of principal component analysis, the orthogonal matrix *P* and principal component matrix *Y* can be obtained by orthogonal transformation of historical data *X* before *k* times, $Y = P^T X$. In addition, according to the eigenvalues of the covariance matrix of *X*, we can find the former *s* principal components $Y_1, Y_2, \ldots, Y_s$ whose significance level is *a*. They can represent the common features of all the components in the random vector *X*, so they should be given priority to be modeled and forecasted.

The former *s* principal components $Y_1, Y_2, \ldots, Y_s$ can be forecasted by CA-CMAC neural network to get principal components Y_i' (I = 1,2,…,*s*). According to the equation $X' = PY'$, we can get the sensors' predicted values *X'* that is after *k* cycles. The specific sensor signal forecasting model is shown in Figure 2.

Figure 2. Signal forecasting model based on PCA.
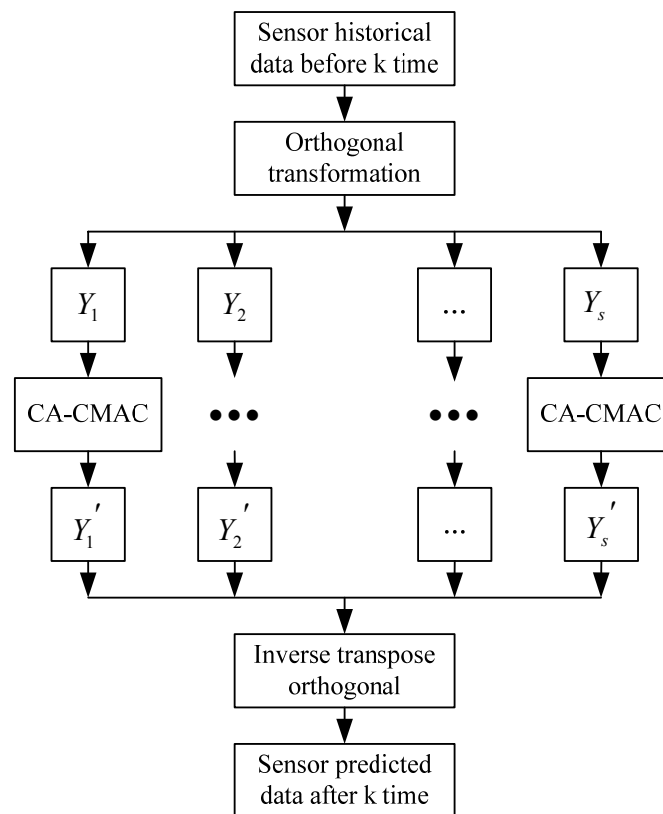
In CA-CMAC neural network training, the inputs of CA-CMAC are the historical values of the principal components Y_i (*i* = 1,2,…,*s*) at the time of (*k*-4, *k*-3, *k*-2, *k*-1), and the desired output is the value of principal components Y_i (*i* = 1,2,…,*s*) at the time *k*. Here the desired output can be also the historical value at time *k*-1, then the inputs of CA-CMAC are historical values at the times (*k*-5, *k*-4, *k*-3, *k*-2), so many training samples can thus be obtained.

Compared with the 5-BP neural network in [8], the 2-RBF network [9,10] and the SOM neural network in [11], the CA-CMAC neural network in Figure 2 has many advantages such as simple structure and high signal forecasting accuracy. Its rapid calculating speed can meet the real-time needs of fault diagnosis systems. In this paper, PCA forecasting model is applied to the detection and isolation of multi-fault sensor system.

3.2. Detection and Isolation of a Multi-fault Sensor System Based on PCA

(1) Detection of Faulty Sensors

When the sensor system is out of order, a fault can be detected according to some deviation between the actual measured data samples and the predicted values of a statistical model. That is, we can determine whether the system is out of order by the squared prediction error (SPE) between the measured values and normal predicted values of sensors. The estimated error vector is expressed as:

$$e(k) = X(k) - \hat{X}(k) \tag{8}$$

where $e(k)$ is the error vector of each sensor at k time, sensor measured values at k time are $X(k) = [X_1(k), X_2(k),\ldots,X_n(k)]$. The reconstructed predicted values at k time $\hat{X}(k) = [\hat{X}_1(k), \hat{X}_2(k), \cdots, \hat{X}_n(k)]$ can be got by the historical data before k time based on PCA, then *SPE* value of sensor system at k time can be expressed as follows:

$$SPE(k) = e^T(k)e(k) = \sum_{j=1}^{n} e_j^2(k) \tag{9}$$

Under normal circumstances, $\hat{X}(k)$ approximately equals $X(k)$ and the error $e(k)$ is small, then the *SPE* value is small too. However, the actual measured values will deviate greatly from the predicted values reconstructed based on PCA when a sensor or more are out of order. Then E_s will increase significantly. The variation curve of *SPE* can be obtained according to Equation (9). If the *SPE* value increases suddenly at some point, it shows that sensor system is out of order at that moment. Regulation of fault detection is defined as:

$$\begin{cases} SPE(k) < \delta_\alpha ----- Free - fault \\ SPE(k) \geq \delta_\alpha ----- fault \end{cases} \tag{10}$$

In Equation (10), δ_α is the fault threshold of *SPE*.

(2) Sensor Fault Isolation Algorithm in Multi-fault Cases

Once the fault of sensor system is monitored, we must find the source of fault accurately in order to exclude the faulty sensors quickly, then the normal values of faulty sensors can be replaced by the predicted values of the model to ensure that the system can operate normally. For the problem of single-fault sensor isolation, a method of linear variable reconstruction for sensor fault isolation has been proposed by Dunia [4,5]. On this basis, we propose an isolation algorithm for multi-fault sensors, fault isolation algorithm of "combined sensors reconstruction". Specifically, we reconstruct single-sensor signals and multi-sensor signals respectively and replace the sensor signals of actual measured values with the reconstructed values. *SPE* is calculated by Equation (11), and we can determine which sensor is out of order by the hopping of *SPE* curve.

① **Reconstruction and Isolation of Sensor Signals**

When the sensor system is out of order, the reconstructed value of each sensor at k time can be obtained by the measured data before k time based on PCA signal reconstructed model, $\hat{X}(k) = [\hat{X}_1(k), \hat{X}_2(k), \cdots, \hat{X}_j(k), \cdots, \hat{X}_n(k)]$. Among the sensors' actual measured values, we just reconstruct the signals of j-th sensor at k time and define the actual measured values as follows: $X(k) = [X_1(k), X_2(k), \cdots, \hat{X}_j(k), \cdots, X_n(k)](j = 1, \cdots\cdots, n)$, then we can get the following equation from Equations (8) and (9):

$$\overline{SPE}_j(k) = \sum_{i=1}^{n} e_i^2(k) \tag{11}$$

Here $\overline{SPE}_j(k)$ represents the $SPE(k)$ value after reconstructing the j-th variable. If only one sensor is out of order in the system, once the measured values of faulty sensor are reconstructed, $\overline{SPE}_j(k)$ will be less than the threshold as the fault has been be excluded by reconstruction. If the faulty variable is not reconstructed, $\overline{SPE}_j(k)$ will still be influenced by the fault and more than the threshold. Accordingly, we can isolate the faulty sensor simply. However, if more than one sensor is out of order, we can not ensure the SPE value is less than the threshold by reconstructing only one faulty sensor signal. That is to say, we can not determine the faulty sensors specifically and have to carry out a "combined sensors reconstruction".

② **Reconstruction and Isolation of Multi-Sensor Signals**

Suppose that among the n sensors to be diagnosed, m sensors are out of order. The signals from m sensors are reconstructed randomly at each time based on the PCA model. The actual measured values of the m sensors at k time are replaced with the reconstructed values:

$$X(k) = [X_1(k), \cdots, \hat{X}_i(k), \cdots, X_2(k), \cdots, \hat{X}_j(k), \cdots, X_n(k)](i, j = 1, \cdots\cdots, n)$$

The $\overline{SPE}_{i,j}(k)$ are calculated after reconstructing m sensors' signals according to Equations (8) and (9). If the $\overline{SPE}_{i,j}(k) \leq \delta_a^2$, the reconstructed m sensors are all out of order. Consequently, the isolation of multi-fault sensors can be realized. However, if $\overline{SPE}_{i,j}(k) > \delta_a^2$ is established, the faulty sensors are not isolated completely and combined reconstruction should be done until the result is in accord with $\overline{SPE}_{i,j}(k) \leq \delta_a^2$. The maximum numbers of combined reconstruction are C_n^m.

4. Simulation Study

4.1. Sensors Model

The simulation system is composed of four sensors and the output signals are defined as:

$$\begin{cases} x_1 = t + e_1 \\ x_2 = t^2 + t + e_2 \\ x_3 = t^3 + 2t + 1 + e_3 \\ x_4 = t^2 + 0.2\cos 2\pi t + e_4 \end{cases} \tag{12}$$

where e_i ($i = 1,2,3,4$) is an independent white noise variable distributed between [-0.1,0.1], t is a variable defined between [-1,1] and $\Delta t = 0.005$. In the simulation, 400 training data points are adopted. It is assumed that two sensors are out of order. According to the proposed algorithm, $n = 4$, $m = 2$.

4.2. Multi-fault Sensor Diagnosis Based on PCA

(1) Sensor Fault Detection

According to the signal prediction model based on PCA, the principal component matrix Y of historical data matrix $X = (x_1, x_2, x_3, x_4)$ can be obtained by orthogonal transformation, so do the first two principal components Y_1, Y_2 whose significance level α is 0.9941. Subsequently, the principal components Y_1', Y_2' at k time are forecasted by the principal components Y_1, Y_2 at the time of (k-4, k-3, k-2, k-1). At last, the estimated values $\hat{X} = (\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4)$ of X can be obtained by inverse orthogonal transformation.

The SPE value under normal circumstance is shown in Figure 3, which shows that the *SPE* value is in a relatively stable state and less than fault threshold ($\delta_\alpha = 0.04$). Thus the sensor data under normal circumstance can be forecasted well by PCA model.

**Figure 3.** SPE value without fault.
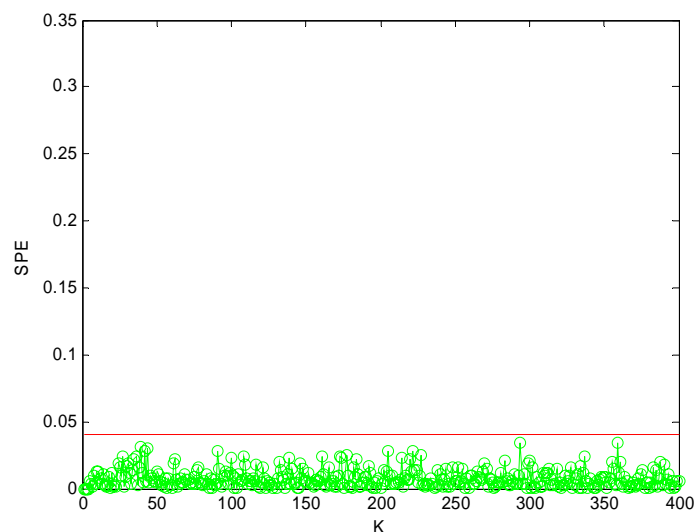
To illustrate the process of fault detection, in this paper, 400 sampling points are used. For variable x_1, a fault is added, which is 14% of its variation range between 150 and 400, and for variable x_4, an added fault is 24% of its variation range between 300 and 400. Then the *SPE* is calculated and the distribution curve is obtained according to Equation (9).

Figure 4 is the *SPE* value of the system when both x_1 and x_4 have a fault. Obviously, starting from the 150th sampling point, the *SPE* value suddenly increases and is more than the fault threshold δ_α. Consequently it can be determined that one sensor is out of order at that time according to the principle of fault detection. At the time of 300th sampling point, the *SPE* value displays another jump, and it shows another sensor is out of order at that time. Accordingly, it can show that not only the sensor system is out of order, but also two sensors (that is $m = 2$) have a fault through the number of jumps of the *SPE* value from Figure 4.

Figure 4. SPE value with fault.
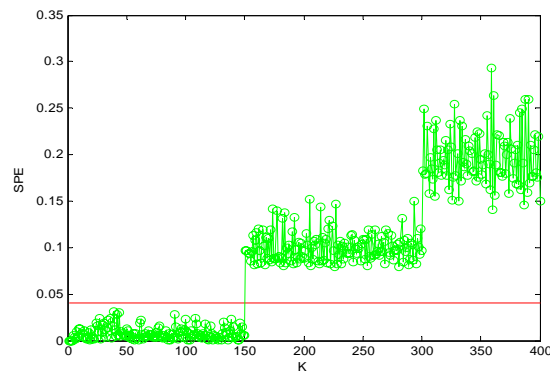
(2) Multi-Sensor Fault Isolation

In order to isolate the faulty sensors, in this paper, we adopt the method of reconstructing sensors two by two as $m = 2$. That is, $x_1x_2, x_1x_3, x_1x_4, x_2x_3, x_2x_4, x_3x_4$ are reconstructed respectively. After the reconstruction, if the *SPE* value of system is less than the fault threshold δ_α, then the two reconstructed sensors at this time are the faulty sensors. The $\overline{SPE_{1,2}}$ value after x_1 and x_2 reconstructed is shown in Figure 5. Then other combinations of variables are reconstructed. Figure 6, Figure 7, Figure 8, Figure 9 and Figure 10 show the values of $\overline{SPE_{1,3}}$, $\overline{SPE_{1,4}}$, $\overline{SPE_{2,3}}$, $\overline{SPE_{2,4}}$, $\overline{SPE_{3,4}}$ after reconstructing $x_1x_3, x_1x_4, x_2x_3, x_2x_4, x_3x_4$ respectively.

Figure 5. SPE value with sensors 1, 2 reconstructed.
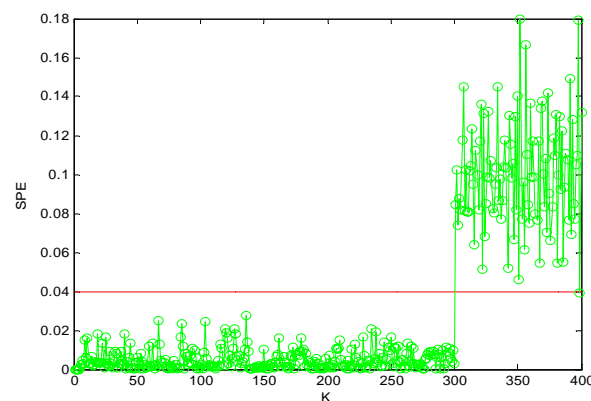
Figure 6. SPE value with sensors 1, 3 reconstructed.
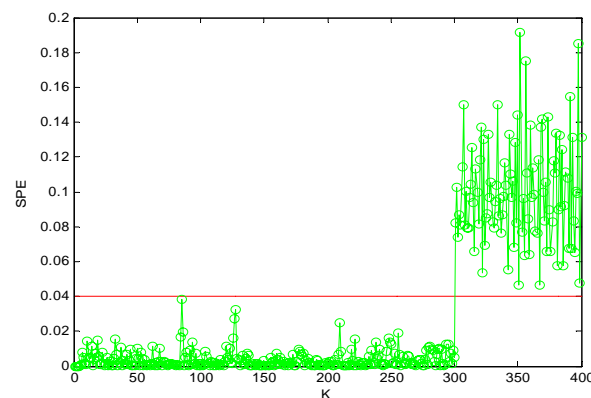
Obviously, after x_1x_4 are reconstructed, the *SPE* value of sensor system is less than the fault threshold. While after reconstructing other combinations, the *SPE* value of sensor system is still more than the fault threshold, which means the fault has not been excluded. Accordingly, it shows that the faulty sensors are x_1 and x_4. That is, sensor 1 and sensor 4 are both out of order and sensor 1 and sensor 4 are the faulty sensors. In Figures 3 to 10, *K* is the time variable (sampling time), the unit of *SPE* is the same as the unit of sensor output variable.

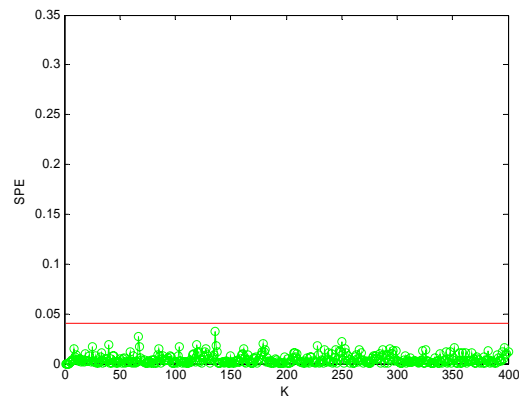**Figure 7.** SPE value with the sensors 1, 4 reconstructed.

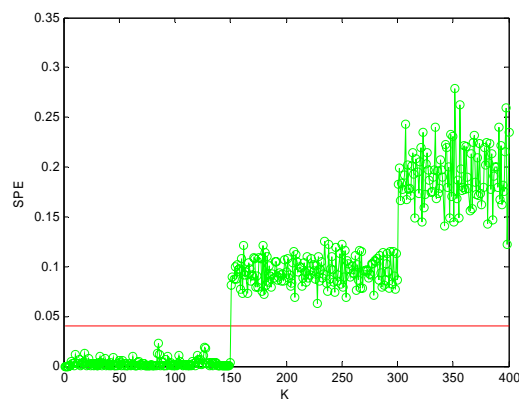**Figure 8.** SPE value with sensors 2, 3 reconstructed.

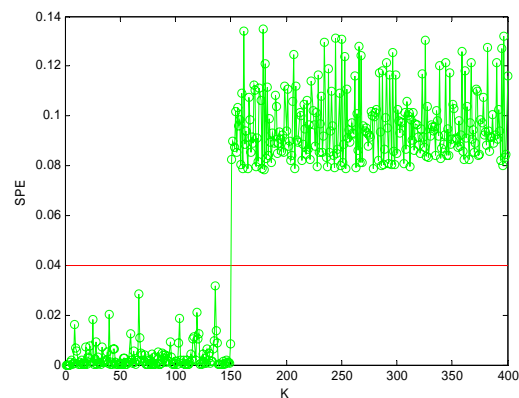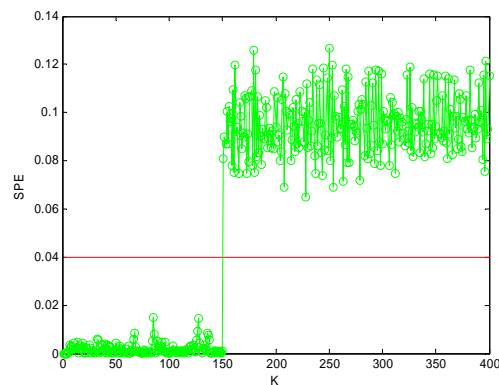**Figure 9.** SPE value with sensors 2, 4 reconstructed.

Figure 10. SPE value with sensors 3, 4 reconstructed.

5. Comparison among Ca-Cmac, Conventional Cmac and Bp Network

To illustrate the advantage of CA-CMAC neural network adopted in PCA model, take the mentioned PCA model as an example to study the learning effect of the conventional CMAC, CA-CMAC and BP neural networks. CMAC and CA-CMAC use the same network structure. The number of input states is 4, while the number of output states is 1. The BP neural network uses a three-layer network (input layer, hidden layer and output layer), and the number of nodes for each layer is 4, 6 and 1, respectively. The hidden layer node is a sigmoid function, while the output layer node is a linear function. When comparing the convergence rate of PCA with different neural networks, we use mean square error (*MSE*) to describe convergence of neural network:

$$MSE = \frac{1}{N}\sum_{k=1}^{N}(Y_{simuk} - Y_{t\arg k})^2 \tag{13}$$

where Y_{simuk} is the output of network, Y_{targk} is the expected value and N is the number of samples.

Figure 11. MSE decline curve in the training course of predicting the principal component 1.
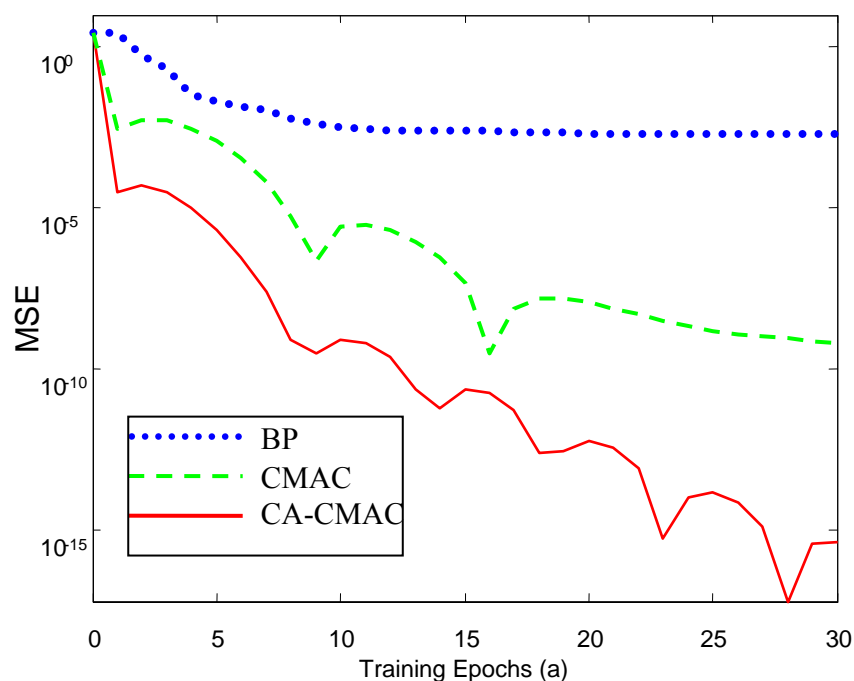
The curve in Figure 11 describes the *MSE* decrease with learning cycles in the training process of predicting principal component 1. From curves of *MSE*, it is obvious that the convergence rates of CMAC and CA-CMAC are much faster than that of the BP neural network.

The data in Table 1 is the *MSE* of principal component 1 varying with the cycles in the training process of the three different neural networks. From Table 1, it shows that the convergence rate of CA-CMAC is much faster than that of the BP network, and it is also faster than the CMAC network. Then it can meet the need of the real-time characteristics of a fault diagnosis system.

For each group training data of input and output, all connected weights in the BP neural network should be adjusted, and its calculation demands are increased; in addition, the BP neural network adopts a gradient descent algorithm, so the calculation speed is slow. On thre other hand, the CMAC is a local neural network. It adjusts only part of the weights and adopts the simple δ algorithm. Its convergence rate is much faster than BP network and the local minimum does not exist, so it has an obvious advantage in training accuracy and training time. In addition, compared to the conventional CMAC, the conception of credit assignment is introduced to CMAC, so the CA-CMAC can avoid the "corrosion" effect, the correcting errors are distributed in accordance with the cell credibility, then the update of weight is much more rational and effective, the learning time decreases greatly and the real-time characteristic of online learning is improved under the condition of the same approximation accuracy.

Table 1. *MSE* values by using different neural networks to train principal component 1.

Neural Network	Training Cycle					
	0	1	2	3	4	5
BP	2.77223	2.67898	0.52841	0.21499	0.03188	0.02098
CMAC	2.90312	0.29E-02	0.54E-02	0.51E-02	0.30E-02	0.12E-02
CA-CMAC	2.90312	0.31E-04	0.51E-04	0.31E-04	0.10E-04	0.21E-05
Neural Network	Training Cycle					
	6	7	8	9	10	11
BP	0.01487	0.01156	0.00580	0.00391	0.00316	0.00282
CMAC	0.34E-03	0.66E-04	0.55E-05	0.23E-06	0.25E-05	0.29E-05
CA-CMAC	0.29E-06	0.24E-07	0.79E-09	0.30E-09	0.74E-09	0.64E-09

6. Conclusions

In this paper, a rapid PCA diagnosis model based on CA-CMAC for multi-fault diagnosis is proposed. Simulation results show that the faulty sensors can be isolated by the method of "combined reconstruction". CA-CMAC provides a noticeable improvement in learning speed and accuracy in comparison to the conventional CMAC and BP neural networks. The proposed algorithm is practical and effective.

Acknowledgments

References

1. Puyati, W.Y.; Walairacht, A. Efficiency Improvement for Unconstrained Face Recognition by Weight Probability Values of Modular PCA and Wavelet PCA. In *Proceedings of 10th International Conference on Advanced Communication Technology*, Gangwon-Do, Korea, 2008; pp. 1449-1453.

2. Peng, D.Z.; Zhang, Y. Dynamics of Generalized PCA and MCA Learning Algorithms. *IEEE Trans. Neural Netw.* **2007**, *18*, 1777-1784.

3. Zhao, K.; Upadhyaya, B.R. Model Based Approach for Fault Detection and Isolation of Helical Coil Steam Generator Systems Using Principal Component Analysis. *IEEE Trans. Nucl. Sci.* **2006**, *53*, 2343-2352.

4. Dunia, R.; Qin, S.J.; Edgar, T.F.; McAvoy, T.J. Identification of Faulty Sensors Using Principal Component Analysis. *AIChE J.* **1996**, *42*, 2797-2812.

5. Ricardo, D.; Qin, S.J.; Thomas, F.E.; McAvoy, T.J. Use of Principal Component Analysis for Sensor Fault Identification. *Comput. Chem. Eng.* **1996**, *20*, 713-718.

6. Hui, D.; Liu, J.H. Drift Reduction of Gas Sensor by Wavelet and Principal Component Analysis. *Sens. Actuat.* **2003**, *96*, 354-361.

7. Li, R.Y.; Rong, G. Fault Isolation by Partial Dynamic Principal Component Analysis in Dynamic Process. *Chinese J. Chem. Eng.* **2006**, *14*, 486-493.

8. Hsieh, W.W. Nonlinear Principal Component Analysis by Neural Networks. *Tellus Ser. A* **2001**, *53*, 599-615.

9. Harkat, M.F.; Mourot, G.; Ragot, J. Nonlinear Pca Combining Principal Curves and Rbf-Networks for Process Monitoring. In *Proceedings 42th IEEE Conference on Decision and Control*, Maui, HI, USA, 2003; pp. 1956-1961.

10. Harkat, M.F.; Mourot, G. Variable Reconstruction Using RBF-NLPCA for Process Monitoring. In *IFAC Symposium on Fault Detection, Supervision and Safety for Technical Process*, Washington, DC, USA, 2003; pp. 55-63.

11. Kramer, M.A. Nonlinear Principal Component Analysis Using Auto-Associative Neural Networks. *AIChE J.* **1991**, *37*, 233-243.

12. Albus, J.S. A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC). *Dyn. Syst. Measur. Control* **1975**, *97*, 220-227.

13. Albus, J.S. Data Storage in Cerebeller Model Articulation Controller (CMAC). *ASME J. Dynam. Syst. Meas. Control* **1975**, *97*, 228-233.

14. Zhu, D.Q.; Kong, M. Adaptive Fault-Tolerant Control of Non-Linear Systems: An Improved Cmac-Based Fault Learning Approach. *Int. J. Control* **2007**, *80*, 1576-1594.

15. Feng, S.S.; Ted, T.; Hung, T.H. Credit Assigned Cmac and Its Application to Online Learning Robust Controllers. *IEEE Trans. Syst. Man Cybern.* **2003**, *33*, 202-213.