NONLINEAR PCA IN FAULT DETECTION AND ISOLATION

Božidar Bratina[1], Boris Tovornik[1]

[1]University of Maribor, Faculty of Electrical Engineering and Computer Science,
2000 Maribor, Smetanova ul. 17, Slovenia

bozidar.bratina@uni-mb.si (Božidar Bratina)

Abstract

When dealing with nonlinear systems, linear fault detection and isolation techniques are sometimes inadequate. To describe nonlinearity of the system often we try to find the optimal fitting function. Many known methods have been developed where multivariate statistical methods are becoming popular in FDI practice as they offer good results according to data processing complexity. One of such is Principle Component Analysis, which is easy to implement, however not always accurate enough as it offers only linear transformation of process data. Therefore in this paper a derivation and implementation of nonlinear PCA for fault detection and isolation is demonstrated on a most popular case study, namely the three-tank laboratory plant using on-line data acquisition and the Matlab/Simulink environment. Derived NLPCA models are based on auto-associative neural network (AANN) with different combinations of encoding and decoding layers, and trained by back-propagation algorithm. Fault detection and isolation scheme is realized in Matlab/ Simulink and was designed to recognize predefined faults that can be introduced into the system. Results at the end show that drift or small shift faults, such as 4% of measured variables, can be identified by the FDI scheme in the system.

Keywords: Neural networks, nonlinear principal components, fault diagnosis.

Presenting Author's biography

Božidar Bratina graduated in 2004 at the Faculty of Electrical Engineering and Computer Science from the University of Maribor. Later that year he started his Ph.D. study at the same faculty in Maribor. His field of research includes Modeling of processes, Fault detection and isolation, Fault diagnosis and Fault recovery.

ISBN 978-3-901608-32-2

1 Introduction

Faults in industrial processes are unavoidable. However with appropriate action consequences can be minimized so every plant operator's dream is to be able to predict and localize faults and malfunctions in the process as quick as possible to ensure minimum plant down-time. Complex modern plants are big scaled, running with a lot of process variables which are supervised usually only by one or two operators, sometimes the amount of process data can be so large and rich that supervisors are blinded with information and a fault goes-by undetected.

To avoid such potentially dangerous situations many support systems for detection, isolation and diagnosis (FDI) of system faults has been developed, which can be data-driven, analytical, and/or knowledge-based. Overview of most important ones can be found in [1] and [2].

If big plants are under consideration then knowledge about the process is usually incomplete and classical analytical FDI methods are inappropriate. Many of them have been developed for special cases, however only few of them are really applicable and can't be simply put into practice. [3], [4], [5]

In this paper an implementation of fault detection and isolation scheme by using nonlinear PCA method based on auto-associative neural networks (AANN) is demonstrated onto a real laboratory plant where no a-priori knowledge is needed. A lot of improvements from the original method described by Kramer [9] have been developed, together with alternatives and modifications of AANNs that were applied in practice. Researchers have developed different ANNs structures, ANN-Fuzzy methods, used Genetic algorithms, etc., to improve operation of their research methods.

Despite different AANN methods, it takes some practical knowledge and experience to put NLPCA into practice on a real plant. In our case a classic PCA was already realized for fault detection and isolation on a laboratory plant, where we tried to overcome false alarm issue with nonlinear fitting function, achieved by AANN.

2 Statistical techniques

When using analytical FDI, a derivation of an adequate model of complex systems is very difficult and takes a lot of effort, therefore large systems are usually simplified so that the FDI techniques can be used. In such case it is more convenient to decide for data-driven methods, which usually don't require modeling steps. Also most of these processes have powerful supervisory systems (SCADA) installed, with the role of visualizing the operation of the process and informing the operator about the process behaviour. These systems present a key tool for development of data-driven FDI algorithms (history of the process).

Upon large process history datasets a statistical model of the process can be obtained by using well established multivariate statistical methods such as principal component analysis [6]. Because it does not require much of a processing power and is simple to implement, it has been widely used for image compression, fault detection, dimensionality reduction of data (gene expression, meteorology, medicine), etc. It can handle high dimensional and correlated process variables, provides a natural solution to the errors-in-variables problem and includes disturbance decoupling. However, main drawback lies in linearity of this technique.

As mentioned before, to improve its fitting to the nonlinear processes many derivatives of PCA has been developed so far. These can be found under the names as kernel PCA (Jade, 2003, Choi et al., 2005), dynamic PCA (Braatz, 2000), adaptive or non-linear PCA; uses artificial neural networks, genetic algorithms, Fuzzy and neuro-fuzzy algorithms, statistical methods (Yu, 1996, Wang, 2001, Chen, 2002), wavelet PCA (Martin, 1999), recursive PCA (Li, 2000), etc.

2.1 Principal Components Analysis

Principal component analysis (PCA) is very popular statistical method for extracting information from measured data, which finds the directions of significant variability in the data by forming linear combinations of variables. The use of PCA also allows the number of variables in a multivariate data set to be reduced, whilst retaining as much as possible of the variation present in the data set.

The data matrix must be auto-scaled by the means and standard deviations of physical variables. The variables are scaled by subtracting the means from the measured values and dividing the results by the standard deviations. Then from this data set, a corresponding squared covariance (correlation matrix) can be calculated. Then the eigenvectors of the covariance matrix of the basis data must be determined and reduction in the dimensionality of the data is possible, where only those directions in the vector space that are most significant for showing variations in the training data are retained. Next the eigenvalues and eigenvectors are computed using singular value decomposition (SVD) and PCA determines an optimal linear transformation of the data matrix X in terms of capturing the variation in the data:

$$T = XP \qquad X = TP^T \tag{1}$$

with $T = [t_1 \quad t_2 \quad \dots \quad t_m]$ where the vectors t_i are called scores or principal components and the matrix $P = [p_1 \quad p_2 \quad \dots \quad p_m]$, where the orthogonal vectors p_i, called loading or principal vectors, are the eigenvectors associated to the eigenvalues λ_i of the covariance matrix (or correlation matrix) Σ of X:

$$PP^T = P^T P = I \tag{2}$$

where $\Lambda = diag\left(\lambda_1, ..., \lambda_m\right)$ is a diagonal matrix with diagonal elements in decreasing magnitude order.

As mentioned, the reduction in dimensionality of the data can be realized in a way that the relative magnitudes of the singular (eigen) values are compared, and then only a few largest singular values and the corresponding eigenvectors are retained. This reduction is important when using statistic measures for fault detection since the smaller singular values effectively act as noise sources and thus impair its reliability. From these selected few largest singular values a reduced matrix of the principal components can be formed, which describes the behaviour of the system in direction of principal components.

There are several criteria by which the number of principal components can be determined. One way is to determine the percentage of variance of each principal component, respectively and then determine how many PCs will be retained. This can be achieved by using a cumulative sum, which gives the percentage of the described variance using eigenvalues of each principal component. It can also be graphically presented in the SCREE graph. Once the number of principal components (l) is determined (normally it is $l<m$), the initial matrix X can be described with l greatest eigenvalues (eigenvectors) of the covariance matrix Σ.

As shown here, the PCA in FDI is used for extracting redundancy relationships between the variables. In most practical cases (noisy measurements), the small eigenvalues indicate the existence of linear or quasi-linear relations among the process variables. However the distinction between significant or not significant eigenvalues may not be obvious (disturbances, nonlinearities and noise). Very important with PCA models is choosing the number of principal components. [7],[8]

2.2 Autoassociative neural networks and PCA

If the system is of very nonlinear nature classical PCA is usually not enough. In 1991 a nonlinear technique for multivariate data analysis was presented by M. A. Kramer, now known as nonlinear PCA - NLPCA. [9] He used a feed-forward neural network to perform identity mapping, where network inputs are reproduced at the output layer. We can say that Kramer's NLPCA is a generalization of classic PCA. The fundamental difference between NLPCA and PCA is that NLPCA allows nonlinear mappings from whereas PCA only allows linear mappings. To perform NLPCA, the NN in Figure 1 contains 3 hidden layers of variables between the input and output layers of variables.

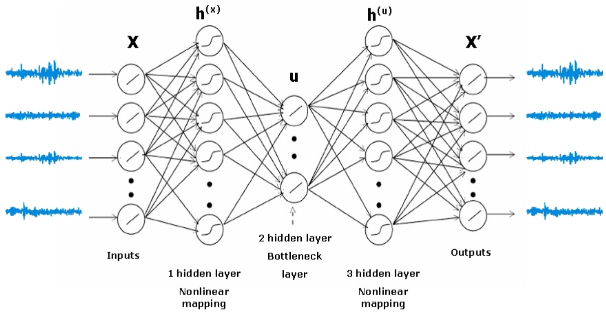

Fig. 1 The structure of Autoassociative ANN

Next to the input layer there is the encoding layer, followed by the bottleneck layer, which is then followed by the decoding layer.

A nonlinear function maps from the higher dimension input space to the lower dimension bottleneck space, followed by an inverse transform mapping from the bottleneck space back to the original space represented by the outputs, which are to be close to the inputs as possible by minimizing the cost function.

As described by Kramer, Hsieh, Hines, etc., a transfer function f_1 maps from x, the input column vector of length l, to the encoding layer, represented by $h^{(x)}$, a column vector of length m, with elements,

$$h_k^{(x)} = f_1\left(\left(W^{(x)} x + b^{(x)}\right)_k\right) \tag{3}$$

where, $b^{(x)}$, a column vector of length m containing the bias parameters, $W^{(x)}$ is an $m \times l$ weight matrix, and $(k = 1, ..., m)$. A transfer function f_2 maps from the encoding layer to the bottleneck layer containing a single neuron, which represents the nonlinear principal component u,

$$u = f_2\left(W^{(x)} h^{(x)} + \overline{b}^{(x)}\right) \tag{4}$$

The transfer function f_1 is generally nonlinear, while f_2 is usually the identity function.

The transfer function f_3 maps from u to the final hidden layer $h^{(u)}$,

$$h_k^{(u)} = f_3\left(\left(W^{(u)} u + b^{(u)}\right)_k\right) \tag{5}$$

$(k = 1, ..., m)$; followed by f_4 mapping from $h^{(u)}$ to x', the output column vector of length l, with

$$x_i' = f_4\left(\left(W^{(u)} h^{(u)} + \overline{b}^{(u)}\right)_i\right) \tag{6}$$

The cost function $J = \left\langle \left\| x - x' \right\|^2 \right\rangle$ is minimized to solve for the weight and offset parameters of the ANN, meaning finding the optimal values of $W^{(x)}$, $b^{(x)}, w^{(x)}, \overline{b}^{(x)}, w^{(u)}, b^{(u)}, W^{(u)}$ and $\overline{b}^{(u)}$. In that way the minimum square error between the NN output and the original data is thus minimized. The choice of the number of hidden layers in an encoding and decoding layer follows a general principle of parsimony. More hidden layers increase the nonlinear modeling capability of the network, on the other hand that could also lead to over-fitted solutions. [10],[11],[12]

2.3 Fault detection scheme

When using PCA method, there are many types of statistical measures to detect abnormal behaviour of the process, such as the Mahalanobis distance or Q statistics.

To isolate faults in the system can become a problem. One possibility is to use a method, depicted in Figure 2, where all possible NLPCA models are determined (fault-free and fault models) from the measurement data. For that reason it is necessary to run the process in all possible regimes and for all measured data sets determine NLPCA models. When certain fault is introduced to the process, the residual of the fault NLPCA model which is the same as the current process model, shouldn't react. Residuals are generated upon comparison between process measured and AANN output values.

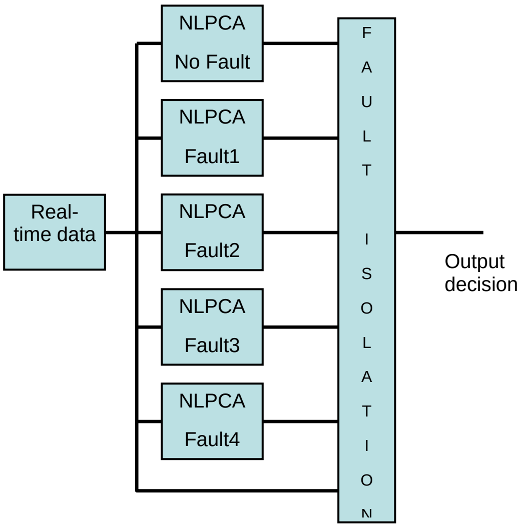

Fig. 2 Isolation of the faults

3 Application to a real laboratory plant

In some previous work [13] we implemented analytical and classic PCA schemes for on-line fault detection to a real laboratory three-tank plant. Since the techniques used were linear by nature, we had to deal with performance issues such as many false alarms, inaccurate isolation, etc. So we wanted to test performance of nonlinear PCA method based on auto-associative neural network when small faults or sensor drift is introduced to the system.

3.1 Hydraulic plant

The process flowsheet of the three-tank laboratory plant is depicted in Figure 3. The upright tanks T_1 and T_2 are mounted above the tank T_3, hence, the inlet to the tanks also depends on the level (hydrostatic pressure) in the tanks T_1 and T_2, respectively (the pumps P_1 and P_2 are not an ideal generators to the system). Also, the outlet pipes are mounted at the bottom of the tank T_3, hence the amount of water in tank T_3 affects the outlet and the inlet flow of the tanks T_1 and T_2.

The performance of the FDI scheme was evaluated by several fault cases introduced to the three-tank laboratory plant. The following faults were introduced: f_{h1} and f_{h2} – displacement of the level sensors in the tank T_1 and T_2, respectively, and f_{P1} and f_{P2} – pipeline of the pumps P_1 and P_2 were partially clogged (closing the inlet valves). Faults tested were abruptly brought about and no multiple faults were predicted or tested.

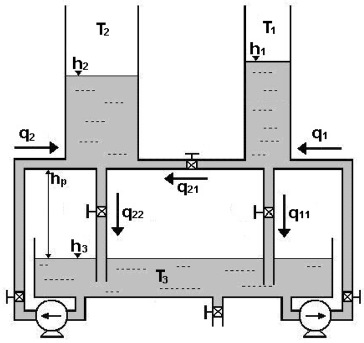

Fig. 3 The laboratory plant flowsheet

3.2 Data acquisition

By implementing FDI methods to a real process, it also must be considered that results highly depend on quality of data acquisition and data extraction from the noise correlated signals. In order to set up as much as modern industrial environment an OPC standard together with TCP/IP protocol was used [13]. The laboratory model was controlled locally by a PLC and touch-screen display, while the process variables (inputs and outputs of the model) were processed in Matlab/Simulink. (Fig 4)

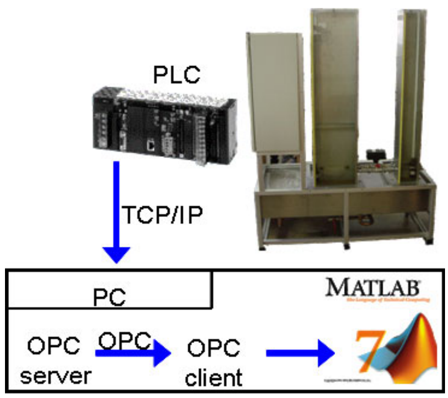

Fig. 4 Data acquisition

3.3 FDI scheme

After setting hardware properly to be able to collect data samples from process directly into Matlab/ Simulink software, process history (values from the process) needed to be recorded. To implement NLPCA for a laboratory three-tank model, various structures of feed-forward neural networks were tested and trained by using back-propagation algorithm. In this case the dimensionality reduction was not the main concern as we used only four process variables (flows and levels).

First the model was run in fault-free regime at different working points so adequate correlation between process variables could be established. Operation of the laboratory model in fault free case is depicted in figure 5.

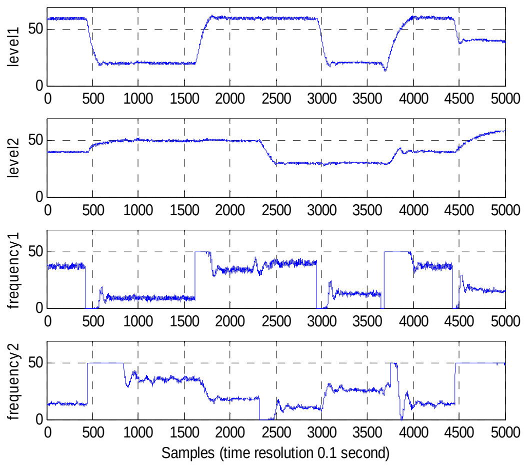

Fig. 5 Normal process operation

The auto-associative neural network was constructed as described in chapter 2 and various structure types were tested. To achieve proper fitting of the network output to desired plant output, we first tried a neural network structure such as 4-6-2-6-4. In this case there are 4 input neurons, 6 hidden neurons – represent mapping or encoding layer, 2 hidden neurons in a "bottleneck layer" – our NLPCAs, 6 hidden neuron in a de-mapping or decoding layer, and 4 output neurons. Neurons in mapping and de-mapping layer used nonlinear output function (logarithmic sigmoid transfer function - logsig or hyperbolic tangent sigmoid transfer function - tansig), and input, bottleneck, and output neurons used linear output function. The neural network was trained using Levenberg-Marquardt backpropagation with gradient descent learning function and for performance function a mean-squared-error was used.

To achieve adequate model of the laboratory plant we tested also a structure 4-8-2-8-4, 4-10-2-10-4, 4-15-2-15-4, and 4-20-2-20-4, where additional hidden neurons were included to improve nonlinear fitting to the plant output. In case of using "tansig" or "logsig" transfer functions, we achieved similar results.

Training of the network was supervised where 5000 learning samples were used for a fault-free case and four faulty cases described in chapter 3.1. According to different AANN structures, an AANN of 4-15-2-15-4 showed adequate results, and was used later for FDI realization in Matlab/ Simulink. In all tested structure cases learning took between 50 and 100 epocs.

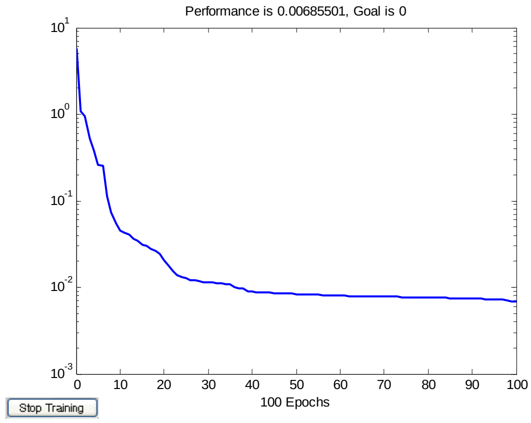

Fig. 6 Learning of the AANN (4-15-2-15-4)

To validate the model and model outputs (levels in the tanks and frequencies of the pumps) values were compared to the real-time process. Process values were preprocessed (removed mean value).

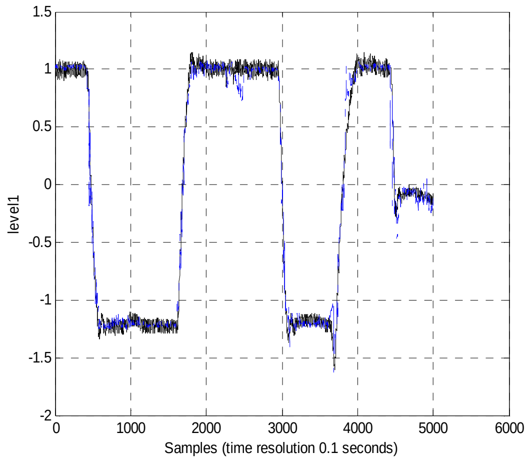

Fig. 7 Process and AANN outputs for level in Tank1

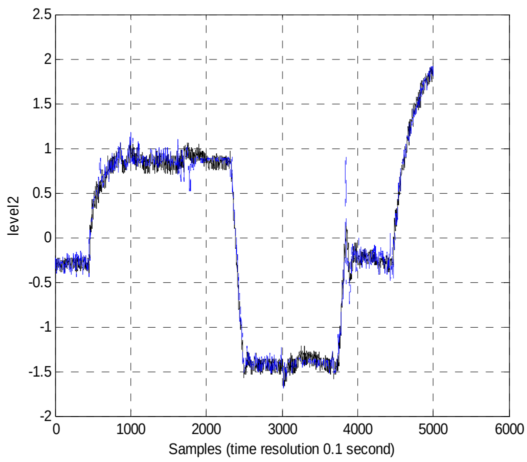

Fig. 8 Process and AANN outputs for level in Tank2

When the AANN model was adequate, the first nonlinear principle component could be extracted from the bottleneck layer. First PC in fault free-case regime is depicted in figure 9.

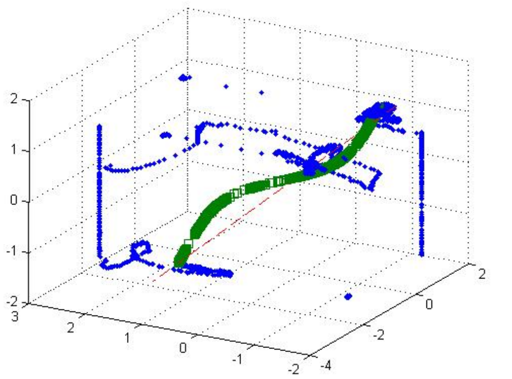

Fig. 9 Nonlinear Principal Component (green)

However for detection and isolation of faults AANN models of predicted fault regimes were necessary. By comparing different model outputs and real time data from the plant, residual signals used for FDI could be formed. These were generated according to deviation between measured and AANN outputs of fault-free and faulty cases where for detection a simple threshold function was used (Fig 10).

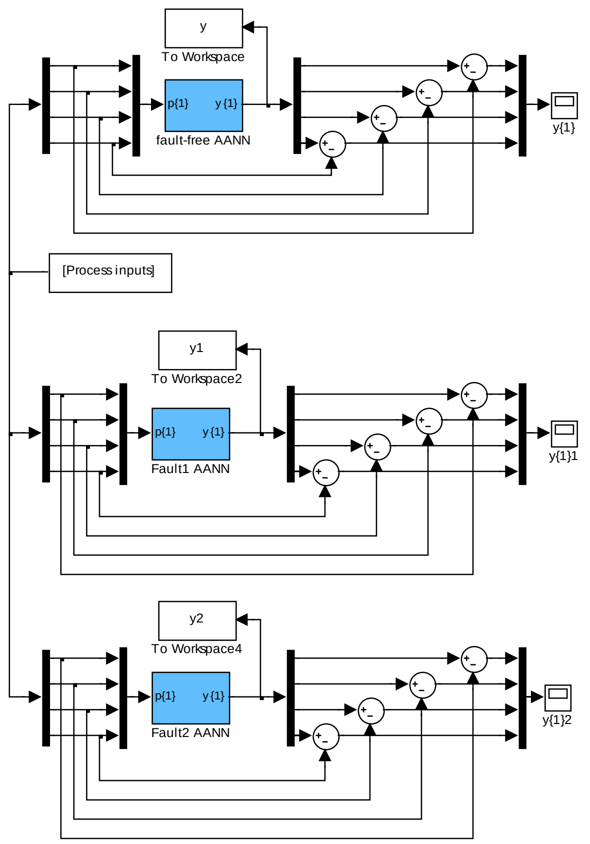

Fig. 10 Forming the residual signals

By properly setting the isolation parameters shift detection on the level sensors was achieved where small faults of 4% could be identified. Also a test for sensor drift was conducted, where it proved that drifts could be detected when they reach a value larger than 4%.

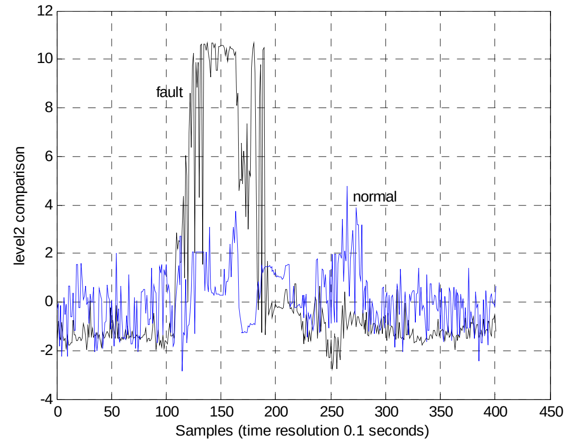

Fig. 11 Fault detection on level sensor in Tank2 (sensor shift)

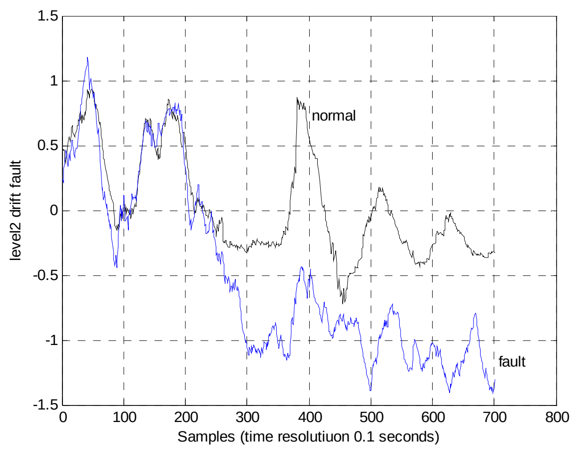

Fig. 12 Fault detection on level sensor in Tank2 (sensor shift)

Faults were introduced into the plant hence the signals were preprocessed in the Matlab/ Simulink. Figures 11 and 12 shows adequate 4% sensor shift and drift detection of the measured signal. The noise 2-3% on the measured signals from the process was filtered to improve detection results.

4 Conclusion

By developing PCA and nonlinear PCA models, studies have shown that fewer non-linear components are needed to describe similar process variance, as described by linear principal components.

In our case the preferred number of hidden neurons used for FDI was 15 for mapping and de-mapping layer. Although the model isn't that complex to best reflect advantages of using nonlinear FDI technique against a linear one, still better performance than with linear PCA FDI scheme was achieved.

However the implementation to the real process required more effort than classic PCA, where AANN models had to be trained properly with rich samples to achieve good results. Fairly good results were achieved where small shift and drift faults with magnitude of 4% could be identified. In comparison to classic PCA, the sensitivity of the diagnostic system is better however it depends on many factors such as noise levels on the measured signal, AANN structures, sample time of FDI scheme, etc.

5 References

[1] V. Venkatasubramanian, R. Raghunathan, K. Yin, S. Kavuri. A review of process fault detection and diagnosis, *Computers and Chemical Engineering*, 27: 293-346, Part I, Part II, Part III, 2003.

[2] V. Uraikul, C. W. Chan, P. Tontiwachwuthikul. Artificial intelligence for monitoring and supervisory control of process systems, *Engineering Applications of Artificial Intelligence*, 20: 115-131, 2007.

[3] J. K. Korbitz, J. M. Kościelny, Z. Kowalczuk, W. Cholewa. Fault Diagnosis. Springer-Verlag. 2004.

[4] L. H. Chiang, E. L. Russell, R. D. Braatz. Fault Detection and Diagnosis in Industrial Control. New York: Springer-Verlag, 2001.

[5] R. Patton, P. Frank, R. Clark (editor). Fault Diagnosis in Dynamic Systems. Prentice Hall, New York. 1989.

[6] I. T. Jolliffe. Principal Component Analysis. Second Edition. Springer, New York. 2004.

[7] J. H. Kim, G. O. Beale. Fault detection and classification in underwater vehicles using T^2 statistic. *Proceedings of Mediterranean Control and Automation Conference*. Dubrovnik, Croatia. 2001.

[8] G. Klančar, I. Škrjanc. A Principal Component Analysis in Fault Detection and Isolation: Hydraulic and Fermentation Process Example. Electrotechnical Review, 69(5): 311-316. 2002.

[9] M. A. Kramer. Nonlinear principal analysis using autoassociative neural networks. *AIChE Journal*, 37: 233-243. 1991.

[10] W. W. Hsieh. Nonlinear canonical correlation analysis by neural networks. *Neural networks*, 13: 1095-1105. 2000.

[11]E. C. Malthouse. Limitations of nonlinear PCA as performed with generic neural networks. *IEEE transactions on neural networks*. 9: 165-173. 1998.

[12] J.W. Hines, R.E. Uhrig, D.J. Wrest. Use of Autoassociative Neural Networks for Signal Validation. Juornal of Intelligent and Robotic Systems, 21: 143-154. 1998.

[13] D. Valh, B. Bratina, B. Tovornik. On-line fault detection and isolation using analytical redundancy. Electrotechnical Review, 73(5): 267-272. 2006.

ISBN 978-3-901608-32-2