International Conference on Automatic control, Telecommunications and Signals (ICATS15)
University BADJI Mokhtar - Annaba - Algeria - November 16-18, 2015

Fault Diagnosis of Dynamic Processes based on Neuronal Principal Component Analysis

Chouaib CHAKOUR,
Mohamed Faouzi HARKAT,
Messaoud DJEGHABA
Institute of Electronics, Badji Mokhtar Annaba University,
P. O. Box 12, Annaba 23000, Algeria.
Email : chakourc@yahoo.fr, mharkat@univ-annaba.dz, messaoud.djeghaba@univ-annaba.dz.

***Abstract*—In this paper a new adaptive principal component analysis (APCA) algorithm is introduced for dynamic process monitoring. Its basic idea is to use a neuronal principal component analysis based on the generalized Hebbian algorithm (GHA). The provided interest of the proposed APCA algorithm is the new form to be followed to update the PCA model. At each time instant, a new observation is available, the PCA model is updated according to it without having to re-explore all previously available data. To identify faults, the reconstruction based contribution (RBC) approach is used and adapted in real times. The results for applying this algorithm on the the Tennessee Eastman process shows its feasibility and advantageous performances.**

***Index Terms*—PCA, Adaptive PCA, Dynamic process, GHA, Fault detection and isolation.**

I. Introduction

Actuator and sensor faults reduce the operating performance and may event cause a complete break-down of the systems. The detection and the handling of faults play an increasing role in modern technology, where the availability of the production tool and the personnel safety are ensured. Multivariate statistical process control (MSPC) based on principal component analysis (PCA) has been widely applied to satisfy these key demands. By revealing linear relation among the process variables, PCA allows the representation of large systems with models of significantly reduced dimensionality. Under the assumption that the industrial process is stationary or time-invariant, the static PCA used for process monitoring is reasonable. But, most industrial processes are time-varying and the monitoring of such processes, therefore, requires the adaptation of the PCA model to accommodate this behavior [1]. The time-varying characteristics of industrial processes include: (i) changes in the mean; (ii) changes in the variance; and (iii) changes in the correlation structure among variables, including changes in the number of significant principal components (PCs).

To address the challenge, several adaptive PCA schemes have been proposed [2-4]. The principle behind the moving window (MW) is well known. As the window slides along the data, a new process model is generated by including the newest sample and excluding the oldest one. Furthermore, the use of a constant number of samples in the window equates to a constant speed of model adaptation. This causes a problem when the window has to cover a large number of data points in order to include sufficient process variation for modeling and monitoring purposes, since the computational speed of MWPCA then drops significantly. If a smaller window size is attempted to improve computational efficiency, data within the window then may not properly represent the underlying relationships between the process variables [2]. An additional danger is that the resulting model may adapt to process changes so quickly that abnormal behavior remains undetected. To overcome this limitation, Wang et al [2] introduces a fast algorithm for moving window principal component analysis (MWPCA) which will adapt a principal component model.

Recursive techniques, on the other hand, update the model for an ever increasing data set that includes new samples without discarding old ones. It offers efficient computation by updating the process model using the previous model rather than completely building it from the original data. Recursive PCA (RPCA) allows older samples to be discarded in favor of newer ones that are more representative of the current process operation. Wold et al [3] and Gallagher et al [1] introduced the use of exponentially weighted moving average (EWMA), exponentially weighted moving covariance (EWMC), and exponentially weighted moving PCA (EWM-PCA). This is achieved by updating the process model at each time when a new vector of measurements becomes available. The past vectors of process measurements are then exponentially weighted in time so that the influence of the most recent measurements is the greatest. However, these approaches do not reflect the process changes well because EWMA, EWMC, and EWM-PCA iteratively apply exponentially decreasing weights without any identification for process changes. In addition, the data set for updating the model is ever increasing, leading to a reduction in the speed of adaptation. An important issue in RPCA is the choice of the weighting factor. The weighting factor determines the influence that the older data has on the model. When the process changes rapidly, the updating rate should be high, whereas when the change is slow and thus the essential process information is valid for a long period, it should be low. Therefore, the forgetting factor should be determined according to the underlying objective of the process monitoring scheme. In Choi et al [4] a new

algorithm for adapting the forgetting factor is proposed.

Two properties are of particular interest in the monitoring of dynamic systems field. One is the speed of adaptation, describing how fast the process model changes with new events. The other is the speed of computation, the time taken for the algorithm to finish one iteration for model adaptation [2]. The leading cause of this two problems is the form where the eigen-decomposition of covariance matrix is achieved. There are many areas of applications in which data may continue to become available over time, where the efficient incrementality is of importance. Since it is computationally expensive to calculate a matrix decompostion, it may not be feasible to recalculate when new data becomes available. Thus, data need not be processed all at one, as in the MWPCA and RPCA algorithms mentioned in the previous paragraphs. Systems that learn in real time need to be able to update data structures quickly. To cope with this drawback, different incremental techniques have been proposed. Adapted for on line learning, artificial neural network methods gained popularity for addressing this problem, propelled by the work of Oja [5] and Sanger [6]. Oja, 1985 demonstrated an incremental solution to finding the first eigenvector from serial arriving data, and Sanger, 1989 later generalized this to finding the first n eigenvectors with the Generalized Hebbian algorithm (GHA).

In this paper, adaptive principal component analysis algorithm (APCA) is introduced to monitor and diagnose dynamic systems. APCA algorithm refers to a paradigm where, at each time instant, a new observation is available, and the model is updated according to it without having to re-explore all previously available data. The basic idea is to use the generalized Hebbian algorithm (GHA), where GHA can extract adaptively principal components without storing and manipulating the covariance matrix.

The rest of the paper is divided as follows. In section 2 a neural network based PCA is presented. Section 3 gives the proposed APCA algorithm for process monitoring. Section 4 gives the reconstruction based contribution approach to identify faults. Results of application of the proposed fault detection and isolation scheme for Tennessee Eastman process are given in section 5. Finally, conclusions are given in section 6.

II. Neural Network based PCA

Conventional PCA seeks the axis that captures most of the variance of the data, where an optimal linear transformation of an input vector $\mathbf{x}(k)$ is determined.

$$\mathbf{y}(k) = W\mathbf{x}(k) \tag{1}$$

where k is the instant of measurement. $\mathbf{x}(k) = [x_1(k), x_2(k) \cdot \cdot \cdot x_m(k)] \in \Re^m$ is a zero mean input vector, $\mathbf{y}(k) = [y_1(k), y_2(k), ..., y_m(k)]^T \in R^m$ is the output vector, called principal components, and $W = [w_1, w_2, ..., w_m]^T \in R^{m\times m}$ is a desired transformation matrix. The orthogonal vectors $w_i = [w_{i1}, w_{i2}, ..., w_{im}]^T$, are called principal vectors.

A. Principal Component Analysis

In the standard numerical approach for estimating the principal components analysis first the autocorrelation matrix C is computed and then its eigenvectors and associated eigenvalues are determined by one of the known numerical algorithms, such as the SVD algorithm. PCA diagonalize the covariance matrix, where

$$C = \frac{1}{m}\sum_{j=1}^{m} x_j x_j^T \tag{2}$$

This is readily performed by solving the eigenvalue equation,

$$\lambda W = CW \tag{3}$$

for eigenvalues $\lambda \geq 0$ and eigenvectors $w_i \in R^m \setminus 0$.

If the input data vector have a large dimension, the covariance matrix is very large, and this may lead to expensive computation. In order to modeling time-varying processes behavior, each available sample measurement must be included to the PCA model. Indeed, the eigenvalues and eigenvectors of the newly updated covariance matrix are calculated to obtain a new PCA representation. A number of approaches have been proposed to calculate and/or adapt the eigenvalues and eigenvectors. The computational complexity to solve such eigen-decomposition problem is of $\mathcal{O}(m^3)$.

B. Generalized Hebbian Algorithm

The neural network approach enables as to find the eigenvectors and the associated eigenvalues directly from the input vector $\mathbf{x}(k)$ without the need to estimate the covariance matrix C. Among the existing iterative methods for PCA, the generalized Hebbian algorithm (GHA) is of particular interest, since it does not only provide a memory-efficient implementation but also has the inherent capability to adapt to time-varying distributions [7]. The generalized Hebbian algorithm estimates the eigenvectors of the covariance matrix with linear order computational and memory complexity $\mathcal{O}(m)$. The essence of these algorithms is a simple Hebbian learning rule:

$$W(k+1) = W(k)+\eta(k)(\mathbf{y}(k)\mathbf{x}(k)^T - LT[\mathbf{y}(k)\mathbf{y}(k)^T]W(k)), \tag{4}$$

and

$$\mathbf{y}(k) = W(k)\mathbf{x}(k), \tag{5}$$

and the corresponding estimated eigenvalues λ_i is calculated as follow:

$$\lambda_i = \frac{1}{m}\sum_{i=1}^{m} y_i(k)^2 \tag{6}$$

where $\eta(k)$ is the step size parameter (learning rate), and $LT[\cdot]$ sets all elements above the diagonal of its matrix argument to zero, there by making it lower triangular. This learn rule converge to the principals axes w_i^*, It was shown in [6] for $i = 1$ and in [7] for $i > 1$ that $W(k) \rightarrow W(k)^*$ as $t1 \rightarrow \infty$, where $t1$ is the iteration number.

The updated GHA has a scalar gain parameter η which is either held constant, leading to slow convergence. In order

to guarantee convergence, several algorithm was presented [13], [14]. The advantage of this algorithms over the Sanger's method is also the explicit calculation and adaptation of the learning rate η. The step size parameter (learning rate) cannot be a constant, but has to decrease over time [15]. Such a choice is common in the stochastic approximation literature, with slow convergence when η is small and divergence for large values. In order to overcome these drawbacks and achieve convergence, the adaptive learning rate is proposed in [14], with:

$$\eta(k) = \frac{\eta_0}{1 + t1/\tau} \tag{7}$$

where η_0 is a free scaler parameter. The tuning parameter τ determines the duration of the initial search phase, with $\eta(k) \approx \eta_0$ when $(\mathrm{t1} \ll \tau)$, before a convergence phase where $\eta(k)$ decrease as $\eta_0/t1$ when $(\mathrm{t1} \gg \tau)$.

III. Adaptive process monitoring

Slow and normal process changes often occur in real processes, which lead to false alarms for a fixed-model monitoring approach. A complete recursive PCA scheme should consider the following issues : (i) Recursive update of the mean and variance of the PCA model, (ii) Recursive determination of the number of principal components (PCs), and (iii) a recursive calculation of the control limits is needed to facilitate the adaptive monitoring.

A. The Updated Mean Vector

At each time a sample become available, the mean vector $\mathbf{m}(k)$ will not be constant and will need to be updated. The mean is updated with the degree of change in the model structure being dependent on the magnitude of the forgetting factors. Choi et al [9] propose a new algorithm for adapting the forgetting factor, where the forgetting factor α directly depend on the change in the mean structure.

$$\mathbf{m}(k) \simeq (1 - \alpha(k))\mathbf{x}^T(k) + \alpha(k)\mathbf{m}(k-1) \tag{8}$$

and, the forgetting factor $\alpha(k)$ which is used to update the mean vector is calculated as :

$$\alpha(k) = \alpha_{\max} - (\alpha_{\max} - \alpha_{\min}) \left[1 - \exp(-\gamma(\frac{\|\Delta\mathbf{m}(k)\|}{\|\Delta m_{nor}\|})^n)\right] \tag{9}$$

Where $\alpha_{\min}$, $\alpha_{\max}$, are the maximum and minimum forgetting value, γ and n are function parameters, and $\|\Delta\mathbf{m}\|$ is the Euclidean vector norm of the difference between two consecutive mean vectors. Here $\|\Delta\mathbf{m}_{nor}\|$ is the averaged $\|\Delta\mathbf{m}\|$ obtained using historical data.

B. Number of Principal Components

Since the number of significant principal components can change over time, there are many ways of determining the number of PCs. In this study, the cumulative percent variance (CPV) method is used. The CPV is a measure of the percent variance explained by the first PCs (l):

$$CPV(l) = \frac{\sum_{i=1}^{l} \lambda_i}{\sum_{i=1}^{m} \lambda_i} \times 100\% \tag{10}$$

Where λ_i is the eigenvalue for each eigenvector. The number of PCs (l) is chosen such that CPV reaches a predetermined value, e.g. 95%.

C. Control Chart (SPE)

The PCA is used to monitor an industrial process, the squared prediction error SPE and the Hotelling T2 are usually used. The SPE is a statistic that measures the lack of fit of the PCA model to the data. The residual vector e is the difference between x and $\hat{x}$, where $\hat{x}$ is the estimated vector of x.

$$\mathbf{e}(k) = \mathbf{x}(k) - \hat{\mathbf{x}}(k) \tag{11}$$

The statistic SPE is given by :

$$SPE(k) = \sum_{i=1}^{m} (e_i(k))^2 \tag{12}$$

Alternatively, the upper control limit for Q-statistic (SPE), with a significance level α is:

$$SPE_{lim} = (\frac{v}{2\mathbf{m}})\chi_{2\mathbf{m}^2/\mathbf{v},\alpha^2} \tag{13}$$

The terms $\mathbf{m}$ and v are the sample mean and variance of the model training set, and $\chi_{2m^2/v,\alpha^2}$ is the critical value of the χ squared variable with $2\mathbf{m}^2/\mathbf{v}$ degrees of freedom at the significant level, α.

D. APCA Monitoring Algorithm

The overall strategy of the proposed APCA algorithm for time-varying process monitoring is shown as follow in the algorithm 2.

IV. Fault isolation

One of the most used approaches in the field of fault diagnosis are the contribution plots. Contribution plots are based on the idea that the variables with the largest contributions to the fault detection index are most likely the faulty variables. When there is no faulty, these contribution are uneven across variables. Therefore, a fault in a normally small contributing variable may not have the largest contribution than other variables unless the fault magnitude is very large. This can be a source of misdiagnosis in the contribution plots method.

Recently, a contribution by reconstruction (RBC) method is presented by Alcala and Qin [10], and is considered as an alternative to the traditional contribution plots.

Algorithm 1 Adaptive PCA Algorithm

1) *Off-line mode* :
   a) Given an initial block of training data $\mathbf{X}$, and obtain the initial values of the sample mean $\mathbf{m}$, variance σ, the loading matrix (eigenvectors) W_0, the Λ eigenvalue matrix, and normalizing the block of training data as $(\mathbf{X}-\mathbf{m})/\sqrt{\sigma}$.
   b) Calculate the control limit of the monitoring statistic and the number of PCs (l_0).
   c) Determine the maximum and minimum values of the forgetting factor $(\alpha_{min}, \alpha_{max})$.
2) *Online mode* :
   At time point k, using the previous values of $m(k-1)$, $SPE_{lim,k-1}$.
   a) Obtain the next testing sample $\mathbf{x}(k)$, and scale it $(\mathbf{x}(k)-\mathbf{m}(k-1))/\sqrt{\sigma}(k-1)$.
   b) Calculate the monitoring statistic $SPE(k)$.
   c) Test if $SPE(k) < SPE_{lim,k-1}$, the testing sample is not an outlier and the system operate properly, go to step 3. Otherwise, consider the current condition to be abnormal and go to step 2.
3) *If the updating condition is satisfied*, do:
   a) Update the mean vector $\mathbf{m}(k)$ and the forgetting factor $\alpha(k)$, as in the Eq. (8) and (9) respectively.
   b) Update the eigenvectors W for a new sample $\mathbf{x}(k)$, as in Eq. (4), where, for the first iteration $W(k)$ is equal to W_0.
   c) Compute the eigenvalues λ_i corresponding to the news eigenvectors w_i, as in Eq. (6).
   d) Find the number (l) of principal components.
   e) Recalculate $SPE(k)$.

A. Reconstruction Based Contribution Method (RBC)

The reconstruction of a fault detection index along a variable direction minimizes the effect of such variable over the detection index [11]. We can use the amount of reconstruction along a variable direction as an amount of contribution of the variable to the fault detection index that is reconstructed [10].

When there is a fault in a variable direction, the fault measurement can be represented as

$$\mathbf{x}(k) = \mathbf{x}^*(k) + \xi_j f, \tag{14}$$

where $\mathbf{x}^*$ is the fault-free part of the measurement, and $\xi_j f$ is the faulty part, which is composed of the fault direction ξ_j, and the magnitude of the fault f.

The reconstructed vector along direction ξ_i is

$$\mathbf{z}_i(k) = \mathbf{x}(k) - \xi_i f_i \tag{15}$$

The task of reconstruction is to find a value of $\hat{f}_i$ such that index(z_i) is minimized. This value if f_i is found to be

$$\hat{f}_i = (\xi_i^T \tilde{C} \xi_i)^{-1} \xi_i^T \tilde{C} \mathbf{x}(k) \tag{16}$$

The reconstruction based contribution of variable x_i to the SPE fault detection index, RBC_i^{SPE} can be calculated as

$$RBC_i^{SPE} = \mathbf{x}^T(k)\tilde{C}\xi_i(\xi_i^T\tilde{C}\xi_i)^{-1}\xi_i^T\tilde{C}\mathbf{x}(k) = \frac{(\xi_i^T\tilde{C}\mathbf{x}(k))^2}{\tilde{c}_{ii}} \tag{17}$$

where $\tilde{c}_{ii} = \xi_i^T\tilde{C}\xi_i$ is the i^{th} diagonal element of $\tilde{C}$.

$\tilde{C}$ is the projection matrix that is defined as

$$\tilde{C} = \tilde{P}\tilde{P}^T \tag{18}$$

$\tilde{P} \in \Re^{n\times(n-l)}$ is a residual loadings of the eigenvectors matrix W, where l is the number of principal components (PCs) retained in the model.

V. Simulation Part

In this section, in order to investigate the potential application of APCA method, it was applied to on-line monitoring in the simulation benchmark of Tenesses Estman.

A. Tennessee Eastman Process (TEP)

The TEP was developed by Downs and Vogel of the Eastman Company to provide a realistic simulation for evaluating process control and monitoring methods. It has become the most important and commonly used benchmark simulation for the development of plant-wide control [12]. There are five major units in TEP simulation (shown Fig.1) a reactor, a separator, a stripper, a condenser, and a compressor. The process has 12 manipulated variables, 22 continuous process measurements, and 19 composition measurements sampled less frequently. Corresponding to different production rates, there are six modes of process operation.

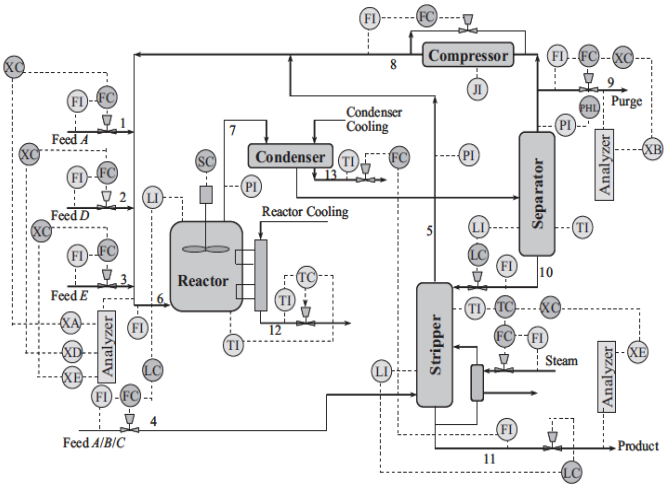

Fig. 1. Tennessee Eastman Process

B. Simulation Results

The TEP process was run for 2 hours, and we collected 2000 samples from 22 variables. The training data is composed of the first 400 samples, were utilized to build the initial PCA model. Also, there are 1600 samples for test. The proposed APCA algorithm is used to monitor online the TEP system, the

PCA model is updated with the evolution of the system using the GHA algorithm. We use the parameters described earlier in the algorithm. For the forgetting factor, we will choose these values of parameters : $\alpha_{max} = 0.99$, $\alpha_{min} = 0.9$, k=0.05 and $n = 3$. As well as, for the adaptive learning rate, we will choose these values of parameters : $\eta_0 = 0.000000001$, $t = 200$ iterations and $\tau = 0.001$. We illustrate in the (Fig.

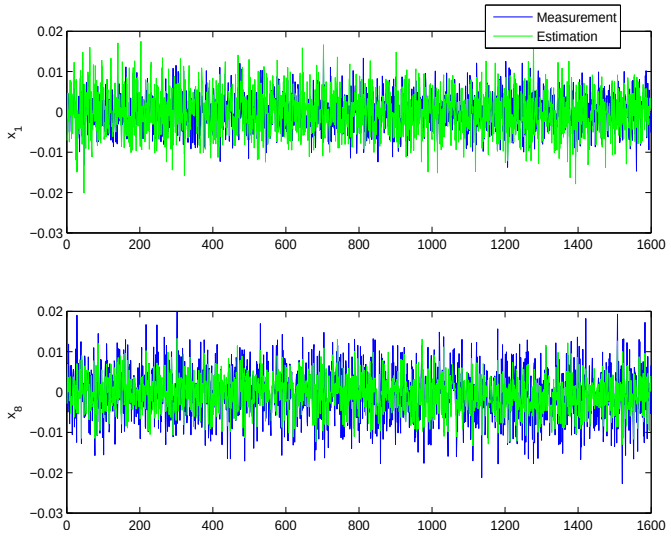

Fig. 2. The x_1, x_8 measurements evolution and its estimations in real time with the APCA algorithm

2) the ability of the proposed APCA algorithm to adapt and follow the operating change of the system with time. Using the APCA algorithm; figure 2 illustrate the estimation in real time of the 'A Feed' and 'Reactor Level' measurements of the the TEP system, which are represented by x_1 and x_8 variables, respectively. In the identification step of the APCA model, the cumulative percent variance method is used to compute the number of principal component retained in the PCA model for each new available observation of the system behavior (Fig. 3). Thus, for greater flexibility to adapting with the evolution of the system, an adaptive forgetting factor is used to update the mean vector in real time according to the change in the mean structure (Fig.4). This change in the mean structure is caused by the change on the operating modes of the system over time. A fault is simulated between samples 1200 and 1600

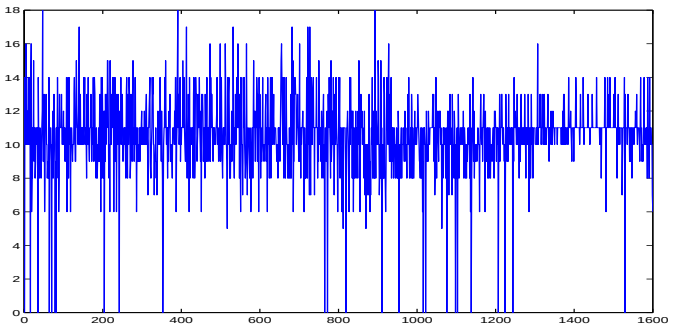

Fig. 3. Number of PCs retained

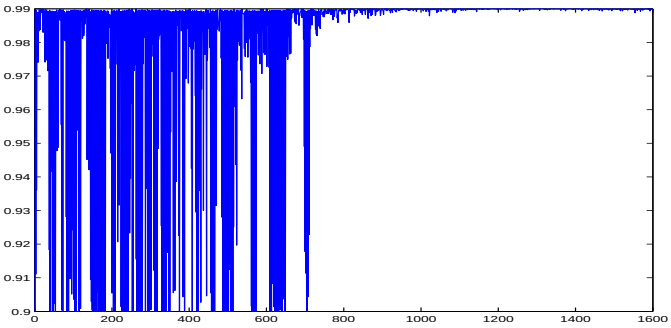

Fig. 4. Forgetting factor

affecting the 'A Feed' measurement, which is represented with the variable x_1. The APCA algorithm is used to monitor and diagnose the abnormal operating of the TEP system. After

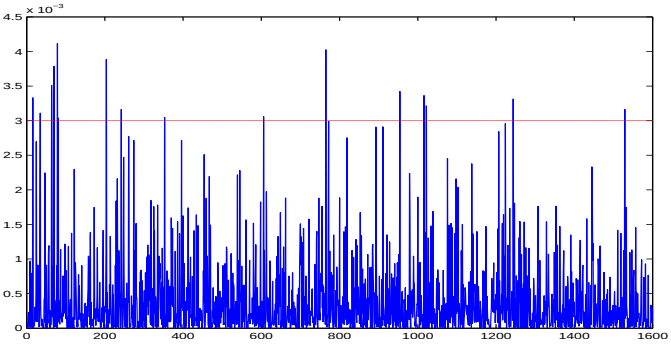

Fig. 5. SPE in fault free case

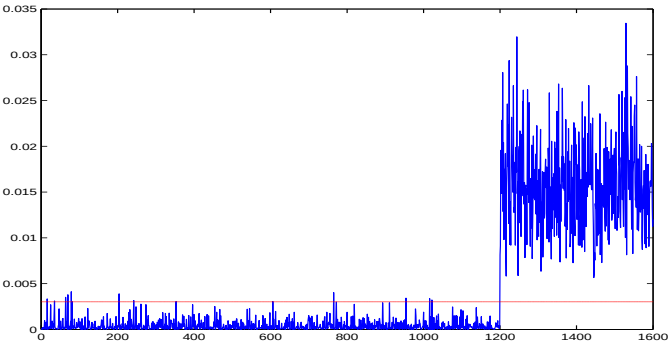

Fig. 6. SPE in the case of fault

time point k = 1200, it is found that the monitoring index SPE continuously exceed its threshold, which indicate a fault has been successfully detected (see Fig.5 and Fig.6). The control limit is calculated at the confidence level 95%.

When a faulty condition is detected, we need to determine the root cause of this problem. The reconstruction based contribution method is used to diagnose the provenance of this faulty. The basic idea is the quantification of fault magnitude a long ξ_i direction as in the equation (19). If the magnitude of the fault is quantified we can reconstruct the fault-free part of the measurement, which is represented with z_i according to the equation (18). The RBC method is used in the core of

the APCA algorithm to diagnose in real time each available faulty measuring samples.

The fault is estimated and is removed along a direction ξ_i to reconstruct the fault free measurement z_i such that the $SPE(z_i)$ index is minimized. We can see in the figure 7 the reconstructed vector z_1 of the measurement x_1 in real time along the direction ξ_1, which is equal to $[10000000000000000000]$. The reconstruction based contribution method guarantees that the faulty variable has the largest contribution. Figure 8 shown the RBC value with SPE fault detection index in different direction ξ_i. The first bar in the figure 8 present the RBC value of the fault in the measurement x_1, where the RBC_1^{SPE} value is the largest, which means that the suspect variable is x_1.

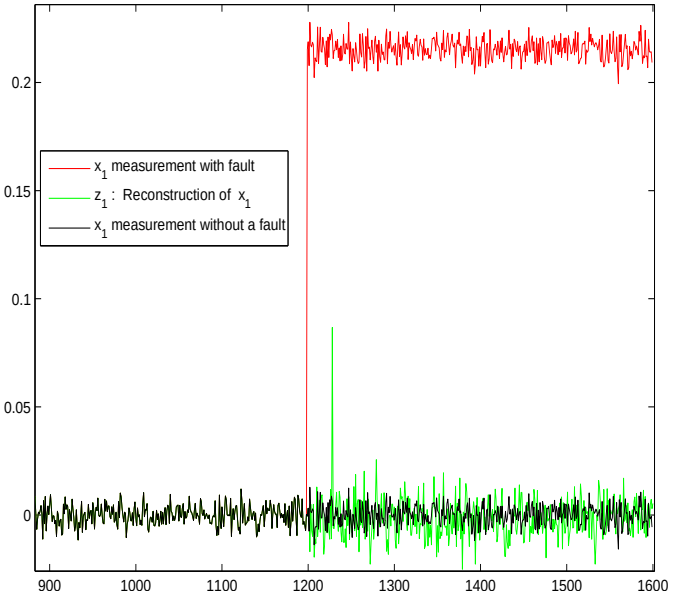

Fig. 7. The reconstruction of x_1 measurement (z_1) in the case of the fault

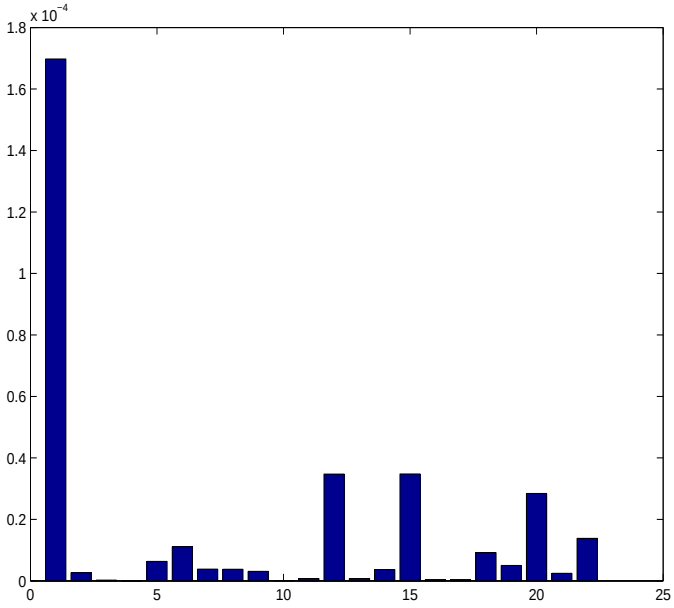

Fig. 8. RBC values for fault in the measurement x_1

VI. Conclusion

In this work, a new adaptive PCA algorithm based on the generalized Hebbian learning is presented for dynamic process modeling. However, GHA can update the PCA model for each available measurement without storing and manipulating the covariance matrix. The presented approach not only improves the computation efficiency but also reduces the storage requirement. The proposed algorithm is applied for sensor and actuator fault detection and isolation of Tennessee Eastman process.

References

[1] N. B. Gallagher, B. M. Wise, S. W. Butler, D. D. White, G. G. Barna. Development and benchmarking of multivariate statistical process control tools for a semiconductor etch process. Proceedings, Banff, Canada, June 9-11, 1997.

[2] Xun Wang, Uwe Kruger, and George W. Irwin. Process Monitoring Approach Using Fast Moving Window PCA. Ind. Eng. Chem. Res. vol 44, pp. 5691-5702, 2005.

[3] S. Wold. Exponentially Weighted Moving Principal Components Analysis and Projections to Latent Structures. Chemom. Intell. Lab. Syst. vol.23, pp. 149, 1994.

[4] Sang Wook Choi, B. Elaine, A. Martin, Julian Morris, and In-Beum Lee. Adaptive Multivariate Statistical Process Control for Monitoring Time-Varying Processes. Industrial Engineering Chemistry Research, vol. 45, pp. 3108 - 3118, 2006.

[5] Erkki Oja. A simplified Neuron Model as a principal component analyzer. Journal of Math. Biology, vol 15, pp.267-273, 1982.

[6] T. D. Sanger. Optimal unsupervised learning in a single layer linear feedforward neural network. Neural Networks, vol. 2, pp.459-473, 1989.

[7] Kwang In Kim, Matthias O. Franz, and Bernhard Sclolkopf. Kernel Hebbian Algorithm for Single-Frame Super-Resolution. Statistical Learning in Computer Vision, Workshop, Prague, Czech Republic, pp. 135-149, May 2004.

[8] Simon Haykin. Neural Networks. chapter 8 pages 440-441, 2005.

[9] Sang Wook Choi, Elaine B. Martin, A. Julian Morris, and In-Beum Lee. Adaptive Multivariate Statistical Process Control for Monitoring Time-Varying Processes. Industrial Engineering Chemistry Research, vol. 45, pp. 3108 - 3118, 2006.

[10] Alcala Carlos, S. Joe Qin. Reconstruction based contribution for process monitoring. Proceeding of the 17th World congress IFAC, Seoul, Korea, July 6-11, 2008.

[11] R. Dunia and S. J. Qin. A unified geometric approch to process and sensor fault identification : the unidimensional fault case. Comput. Chem. Eng., vol 22, pp. 927-934, 1998.

[12] T. Larsson, K. Hestetun, E. Hovland, and S. Skogestad. Self-optimizing control of a large-scale plant : the tennessee eastman process. Ind. Eng. Chem. Res, vol 40, pp. 4889-4901, 2001.

[13] Andrzej Cichocki, Wlodzimierz Kasprzak and Waldyslaw Skarbek. Adaptive learning algorithm for principal component analysis with partial Data. proceeding cybernitics and systems, volume 2, vienna, Austria, pp. 1014-1019, 1996.

[14] C. Darken, J. Change, and J. Moody. Learning rate schedules for faster stochastic gradient search. Proceeding IEEE Workshop Neural network for signal processing, 1992.

[15] Paul Honeine. On line kernel principal component analysis : Reduced order model. IEEE transactions on patern analaysis and machine intelligence, vol. 34, 2012.