Combined Input Training and Radial Basis Function Neural Networks based Nonlinear Principal Components Analysis Model Applied for Process Monitoring

Messaoud Bouakkaz and Mohamed-Faouzi Harkat
University Badji Mokhtar-Annaba, P. O. Box 12, Annaba 23000, Algeria

Keywords: Nonlinear PCA, IT-net, RBF-neural Network, Process Monitoring, Fault Detection and Isolation.

Abstract: In this paper a novel Nonlinear Principal Component Analysis (NLPCA) is proposed. Generally, a NLPCA model is performed by using two sub-models, mapping and demapping. The proposed NLPCA model consists of two cascade three-layer neural networks for mapping and demapping, respectively. The mapping model is identified by using a Radial Basis Function (RBF) neural networks and the demapping is performed by using an Input Training neural networks (IT-Net). The nonlinear principal components, which represents the desired output of the first network, are obtained by the IT-NET. The proposed approach is illustrated by a simulation example and then applied for fault detection and isolation of the TECP process.

1 INTRODUCTION

Principal component analysis (PCA) is among the most popular methods for extracting information from data, which has been applied in a wide range of disciplines. In process monitoring with Principal component analysis, PCA is used to model normal process behavior and faults are then detected by referencing the measured process behavior against this model.

It is known that the multivariate projection technique of PCA is linear, therefore it is only applicable for extracting information from linearly correlated process data. However, many industrial processes exhibit nonlinear behavior. For such nonlinear systems, linear PCA is inappropriate to describe the nonlinearity within the process and it can produce excessive number of false alarms or alternatively, missed detection of process faults, which significantly compromise the reliability of the monitoring systems.

To cope with this problem, extended versions of PCA have been developed such as Nonlinear PCA (NLPCA). Whilst linear PCA identifies the linear correlations between process variables, the objective of nonlinear PCA is to extract both linear and nonlinear relationships. Hastie and Stuetzle (Hastie and Stuetzle, 1989), proposed a principal curve methodology to provide a nonlinear summary of a *m*-dimensional data set. However, this approach is non-parametric and can not be used for continuous mapping of new data. To overcome the parametrization problem, several nonlinear PCA based on neural networks have been proposed (Kramer, 1991), (Dong and McAvoy, 1996), (Tan and Mavrovouniotis, 1995).

Tan and Mavrovouniotis (Tan and Mavrovouniotis, 1995) formulated an alternative scheme of nonlinear PCA based on an input-training neural network (IT-Net). Under this approach, only the demapping section of the NLPCA model is considered.

Compared with the other neural networks, when it is in training, its inputs which represent the desired principal component are not fixed but adjusted simultaneously with the internal network parameters, and it can perform all functions of a five layer neural network. However, IT-Net has its own limitation. For example, for a new data set or observation, calculation of its corresponding nonlinear principal component require more computation due to the necessity of an on-line nonlinear optimizer.

To improve this approach, a NLPCA model combinin a principal curve algorithm (Hastie and Stuetzle, 1989) and two cascade three-layer neural networks is proposed to identify mapping and demapping models (Dong and McAvoy, 1996).

Harkat *et al.* (Harkat et al., 2003) proposes a similar approach which uses two RBF networks for nonlinear principal component mapping and demapping, respectively. First, the principal curve algo-

Bouakkaz M. and Harkat M..
Combined Input Training and Radial Basis Function Neural Networks based Nonlinear Principal Components Analysis Model Applied for Process Monitoring.
DOI: 10.5220/0004152304830492
In *Proceedings of the 4th International Joint Conference on Computational Intelligence* (NCTA-2012), pages 483-492
ISBN: 978-989-8565-33-4

rithm is used to estimate the principal components. Then supervised learning is used to train the two RBF networks. The methodology proposed in this paper avoids the use of the principal curve algorithm by replacing the RBF demapping network with an IT-Net, which is able to estimate the principal components during learning.

The NLPCA approach proposed in this study uses *SPE* index for fault detection. In the linear version of PCA, the reconstruction approach, which tries to reconstruct the i^{th} variable from all other variables, is used for fault isolation (Dunia et al., 1996). Based on the same idea, we develop a nonlinear version of reconstruction approach using NLPCA model for fault isolation and reconstruction of the faulty measurements as in the linear case (Harkat et al., 2003), (HAR,).

The outline of this paper is as follows. Section 2 presents a Principal Component Analysis approach. Section 3 gives briefly reviews of some existing NLPCA methods. Section 4 describes the proposed NLPCA model combining the IT-Net and RBF neural networks. Section 5 describes the detection and isolation approach. Section 6 give an illustration example while section 7 gives the results of application of the proposed approach to the Tennessee Eastman process, and finally conclusions are presented in the last section.

2 PRINCIPAL COMPONENT ANALYSIS (PCA)

Principal component analysis (PCA) is a dimension reduction technique used in multivariate statistical analysis which deals with data that consist of measurements. The number of variables in such cases is often impracticably large, and one way to reducing it is to take linear combinations of variables and discard those with small variances. PCA looks for a few linear combinations which can be used to summarize the data while losing as little information as possible.
Let X represents a $N \times m$ matrix of data. PCA is an optimal factorization of X into matrix T (principal components $N \times \ell$) and P (loadings $m \times \ell$) plus a matrix of residuals E ($N \times m$).

$$X = TP^T + E \tag{1}$$

where ℓ is the number of factors ($\ell < m$). The Euclidean norm of the residual matrix E must be minimized for a given number of factors. This criterion is satisfied when the columns of P are eigenvectors corresponding to the ℓ largest eigenvalues of the covariance matrix of X. PCA can be viewed as a linear mapping from $\Re^m$ to a lower dimensional space $\Re^\ell$. The mapping has the form

$$\mathbf{t} = P^T\mathbf{x} \tag{2}$$

where $\mathbf{x}^T$ represents a single row of X and $\mathbf{t}^T$ represents the corresponding row of T. The loadings P are the coefficients for the linear transformation. The projection can be reversed back to $\Re^m$ with

$$\hat{\mathbf{x}} = P\mathbf{t} \tag{3}$$

where $\hat{\mathbf{x}}$ is the estimated vector of data.
Nonlinear PCA is an extension of linear PCA. Whilst PCA identifies linear relationships between process variables, the objective of nonlinear PCA is to extract both linear and nonlinear relationships. This generalization is achieved by projecting the process variables down onto curves or surfaces (Fig.2) instead of lines or planes (Fig.1).

In both cases the objective function to be minimized is the sum of squared orthogonal deviations:

$$\min \sum_{i=1}^{N} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2 = \min \sum_{i=1}^{N} \|\mathbf{x}_i - \mathscr{F}(\mathscr{G}(\mathbf{x}_i))\|^2 \tag{4}$$

where $\mathbf{x}_i$ is the ith row of X, $\mathscr{G}$ is the mapping function and $\mathscr{F}$ represents the demapping function. In this case, the nonlinear mapping has the form

$$\mathbf{t} = \mathscr{G}(\mathbf{x}) \tag{5}$$

and the inverse transformation is implemented by the second nonlinear vector function $\mathscr{F}$ that has the form

$$\hat{\mathbf{x}} = \mathscr{F}(\mathbf{t}) \tag{6}$$

Given an $N \times m$ matrix representing N measurements made on m variables, reduction of data dimensionality aims to map the original data matrix to a much smaller matrix of dimension $N \times \ell$ ($\ell < m$), which is able to reproduce the original matrix with minimum distortion through a demapping projection. The reduced matrix describes principal component variables extracted from the original matrix (Fig.3).

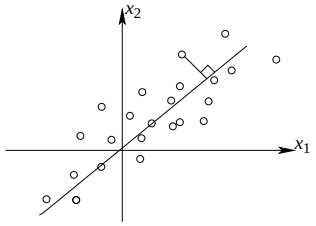

Figure 1: The linear principal component minimizes the sum of squared orthogonal deviations using a straight line.

We provide in the next a brief overview of neural network based NLPCA proposed over the last two decades and its implementation.

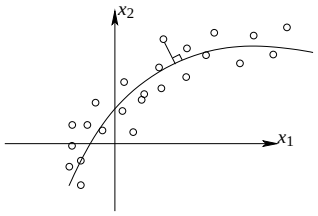

Figure 2: The nonlinear principal component minimizes the sum of squared orthogonal deviations using a smooth curve.

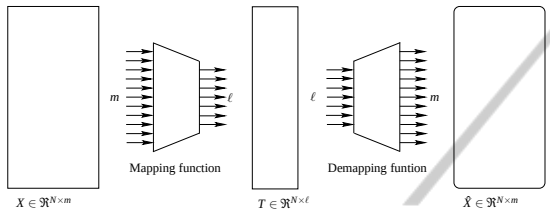

Figure 3: Reduction of data dimentionality.

3 NONLINEAR PCA

The objective, is to extract the nonlinear information from the nominal data set, namely, how to find matrix of nonlinear principal component scores T and a suitable nonlinear function $\mathscr{F}(t)$ to satisfy the equation (4). In this field, many neuronal NLPCA approaches have been developed, (Hastie and Stuetzle, 1989), (Kramer, 1991), (Tan and Mavrovouniotis, 1995), (Dong and McAvoy, 1996), (Harkat et al., 2003).

3.1 Five-layer Neural Network based NLPCA

To perform NLPCA, the Neural Network in Fig.4 contains three hidden layers of neurons between the input and output layers of variables (Kramer, 1991).

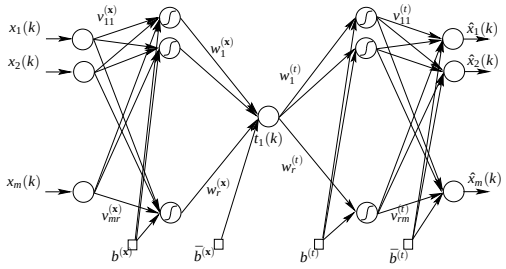

Figure 4: Five-layer NLPCA neural network for extraction one nonlinear principal component.

A transfer function $\mathscr{G}_1$ maps from x, the input column vector of length m, to the first hidden layer, represented by $h^{(x)}$, a column vector of length r, with elements

$$h_j^{(x)} = \mathscr{G}_1\left(\sum_{i=1}^{m} v_{ij}{}^{(x)} x_j + b_j^{(x)}\right) \tag{7}$$

The mapping function $\mathscr{G}$ is defined as

$$t = \mathscr{G}(x) = \sum_{j=1}^{r} w_j^{(x)} h_j^{(x)} + \bar{b}^{(x)} \tag{8}$$

Next, a transfer function $\mathscr{F}_1$ maps from t to the final hidden layer $h^{(t)}$, a column vector of length r, with elements

$$h_j^{(t)} = \mathscr{F}_1\left(w_j^{(t)} t + b_j^{(t)}\right) \tag{9}$$

and the demapping function $\mathscr{F}$ is given by

$$\hat{x} = \mathscr{F}(t) = \sum_{j=1}^{r} v_{ji}^{(t)} h_j^{(t)} + \bar{b}_i^{(x)} \tag{10}$$

where $\hat{x}$ representing the estimation vector of the original data x The transfer functions $\mathscr{G}_1$ and $\mathscr{F}_1$ are generally nonlinear. The *MSE* (mean square error) between the neural network output $\hat{x}$ and the original data x is minimized by finding the optimal values of $V^{(x)}$,$b^{(x)}$, $w^{(x)}$, $\bar{b}^{(x)}$, $w^{(t)}$, $b^{(t)}$, $V^{(t)}$ and $\bar{b}^{(t)}$.

3.2 Three-layer Neural Network NLPCA

Dong and McAvoy (Dong and McAvoy, 1996) presented an NLPCA method which integrates the principal curve algorithm and neural networks. The basic idea is to reduce the five-layer auto-associative network to a three-layer networks. In a such approach, two three-layer neural networks have been used. The inputs of the first neural network are the original data, and the outputs are the nonlinear principal scores obtained by principal curves. The inputs of the second network inputs are the ℓ nonlinear principal scores obtained by principal curves, and the outputs are the corrected data. Each neural network can be trained separately by any appropriate algorithm.

3.3 RBF-NLPCA Model

The nonlinear principal component analysis model for mapping and demapping can be obtained by using two RBF-Networks (Fig.5) and (Fig.6).

To identify the RBF-NLPCA model, we determinate the parameters of radial basis functions (centers and dispersions) and the weight parameters for the two RBF-networks. It should be noted that the nonlinear principal component matrix T being unknown, the

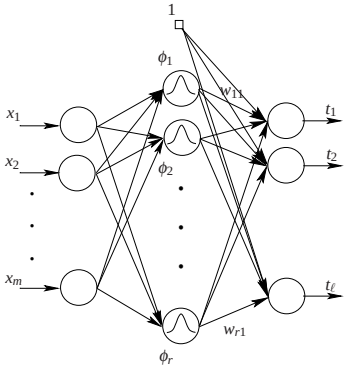

Figure 5: RBF-Network for mapping from $\Re^m \rightarrow \Re^\ell$.

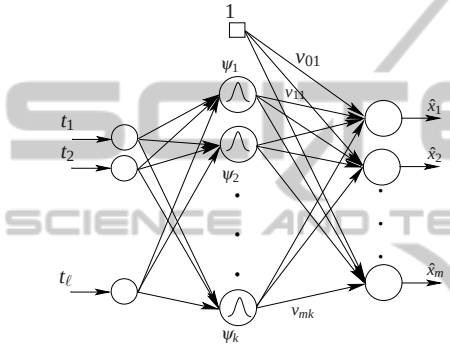

Figure 6: RBF-Network for mapping from $\Re^\ell \rightarrow \Re^m$.

training of the two RBF-networks separately is then impossible.

To overcoming this problem, two solutions have been proposed for estimating nonlinear principal component (Webb et al., 1999), (Harkat et al., 2003). The difference between the two methods is how to calculate the nonlinear principal component necessary for training of the two RBF neural networks. In the first solution, proposed by Webb (Webb et al., 1999), use the maximizing of the variance (Webb et al., 1999), and the second is the combining the RBF-networks and principal curves.

1. Webb (Webb et al., 1999) proposed an approach to nonlinear principal component analysis using radial basis function (RBF) networks. The first network projects data onto a lower dimension space Fig.5, such that the nonlinear features are captured in the sense of variance maximization of its outputs. So, Network parameters are determined by the solution of a generalized symmetric eigenvector equation by maximizing the variance of its outputs and then nonlinear principal components are the outputs given by this identified network.

   By preserving the original dimension of the data, the second network try to perform the inverse transformation Fig.6 (reproducing the original data) by minimizing the squared prediction error between the original data samples and its corresponding outputs. The two networks are trained separately and the outputs of the first one are the inputs of the second.

   However, the optimization task of the first network become so difficult because for this optimization we need to minimize estimation error which lead to train the second RBF-network and compare its outputs to the inputs of the first one.

2. The nonlinear component matrix T is estimated by using the principal curve algorithm (Hastie and Stuetzle, 1989), (Verbeek, 2001). Then, each RBF-network can be trained separately. So, the training problem is transformed into two classical nonlinear regression problems. We consider the RBF neural network illustrated in Fig.5. The aim is to use this network to define the nonlinear function $\mathscr{G}(.) : \Re^m \rightarrow \Re^\ell$ of $x \in \Re^m$. The outputs are computed as a linear weighted sum of the hidden node outputs:

$$t_j = \mathscr{G}(x) = \sum_{i=1}^{r} w_{ij}\phi_i(x) \qquad (11)$$

Where t is the output vector of ℓ outputs, $(w_{ij},\ i = 1,\cdots i,\cdots r)$ are the output layer weight parameters matrix W elements to be determined, connecting hidden node i to output j, and $(\phi_i,\ i = 1,\cdots i,\cdots r)$ is the gaussian function given by:

$$\phi_i(x) = \left(\frac{\|x - c_i\|^2}{2\sigma_i^2}\right) \qquad (12)$$

Where c_i and σ_i^2 respectively denote centers and dispersions, In this paper, the centers, c_i are obtained with C Fuzzy-means clustering method and the dispersions σ_i^2 are determined as the distance between c_i and the closest $c_j (i \neq j, j = \{1,\cdots,r\})$. So the training of the first network needs only to update the parameter weights. The parameter weights are obtained by a least square solution minimizing according to definition (13).

$$w = (\phi^T\phi)^{-1}\phi^T T \qquad (13)$$

Where the i^{th} column of ϕ^T is $\phi(x_i)$ and $\phi(.) = (\phi_1,\cdots,\varphi_i,\cdots,\phi_r)$, $w^T = (w_1,\cdots,w_r)$, and the i^{th} column of T^T is t_i.

By preserving the original dimension of the data, the second network tries to perform the inverse transformation from the reduced data (Fig. 6). We define the inverse transformation $\mathscr{F} : \Re^\ell \rightarrow \Re^m$:

$$\hat{x}_i = \mathscr{F}(t) = \sum_{j=1}^{k} v_{ji}\psi_j(t) + v_0 \qquad (14)$$

for some radial basis ψ_j, $(j = 1,...,k)$, weights $V = (v_0,...,v_k)$, where k is the number of kernels and $v_i \in \Re^{\ell}$, $(i = 0,\ldots,k)$.

The NLPCA model training procedure involves three steps (Harkat et al., 2003):

1. Find principal curves by successively applying the principal curve algorithm (Hastie and Stuetzle, 1989), (Verbeek, 2001) to observed data and residuals. Then in the first step T_1 denotes the first nonlinear principal component, so: $X = \mathscr{F}_1(T_1) + E_1$, where E_1 is the residual. When more than one nonlinear principal component is needed we do the same calculation from the residual data (LeBlanc and Tibshirani, 1994).
2. Train an RBF network that maps the original data onto the nonlinear principal components (obtained by the principal curves algorithm).
3. Train the second RBF network that maps the nonlinear principal components onto the original data.

3.4 IT-Net NLPCA

Compared with the other neural networks, the IT-Net has its own features, when it is in training, its inputs are not fixed but adjusted simultaneously with internal network parameters (linking weights between each layer and active bias of each node). IT-Net alternates between improving model parameters for given inputs and finding the optimal inputs for given model parameters. For a trained IT-Net, its final inputs consist of the matrix of nonlinear principal component scores T and at the same time IT-Net approximates the nonlinear function (6).

To perform NLPCA based IT-Net, the structure presented in Fig.7, contains one hidden layer between input and output layers.

Let, $V \in \Re^{r\times\ell}$, $b_1 \in \Re^{r\times 1}$, $W \in \Re^{m\times r}$, and $b_2 \in \Re^{m\times 1}$ be the weights from the input layer to the hidden layer, biases of the nodes in the output layer, the weights from the hidden layer to the output layer and the biases of the neurons in the hidden layer, respectively. For the IT-Net (Fig.7) with linear nodes in both the input and output layer, and a nonlinear function in hidden layer, a suitable function is sigmoid (σ):

A transfer function σ maps from t (input column vector of length ℓ) to the hidden layer, represented by h (column vector of length r), with elements,

$$h_j = \sigma\left(\sum_{i=1}^{\ell} v_{ji}t_i + b_{1j}\right) \qquad (15)$$

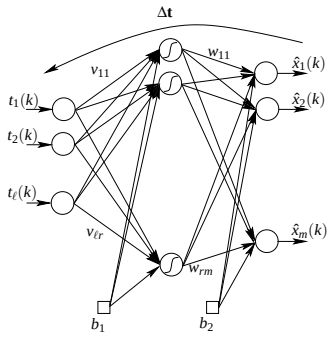

Figure 7: Structure of IT-Net.

where t_i represents the elements of the input vector t. The IT-Net outputs $\hat{\mathbf{x}}$ representing the approximation of the original data $\mathbf{x}$, a column vector of length m, with elements:

$$\hat{\mathbf{x}}(k)_k = \left(\sum_{i=1}^{r} w_{ji}h_j + b_{2j}\right) \qquad (16)$$

So, the demapping function is defined as follow:

$$\mathscr{F}(t) = W\left(\sigma\left(Vt + b_1\right)\right) + b_2 \qquad (17)$$

In this study, we propose to use a gradient descent back propagation algorithm with momentum η and adaptive learning rates r_t, r_w, r_v, r_{b1}, and r_{b2} to update the inputs, the weights between each layer, and active bias of each hidden layer node. The steepest descent directions for the optimization of the network weights are defined as follow:

$$\frac{\partial J}{\partial w_{kj}} = (x_k - \hat{x_k})h_j \qquad (18)$$

$$\frac{\partial J}{\partial v_{ji}} = \sum_{k=1}^{m}(x_k - \hat{x_k})w_{kj}dh_j t_i \qquad (19)$$

The steepest descent direction for the optimization of the network inputs is defined as follow:

$$\frac{\partial J}{\partial t_i} = \sum_{k=1}^{m}(x_k - \hat{x_k})\sum_{j=1}^{r} w_{kj}dh_j v_{ji} \qquad (20)$$

where

$$dh_j = \sigma'\left(\sum_{i=1}^{\ell} v_{ji}ti + b_{1j}\right) \qquad (21)$$

Then the inputs and parameters of IT-Net can be updated according to the following equations:

$$\Delta t_i(k+1) = r_t\left[(1-\eta)\frac{\partial J}{\partial t_i(k)}\right] + \eta\left(\Delta t_i(k)\right) \qquad (22)$$

$$\Delta w(k+1) = r_w\left[(1-\eta)\frac{\partial J}{\partial w(k)}\right] + \eta\left(\Delta w(k)\right) \qquad (23)$$

$$\Delta v(k+1) = r_v \left[(1-\eta) \frac{\partial J}{\partial v(k)} \right] + \eta \left(\Delta v(k) \right) \quad (24)$$

$$\Delta b_1(k+1) = r_{b1} \left[(1-\eta) \frac{\partial J}{\partial b_1(k)} \right] + \eta \left(\Delta b_1(k) \right) \quad (25)$$

$$\Delta b_2(k+1) = r_{b2} \left[(1-\eta) \frac{\partial J}{\partial b_2(k)} \right] + \eta \left(\Delta b_2(k) \right) \quad (26)$$

where

$$\Delta T_{N\ell}(k+1) = T_{N\ell}(k+1) - T_{N\ell}(k) \quad (27)$$

$$\Delta z(k+1) = z(k+1) - z(k) \quad (28)$$

where z represents each internal network parameter. where adaptive learning rates r_t, r_z are adapted as proposed in (Zhu and Li, 2006).

$$r_z(k) = 2^{\lambda_z} r_z(k-1) \quad (29)$$

$$\lambda_z(k) = sign \left[\frac{\partial J}{\partial z(k)} \frac{\partial J}{\partial z(k-1)} \right] \quad (30)$$

$$r_t(k) = 2^{\lambda_t} r_t(k-1) \quad (31)$$

$$\lambda_t(k) = sign \left[\frac{\partial J}{\partial t(k)} \frac{\partial J}{\partial t(k-1)} \right] \quad (32)$$

Where, $\frac{\partial J}{\partial t}$ and $\frac{\partial J}{\partial z}$ are steepest descent direction for optimizing network inputs and weights between each layer, respectively, and $\frac{\partial J}{\partial t_{N\ell}(k-1)}$ and $\frac{\partial J}{\partial z(k-1)}$ are its previous steepest descent directions.

3.5 IT-Net Algorithm

The algorithm of IT-Net can follows the following steps:

1. **Initialization:** Scaling each variable of X matrix to zero mean and unit variance, initialize input matrix T of scaled X, and weights (W and V), bias (b_1 and b_2) with random values in the range $[0\ 1]$.
2. **Computation:** For each input t_{ki}, (i=1,2,.....,ℓ) and $(k = 1, ..., N)$, calculate the corresponding IT-Net output $\hat{x}_{kj}$, $(j = 1, 2, ..., m)$ by using (16) and calculate the errors between the output and the scaled x,
3. **Update:** the weights W, V, b_1, b_2 and input vector t by using equations (18) to (32),
4. For next input $t_{(k+1)i}$, $i = (1, 2, ..., \ell)$, repeat Step 2 to Step 3 until all inputs have been adjusted and calculate the total error between the outputs $\hat{x}$ and the scaled sample x. if the maximum number of epochs (repetitions) is reached or if the error is smaller than aim error, and at the same time the error does not change any more, exit the iteration. Otherwise, go to the next step,
5. **Repeat:** the iteration from Step 2 to Step 4. After the iteration is finished, we acquire the input matrix T and a neural network model with fixed weights.

For a new observation, to calculate its corresponding nonlinear component requires the application of a new training of network, but we do not need to train the weights any more (fixed values) but only train inputs. The new input can be viewed as the nonlinear principal component of the testing sample, which is time consuming and can not be used in on-line process monitoring.

4 THE PROPOSED NLPCA MODEL

To take advantage of the Radial Basis Functions (RBF) (Fig.5) and Input Training neural networks (IT-Net) (Fig.7), a new NLPCA model combining these two neural networks is proposed for fault detection, isolation and reconstruction. These two networks are used in cascade. The first RBF neural network is used for mapping and the IT-Net for demapping, respectively. The proposed method involves two steps :

1. Find nonlinear principal components T by training the IT-Net, at convergence we have also the demapping function $\mathscr{F}$,
2. Train the RBF network that maps the original data to nonlinear principal components (obtained from the IT-Net).

5 FAULT DETECTION AND ISOLATION

5.1 Fault Detection

Abnormal situations that occur due to sensor drifts induce changes in sensor measurements. Nonlinear principal component analysis is used to model normal process behavior and faults are then detected by checking the observed behavior against this model.

Once a NLPCA representation has been built from data representing normal operation, the k^{th} sample vector $\mathbf{x}(k)$ may be decomposed by NLPCA into two portions,

$$\mathbf{x}(k) = \hat{\mathbf{x}}(k) + \tilde{\mathbf{x}}(k) = \hat{\mathbf{x}}(k) + \mathbf{e}(k) \quad (33)$$

where vectors $\hat{\mathbf{x}}(k)$ and $\tilde{\mathbf{x}}(k)$ are the estimation of $\mathbf{x}(k)$ from the NLPCA model at instant k and residual,

respectively. The $\hat{\mathbf{x}}(k)$ is the projection on the nonlinear principal component subspace according to (6). In this paper, fault detection is performed by monitoring the residuals, so we use the *SPE* detection index (squared prediction error) which is a statistic that measures the lack of fit of the NLPCA model to the data. At time *k*, the *SPE* is given by:

$$SPE(k) = \mathbf{e}^T(k)\mathbf{e}(k) = \sum_{j=1}^{m} e_j^2(k) \tag{34}$$

where $SPE(k)$ is the value of squared prediction error for the k^{th} sample of the process variables. Fault detection compares the current *SPE* with a threshold δ^2, the process is considered normal if:

$$SPE(k) \leq \delta^2 \tag{35}$$

δ denoted the confidence limit for *SPE*. To improve fault detection by reducing the rate of false alarms (due to noise), *EWMA* (Exponentially Weighted Moving Average) filter can be applied to the residuals. The filtered residuals $\mathbf{e}_f$ are thus obtained:

$$\mathbf{e}_f(k) = (I-\Lambda)(\mathbf{e}_f(k-1)) + \Lambda\mathbf{e}(k) \tag{36}$$

$$SPE_f(k) = \mathbf{e}_f^T(k)\mathbf{e}_f(k) \tag{37}$$

where $\mathbf{e}_f(k)$ and $SPE_f(k)$ are the filtered residuals and $SPE(k)$ respectively. $\Lambda = \gamma I$ denotes a diagonal matrix whose diagonal elements are forgetting factors for the residuals. If $SPE_f(k)$ is above the confidence limits, a fault is detected.

5.2 Fault Isolation and Reconstruction

After the presence of fault has been detected, it is important to identify the fault and apply the necessary corrective actions to eliminate the abnormal data. Like variable reconstruction approach for sensor fault isolation proposed by Dunia (Dunia et al., 1996) in the linear case, we propose an extension of this approach in the nonlinear case (Harkat et al., 2007). This approach assumes that each sensor may be faulty (in the case of a single fault) and suggests to reconstruct the assumed faulty sensor using the NLPCA model from the remaining measurements. By examining the residuals given by NLPCA model before and after reconstruction, we can determine the faulty sensor.

5.2.1 Nonlinear reconstruction

For the i^{th} variable, its reconstruction z_i is defined, as in the linear case, by an iterative approach, the estimated value $\hat{x}_i$ is re-estimated by the NLPCA model until convergence (Fig.8).

$$z_i = \xi_i^T \mathscr{F}(\mathscr{G}(\mathbf{x}_i)) \tag{38}$$

where $\mathbf{x}_i = [x_1\ x_2\ ...\ \hat{x}_i\ ...\ x_m]^T$ and ξ_i is the i^{th} column of the identity matrix. The iterative expression given by equation (38) must be started using some better initial value $z_i^{(0)}$. We suggest to use the measure x_i as initial value $z_i^{(0)} = x_i$. Note that the reconstruction expression (38) converge quickly converges quickly (i.e. in one or two iterations), for all the examples that have been treated.

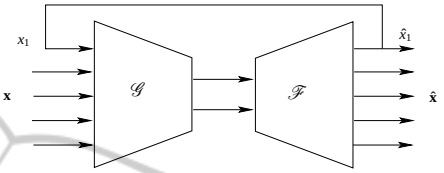

Figure 8: Depicts schematically an iterative reconstruction for x_1

So, let $SPE_{fj}(k)$ be the index $SPE_f(k)$ calculated after reconstruction of the j^{th} variable. Therefore, if the faulty variable is reconstructed, the index SPE_{fj} is in the control limit because the fault is eliminated by reconstruction. If the reconstructed variable is not faulty, the index SPE_{fj} being always affected by the fault, SPE_j is outside its control limit. In summary, when fault is detected, all indices SPE_{fj}, $(j = 1,...,m)$ are computed, and if $SPE_{fj} \leq \delta_\alpha^2$, the j^{th} sensor is considered as the faulty one.

6 SIMULATION EXAMPLE

We use a simple three dimensional case to illustrate the proposed NLPCA model efficacy. Three variables x_1, x_2 and x_3 are functions of one parameter u, and Gaussian noise added.

$$\begin{cases} x_1 = u^2 + 0.3\sin 2\pi u + \varepsilon_1 \\ x_2 = u + \varepsilon_2 \\ x_3 = u^3 + 1 + \varepsilon_3 \end{cases}$$

The ε_i components are independent white noise variables, uniformly distributed over $[-0.1\ 0.1]$ and u is a variable that is defined over $[-1\ 1]$. The two networks are trained using 400 data point. One nonlinear principal component is retained for this model which explains 97% of the variance of data.

The estimated curve using the proposed NLPCA model is shown in Fig.9. Fig.10 indicates the evolution of the first nonlinear principal component obtained by three NLPCA models; the proposed model (with 1-7-3 IT-Net structure, 3-8-1 RBF structure), Verbeek algorithm (Hastie and Stuetzle, 1989) and five-layer neural network (Hsieh and Li, 2001).

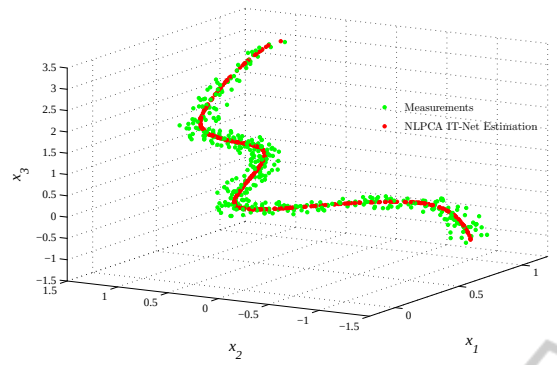

Figure 9: Data and estimation by IT-Net for nonlinear system

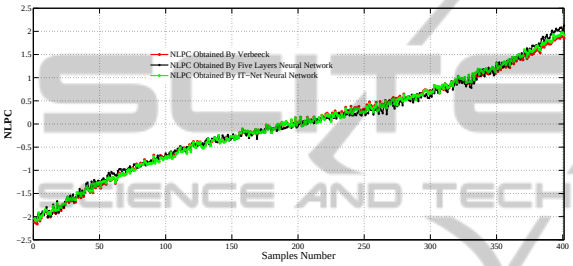

Figure 10: Evolution of the first nonlinear principal component.

For this example, Fig.11 represents the evolution of SPE_f in normal operating condition and in the presence of a fault simulated on the variable x_1 from sample 300 to 400. This fault is detected by this index.

Fig.12 indicates the time evolution of SPE_{f1}, SPE_{f2} and SPE_{f3} indices after reconstruction of variables x_1, x_2 and x_3, respectively. The SPE_{f1} calculated after the reconstruction of x_1 is under its control limit which indicates that x_1 is the faulty variable. Then we can reconstruct this variable in order to give a replacement value for the faulty measurements. Fig.13 shows, for variable x_1, the fault free measurements, the faulty measurements and the replacement values obtained by IT-Net and RBF neural networks NLPCA model. It is clear, that the reconstruction measurements are good estimations of fault free measurements.

7 APPLICATION

The Tennessee Eastman (TECP) process shown in Fig.14 is a simulation of a real plant, developed by Downs and Vogel of the Eastman chemical Company to provide a realistic simulation for evaluating process

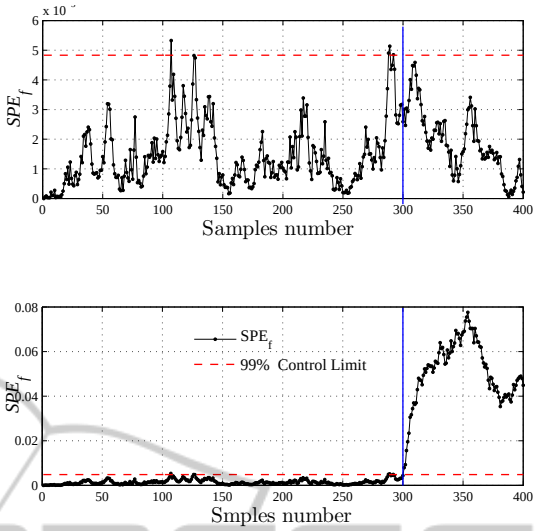

Figure 11: Fault detection for NLPCA IT-Net (*a*: SPE_f plot for normal operating conditions, *b* : SPE_f plot with a simulated fault on variable x_1).)

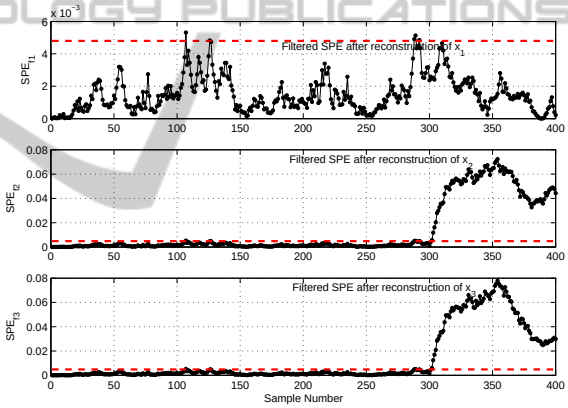

Figure 12: Fault isolation and faulty variable reconstruction (*a*,*b*,*c*: SPE_{fi} plot after reconstruction each variable: x_1, x_2, x_3 respectively.

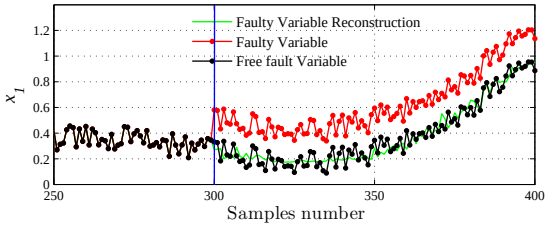

Figure 13: Faulty variable x_1 reconstruction.

control and monitoring methods, and widely used as a source of data (Downs and Vogel, 1993), (Vogel, 1994).

The process has five major units: a reactor, a product condenser, a vapor-liquid separator, a recycle com-

Table 1: Tennessee Eastman process Variables.

Measurements	Variable name	Units
m_1	D Feed	Kg/h
m_2	E Feed	Kg/h
m_3	A+C Feed	kscmh
m_4	Reactor Feed Rate	kscmh
m_5	Reactor Level	%
m_6	Reactor Temp	°C
m_7	Product Sep Underflow	m^3/hr
m_8	Stripper underflow	Kpa
m_9	Stripper Temp	°C
m_{10}	Steam Flow	Kg/s
m_{11}	Reactor Cool Temp	°C
m_{12}	Cond Cool Temp	°C

pressor and product stripper. The process produces products, G and H from four reactants A, C, D and E, also presents an inert B, and by-product F.

The process here consists of 12 manipulated variables from the controller and 41 measurements, of which 22 are continuous and 19 compositions measured by the Gaz chromatographic measurements can not be collected continuously. In this paper only 12 continuous outputs are used in our study as demonstrated in table 1.

There are two important factors that should be considered, one is that the process is nonlinear, and the second, the process operates in different modes (Downs and Vogel, 1993), a 50:50 G:H mass ratio, and others are in a 10:90 and a 90:10 G:H mass ratio. Here, the process is operated in mode at 90:10 mass ratio. We generate a set of data according to this condition. NLPCA is used to model the data. The first 400 samples are the normal data, and the next 100 samples involve data with a drift fault simulated on the variable m_6 which mean that the relationship among the process variables changes.

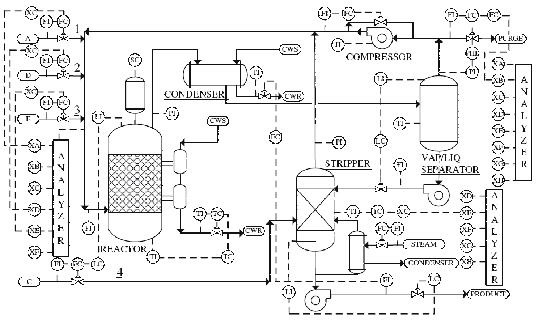

Figure 14: A diagram of the Tennessee Eastman and the base control problem simulator.

Four (4) nonlinear principal components are retained for the NLPCA model, which explains 98.68% of the variance of data.

Based on the obtained NLPCA model, the indices for detecting sensor faults and isolating faulty sensor can be calculated on-line. Fig.15 shows time evolution of the squared prediction error SPE_f for normal operating conditions.

To apply the sensor data validation method (sensor fault detection, isolation and reconstruction), a fault is introduced for the variable m_4 between samples 400 and 500.

SPE_f in 15 almost immediately allows to detecting the fault. To identify sensor, a reconstruction approach is applied and SPE_{fj} ($j = 1, 2, ...12$) are computed. In Fig.17, the index SPE_{f4} (computed after the reconstruction of m_4) is under its control limit. Which indicates that the faulty variable is m_4. Variable m_4 being identified as the faulty variable, then we can reconstruct this variable in order to give a replacement value for the faulty measurements. Fig.18 shows the fault-free measurements, the faulty measurements and the replacement values obtained by reconstruction for variable m_4. It is clear that the reconstructed measurements are good estimations of the fault-free measurements.

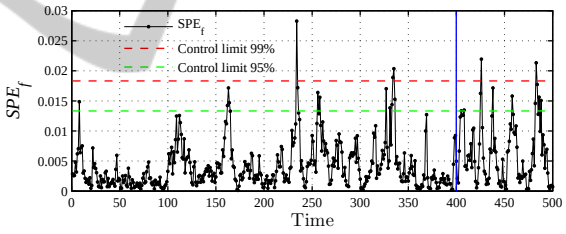

Figure 15: Time evolution of SPE_f for normal condition.

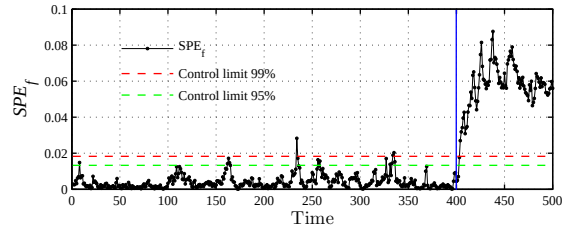

Figure 16: Time evolution of SPE_f with a fault on variable m_4.

8 CONCLUSIONS

In this paper we have presented a new nonlinear principal component analysis model. The proposed NLPCA model is obtained by combining two cascade neural networks. An RBF-Network for map-

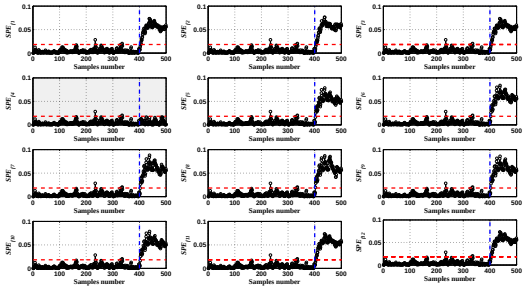

Figure 17: Time evolution of SPE_{fi} after reconstruction of each the variable m_i.

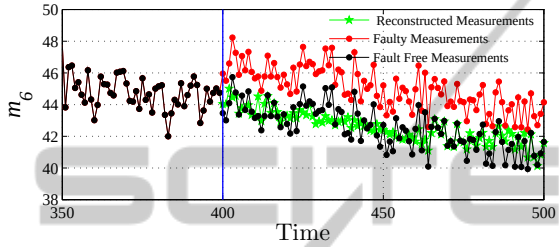

Figure 18: Reconstruction of the faulty measurements.

ping function and an IT-Net for demapping function. The principal components are calculated using the IT-Net which represent the demapping function, then the RBF network is trained to perform the mapping function. This two functions determines the NPLCA model.

An NLPCA model is built, using data obtained when the process is under normal condition. An extension of the nonlinear reconstruction approach is proposed. The variable reconstruction consists in estimating a variable from others process variables using the NLPCA model, i.e. using the redundancy relations between this variable and the others. This approach is presented as an extension of the reconstruction in the linear case (Dunia et al., 1996) and leads to an iterative expression of the reconstructed measurements.

The proposed approach for sensor fault detection and isolation, using nonlinear reconstruction method, is presented and successfully applied to a Tennessee Eastman process. The proposed approach also gives replacement values of faulty measurements.

REFERENCES

Dong, D. and McAvoy, T. (1996). Nonlinear principal component analysis based on principal curves and neural networks. Computers and Chemical Engineering 20, 65 78.

Downs, J. and Vogel, E. (1993). A plant-wide industrial control problem. Computers and chemical engineering Journal 17, 245-255.

Dunia, R., Qin, S., Ragot, J., and McAvoy, T. (1996). Identification of faulty sensors using principal component analysis. AIChE Journal 42, 2797-2812.

Harkat, M., Djellel, S., Doghmane, N., and Benouareth, M. (2007). Sensor fault detection, isolation and reconstruction using nonlinear principal component analysis. Intarnational Journal of Automation and Computing, 4,.

Harkat, M., Mourot, G., and Ragot, J. (2003). Variable reconstruction using rbf-nlpca for process monitoring. In *IFAC Symposium on Fault Detection, Supervision and Safety for Technical Process, SAFEPROCESS*. Washington, USA.

Hastie, T. and Stuetzle, W. (1989). Principal curves. Journal of the American Statistical Association 84, 502-516.

Hsieh, W. and Li, C. (2001). Nonlinear principal component analysis by neural networks. Tellus Journal 53A, 599-615.

Kramer, M. (1991). Nonlinear principal component analysis using auto-associative neural networks. AIChE Journal 37, 233-243.

LeBlanc, M. and Tibshirani, R. (1994). Adaptive principal surfaces. Journal of American Statistical Association 89(425), 53-64.

Tan, S. and Mavrovouniotis, M. (1995). Reduction data dimensionality through optimizing neural network inputs. AIChE Journal 41, 1471-1480.

Verbeek, J. (2001). A k-segments algorithm for finding principal curves. IAS Technical Journal.

Vogel, N. R. E. (1994). Optimal steady-state operation of the tennessee eastman challenge process. Computers and chemical engineering Journal 19, 949-959.

Webb, A., Vlassis, N., and Krose, B. (1999). A loss function to model selection in nonlinear principal components. Neural Networks Journal 12, 339-345.

Zhu, Q. and Li, C. (2006). Dimensionality reduction with input training neural network and its application in chemical process modeling. Chinese Journal.